

# Towards Data-Driven Control for Modern Wind Turbines

Ivo Houtzager



# **TOWARDS DATA-DRIVEN CONTROL FOR MODERN WIND TURBINES**

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  
vrijdag 2 december 2011 om 10:00 uur

door

**Ivo HOUTZAGER**

werktuigkundig ingenieur  
geboren te Krimpen aan den IJssel

Dit proefschrift is goedgekeurd door de promotor:

Prof. dr. ir. M. Verhaegen

Copromotor:

Dr. ir. J. W. van Wingerden

Samenstelling promotiecommissie:

|                                  |                                           |
|----------------------------------|-------------------------------------------|
| Rector Magnificus,               | voorzitter                                |
| Prof. dr. ir. M. Verhaegen,      | Technische Universiteit Delft, promotor   |
| Dr. ir. J. W. van Wingerden,     | Technische Universiteit Delft, copromotor |
| Prof. dr. L. Y. Pao,             | University of Colorado                    |
| Prof. dr. ir. M. Steinbuch,      | Technische Universiteit Eindhoven         |
| Prof. dr. ir. J. Hellendoorn,    | Technische Universiteit Delft             |
| Prof. dr. ir. G. A. M. van Kuik, | Technische Universiteit Delft             |
| Prof. dr. M. Lovera,             | Politecnico di Milano                     |



This dissertation has been completed in partial fulfilment of the requirements of the Dutch Institute of Systems and Control (DISC) for the graduate studies.



The work presented in this thesis has been partly supported by the Dutch Technology Foundation (STW) under project number: TMR5636.



The work presented in this thesis has been partly supported by the We@Sea consortium under project number: 2007-014.

ISBN: 978-90-5335-490-2

Copyright © 2011 by I. Houtzager.

All rights reserved. No part of the material protected by this copyright notice may be reproduced or utilized in any form or by any means, electronic or mechanical, including photocopying, recording or by any information storage and retrieval system, without written permission from the copyright owner.

Printed by Ridderprint in The Netherlands





# Acknowledgements

**I**N the last four years, my family and friends frequently comment me about what an extraordinary job I do by myself. They assume that the person writing this thesis is doing all the work on his own, when in reality various people contributed to the research in different ways. Doing something as complicated as a Ph.D. research is not something you do by yourself. Doing something as complicated as a Ph.D. research is not something you do by yourself, I know.

First, I would like to thank all my colleagues within the DCSC department. Michel, I am grateful you gave me the opportunity of achieving this and improving my writing and presentation skills. Jan-Willem, I really appreciated your tireless support and guidance and it was pleasure working with you. Your enthusiasm, constructive criticism, and willingness to make time for me whenever I needed, made me confident in doing my research. Gijs, Justin, Jianfei, Xiukun, Rufus, Balasz, Navin, Pieter, and Patricio, your comments, suggestions, and feedback have been very valuable and I enjoyed working with you. Other colleagues I would particularly like to mention are Paolo, Andrea, Jacopo, Pawel, Ali, Amol, Stefan, Aleksandar, Ruxandra, and Federico, with whom I spent enjoyable times both inside and outside the office.

Furthermore, I would like to thank all my colleagues within the DUWIND organization. Gijs, Wim, Teun, Thanasis, and Carlos, I really appreciate the fruitful discussions about wind turbines and your support during the experiments. I would like to thank Stoyan from ECN for providing me with the simulation data for the identification experiments. I am grateful to the members of my Ph.D. committee for providing me with constructive remarks, which helped me to improve this thesis. I would like to thank the members of the STW user committee for their input during project meetings and for hosting some of these meetings. I would also like to thank those who did not help me directly with this work, but contributed to make these years funnier and less boring. Sebastiaan, Barend, Bas, Edwin, Rianne, Erik and other KrRB lifeguards, with whom I always enjoyed swimming on Thursday evening. Charles and Alida, thank you very much for providing excellent beers afterwards.

Last, but certainly not least, I would like to thank my parents Bart and Henny and my brother Leon and his partner Sjoerd for their support, encouragement, and understanding.

*Ivo Houtzager  
Rijswijk, November 2011*





# Contents

|                                                                                                      |            |
|------------------------------------------------------------------------------------------------------|------------|
| <b>Acknowledgements</b>                                                                              | <b>vii</b> |
| <b>1 Introduction</b>                                                                                | <b>1</b>   |
| 1.1 Introduction to modern wind turbines . . . . .                                                   | 1          |
| 1.1.1 Phenomenon of the wind . . . . .                                                               | 2          |
| 1.1.2 The history of wind energy . . . . .                                                           | 3          |
| 1.1.3 State-of-the-art in wind turbine designs . . . . .                                             | 4          |
| 1.1.4 Trends in wind turbine designs . . . . .                                                       | 5          |
| 1.2 Introduction to modern wind turbine control . . . . .                                            | 7          |
| 1.2.1 State-of-the-art control for power maximization . . . . .                                      | 7          |
| 1.2.2 State-of-the-art control for load minimization . . . . .                                       | 11         |
| 1.2.3 Developments in active load reduction . . . . .                                                | 12         |
| 1.3 Introduction to data-driven modelling and control . . . . .                                      | 15         |
| 1.3.1 First-principles modelling . . . . .                                                           | 15         |
| 1.3.2 Data-driven modelling . . . . .                                                                | 16         |
| 1.3.3 Data-driven control . . . . .                                                                  | 18         |
| 1.4 Problem formulation . . . . .                                                                    | 19         |
| 1.5 Main contributions and outline . . . . .                                                         | 20         |
| <b>2 Predictor-Based Subspace Identification for the Model-Based Control of Wind Turbine Systems</b> | <b>25</b>  |
| 2.1 Introduction . . . . .                                                                           | 25         |
| 2.2 System description, assumptions and notations . . . . .                                          | 28         |
| 2.2.1 System description and assumptions . . . . .                                                   | 28         |
| 2.2.2 Innovation form descriptions . . . . .                                                         | 28         |
| 2.2.3 Lifted system description and notations . . . . .                                              | 29         |
| 2.3 Batchwise identification with VARX-based PBSID <sub>opt</sub> . . . . .                          | 32         |

|          |                                                                                                   |           |
|----------|---------------------------------------------------------------------------------------------------|-----------|
| 2.3.1    | VARX predictor with finite past window . . . . .                                                  | 32        |
| 2.3.2    | The relation between the VARX predictor and the state . . . . .                                   | 33        |
| 2.3.3    | The VARX-based PBSID <sub>opt</sub> algorithm . . . . .                                           | 34        |
| 2.4      | Batchwise identification with VARMAX-based PBSID <sub>opt</sub> . . . . .                         | 36        |
| 2.4.1    | VARMAX predictor with finite past window . . . . .                                                | 36        |
| 2.4.2    | The relation between the VARMAX predictor and the state . . . . .                                 | 37        |
| 2.4.3    | The VARMAX-based PBSID <sub>opt</sub> algorithm . . . . .                                         | 38        |
| 2.5      | Simulation study . . . . .                                                                        | 40        |
| 2.5.1    | Identification results with the SUSCON wind turbine model . . . . .                               | 40        |
| 2.5.2    | Identification results with the TURBU wind turbine model . . . . .                                | 43        |
| 2.5.3    | Identification results with the PHATAS wind turbine model . . . . .                               | 48        |
| 2.6      | Conclusion . . . . .                                                                              | 52        |
| <b>3</b> | <b>Recursive Identification with Application to the Real-Time Closed-Loop Tracking of Flutter</b> | <b>53</b> |
| 3.1      | Introduction . . . . .                                                                            | 53        |
| 3.2      | Recursive identification with RPBSID <sub>pm</sub> . . . . .                                      | 55        |
| 3.2.1    | Step 1: Adaptive filtering of the Markov parameters . . . . .                                     | 55        |
| 3.2.2    | Step 2: The selection of the weight and state vector basis . . . . .                              | 56        |
| 3.2.3    | Step 3: Adaptive filtering of the system matrices . . . . .                                       | 58        |
| 3.2.4    | The RPBSID <sub>pm</sub> algorithm . . . . .                                                      | 58        |
| 3.3      | Practical implementation in real time . . . . .                                                   | 59        |
| 3.3.1    | Regularization of the Markov estimation problem . . . . .                                         | 60        |
| 3.3.2    | Computational cost . . . . .                                                                      | 61        |
| 3.4      | Simulation study . . . . .                                                                        | 64        |
| 3.4.1    | Slowly time-varying case . . . . .                                                                | 64        |
| 3.4.2    | Abrupt-change case . . . . .                                                                      | 66        |
| 3.5      | Experimental study on 2D-airfoil system . . . . .                                                 | 66        |
| 3.5.1    | Description of the experiment . . . . .                                                           | 67        |
| 3.5.2    | Experimental modelling and control . . . . .                                                      | 69        |
| 3.5.3    | Experimental results . . . . .                                                                    | 71        |
| 3.6      | Conclusion . . . . .                                                                              | 77        |

|          |                                                                                            |            |
|----------|--------------------------------------------------------------------------------------------|------------|
| <b>4</b> | <b>Rejection of Periodic Wind Disturbances on an Experimental Smart Rotor Test Section</b> | <b>79</b>  |
| 4.1      | Introduction . . . . .                                                                     | 79         |
| 4.2      | System description, assumptions and notations . . . . .                                    | 81         |
| 4.2.1    | System description and assumptions . . . . .                                               | 81         |
| 4.2.2    | Lifted system description and assumptions . . . . .                                        | 82         |
| 4.3      | Repetitive control with lifted LQG design . . . . .                                        | 83         |
| 4.3.1    | Output-feedback formulation . . . . .                                                      | 84         |
| 4.3.2    | Multiple memory loops . . . . .                                                            | 85         |
| 4.3.3    | The lifted LQG solution . . . . .                                                          | 86         |
| 4.4      | Experimental study on “smart” rotor section . . . . .                                      | 89         |
| 4.4.1    | Description of the experiment . . . . .                                                    | 89         |
| 4.4.2    | Experimental modelling and control . . . . .                                               | 90         |
| 4.4.3    | Experimental results . . . . .                                                             | 92         |
| 4.5      | Conclusion . . . . .                                                                       | 101        |
| <b>5</b> | <b>Wind Turbine Load Reduction by Rejecting the Periodic Load Disturbances</b>             | <b>103</b> |
| 5.1      | Introduction . . . . .                                                                     | 103        |
| 5.2      | Description of the simulations . . . . .                                                   | 104        |
| 5.2.1    | Specifications of the reference wind turbine . . . . .                                     | 104        |
| 5.2.2    | Simulation environment . . . . .                                                           | 105        |
| 5.2.3    | Baseline control for above-rated operation . . . . .                                       | 106        |
| 5.2.4    | Individual pitch control for load reduction . . . . .                                      | 108        |
| 5.2.5    | Wind conditions during the simulations . . . . .                                           | 112        |
| 5.3      | Lifted Repetitive Control Design . . . . .                                                 | 113        |
| 5.3.1    | System descriptions and assumptions . . . . .                                              | 114        |
| 5.3.2    | Lifted repetitive control by LQG design . . . . .                                          | 116        |
| 5.3.3    | Model linearisation for the lifted RC design . . . . .                                     | 119        |
| 5.3.4    | Extensions to RC for uncertain period time . . . . .                                       | 120        |
| 5.3.5    | Guidelines for selection of the parameters . . . . .                                       | 122        |
| 5.4      | Simulation results and analysis . . . . .                                                  | 123        |
| 5.4.1    | Performance of individual pitch controller . . . . .                                       | 123        |
| 5.4.2    | Performance of lifted repetitive controller . . . . .                                      | 125        |
| 5.5      | Conclusion . . . . .                                                                       | 130        |
| <b>6</b> | <b>Conclusions and Recommendations</b>                                                     | <b>133</b> |
| 6.1      | Conclusions . . . . .                                                                      | 133        |
| 6.2      | Recommendations . . . . .                                                                  | 135        |

|          |                                                                              |            |
|----------|------------------------------------------------------------------------------|------------|
| <b>A</b> | <b>On the Consistency of VARX Model Estimation in Closed Loop</b>            | <b>139</b> |
| A.1      | Consistency when the true system is the VARX model . . . . .                 | 139        |
| A.2      | Consistency when the true system is the innovation state-space model .       | 140        |
| A.2.1    | Controller description and assumptions . . . . .                             | 140        |
| A.2.2    | The relation between the VARX and Markov parameters . . . . .                | 141        |
| A.2.3    | The asymptotic behaviour for a finite VARX predictor . . . . .               | 141        |
| <b>B</b> | <b>Spectral Analysis for the Verification of Models in Closed Loop</b>       | <b>143</b> |
| B.1      | Transfer function description and assumptions . . . . .                      | 143        |
| B.2      | Consistency with conventional spectral analysis . . . . .                    | 144        |
| B.3      | Spectral analysis for closed-loop estimation . . . . .                       | 144        |
| <b>C</b> | <b>Sequentially Semi-Separable Matrix Descriptions and Computations</b>      | <b>145</b> |
| C.1      | Sequentially semi-separable matrix descriptions . . . . .                    | 145        |
| C.2      | Sequentially semi-separable matrix computations . . . . .                    | 147        |
| C.3      | SSS matrix construction for efficient repetitive control synthesis . . . . . | 152        |
|          | <b>Bibliography</b>                                                          | <b>153</b> |
|          | <b>List of Abbreviations</b>                                                 | <b>169</b> |
|          | <b>Summary</b>                                                               | <b>171</b> |
|          | <b>Samenvatting</b>                                                          | <b>173</b> |
|          | <b>Curriculum Vitae</b>                                                      | <b>175</b> |

# Introduction

---

To reduce the cost per kWh, the wind turbine manufactures want to increase the energy yield of wind farms as much as possible. One of the current developments is to increase the rotor diameter to maximize the energy yield, because of the quadratic relationship between rated power and rotor diameter, and also because of the increased wind resources at higher altitudes. Consequently, modern wind turbines have become the largest rotating machines on earth, with a rotor blade larger than the wingspan of an Airbus A380. Due to the increase of rotor size, and also due to fewer restricting planning constraints, wind farms are more often being relocated offshore where maintenance is difficult and expensive. Therefore, the need of an active reduction of loads on wind turbines is becoming more important.

This thesis focuses on the development of data-driven modelling and model-based control algorithms for the load reduction on modern wind turbines. To provide some additional background information, and to better motivate the research, this introductory chapter will start with a brief overview of the basic principles and trends in the field of wind energy. After this overview, the state-of-the-art control in modern wind turbines and the developments for load reduction are reviewed. Further, data-driven modelling and control is introduced. The chapter finishes with the problem formulation, the main contributions, and a brief outline of the remaining chapters of this thesis.

---

## 1.1 Introduction to modern wind turbines

**B**EFORE addressing the more technical control and modelling aspects of wind turbine technology, an attempt is made to give a short general introduction to modern wind turbines. This involves a very brief explanation of the phenomenon of wind, an historical part explaining the development of wind energy, the state-of-the-art in wind turbine designs, as well as a part dealing with trends in wind turbine designs. Since wind energy is still a rapidly developing area, the overview will be

far from exhaustive. It is also by no means the intention to give a full historical review of wind turbines, but merely to mention some major milestones in their development and to give examples of the historical exploitation of wind energy. For more background information the reader is referred to the books Burton et al. (2011); Manwell et al. (2010).

### 1.1.1 Phenomenon of the wind

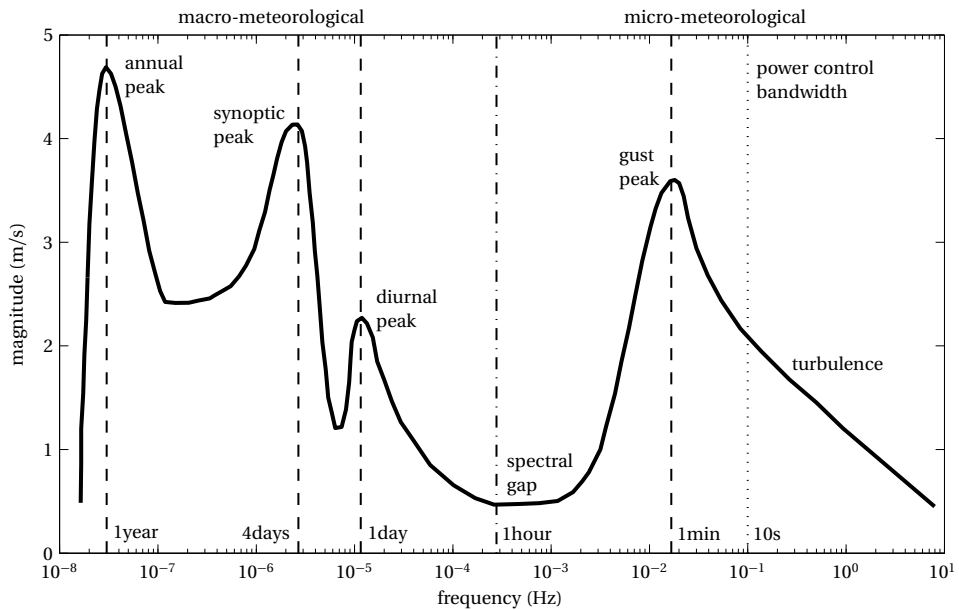
The dictionary describes wind as “an airflow in the atmosphere” (Van Dale, 2009). Around Earth there is an amount of air located in a shell, which covers the whole planet: the atmosphere. The layer rests on the surface of the Earth, because it is subjected to gravity. The air pressure is the weight of an air column, which reaches from the surface of Earth to the border of the atmosphere. On a specific time there are areas with a low and high pressure. The sun warms a specific region of the earth, this warm air has a lower density than cold air, so air will move and pressure differences arrive. As soon as there is a difference in pressure between two locations, nature tries to restore the balance. Air flows from high pressure areas to the low pressure areas. This movement is called wind. The wind speed will be faster if the pressure difference over a certain distance is larger.

The force from pressure difference applied on an air particle is not the only one; also the friction between the air current and the surface, the shape of pressure areas, and the Earth rotation gives forces that influence the displacement of air. All of these influences in horizontal wind speed can be represented in a Van der Hoven spectrum. In Figure 1.1, a typical Van der Hoven spectrum for the horizontal wind speed is given. A high value indicates a significant change in wind speed over the corresponding time period. Although this figure is obviously height and site-specific, there are distinctive similarities. An overall phenomenon is the spectral gap between time periods of 10 minutes to 2 hours, that separates the long-term macro-meteorological effects, due to the gradient between pressure systems and diurnal winds, with short-term micro-meteorological effects, due to local wind gusts and turbulence.

In the lower layers of the atmosphere the wind speed is strongly influenced by the friction with the surface of the Earth. The roughness of the landscape, like buildings and trees, can reduce the wind speed and increase the turbulence intensity considerably. The wind shear effect is a roughness-induced variation of the wind speed at different heights above the surface, and represents for the effect of the wind getting a lower velocity as we move closer to the surface. Turbulence is very fast irregular wind flow, like whirls or vortices, with different frequencies (to a maximum period of 1 min), amplitudes, directions, and may intersect each other. Water surfaces are considered very smooth, therefore wind speed at sea is faster and less turbulent than at the same height on land (DWIO, 2009).

Wind turbines interact with these wind effects to capture its kinetic energy and converting it into usable energy. For this reason, these wind effects play also an important role in the control objectives for wind turbines. A clear distinction in control objectives can be made between the long-term wind effects to be tracked for improving the power output and with the short-term turbulence to be rejected for

reducing the fatigue loads. Control is successfully used in wind turbines to track the long-term macro-meteorological wind effects and even the short-term wind gusts for an improved power output. However, the irregular wind flows of turbulence decreases the effectiveness of wind turbines, and imposes a lot of wear and tear on the wind turbine. To reduce these fatigue loads, feedback control can be applied to damp the natural frequencies of the wind turbine structure that are in the same frequency range as the turbulence spectrum.



**Figure 1.1:** Typical Van der Hoven spectrum for the horizontal wind speed. (Green Rhino Energy, 2011; Kühn, 2001; Van der Hoven, 1957)

### 1.1.2 The history of wind energy

Wind energy was the most important source of energy in the Netherlands, before the invention of the steam engine. During The Golden Age (17th century) of the Netherlands, wind energy played an important part in the economic growth. Windmills were used to saw log timber, grind grain, and to pump water for irrigation or to prevent the ocean from flooding the low-lying land. Especially, the introduction of the wind-powered sawing of log timber into planks was important for the economic growth, as it enabled to get thinner planks at a 30 times faster rate than before (Bonke et al., 2004). This invention eventually led to the end of the wood-sawyers guilds<sup>1</sup>, who were manually sawing the log timber, but resulted in considerable more Dutch sail ships. These ships were even lighter and faster than any of the competing countries, and this property eventually led to the nickname: “flying Dutchmen” (Top, 2005).

<sup>1</sup>Dutch: “houtzagers gilden”. In that time it was very common in Holland to take your profession as surname, as probably did my name-related ancestors.

In the late nineteenth century the first wind turbines for generating electricity were developed and constructed by C.F. Brush, which is one of the founders of today's General Electric. This wind turbine had a rotor diameter of 17 meters and 144 rotor blades and had an electric output of only 12kW. It was the Dane P. la Cour, who later discovered with experiments, that faster moving wind turbines with just a few rotor blades are more efficient for electricity production than slow moving wind turbines. Wind turbines for generating electricity need to operate at high speeds, but do not require much torque or turning force. Between 1940-1950, the first "modern" two-bladed and three-bladed wind turbines were developed in Denmark.

Nowadays, the development of wind energy seems to be unstoppable. Wind energy was rediscovered after the oil crisis in 1973, and currently is a competitive form of renewable energy. It is clean, economic, practical and has renewable interaction with the environment and draws attention to politics, business circles and individuals. It is becoming an aesthetically pleasing symbol of a better future, especially when compared with the effects of acid rain, global climate change, radioactivity, land and water contamination associated with conventional energy sources. However, wind energy is currently still more expensive than the fossil fuel alternatives; like, natural gas, oil, and coal. With the future improvements in wind power output, the rising cost of fossil fuels, and that current rates of exploitation are expected to deplete within the next centuries, it is only a question of when, not if, wind energy can compete against the conventional energy sources.

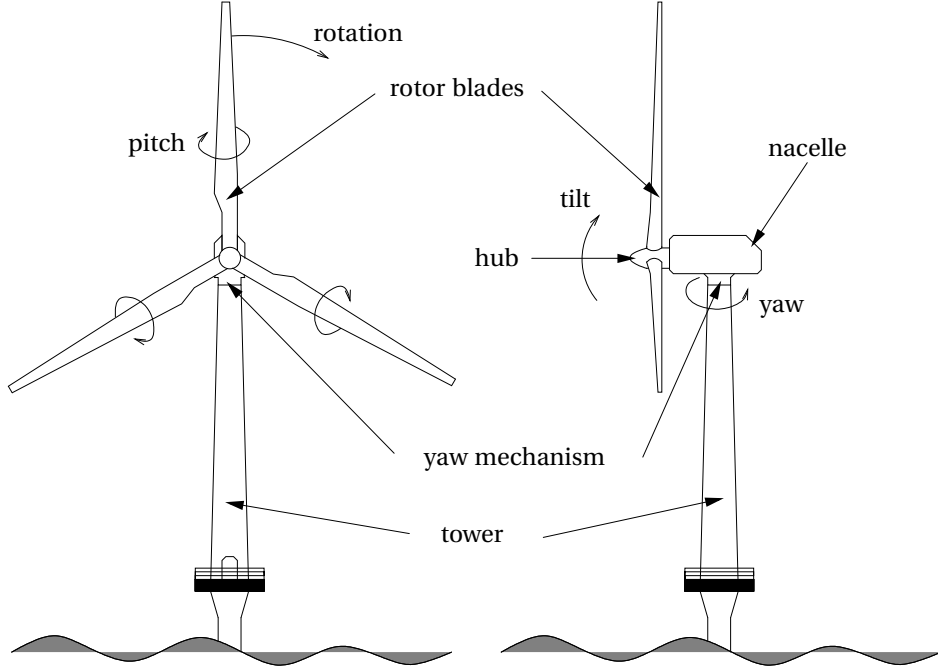
### 1.1.3 State-of-the-art in wind turbine designs

The majority of modern wind turbines have three blades and have the rotor positioned upwind with a horizontal axis. This Horizontal-Axis Wind Turbine (HAWT) design is also usually called the "Danish" concept. The main components of a HAWT that are visible from surface are the tower, nacelle, and rotor (with hub and blades), as shown in Figure 1.2. The rotor blades capture the kinetic energy of the wind and transform it into torque. The nacelle houses the generator driven by the rotor, with possibly a gearbox in between. A yaw mechanism keeps the rotor facing the wind. This is called an upwind design and has the advantage that the wind shade of the tower, also called the tower shadow, is smaller before the tower than behind. The disadvantage of upwind turbines is that the rotor blades need to be made stiffer, and placed at some distance from the tower to prevent the blades hitting the support structure. They provide excellent methods for adjusting the rotor speed for optimal power output at different wind speeds and minimizing the loads by regulating the pitch of the rotor blades and the generator torque.

The advantages of a three-bladed rotor are that the rotor is better balanced and has a greater aesthetic appeal. The disadvantages are that they cost more, weigh more, and can be difficult to install. One-bladed and two-bladed wind turbines exist and are cheaper and lighter, by saving the cost of one or two blades. They can rotate at higher speeds, which reduces the cost of the gearbox, and they are easier to install. However, these wind turbines can be noisier and endure heavy loads when the blades pass the front of the tower. One of the current solutions to avoid these loads is to install a teeter



mechanism, which is a special hub bearing that allows for the teeter movement of the entire rotor (similar to the tilt movement when the rotor is in the upright position).



**Figure 1.2:** Schematic representation of a modern offshore wind turbine

#### 1.1.4 Trends in wind turbine designs

The power available in the wind  $P_w(t)$  passing through an area  $A$  with a mean speed  $V(t)$  over the area can be given as:

$$P_w(t) = \frac{1}{2} \rho A V(t)^3, \quad (1.1)$$

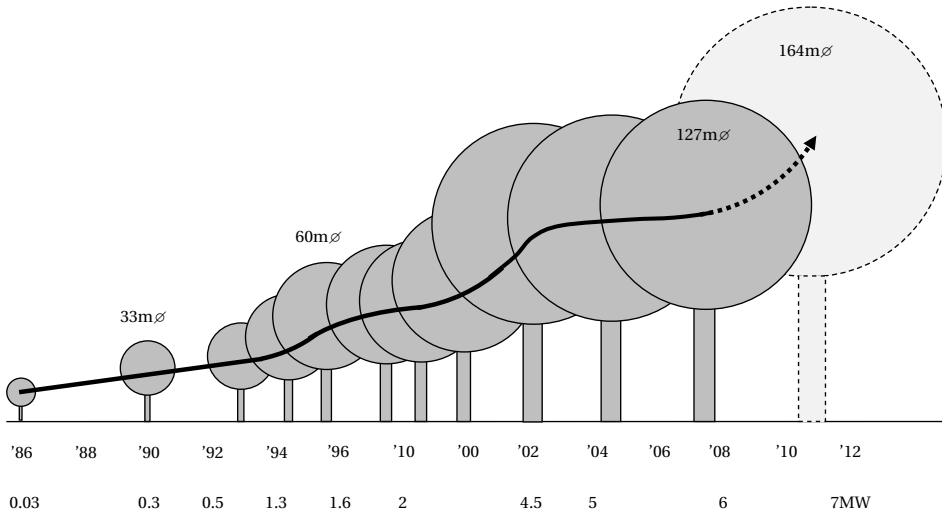
where  $\rho$  is the density of air. A simple way of modelling the ability of a wind turbine to capture wind energy is by defining a power coefficient as the ratio of extracted power and wind power as:  $C_p = P/P_w$ . With this definition, the power output  $P(t)$  of a wind turbine can be given by the well-known expression (Burton et al., 2011; Pao and Johnson, 2011);

$$P(t) = \frac{1}{2} \rho \pi R^2 C_p V(t)^3, \quad (1.2)$$

where  $R$  is the rotor radius. A doubling of the rotor diameter leads to a four-time increase in power output. The influence of the wind speed is even greater. With a double increase of wind speed leads to an eight-fold increase in power. So, the trend with wind turbines is to increase the rotor diameter as much as possible and place the wind farms in areas with the highest wind speeds; namely, offshore locations, as the best onshore locations are already taken. The influence of the power coefficient  $C_p$  is

limited by the maximum achievable value of  $C_{p,\max} \approx 0.59$ , called the Betz limit (Betz, 1966). The power coefficient of modern wind turbines reaches values of about 0.45.

In Figure 1.3, the trend is illustrated with an 1.6 MW wind turbine with a rotor diameter of 60 m realized in 1996 to a 6 MW ENERCON E-126 wind turbine with a rotor diameter of 127 m realized in 2008. A 7 MW VESTAS V-164 wind turbine with a rotor diameter of 164 m is announced to be realized in 2012. Thus it is clearly visible that wind turbine manufactures increase the rotor diameter to maximize the energy yield. Modern wind turbines have become the largest rotating machines on earth, with a rotor blade larger than the wingspan of an Airbus A380 (79.8 m).



**Figure 1.3:** The rotor diameter and power output at the moment of market introduction (van Kuik, 2001; van Wingerden, 2008).

Offshore locations offer significant advantages. First, the mean wind speeds are higher and there is less turbulence. Second, the expansion on offshore locations are less restricted than on land, due to less planning and noise constraints and fewer land use pressures. For these reasons, the turbine size can be increased more and larger wind farms can be developed. Early experiences (since 2001) has demonstrated the technical feasibility of large offshore wind farms, in spite of that wind power at sea calls for maintenance that is different from those onshore, because the accessibility to the turbines is more difficult. Therefore, the more demanding offshore environment requires more reliable wind turbines.

One of the research trends in wind turbine designs is to equip wind turbines only with two blades and position the rotor downwind instead (Fairley, 2009). As discussed in Subsection 1.1.3, the two-bladed downwind designs have a number of significant advantages compared with the current three-bladed upwind design. The use of only two blades will give a lighter rotor that can operate at higher rotational speeds, is cheaper to produce (minus one blade), and will be easier to install. Placing the rotor downwind will eliminate the tower clearance issues, such that more flexible blades and higher rotational speeds are possible. The success of this design will heavily

depend on active control techniques (including the work in this thesis) to compensate the disadvantages; namely, the increased tower shadow, and the unbalance of the rotor. A teeter mechanism can ensure that the overall load on the two blades remain the same, since a difference in load will cause a change in teeter angle. Thus it ensures that the shaft moment will be zero around the teeter axis. The load reducing effect has previously shown to be approximately 20% related to the flapwise rotor blade moment (Rasmussen and Kretz, 1992). However, the physical properties of the teeter bearing become more complex when it is used for larger rotating systems due the increasing gyroscopic effects and aerodynamic forces (Larsen et al., 2007).

## 1.2 Introduction to modern wind turbine control

This section provides a technical insight of the state-of-the-art control in modern wind turbines and the future developments in active load reduction. The exact control methods and strategies used in commercial wind turbines are protected by the manufacturers. However, it is expected that the work of Bossanyi (2000, 2003a); van der Hooft et al. (2003) forms the basis for the widespread control approaches in the industry. For this reason, this outlined control procedure will form the baseline for comparing the control strategies developed in this thesis. The control objectives considered in wind turbines are twofold: power maximization and load minimization.

With the background information in Subsection 1.1.1, the control objectives could be formulated as tracking the long-term wind effects for maximising the power output and rejecting the short-term turbulence induced vibrations for minimizing the fatigue loads, see Figure 1.1. From a practical point of view the above objectives are not very convenient, as the wind speed measurements are generally not available and the description of fatigue damage is highly non-linear and its prediction using rainflow counting and the Miner's rule gives a large uncertainty (Veldkamp, 2006). If the wind speed is measured, for example by an anemometer on the nacelle, this measurement is generally not useful for control. The measurement will be affected by the wind turbine wake and turbulence, therefore it is not representative for the mean wind speed over the rotor swept area. The non-linearity of the fatigue process makes it difficult to formulate an appropriate cost to be minimized (Hammerum et al., 2007). For both reasons, the objectives are pursued by other measures and the use of models.

### 1.2.1 State-of-the-art control for power maximization

Wind turbines with variable-speed operation are the most dominating type of newly installed wind turbines in the last years. This is due to their very attractive features, given by the presence of the power converter. When implementing a frequency converter between the generator and the electrical grid, it is possible to decouple the rotational speed from the grid frequency. Compared with fixed-speed turbines, the variable-speed turbines have the advantage that at lower wind speeds, called below rated, the rotor speed can vary to maintain an optimal power production. At low wind speeds the reduced rotor speed also reduces the aerodynamically generated acoustic noise and the stresses on the mechanical structure, since sudden wind gusts can be absorbed by an increase in rotor speed.

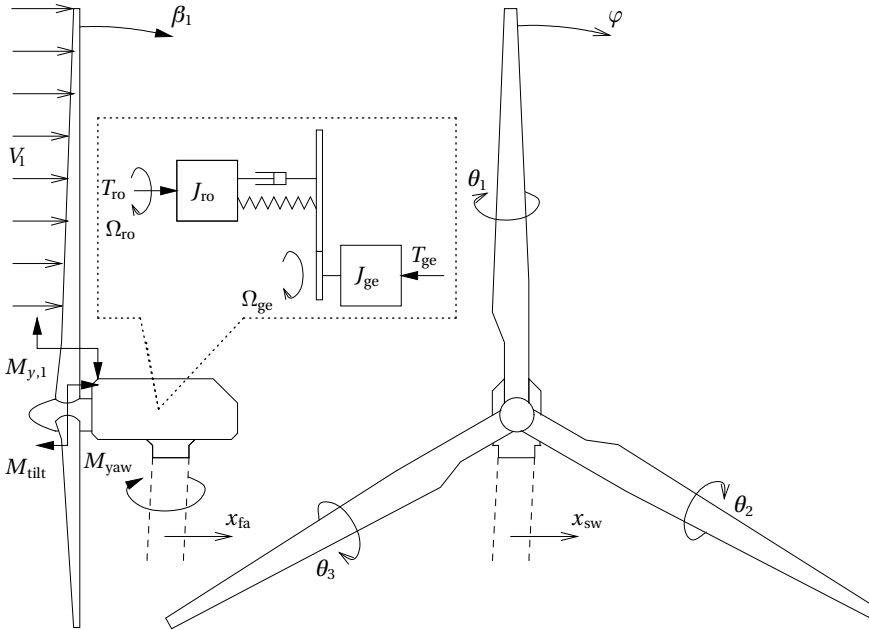
The use of variable-speed wind turbines has increased the importance of power control, also called torque control, depending if the reference input of the generator is the power  $P(t)$  or the generator torque  $T_{ge}(t)$ , see also Figure 1.4. These variables are related by the fundamental relation as:

$$P(t) = T_{ge}(t) \Omega_{ge}(t), \quad (1.3)$$

where  $\Omega_{ge}(t)$  is the rotational generator speed. The basis for power control is the wind turbine model in (1.2). This wind turbine model is simplified by introducing a static power coefficient  $C_p$  to represent the efficiency of the wind turbine under certain operational conditions. For this purpose, the effects of the dynamics to the efficiency are neglected. The strategy of the power controller is to achieve the optimal power coefficient  $C_{p,opt}$  by moving along the power coefficient surface  $C_p(\lambda(t), \theta(t))$  by adjusting the collective pitch angle  $\theta(t)$  and the tip speed ratio  $\lambda(t)$ . The tip speed ratio is the ratio between the mean wind speed and rotor speed and is defined as:

$$\lambda(t) = \frac{v \Omega_{ge}(t) R}{V(t)}, \quad (1.4)$$

where  $v$  is the gearbox ratio.



**Figure 1.4:** Schematic representation of the variables in wind turbine control

In Figure 1.5, the power coefficient surface is given for the UPWIND 5MW (also referred to as the NREL 5MW) wind turbine described in Jonkman et al. (2008). The power coefficient surface  $C_p(\lambda(t), \theta(t))$  can be calculated with the blade element momentum method used in most wind turbine modelling software. If the pitch angles are kept constant at fine pitch (in this case at  $\theta_{fine} = 0$  deg), the optimal

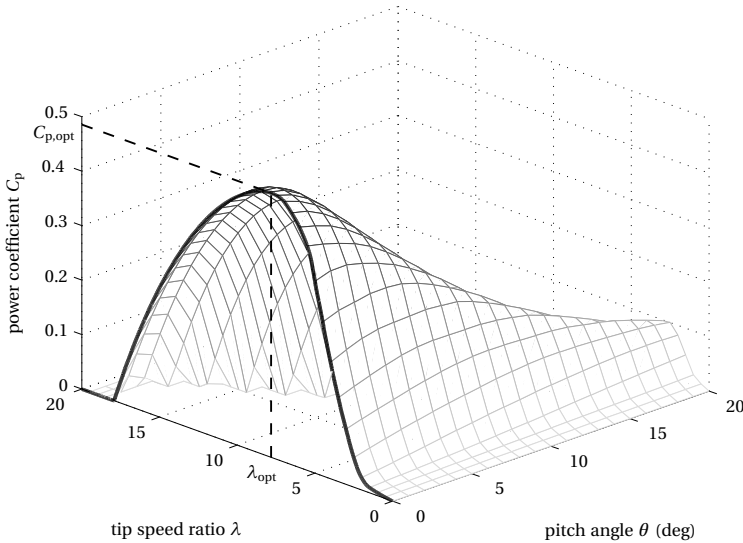
power coefficient  $C_{p,\text{opt}}$  can be found at an optimal value of the tip speed ratio  $\lambda_{\text{opt}}$ , see Figure 1.5. Now, the optimal generator speed  $\Omega_{\text{ge}}(t)$  can be described with the following equation:

$$\Omega_{\text{ge}}(t) = \frac{\lambda_{\text{opt}} V(t)}{v R}. \quad (1.5)$$

The optimal static relationship between the varying wind speed  $V(t)$  and the maximum power capture from the wind is already given by the model in (1.2). Rewriting (1.5) and substituting into (1.2) gives the optimal reference torque  $T_{\text{ge,opt}}(t)$  by tracking the generator speed  $\Omega_{\text{ge}}(t)$  as:

$$T_{\text{ge,opt}}(t) = K_\lambda \Omega_{\text{ge}}(t)^2 = \frac{\rho \pi R^5 C_{p,\text{opt}}}{2 v^3 \lambda_{\text{opt}}^3} \Omega_{\text{ge}}(t)^2. \quad (1.6)$$

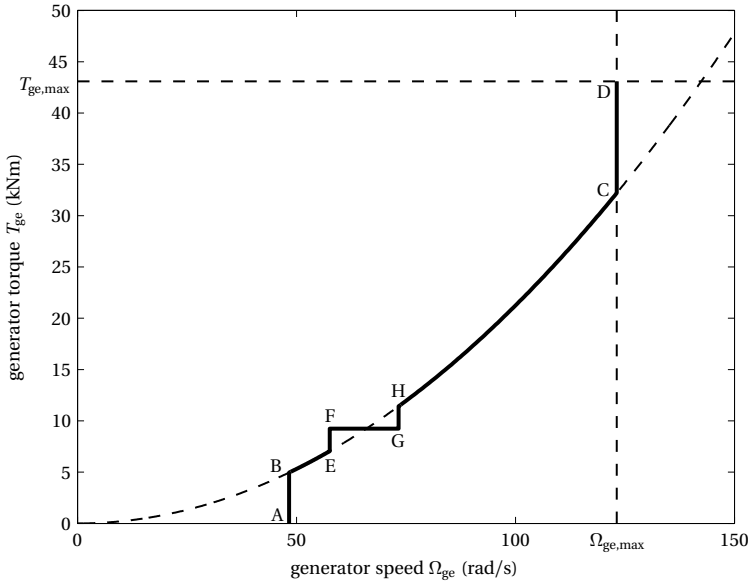
The fine pitch is normally selected to be around 2–3 deg higher to compensate for possible blade contamination. When the blades are covered with insects or ice, the fine pitch can be lowered to obtain some additional “spare” power, such that the gain  $K_\lambda$  can be kept at its nominal value (Spruce, 2004).



**Figure 1.5:** Power coefficient surface for the UPWIND 5MW wind turbine.

The quadratic rule (1.6) with some additional filters gives smooth and stable control in almost all variable-speed wind turbines (Johnson et al., 2006). However, for heavy rotors the large inertia could prevent it from changing fast enough to follow the wind optimally, because the dynamics of the rotor are neglected in the model. It is possible to manipulate the generator torque to cause the rotor to accelerate faster when required. In Bossanyi (2000), a feedback control method is described where the required generator speed is estimated using aerodynamic torque reconstruction. In Holley et al. (1999) it is shown that this would capture maximal 3% more energy from fast wind gusts, but it demands huge torque variations, making it in most cases impractical. Still an improvement can be expected by the identification of the power coefficient surface, due to the sometimes large difference between model and reality.

The power control strategy is thus fundamentally based on the static optimal curve in (1.6), and illustrated in Figure 1.6. This quadratic relationship is not used in the whole below rated area. For very low wind speeds it is not worthwhile to produce energy. With the UPWIND 5MW wind turbine, this is the case for wind speeds lower than 4 m/s. For start-up, the generator speed in trajectory A–B is tracked using a PI torque controller. Because of design restrictions, like tower clearance and acoustic noise, an upwind wind turbine reaches the maximum rotor speed at a relatively low wind speed. If the wind speed increases further, it is desirable to increase the generator torque without any further speed increase, in order to capture more energy from the wind. For this purpose, a PI torque controller is implemented to track the generator speed in the trajectory C–D. Finally, in above-rated condition D the generator torque is kept constant and the rated generator speed is tracked by adjusting the pitch, for example when the wind speed is still increasing the pitch angle is increased to lower the power coefficient to keep the rotor speed at rated. In Figure 1.5, it is clearly visible that the power coefficient is lower for an increasing collective pitch angle.



**Figure 1.6:** Generator torque-speed curve for the UPWIND 5MW wind turbine.

In Bossanyi (2000, 2003a), also an improvement in load reduction is recommended to the generator torque-speed trajectory A–B–C–D. The improvement is that the logic describing the different sections of the generator torque-speed curve can also be easily extended to implement so-called “speed exclusion zones”. These zones can be used to avoid speeds at which blade passing frequency would excite the tower resonance, by adding additional generator speed set-points with some logic for switching between them, see the trajectory E–F–G–H in Figure 1.6.

To catch the maximum wind power, horizontal-axis upwind wind turbines have to be directed towards the wind. This is accomplished actively with an electrical

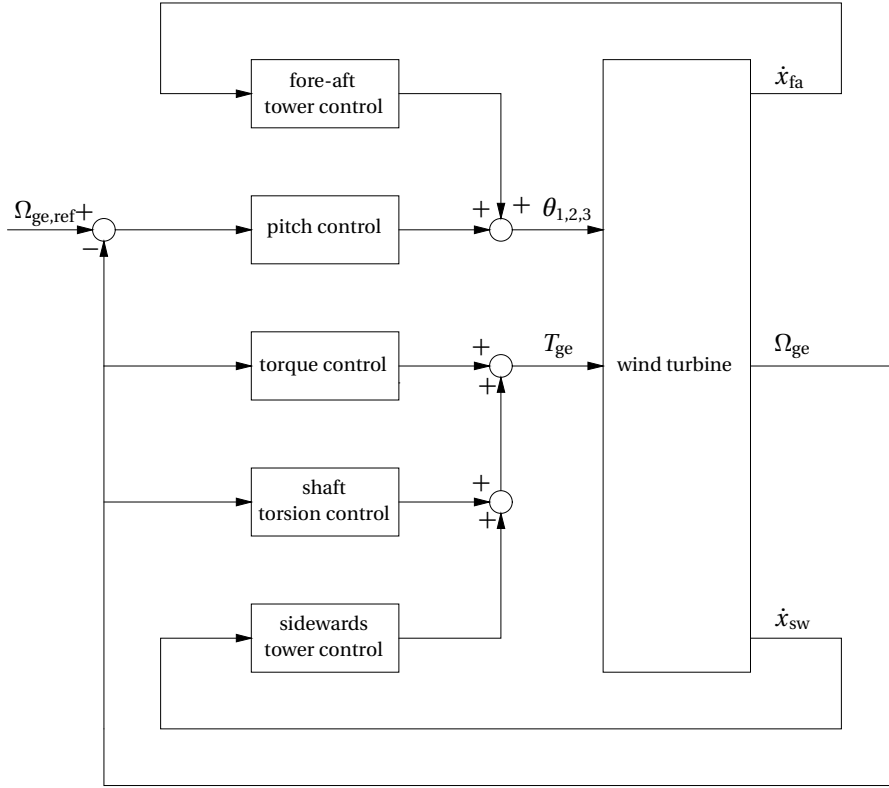
or hydraulic servo system that rotates the nacelle with rotor on the yaw bearing, see Figure 1.2. The wind turbine is said to have a yaw error, if the rotor is not perpendicular to the wind. Almost all manufactures of upwind turbines do not control their yaw error continuously; they prefer to brake the yaw mechanism whenever it is unused. The servo is activated only when the mean relative wind direction exceeds some predefined limits. A misaligned wind turbine would extract less energy compared with a wind turbine which has no yaw error. More important is that a wind turbine with yaw error is subjected to larger periodic loads than wind turbines which are yawed in a perpendicular direction against the wind. Yaw control is separated from other control, because of the many mechanical limitations. For example, the limitation in yaw angle due to the entanglement of the power cables in the tower.

### 1.2.2 State-of-the-art control for load minimization

Until the advent of MW-scale wind turbines in the 90's, stall regulation predominated to keep the loads within acceptable limits, as all other control options were considered too complex. Stall-regulated wind turbines have a rotor with blades that stall the flow around the blade in high wind speeds. This means that increasing wind speeds automatically induce increasing drag forces that limit the captured energy, therefore these wind turbines are stable by design. With new control technology, it became possible to introduce active stall-regulated wind turbines that automatically adjust the pitch to assist the stall at keeping the rotor speed constant when generating (occasionally double speed, with a lower rotor speed for lighter winds). With the increasing size of wind turbines, the stall-induced vibrations (vibrations which occur as the blade enters stall) were becoming problematic, for example the negative aerodynamic damping in deep stall causes the tower vibrations to rise to the shutdown level (Spruce and Turner, 2010).

Collective pitch regulation is now the favoured option for the largest modern wind turbines. This considerable improvement allowed that blade designs are significantly lighter, due to the lower load spectrum, and have more energy yield, as their design is not any longer optimized for stall characteristics. This is also one of the reasons that allowed modern wind turbines to run at variable rotational speed, but created a dependence on an active collective pitch control system to keep the loads within acceptable limits, therefore these modern wind turbines are not stable by design. Currently, the number of control loops in modern wind turbines for load reduction have increased considerably, see Figure 1.7. Most modern wind turbines have these control loops decomposed with care into the following, separate, and almost uncoupled Single-Input and Single-Output (SISO) control loops (Bossanyi et al., 2009):

- Generator speed regulation using the generator torque
- Generator speed regulation using the collective pitch
- Torsional shaft damping using the generator torque
- Fore-aft tower damping using the collective pitch
- Sidewise tower damping using the generator torque



**Figure 1.7:** Block diagram of the feedback loops for wind turbine control

In Chapter 5, each controller of these control loops is described in more detail for the UPWIND 5MW wind turbine. Many of these controllers can be designed using the classical loop-shaping methods of controller tuning. With some help of wind turbine modelling software, a linearised model of each of the SISO open-loop dynamics is extracted. The loop is then closed by a linear control transfer function with the measured signal as input and the control action as output. The classical loop-shaping techniques are then used to tune the controller transfer function, such that the closed-loop behaviour is sufficiently responsive and yet sufficiently stable, for example by examination of the stability margins and the use of root-locus, Nyquist, Bode and step response plots (Bossanyi et al., 2009).

### 1.2.3 Developments in active load reduction

With the increase in rotor size and the relocation offshore, where the environmental conditions are heavier and maintenance is difficult and expensive, the importance of active reduction of fatigue loads will increase. The purpose of such active control is to increase the lifespan of the wind turbine, without making the construction heavier, so that the wind turbine becomes more cost-effective. The collective pitch control



method assumes that wind gusts are active on the whole rotor area. Therefore, it can only handle slow wind changes that have an effect on the entire rotor. As the rotor size increases, the wind speed variation across the rotor swept area becomes larger. For further load reduction it is necessary to react to wind speed variations in a more detailed way; for example, each blade individually with Individual Pitch Control (IPC) (Bossanyi, 2003b, 2005; Selvam et al., 2009; van Engelen, 2006), or even at several radial distances spanwise along each blade with “Smart” Rotor Control (SRC) (Andersen et al., 2010; Barlas and van Kuik, 2010; Buhl et al., 2005; Lackner and van Kuik, 2010b).

The goal for both concepts is to increase the lifetime of the wind turbine. When the lifetime constraint is met, the concept provides the designer with an increased envelope which can be used to, for instance, mount the rotor downwind (for eliminating tower clearance issues), and allow lighter components or increase the mean loading on the blades which increases the power conversion. However, the lifetime of a wind turbine becomes also dependent on the wear and tear of the pitch/smart actuators. Therefore it is required that the control variations on these actuators are kept to a minimum. A preliminary study in Tavner et al. (2011) investigated the effects of the stochastic wind fluctuations, i.e. turbulence, on pitch mechanism failures. It was observed from statistical analysis that the cross-correlations between failures and the turbulence intensity coefficients are much larger than to the mean wind speed deviation coefficients. This could indicate that the current pitch control are tuned in such a way that the pitch actuators are acting to heavily on the turbulence.

### Individual pitch control

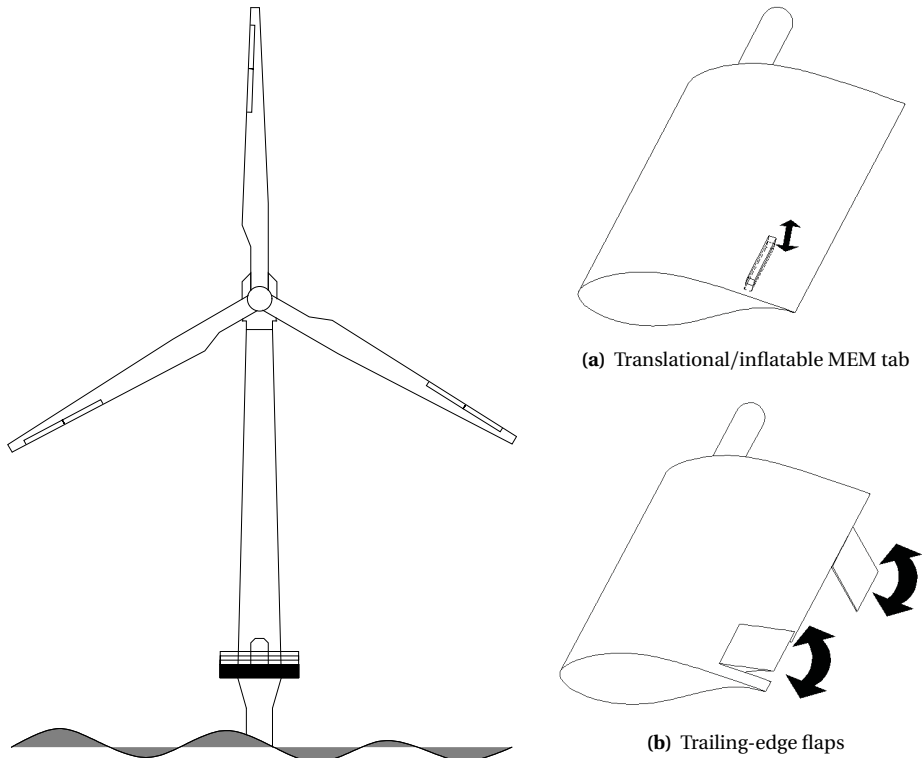
The idea for Individual Pitch Control (IPC) seems very attractive and almost readily implementable, since many of the larger wind turbines already have individual pitch actuators, although controlled collectively, for the regulation of the rotor speed. However, the performance of the IPC method is restricted by the limited bandwidth of the pitch actuator and they only affect the load on the whole blade (Lackner and van Kuik, 2010a). For this IPC concept, researchers are mainly investigating the use of feedback control after transforming (and decoupling) the rotating frame to a fixed non-rotating frame (Bossanyi, 2003b, 2005; van Engelen, 2006). This so-called Coleman transformation makes it possible to apply Single-Input and Single-Output (SISO) loop-shaping control techniques to generate smooth near-sinusoidal control signals to compensate for the 1P and 2P disturbance harmonics (Bossanyi, 2009).

For higher harmonics the use of loop-shaping techniques become more challenging, because more coupling exists between the transformed measurement channels. Model-based control design techniques should be used to overcome this problem, as proposed in Selvam et al. (2009). Also at these higher harmonics, a larger amount of phase lag (delay) from the input-output dynamics have to be compensated. As the generated control signals for the 1P and 2P harmonics are near sinusoidal, the phase lag can be compensated during the calculation of the reverse of the Coleman transformation. However, this becomes considerably more challenging at higher harmonics. The Coleman transformation with loop-shaping has also successfully been applied on an experimental two-bladed wind turbine in Bossanyi and Wright

(2009); Bossanyi et al. (2010), but the use of model-based control design methods is challenging as the dependence of the dynamics on the azimuth is not necessarily lowered.

### “Smart” rotor control

The “Smart” Rotor Control (SRC) concept is borrowed from the helicopter industry (Straub et al., 2009), where multiple actuators with a significantly higher bandwidth, like trailing-edge flaps or translational/inflatable Micro-Electro-Mechanical (MEM) tabs, are proposed to reduce the loads by manipulating the airflow locally and consequently the aerodynamic forces. In Figure 1.8(b) an illustrative example is given of a wind turbine with “smart” rotor blades. The rejection of disturbances using SRC is already a research topic for several years. The feasibility of load reduction using feedback control applied on an experimental “smart” airfoil with trailing-edge flaps was demonstrated in Bak et al. (2007); van Wingerden et al. (2008).



**Figure 1.8:** An illustrative example of a wind turbines with the “smart” rotor concept. At the tip of the blade a number of additional control devices are placed; for example, the MEM-tabs and trailing-edge flaps illustrated in (a) and (b), respectively.

With the increase of sensors and actuators on each blade, the loopshaping control techniques with the Coleman decoupling are not directly applicable. Possible solu-

tions are to use a different representation of the Coleman transformation together with model-based control as proposed in Thomsen et al. (2010), or to use the distributed feedback controllers proposed in Engels et al. (2010); Rice and Verhaegen (2010a). The difficulty for the synthesis of these kinds of controllers is that a special structured model is required, which is not always directly available. Also feedback controllers will typically give a lot of control variations to the actuators. Therefore, the wind turbine designers are reluctant to implement these new concepts directly in practice. An alternative is to compensate only the periodic disturbances using feedforward control, as successfully demonstrated on two rotating blades in van Wingerden et al. (2010, 2011). In this method the amplitudes of 1P, 2P, 3P, and 4P sinusoidal basis functions are adaptively fitted using the blade-root bending moment measurements from the previous rotation, and the frequencies are obtained by measuring the rotational speed. Furthermore, the method can easily be adapted to be used in combination with feedback control for both IPC and SRC. In Chapter 5, we will extend the development of this adaptive feedforward control by proposing the use of repetitive control in the form of a lifted feedback controller, which takes in addition the periodic time-varying MIMO dynamics and non-sinusoidal periodic load disturbances into consideration.

## 1.3 Introduction to data-driven modelling and control

This section provides a brief explanation of the modelling methodologies used for wind turbine control design and its verification. It is by no means the intention to give a full detailed overview of all the modelling aspects, but merely to mention the advantages, disadvantages, and challenges in the different modelling methodologies. The use of models has a long tradition in the wind energy community. In the industry, by far the most important purpose of the use of models is to lower the risk of failures. In Failure Modes and Effects Analysis (FMEA) (IEC, 1985), the risk of failure is defined as:  $\text{risk} = \text{severity} \times \text{occurrence} \times \text{detectability}$ . The severity of failures on the availability of the wind turbine system is very high, because any failures can easily cause the loss of its primary function during a long period and can even result in unsafe operation. For this reason, manufactures are forced to keep the occurrence and detectability numbers low (low detectability number means almost certain detection), and models (and also control) play an important role to achieve this. This role will increase as the risk of failure grows due the trend in rotor size and the relocation offshore.

### 1.3.1 First-principles modelling

Currently, the laws of physics are used to describe a relevant number of properties of the wind turbine in a mathematical language. These so-called first-principles models are extensively used to determine the static and dynamic loads under extreme conditions, to predict the lifetime of components by calculating the damage equivalent loads, to optimize the structural design, to tune and verify controllers, and condition monitoring. In the wind energy community there is a wide variety of different

modelling codes. Most of these modelling codes are very complex and have been especially developed to deal with design calculations and time-domain simulations. For detailed information and a comparison the reader is referred to Molenaar (2003).

The main advantage of first-principles modelling is that you can have a model before the actual wind turbine is built and consequently the model can be used for its design. For the purpose of lifetime prediction, controller tuning and verification, and condition monitoring, these first-principles models need to be calibrated and validated to the real system. One of the reasons is that the amount of detail in the model is normally the choice of the designer, who normally tends to “overmodel” the system to make sure to capture all the dynamics (van Wingerden, 2008). Further, it is hard to have exact knowledge of the material and aerodynamic properties, and small differences may produce significant different dynamics (Witteveen et al., 2007). The calibration and validation of these models is very time consuming and labour intensive process, therefore only a small part of the modelling codes are verified and even a smaller part are validated (Molenaar, 2003). This means that during the commissioning of the production wind turbine often requires a complete retuning of controllers and models due to differences between the model and the true system (van der Veen et al., 2011). A common approach for calibration is to do actual measurements at a number of different operation positions and to compute the Power Spectral Density (PSD) diagrams and the Campbell diagram to indicate the periodic disturbances and time-varying dynamics (Rossetti et al., 2008).

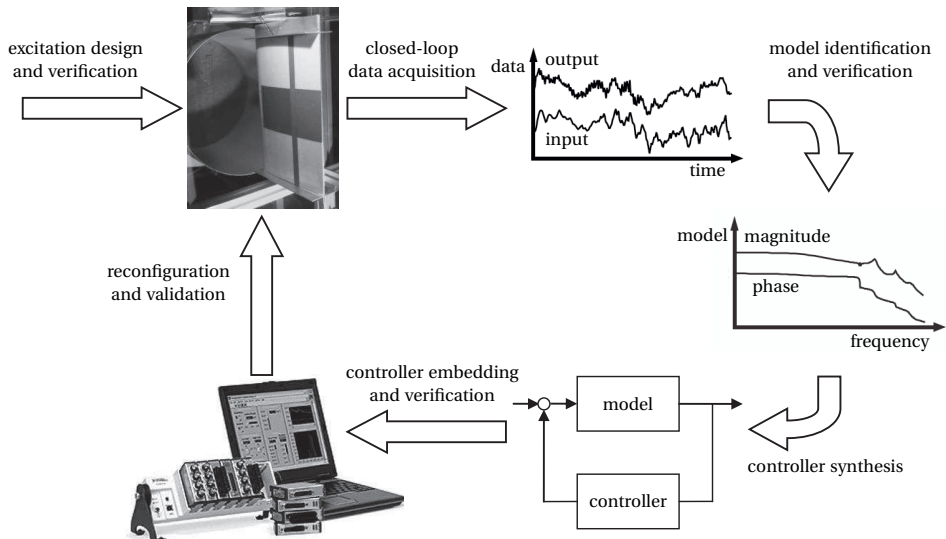
The main disadvantage of first-principles models is that they are not tailored for model-based control synthesis, because they typically contain irrelevant dynamics and are highly non-linear. Although there are a number of dedicated design tools that have the capability to derive models for model-based controller synthesis, it is still far from trivial how to isolate properties belonging to the dynamics of the wind turbine that are needed for model-based controller synthesis. For optimizing the controller it is more interesting to directly model the dynamics between the actuators, disturbances, and sensors because in these signals gain and phase information is present, which is basically the information needed for controller synthesis (van Wingerden, 2008). System identification can be used to create a more direct up-to-date model than the existing one by using measurement data. System identification is well established, but only a few applications are reported in the wind energy (Bongers and van Baars, 1991; Font et al., 2010; Iribas, 2006; van Baars and Bongers, 1994), mostly for the purpose of the validation of first-principles models.

### 1.3.2 Data-driven modelling

With the introduction of new rotor concepts with more sensors and actuators, model-based controller design becomes more important (maybe even necessary) for the wind energy community. Model-based controller synthesis is more suitable for multivariable control problems, but its effectiveness largely depends on the derivation of accurate and suitable models (preferably with an uncertainty description) and a characterization of the noises and disturbances acting on the system. The academic wind energy community has already a strong focus towards modern model-based control, for example in Bianchi et al. (2007); Bongers (1994); Østergaard (2008);

Steinbuch (1989); Stol (2001), but with current wind turbine designs the industry is not necessitated to use these model-based methods in practice, as the loopshaping methods suffice the requirements. Nevertheless, as wind turbines become larger and more flexible, it is possible that model-based control methods, perhaps in conjunction with these new rotor concepts, will increasingly find a way towards industry.

The trend is to use system identification not only as an update step to have a more accurate first-principles model, but to use identification methods directly in the (control) design of a wind turbine system. This so-called data-driven or experimental modelling directly obtains an identified model of the true system and avoids the non-trivial controller retuning, see Figure 1.9. Obviously, the quality of the identified model depends on the information in the data. More “informative” data can be generated by adding additional excitation to the inputs. If a more accurate model is obtained around the bandwidth, a less conservative model-based controller can be synthesized and implemented. To further improve the performance, the steps in the design of the excitation signals and the model-based controller can be repeated Gevers (2005); Hjalmarsson (2005). Although the number of significant advantages, system identification can not totally replace first-principles modelling. Apart that first-principles models are needed for the general design of the system and probably the synthesis of an initial base-line controller, they will keep playing an important role in the data-driven wind turbine modelling methodology to evaluate the designed excitation signal for excessive loads and to verify the behaviour of any synthesized model-based controller under extreme conditions.



**Figure 1.9:** Control design cycle with data-driven modelling.

Apart that the identification problem is normally multivariable, there are a number of additional challenges with the identification of wind turbine systems. In the specific case of wind turbine systems, the identification has to be performed with taking the

existing stabilizing controller into consideration. Most identification techniques for multivariable systems are all based on the open-loop setting and will give biased results in the closed-loop setting, see Chapter 2. The resulting identified models are also only valid around one operating point. For a fixed operation point of a wind turbine, the dynamics can be assumed to be linear time invariant, such that well-established identification methodologies can be used to obtain a model for control. Identification methods for the estimation of time-varying dynamics are available in the literature, for example the LPV identification methods in Tóth (2008); van Wingerden and Verhaegen (2009); van Wingerden et al. (2009). Mainly due to the “curse of dimensionality” they are not yet suited for realistic systems (van Wingerden, 2008). Further, the disturbances in the wind turbine are periodic in nature, therefore ideas similar to those in van Baars et al. (1993) (periodic basis functions) and in Kanev (2008) (Coleman transformation) needs to be applied for compensation of these periodic disturbances. Contributions in the form of identification algorithms that can deal with the above mentioned problems should be very general, therefore they could also be used outside the field of wind energy. Possible applications are active magnetic bearings (Balini, 2011; Blom, 2011), and adaptive optics (Hinnen, 2006; Song, 2011).

### 1.3.3 Data-driven control

The control system itself has the second largest annual failure rate for wind turbines (number one is the electrical system), although the annual downtime for this component is relatively low compared with faults in mechanical components (Faulstich et al., 2011; Ribrant and Bertling, 2007). One of the possible problems is the less straightforward integration with the supervisory control during changes in operational conditions and the safety system to reduce extreme actuator and structural loads. If model-based controllers are going to be used, this integration with the supervisory control system requires even further sophistication than with classical controllers, where adjustments such as the addition of notch filters, saturation with anti-windup, gain scheduling, and even startup/shutdown are more straightforward. A promising methodology under research is the so-called data-driven control design (Dong, 2009). In the data-driven control methodology, the intention is to use reliable data-driven methods that can update the controllers and the monitoring filters with low and intuitive supervision by detecting changes in the data. Basically, the data-driven control approach is to let the wind turbine speak for itself (through data), perhaps by asking first (through excitation signals), and adapt the control signals accordingly.

Similar to the control design based on data-driven modelling, the data-driven control methodology also enables that the controller is designed using data from the actual system to be controlled. In contrast, the data-driven controller constantly optimizes the closed-loop performance criterion based on actual data from the system, instead of on a fixed identified model of that system under nominal operating conditions. The data-driven control design can evade the need to accurately model the system and the need to be robust to all hypothetical disturbances, parasitic dynamics and operating conditions (the worst-case scenario in Woodley (2001)), and can consider only those that actually and most probably occur (the cautious scenario in Dong and Verhaegen (2009)). Not only for higher performance, but also to prevent

“impossible” choices for the controller, especially during the very uncertain transient periods. By taking additional signal constraints in the control objective, it is possible to account for input saturation, fatigue loads (in terms of signal variances), sensor and actuator faults (if redundant), and the lack of information in the data through the optimisation of the excitation signal (Dong, 2009; Hallouzi, 2008).

Besides the computational complexity of the algorithms involved, which may require large online computations, it is difficult to allow industrial implementation of such control systems. Most steps of the control design cycle depicted in Figure 1.9 will be automated in the data-driven control methodology, therefore it is very hard to verify each step individually before online implementation. The acceptance of a data-driven controller depends heavily on the global stability without cumbersome assumptions on the system and disturbances. Although some data-driven control design methods incorporate some sort of stability proof (Kulcsár and Verhaegen, 2011), they generally rely on system knowledge like the order of the plant, or they require sufficiently exciting input signals of some order. Additionally, the closed-loop is tuned to specific conditions and the effects of a change in these conditions on the controller and its adaptation are unclear in general, therefore the tuning happens not in a very insightful way. As the current theory and simulations are limited, research experiments will play an important role to get more experience, thrust, and knowledge of the adaptation and stability behaviour in reality. The way towards data-driven control may lie in experiments with data-driven methods for condition monitoring and fault detection of wind turbines (Wei and Verhaegen, 2008; Wei et al., 2009, 2010), thus without closing the feedback loop to overcome the stability requirement.

## 1.4 Problem formulation

Further developments towards control techniques for the load reduction of modern wind turbines can achieve an increased lifetime of components and make the scaling to larger rotor diameters possible, and therefore improve the cost effectiveness of wind turbines. Developments in active control technologies played an important role in the history of load reduction for wind turbines, and this is probably also the case in the future. For example, in Section 1.2.3 new rotor concepts were described for future wind turbines. The success of these designs will heavily depend on new developments in active control technology.

This thesis focuses on the development of data-driven modelling and model-based control algorithms for the load reduction of modern wind turbines, where the definition of modern wind turbines include the new developments in rotor concepts. The focus will be on novel algorithms for the rejection of periodic disturbances. The load disturbances seen by an individual rotor blade are to a large extent periodic and repetitive; for instance, tower shadow, wind shear, yawed error, and gravity are depending on the azimuth angle and rotational speed, and will change slowly over time. The contribution of these periodic disturbances will become even larger in the future, due to the increasing length and mass of the rotor blades and the possible downwind location of a two-bladed rotor. By compensating only the periodic disturbances, it is expected that the required control action would not create too much fatigue damage

on the actuators, as they are much smoother than the turbulence induced vibrations. To summarize, the main problem formulation is:

**Problem Description 1.1** (Periodic disturbance rejection). *Develop control algorithms, suited for real-time implementation, for the rejection of periodic load disturbances in modern wind turbines, and verify/validate the feasibility of these algorithms for this purpose.*

In pursuit of this goal, the novel control synthesis would require a mathematical model description of the system to be controlled. As discussed in Section 1.3, the way to obtain these models is from measured data using identification techniques. Being that modern wind turbines are not asymptotically stable in open-loop (only closed-loop experiments are supported in reality), that new rotor concepts would require an increasing number of actuators and sensors, and that the operational conditions of modern wind turbines slowly change over time, makes it necessary to formulate a second problem as:

**Problem Description 1.2** (Data-driven modelling). *Develop identification algorithms that do not require any controller related information to consistently identify a linear model from systems with multi inputs and multi outputs, and are able to track the dynamics under slowly changing operational conditions.*

## 1.5 Main contributions and outline

This section provides first a brief overview of the outline and then the main contributions of this thesis. This thesis contains six chapters, where the first is this introduction. Each of the subsequent chapters, except for Chapter 6, are basically extended versions of papers that have been published in or accepted to well-established journals and conferences in the field of wind energy, system identification, and control. Chapters 2 and 3 are dedicated to the identification problem of modern wind turbine systems. Chapters 4 and 5 are dedicated to the load reduction control problem of modern wind turbine rotor concepts. Chapter 4 follows the data-driven control design cycle using the identification methods in Chapter 2. The main contributions and the detailed content in these main chapters are given in the following subsections. The summary of the chapter goals, the means of which algorithms and tools they are achieved with, and the proof of concept of these algorithms and tools are schematically given in Table 1.1. The main references where the content of each chapter has been published are reported in a local bibliography as well. Chapter 6 concludes this dissertation, summarizes the main conclusions that can be drawn, and will provide suggestions for future research.

## Chapter 2: Predictor-Based Subspace Identification for the Model-Based Control of Wind Turbine Systems

In this chapter, the “optimized” Predictor-Based Subspace IDentification (PBSID<sub>opt</sub>) algorithm will be presented for closed-loop wind turbine model estimation (Houtzager et al., 2009, 2010). Further, a novel VARMAX-based PBSID<sub>opt</sub> method is developed



that relaxes the requirement that a certain window has to be large for asymptotically consistent estimates. The PBSID<sub>opt</sub> methods are very attractive for the wind power community, since it provides the user with the ability to identify LTI systems with multiple inputs and multiple outputs, and does not require any controller related information. The PBSID<sub>opt</sub> methods are verified on a data source from controller-in-the-loop simulations of a modern wind turbine. The developed algorithms have been implemented by the author in software, which are contributed in the PBSIDToolbox (Houtzager, 2010).

## Bibliography

- I. Houtzager. Predictor-Based Subspace Identification Toolbox version 0.4. Website, 2010. URL <http://www.dcsc.tudelft.nl/~datadriven/pbsid/>.
- I. Houtzager, J. W. van Wingerden, and M. Verhaegen. VARMAX-based closed-loop subspace model identification. In *48th IEEE Conference on Decision and Control*, Shanghai, China, 2009.
- I. Houtzager, B. A. Kulscár, J. W. van Wingerden, and M. Verhaegen. System identification for the control of wind turbine systems. In *3rd Conference on the Science of Making Torque from Wind*, Heraklion, Crete, Greece, 2010.

## Chapter 3: Recursive Identification with Application to the Real-Time Closed-Loop Tracking of Flutter

In this chapter, a novel Recursive Predictor-Based Subspace Identification (RPBSID<sub>pm</sub>) algorithm will be presented to identify LTI systems with multiple inputs and multiple outputs (Houtzager et al., 2009). The algorithm is suited for real-time implementation and will provide consistent estimates from data gathered in open loop or closed loop. The real-time implementation and the ability to work with MIMO systems operating in closed loop makes this approach suitable for online estimation of unstable dynamics. The ability to do so is demonstrated by the detection of flutter on a “smart” rotor test section (Houtzager et al., 2011). The developed algorithms have been implemented by the author in software, which are contributed in the PBSIDToolbox (Houtzager, 2010).

## Bibliography

- I. Houtzager. Predictor-Based Subspace Identification Toolbox version 0.4. Website, 2010. URL <http://www.dcsc.tudelft.nl/~datadriven/pbsid/>.
- I. Houtzager, J. W. van Wingerden, and M. Verhaegen. Fast-array recursive closed-loop subspace model identification. In *15th IFAC Symposium on System Identification*, Saint-Malo, France, 2009.

- I. Houtzager, J. W. van Wingerden, and M. Verhaegen. Recursive predictor-based subspace identification with application to the real-time closed-loop tracking of flutter. *Accepted for IEEE Transactions on Control System Technology*, 2011.

## Chapter 4: Rejection of Periodic Wind Disturbances on an Experimental Smart Rotor Test Section

In this chapter, an extended lifted Repetitive Control (RC) algorithm will be presented for periodic wind disturbance rejection of linear systems with multiple inputs and multiple outputs and with both repetitive and non-repetitive disturbance components. This extended RC algorithm can reject periodic wind disturbances in modern wind turbines, and this is demonstrated on an experimental “smart” rotor test section (Houtzager et al., 2011a,b). The developed algorithms have been implemented by the author in software, which are contributed in the LRCToolbox (Houtzager, 2011a, LRCToolbox) and the SSSToolbox (Houtzager, 2011b).

### Bibliography

- I. Houtzager. Lifted Repetitive Control Toolbox version 0.2. Website, 2011a. URL <http://www.dsc.tudelft.nl/~datadriven/lrc/>.
- I. Houtzager. Sequentially Semi-Separable Matrix Toolbox version 0.8. Website, 2011b. URL <http://www.dsc.tudelft.nl/~datadriven/sss/>.
- I. Houtzager, J. W. van Wingerden, and M. Verhaegen. Rejection of periodic wind disturbances on a smart rotor test section using lifted repetitive control. *Provisionally accepted for IEEE Transactions on Control System Technology*, 2011a.
- I. Houtzager, J. W. van Wingerden, and M. Verhaegen. Rejection of periodic wind disturbances on an experimental “smart” rotor section using lifted repetitive control. In *IEEE Multi-Conference on Systems and Control*, Denver, CO, USA, 2011b.

## Chapter 5: Wind Turbine Load Reduction by Rejecting the Periodic Load Disturbances

In this chapter, a novel lifted Repetitive Control (RC) algorithm that can reject the periodic load disturbances for modern fixed-speed wind turbines and modern variable-speed wind turbines operating above-rated equipped with individually pitch controlled blades. For verification, the load reduction capabilities of RC compared with typical IPC are investigated in the operation of the UPWIND 5MW wind turbine in aerolastic modelling software (Houtzager et al., 2011). The developed algorithms have been implemented by the author in software, which are contributed in the LRCToolbox (Houtzager, 2011b) and the DISCON tool (Houtzager, 2011a).

## Bibliography

- I. Houtzager. External controller design tool for GH BLADED and SIMULINK. Website, 2011a. URL <http://www.dcsc.tudelft.nl/~datadriven/discon/>.
- I. Houtzager. Lifted Repetitive Control Toolbox version 0.2. Website, 2011b. URL <http://www.dcsc.tudelft.nl/~datadriven/lrc/>.
- I. Houtzager, J. W. van Wingerden, and M. Verhaegen. Wind turbine load reduction by rejecting the periodic load disturbances. *Provisionally accepted for Wind Energy*, 2011.

**Table 1.1:** Scheme of the outline of this thesis

|           | problem                                         | papers                                             | methods                                                           | software tools                                                                               | proof of concept                                                                                                                         |
|-----------|-------------------------------------------------|----------------------------------------------------|-------------------------------------------------------------------|----------------------------------------------------------------------------------------------|------------------------------------------------------------------------------------------------------------------------------------------|
| Chapter 2 | Problem 1.1: Data-driven modelling              | Houtzager et al. (2010); Houtzager et al. (2009b)  | PBSID <sub>opt</sub>                                              | PBSIDToolbox (Houtzager, 2010)                                                               | simulations with ECN's SUSCON, TURBU, and PHATAS modelling software (and experiments with "smart" rotor test section in Chapter 3 and 4) |
| Chapter 3 | Problem 1.1: Data-driven modelling              | Houtzager et al. (2009a); Houtzager et al. (2011a) | RPBSID <sub>pm</sub> (recursive version of PBSID <sub>opt</sub> ) | PBSIDToolbox Houtzager (2010)                                                                | experiments with "smart" rotor test section                                                                                              |
| Chapter 4 | Problem 1.2: Rejection of periodic disturbances | Houtzager et al. (2011d); Houtzager et al. (2011b) | RC (using lifted model descriptions)                              | PBSIDToolbox (Houtzager, 2010); LRCToolbox (Houtzager, 2011b); SSSToolbox (Houtzager, 2011c) | experiments with "smart" rotor test section                                                                                              |
| Chapter 5 | Problem 1.2: Rejection of periodic disturbances | Houtzager et al. (2011c)                           | RC (with extension to periodic LTV)                               | LRCToolbox (Houtzager, 2011b); DISCON (Houtzager, 2011a)                                     | simulations with GH BLADED modelling software                                                                                            |

# Predictor-Based Subspace Identification for the Model-Based Control of Wind Turbine Systems

---

The chapter presents the optimized predictor-based subspace identification method and shows the effectiveness for the estimation of wind turbine models in closed loop. This identification technique does not require any controller related information, consequently the identified model becomes consistent no matter the wind turbine operates with or without controller in the loop. Since the wind turbine is not asymptotically stable in open-loop, only closed-loop experiments are supported in reality. This fact makes the proposed method very attractive for the wind power community. Further, a novel extension to the optimized predictor-based subspace identification method is proposed that relaxes the requirement that the past window has to be large for asymptotically consistent estimates. For this purpose, a finite description of the input-output relation is formulated that can be solved efficiently using the extended least squares recursion. The effectiveness and robustness of the proposed methods are discussed in a simulation study, where the algorithms are used on a data source based on controller-in-the-loop simulations of a typical modern wind turbine.

---

## 2.1 Introduction

FOR the wind energy community, model-based controller design becomes more and more important. Model-based controller synthesis necessitates a nominal description of the real plant. Nominal description of the plant can either be derived from physical principles or using measured data, respectively. Therefore, the latter is considered as a preliminary phase on the way towards a model-based controller design. System identification of wind turbines is not only important at the design of

a new wind turbine setup, but also when existing devices have to be re-identified in order to create a more up-to-date and accurate model than the existing one. In the specific case of wind turbine systems, the identification has to be performed in taking consideration the existing stabilizing controller as well. Further, the disturbances in the wind turbine are periodic in nature, therefore ideas similar to those in van Baars et al. (1993) (periodic basis functions) and in Kanev (2008) (Coleman transformation) needs to be applied for compensation.

Subspace IDentification (SID) methods are known to be computationally efficient methods to identify Linear Time-Invariant (LTI) state-space models from Multi-Input and Multi-Output (MIMO) measurements of a dynamic system. In Van Overschee and De Moor (1996), an unified methodology is suggested in which many of the SID methods fall, such as Canonical Variate Analysis (CVA) (Larimore, 1990), ORThogonal decomposition (ORT) (Katayama, 2005), N4SID (Van Overschee and De Moor, 1996), and MOESP (Verhaegen and Verdult, 2007) methods. These methods store input and output data in structured block Hankel matrices, such that it is possible to retrieve certain subspaces that are related to the system matrices. These methods are very successful for offline batchwise identification, because the key linear algebra steps are an RQ factorization, an SVD, and the solution of a linear least-squares problem, therefore the problem of forming a non-linear optimization is circumvented. Especially the use of an RQ factorization resulted in computationally efficient implementations (Verhaegen and Verdult, 2007). The disadvantage with these traditional SID methods is that they give biased results when the system to be modelled operates in closed loop, because the future inputs are correlated with the noise, due to the feedback controller. The traditional SID methods have been updated to give unbiased results in closed-loop operation, but these methods can only be applied if the controllers and/or the external signals are known in advance (Oku et al., 2006; Verhaegen, 1993), or by an iterative procedure to the optimal instrumental variables (Gibson and Mercère, 2006).

The first alternative to SID are the Prediction Error Methods (PEM) (Ljung, 1999). These methods use a pre-defined parametrization of the model, for example an Auto Regressive with eXogenous inputs (ARX) model structure, where the parameters are obtained by minimizing a quadratic cost function. PEM can provide asymptotical consistent estimates in closed loop if the model structure and order fits the dynamics of the true system (Van den Hof and Schrama, 1995), but the order is difficult to estimate in practice, the methods are computational expensive for MIMO systems, and the model structures other than ARX require the forming of a non-linear optimization problem. The second alternative to SID is the Observer/Kalman-filter IDentification (OKID) method (Juang and Pappa, 1985; Phan et al., 1992). This method obtains first Markov parameters, by estimating an Vector Auto-Regressive with eXogenous inputs (VARX) model, which is in the case of SISO measurements similar to a high-order ARX model. VARX models, with high order, can also provide asymptotically consistent estimates even on closed-loop data if there is sufficient excitation from an external signal or a controller of sufficiently high order, and that the system to be modelled or the feedback controller does not contain direct feedthrough (Chiuso and Picci, 2005). Then with the Markov parameters, a Hankel matrix is constructed. Finally, the system matrices of the state-space model are directly estimated from the Hankel matrix by using the Eigensystem Realization Algorithm (ERA) (also known as the Ho-Kalman

decomposition). To obtain a good estimate of the system matrices, a large number of Markov parameters are needed to construct a sufficiently large Hankel matrix, making OKID normally less efficient and accurate than SID in open loop.

Recently, a number of significant advances have been presented to identify LTI state-space models from measurements of dynamic systems operating in closed loop. The SSNEW (Ljung and McKelvey, 1996; Qin and Ljung, 2003), SSARX (Jansson, 2003), and Predictor-Based Subspace IDentification (PBSID) (Chiuso, 2007) methods are all extensively utilizing the Vector Auto-Regressive with eXogenous inputs (VARX) model parametrized by Markov parameters. The Markov parameters are used to construct a Toeplitz matrix instead, where from multiplication with past input and output data, and an SVD, an estimation of the state sequence can be obtained. With the state sequence, it is straightforward to recover the system matrices. The proposed algorithm is based on the optimized version of the Predictor-Based Subspace IDentification (PBSID) method, the so-called PBSID<sub>opt</sub> method (Chiuso, 2007). All these methods have a lot of common characteristics, and it is shown in Chiuso (2006, 2007); Chiuso and Picci (2005) that the asymptotic variance of most these methods can be a unified up to a particular weight. Furthermore, it was shown that the performance of PBSID<sub>opt</sub> compares favourably with PBSID and the other SID algorithms.

The main contributions of this chapter are twofold. The first contribution is the implementation of the PBSID<sub>opt</sub> to identify LTI models from data measured in open or closed loop. A slightly modified description is used then in the original work of Chiuso (2007), to improve the algorithm for practice. The second contribution is a novel improvement on the first part of the PBSID<sub>opt</sub> method; the estimation of the Markov parameters. Inspired by Phan et al. (1995), an Vector Auto-Regressive Moving Average with eXogenous inputs (VARMAX) model is utilized, which enables consistent estimates of the model parameters of the finite-order input-output model. This relaxes the asymptotic consistency results in the case of VARX models, resulting in a very large past window in practice. A large past window has a number of disadvantages. First is that the increasing number of parameters to be fit gives an increase in the variance of the estimate. Second is that the matrix with stacked input and output data becomes more ill conditioned, because normally in practice the input signals do not persistently excite the system enough, which means that the estimate becomes very sensitive to perturbations on the measurement data. Third is that the memory usage increases considerably for increasing past window. The estimation of a VARMAX model comes at the cost of a non-linear problem; however most of the times it can be estimated efficiently. Instead of the residual whitening operations in Phan et al. (1995), we propose the use of the Extended Least Squares (ELS) recursion described in Ljung (1977, 1999) to solve the estimation problem.

The outline of this chapter is as follows. In Section 2.2 the problem formulation, the assumptions made, and some notations are presented. In Section 2.3 the theoretical framework is presented for the VARX-based predictor-based subspace identification problem and a batchwise algorithm for LTI systems is presented. In Section 2.4, the theoretical framework is presented for the VARMAX-based predictor-based subspace identification problem and a batchwise algorithm for LTI systems is proposed. In Section 2.5, the effectiveness of the methods are shown by simulation studies with modern wind turbine models. Finally, the conclusions are presented.

## 2.2 System description, assumptions and notations

In this section the system description, the assumptions made, and some notations are presented.

### 2.2.1 System description and assumptions

The dynamics of the system  $S$  to be modelled can be written as the following state-space model:

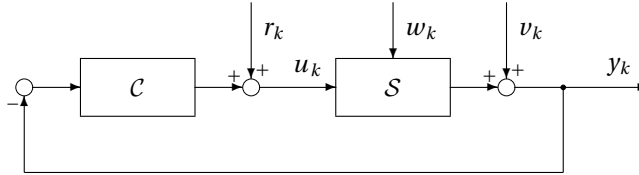
$$S \begin{cases} x_{k+1} &= Ax_k + Bu_k + w_k, \\ y_k &= Cx_k + Du_k + v_k, \end{cases} \quad (2.1)$$

where  $x_k \in \mathbb{R}^n$ ,  $u_k \in \mathbb{R}^r$ ,  $y_k \in \mathbb{R}^\ell$ ,  $w_k \in \mathbb{R}^n$ , and  $v_k \in \mathbb{R}^\ell$ , are the state, input, output, process noise, and measurement noise vectors. The state-space matrices  $A \in \mathbb{R}^{n \times n}$ ,  $B \in \mathbb{R}^{n \times r}$ ,  $C \in \mathbb{R}^{\ell \times n}$ , and  $D \in \mathbb{R}^{\ell \times r}$  are also called the state, input, output, and direct feedthrough matrices, respectively.

We consider the case that the system  $S$  to be modelled operates in closed loop, see Figure 2.1. In Assumption 2.1, we adopt mild conditions commonly used for the direct closed-loop state-space identification problem (Aling, 1990).

#### Assumption 2.1.

- The process noise  $w_k$  and measurement noise  $v_k$  are zero-mean white or coloured Gaussian sequences.
- The external excitation  $r_k$  is a zero-mean sequence uncorrelated with the process noise  $w_k$  and measurement noise  $v_k$ , and is sufficiently persistently exciting (Ljung, 1999).
- The feedback loop contains at least one sample delay, i.e., the system  $S$  or the controller  $C$  has no direct feedthrough.



**Figure 2.1:** Block diagram of a system operating in closed loop

### 2.2.2 Innovation form descriptions

Using Kalman filter theory, we can reformulate (2.1) to the innovation form (Ljung, 1999; Verhaegen and Verdult, 2007):

$$S_{\mathcal{I}} \begin{cases} \hat{x}_{k+1} &= A\hat{x}_k + Bu_k + Ke_k, \\ y_k &= C\hat{x}_k + Du_k + e_k, \end{cases} \quad (2.2)$$



where  $\hat{x}_k \in \mathbb{R}^n$  denotes the predicted state vector with  $E[x_k - \hat{x}_k] = 0$  and  $E[(x_k - \hat{x}_k)(x_k - \hat{x}_k)^T]$  minimal, and  $e_k \in \mathbb{R}^\ell$  denotes the white innovation sequence with  $E[e_k e_k^T] = R$ , and  $K \in \mathbb{R}^{n \times \ell}$  denotes the Kalman gain matrix. We can also rewrite the system  $\mathcal{S}_T$  in (2.2) in the so-called predictor form as:

$$\mathcal{S}_P \begin{cases} \hat{x}_{k+1} &= \tilde{A}\hat{x}_k + \tilde{B}u_k + Ky_k, \\ y_k &= C\hat{x}_k + Du_k + e_k, \end{cases} \quad (2.3)$$

with the matrix  $\tilde{A} = A - KC$  asymptotically stable, and the matrix  $\tilde{B} = B - KD$ . Similar as in Phan et al. (1995), we can also introduce another deadbeat observer matrix  $M$  that will create additional freedom for the optimizer later on. By considering a deadbeat gain matrix  $M$ , which is expected to exist as only an observable part of the system can be identified from the measurement data, we can rewrite (2.2) in the so-called deadbeat predictor form as:

$$\mathcal{S}_D \begin{cases} \hat{x}_{k+1} &= \bar{A}\hat{x}_k + \bar{B}u_k + My_k + \bar{K}e_k, \\ y_k &= C\hat{x}_k + Du_k + e_k, \end{cases} \quad (2.4)$$

with the matrix  $\bar{A} = A - MC$  asymptotically stable and nilpotent, i.e.  $\bar{A}^j = 0$  for  $j \geq n$ , the matrix  $\bar{B} = B - MD$ , and the matrix  $\bar{K} = K - M$ .

The retrieval of the system matrices (including the Kalman gain) is the goal of the identification procedure. We can only estimate the observable and controllable part (to both the deterministic and stochastic inputs) of the system  $\mathcal{S}$ , therefore we assume that this state-space description is a minimal realization. It is noted that for coloured noise sequences this includes the noise filter dynamics, and therefore the state vector can be larger than the order of the deterministic part only. Further, it is well-known that an invertible linear transformation of the state vector does not change the input-output behaviour of a state-space system. Therefore, we can only determine the system matrices up to an unknown similarity transformation  $T \in \mathbb{R}^{n \times n}$ :  $T^{-1}AT$ ,  $T^{-1}B$ ,  $T^{-1}K$ , and  $CT$ .

### 2.2.3 Lifted system description and notations

We define a past window denoted by  $p \in \mathbb{N}^+$  and a future window denoted by  $f \in \mathbb{N}^+$ , where  $n/\ell \leq f$ . These windows are used to define the following stacked vectors:

$$\bar{y}_{k-p,p} = \begin{bmatrix} y_{k-p} \\ y_{k-p+1} \\ \vdots \\ y_{k-1} \end{bmatrix}, \quad \bar{y}_{k,f} = \begin{bmatrix} y_k \\ y_{k+1} \\ \vdots \\ y_{k+f-1} \end{bmatrix}.$$

The stacked vectors  $\bar{u}_{k-p,p}$ ,  $\bar{u}_{k,f}$ ,  $\bar{e}_{k-p,p}$ , and  $\bar{e}_{k,f}$  are defined in a similar way. For the batchwise case, i.e., the case when a batch of  $N$  data is available, we can also define the stacked matrix  $Y$ :

$$Y = [y_p, \quad \dots, \quad y_{N-1}]. \quad (2.5)$$

The stacked matrices  $U$ ,  $X$  are defined in a similar way. Further, we can define the stacked matrix  $\bar{Y}_p$ :

$$\bar{Y}_p = [\bar{y}_{0,p}, \quad \dots, \quad \bar{y}_{N-p,p}]. \quad (2.6)$$

Again, we can obtain in a similar way the stacked vectors  $\bar{U}_p$ .

Using the definition of the stacked vectors, we can “lift” the system in (2.2) to the following lifted description:

$$\begin{cases} \hat{x}_k &= A^p \hat{x}_{k-p} + \mathcal{L} \bar{u}_{k-p,p} + \mathcal{K} \bar{e}_{k-p,p}, \\ \bar{y}_{k,f} &= \Gamma \hat{x}_k + \mathcal{G} \bar{u}_{k,f} + \mathcal{H} \bar{e}_{k,f}, \end{cases} \quad (2.7)$$

where  $\mathcal{L} \in \mathbb{R}^{n \times pr}$  and  $\mathcal{K} \in \mathbb{R}^{n \times p\ell}$  are the extended controllability matrices, and are given by:

$$\mathcal{L} = [A^{p-1}B \quad \dots \quad AB \quad B], \quad \mathcal{K} = [A^{p-1}K \quad \dots \quad AK \quad K],$$

and  $\mathcal{G} \in \mathbb{R}^{f\ell \times fr}$  and  $\mathcal{H} \in \mathbb{R}^{f\ell \times f\ell}$  are the impulse matrices with a lower block triangular structure,  $\Gamma \in \mathbb{R}^{f\ell \times n}$  is the extended observability matrix, and are given by:

$$\Gamma = \begin{bmatrix} C \\ CA \\ \vdots \\ CA^{f-1} \end{bmatrix}, \quad \mathcal{G} = \begin{bmatrix} D & 0 & \dots & 0 \\ CB & D & \ddots & \vdots \\ \vdots & \ddots & \ddots & 0 \\ CA^{f-2}B & CA^{f-3}B & \dots & D \end{bmatrix},$$

$$\mathcal{H} = \begin{bmatrix} I & 0 & \dots & 0 \\ CK & I & \ddots & \vdots \\ \vdots & \ddots & \ddots & 0 \\ CA^{f-2}K & CA^{f-3}K & \dots & I \end{bmatrix}.$$

Note that  $I$  is used to represent an identity matrix. Similarly, the system in (2.3) can be “lifted” to the following description:

$$\begin{cases} \hat{x}_k &= \tilde{A}^p \hat{x}_{k-p} + \tilde{\mathcal{L}} \bar{u}_{k-p,p} + \tilde{\mathcal{K}} \bar{y}_{k-p,p}, \\ \bar{y}_{k,f} &= \tilde{\Gamma} \hat{x}_k + \tilde{\mathcal{G}} \bar{u}_{k,f} + (I - \tilde{\mathcal{H}}) \bar{y}_{k,f} + \bar{e}_{k,f}, \end{cases} \quad (2.8)$$

where the lifted matrices  $\tilde{\mathcal{L}} \in \mathbb{R}^{n \times pr}$ ,  $\tilde{\mathcal{K}} \in \mathbb{R}^{n \times p\ell}$ ,  $\tilde{\Gamma} \in \mathbb{R}^{f\ell \times n}$ ,  $\tilde{\mathcal{G}} \in \mathbb{R}^{f\ell \times fr}$ , and  $\tilde{\mathcal{H}} \in \mathbb{R}^{f\ell \times f\ell}$  are defined in an almost similar way without  $(\cdot)$ . For the matrices  $\tilde{\mathcal{L}}$ ,  $\tilde{\mathcal{K}}$ ,  $\tilde{\Gamma}$ , and  $\tilde{\mathcal{G}}$  the only difference is that the generator matrices  $A$  and  $B$  are replaced with  $\tilde{A}$  and  $\tilde{B}$ . The matrix  $\tilde{\mathcal{H}}$  is defined as:

$$\tilde{\mathcal{H}} = \begin{bmatrix} I & 0 & \dots & 0 \\ -CK & I & \ddots & \vdots \\ \vdots & \ddots & \ddots & 0 \\ -C\tilde{A}^{f-2}K & -C\tilde{A}^{f-3}K & \dots & I \end{bmatrix}. \quad (2.9)$$

More interesting is the relation between the innovation and predictor form given in Lemma 2.1.

**Lemma 2.1.** *It can be shown that  $\tilde{\mathcal{H}}$  is invertible, and the following transformation holds:*

$$\begin{bmatrix} \Gamma & \mathcal{G} & \mathcal{H} & I \end{bmatrix} = \tilde{\mathcal{H}}^{-1} \begin{bmatrix} \tilde{\Gamma} & \tilde{\mathcal{G}} & I & \tilde{\mathcal{H}} \end{bmatrix}. \quad (2.10)$$

*Proof.* Proof follows through straightforward multiplications of the Markov parameters (matrix block elements).  $\square$

Similarly, the system in (2.4) can be “lifted” to the following description:

$$\begin{cases} \hat{x}_k &= \bar{A}^p \hat{x}_{k-p} + \bar{\mathcal{L}} \bar{u}_{k-p,p} + \bar{\mathcal{P}} \bar{y}_{k-p,p} + \bar{\mathcal{K}} \bar{e}_{k-p,p}, \\ \bar{y}_{k,f} &= \bar{\Gamma} \hat{x}_k + \bar{\mathcal{G}} \bar{u}_{k,f} + (I - \bar{\mathcal{F}}) \bar{y}_{k,f} + \bar{\mathcal{H}} \bar{e}_{k,f}, \end{cases} \quad (2.11)$$

where the lifted matrices  $\bar{\mathcal{L}} \in \mathbb{R}^{n \times pr}$ ,  $\bar{\mathcal{P}} \in \mathbb{R}^{n \times pl}$ ,  $\bar{\mathcal{K}} \in \mathbb{R}^{n \times pl}$ ,  $\bar{\Gamma} \in \mathbb{R}^{f\ell \times n}$ ,  $\bar{\mathcal{G}} \in \mathbb{R}^{f\ell \times fr}$ ,  $\bar{\mathcal{F}} \in \mathbb{R}^{f\ell \times f\ell}$ , and  $\bar{\mathcal{H}} \in \mathbb{R}^{f\ell \times f\ell}$  are defined in an almost similar way without  $(\cdot)$ . For the matrices  $\bar{\mathcal{L}}$ ,  $\bar{\mathcal{K}}$ ,  $\bar{\Gamma}$ , and  $\bar{\mathcal{G}}$  the only difference is that the generator matrices  $A$ ,  $B$ , and  $K$  are replaced with  $\bar{A}$ ,  $\bar{B}$ , and  $\bar{K}$ , respectively. The matrix  $\bar{\mathcal{P}}$ ,  $\bar{\mathcal{F}}$ , and  $\bar{\mathcal{H}}$  are defined as:

$$\begin{aligned} \bar{\mathcal{P}} &= \begin{bmatrix} \bar{A}^{p-1}M & \cdots & \bar{A}M & M \end{bmatrix}, \\ \bar{\mathcal{F}} &= \begin{bmatrix} I & 0 & \cdots & 0 \\ -CM & I & \ddots & \vdots \\ \vdots & \ddots & \ddots & 0 \\ -C\bar{A}^{f-2}M & -C\bar{A}^{f-3}M & \cdots & I \end{bmatrix}. \end{aligned} \quad (2.12)$$

$$\bar{\mathcal{H}} = \begin{bmatrix} I & 0 & \cdots & 0 \\ C\bar{K} & I & \ddots & \vdots \\ \vdots & \ddots & \ddots & 0 \\ C\bar{A}^{f-2}\bar{K} & C\bar{A}^{f-3}\bar{K} & \cdots & I \end{bmatrix}. \quad (2.13)$$

More interesting is the relation between the innovation, predictor, dead predictor form given in Lemma 2.2 and Lemma 2.3.

**Lemma 2.2.** *It can be shown that  $\bar{\mathcal{F}}$  is invertible, and the following transformation holds:*

$$\begin{bmatrix} \Gamma & \mathcal{G} & \mathcal{H} & I \end{bmatrix} = \bar{\mathcal{F}}^{-1} \begin{bmatrix} \tilde{\Gamma} & \tilde{\mathcal{G}} & \tilde{\mathcal{H}} & \tilde{\mathcal{F}} \end{bmatrix}. \quad (2.14)$$

*Proof.* Proof follows through straightforward multiplications of the Markov parameters (matrix block elements).  $\square$

**Lemma 2.3.** *It can be shown that  $\tilde{\mathcal{H}}$  is invertible, and the following transformation holds:*

$$\begin{bmatrix} \tilde{\Gamma} & \tilde{\mathcal{G}} & I & \tilde{\mathcal{H}} \end{bmatrix} = \tilde{\mathcal{H}}^{-1} \begin{bmatrix} \tilde{\Gamma} & \tilde{\mathcal{G}} & \tilde{\mathcal{H}} & \tilde{\mathcal{F}} \end{bmatrix}. \quad (2.15)$$

*Proof.* Proof follows through straightforward multiplications of the Markov parameters (matrix block elements).  $\square$

## 2.3 Batchwise identification with VARX-based PBSID<sub>opt</sub>

In this section, the theoretical framework for the VARX-based predictor-based subspace identification method to identify LTI state-space models from measured data is presented. The goal is to obtain the predicted state sequence, because if this is obtained the state-space system matrices can easily be estimated by solving two linear problems (Van Overschee and De Moor, 1996; Verhaegen and Verdult, 2007).

### 2.3.1 VARX predictor with finite past window

We define the one-step-ahead Vector Auto-Regressive with eXogenous inputs (VARX) predictor as:

$$\hat{y}_{k|k-1} = \sum_{i=0}^p \tilde{\Xi}^{(u_{k-i})} u_{k-i} + \sum_{i=1}^p \tilde{\Xi}^{(y_{k-i})} y_{k-i}, \quad (2.16)$$

where  $\hat{y}_{k|k-1}$  is the predicted output for time instant  $k$  using the inputs of time instants  $k, \dots, k-p$  and using the outputs of time instants  $k-1, \dots, k-p$ . Further,  $\tilde{\Xi} \in \mathbb{R}^{\ell \times p(r+\ell)+r}$  is the set of VARX parameters to be estimated:

$$\tilde{\Xi} \triangleq [\tilde{\Xi}^{(u_{k-p})} \quad \dots \quad \tilde{\Xi}^{(u_k)} \quad \tilde{\Xi}^{(y_{k-p})} \quad \dots \quad \tilde{\Xi}^{(y_{k-1})}]. \quad (2.17)$$

In Chiuso and Picci (2005) it was shown that estimated VARX predictors can provide consistently estimated Markov parameters of the system  $\mathcal{S}_{\mathcal{P}}$  in (2.3) even with closed-loop data. To see this implicit relation between the Markov parameters and the VARX predictor, we first need to introduce an approximation for the state. The contribution of the initial state  $\hat{x}_{k-p}$  in (2.8) can be made arbitrarily small by making  $p$  large, because the estimated system  $\mathcal{S}_{\mathcal{P}}$  in (2.3) is asymptotically stable by definition. It is noted that in the case of coloured noise, the Kalman predictor is asymptotically stable when the poles of the stochastic part which are not shared with the deterministic part are asymptotically stable and zeros of the stochastic part are not on the unit disk (Anderson and Moor, 2005). We assume that the past window is chosen large enough such that the contribution of the initial state can be neglected, in other words the matrix  $\tilde{A}$  is nilpotent.

**Assumption 2.2** (Nilpotency). *The transition matrix is deadbeat with degree  $p$ , i.e., the matrix  $\tilde{A}^j = 0$  for all  $j \geq p$ , see Chiuso and Picci (2005).*

In a number of closed-loop SID methods it is well known to make this approximation, see Chiuso (2007); Jansson (2003). Now the following implicit relation is proposed between the Markov parameters and the VARX predictor.

**Proposition 2.1.** *Under Assumptions 2.1 and 2.2, consider the linear estimation of (2.17) asymptotically with an infinite data sequence ( $N \rightarrow \infty$ ), then the VARX parameters in (2.17) are related to the Markov parameters in (2.8), and described by:*

$$\tilde{\Xi}^{(u_{k-i})} = \begin{cases} D, & \text{if } i = 0 \\ C\tilde{A}^{i-1}\tilde{B}, & \text{if } i > 0 \end{cases}, \quad \tilde{\Xi}^{(y_{k-i})} = C\tilde{A}^{i-1}K.$$

*Proof.* See Appendix A and Chiuso and Picci (2005).  $\square$

It is noted that there are two cases for the VARX predictor where a direct implicit relation for a finite past window exists without considering an approximation. The first case is if the system  $\mathcal{S}_{\mathcal{I}}$  in (2.2) is of the ARX type, which is not common in practical situations, then the transition matrix becomes nilpotent for all  $p \geq n$ . The second case is if the system  $\mathcal{S}$  in (2.1) has no process noise, i.e.,  $w_k = 0$ , then the feedback gain  $K$  in the system  $\mathcal{S}_{\mathcal{P}}$  in (2.3) becomes a deadbeat gain  $M$ , which also has a nilpotent transition matrix if  $p \geq n$ .

### 2.3.2 The relation between the VARX predictor and the state

To extract the predicted state sequence using the estimated Markov parameters, we introduce in this procedure again the approximation for the state equation in (2.8). With Assumption 2.2, the state  $x_k$  is given by:

$$\hat{x}_k = \tilde{\mathcal{L}} \tilde{u}_{k-p,p} + \tilde{\mathcal{K}} \tilde{y}_{k-p,p}. \quad (2.18)$$

With the approximation given in (2.18), we can rewrite the system in (2.8) as:

$$\tilde{y}_{k,f} = \tilde{\Gamma} \tilde{\mathcal{L}} \tilde{u}_{k-p,p} + \tilde{\Gamma} \tilde{\mathcal{K}} \tilde{y}_{k-p,p} + \tilde{\mathcal{G}} \tilde{u}_{k,f} + (I - \tilde{\mathcal{H}}) \tilde{y}_{k,f} + \tilde{e}_{k,f}, \quad (2.19)$$

where  $\tilde{\Gamma} \tilde{\mathcal{L}} \in \mathbb{R}^{f\ell \times pr}$  and  $\tilde{\Gamma} \tilde{\mathcal{K}} \in \mathbb{R}^{f\ell \times p\ell}$  are the products between the extended observability and the extended controllability matrices, and are given by:

$$\begin{aligned} \tilde{\Gamma} \tilde{\mathcal{L}} &= \begin{bmatrix} \tilde{\Xi}^{(u_{k-p})} & \tilde{\Xi}^{(u_{k-p+1})} & \dots & \tilde{\Xi}^{(u_{k-p+f-1})} & \dots & \tilde{\Xi}^{(u_{k-1})} \\ 0 & \tilde{\Xi}^{(u_{k-p})} & \dots & \tilde{\Xi}^{(u_{k-p+f-2})} & \dots & \tilde{\Xi}^{(u_{k-2})} \\ \vdots & \ddots & \ddots & \vdots & \ddots & \vdots \\ 0 & \dots & 0 & \tilde{\Xi}^{(u_{k-p})} & \dots & \tilde{\Xi}^{(u_{k-f})} \end{bmatrix}, \\ \tilde{\Gamma} \tilde{\mathcal{K}} &= \begin{bmatrix} \tilde{\Xi}^{(y_{k-p})} & \tilde{\Xi}^{(y_{k-p+1})} & \dots & \tilde{\Xi}^{(y_{k-p+f-1})} & \dots & \tilde{\Xi}^{(y_{k-1})} \\ 0 & \tilde{\Xi}^{(y_{k-p})} & \dots & \tilde{\Xi}^{(y_{k-p+f-2})} & \dots & \tilde{\Xi}^{(y_{k-2})} \\ \vdots & \ddots & \ddots & \vdots & \ddots & \vdots \\ 0 & \dots & 0 & \tilde{\Xi}^{(y_{k-p})} & \dots & \tilde{\Xi}^{(y_{k-f})} \end{bmatrix}. \end{aligned} \quad (2.20)$$

These are upper block triangular matrices, because the introduced zeros follow from Assumption 2.2. When  $f \leq p$ , this implies that in the asymptotic case the approximation of the matrices  $\tilde{\Gamma} \tilde{\mathcal{L}}$  and  $\tilde{\Gamma} \tilde{\mathcal{K}}$  can be fully constructed by the Markov parameters  $\tilde{\Xi}$  obtained from the least squares problem given in (2.16). Observe that the product between the state and the extended observability matrix is given by:

$$\tilde{\Gamma} \hat{x}_k = \tilde{\Gamma} \tilde{\mathcal{L}} \tilde{u}_{k-p,p} + \tilde{\Gamma} \tilde{\mathcal{K}} \tilde{y}_{k-p,p}, \quad (2.21)$$

and this observation is the key idea behind the PBSID<sub>opt</sub> method. To summarize, after the construction of the matrices  $\tilde{\Gamma} \tilde{\mathcal{L}}$  and  $\tilde{\Gamma} \tilde{\mathcal{K}}$ , we obtain a product of the extended observability matrix and the state sequence. Both of them can be estimated by solving a low-rank approximation problem. Generally, the relation in (2.21) can be pre-multiplied with any pre-defined weight matrix  $W \in \mathbb{R}^{f\ell \times f\ell}$  from the left side. Applying a weight leads to a weighted version of the PBSID<sub>opt</sub> method and also to a weighted version of the SSNEW method in Ljung and McKelvey (1996), see also Chiuso (2007).

### 2.3.3 The VARX-based PBSID<sub>opt</sub> algorithm

In this subsection the VARX-based predictor-based subspace identification method for LTI systems operating in open or closed loop is presented. The batchwise system identification problem can be formulated as:

**Problem Description 2.1** (Batchwise system identification).

*Given the input sequence  $u_k$ , output sequence  $y_k$  over a time  $k = \{0, \dots, N-1\}$  find, if they exist, system matrices  $A$ ,  $B$ ,  $C$ ,  $D$ , and  $K$  up to an unknown similarity transformation  $T$ .*

Solving the problem with the theory of the previous subsections, results in the so-called “optimized” predictor-based subspace identification (PBSID<sub>opt</sub>) method which was originally presented in Chiuso (2007). The algorithm can be divided in three steps, where in each step linear problems need to be solved. A slightly modified description is given in Algorithm 2.1. First, the innovation sequence needed for the estimation of the system matrices in Step 3 is estimated using the relation  $\hat{E} = Y - C\hat{X} - DU$ . In the original recipe the estimated VARX model was used using the relation  $\hat{E} = Y - \hat{\Xi}\Psi$ , where  $\Psi = [\bar{U}_p^T \quad U^T \quad \bar{Y}_p^T]^T$ . Second, the weighting matrix  $W = I \otimes R^{1/2}$  is not applied. This weight was originally applied to get similar weighting as the CCA subspace algorithm (Chiuso, 2010). The experience with multiple systems in practice (especially if the innovation is not perfectly white) is that both suggestions are not beneficial for the quality of the estimate, although they work well on small simulation models. The last modification is that a guaranteed stabilizing Kalman matrix  $K$  is estimated using the Riccati equation, as described in Verhaegen and Verdult (2007). For validation purposes, it is important to have a stable Kalman predictor of the identified model.

In Assumption 2.2 an approximation is made which requires the past window to be chosen as large as possible. However, in practical experiments it is most of the time needed to design the excitation signal such that it does not exceed the load specifications and ensures that the system to be identified operates around a particular operation point. In this case, especially with a low-order controller in the feedback loop, the matrix  $\Psi$  becomes ill conditioned, because the input signals do not persistently excite the system. Apart from just selecting a lower past window size, a number of other solutions exist to get a well-conditioned matrix  $\Psi$  for the least squares estimation. The first possible solution is generating more persistently exciting control signals by changing the design properties of the controller. For example automatically generated constraints can be added to a Model Predictive Control (MPC) design problem when the matrix  $\Psi$  becomes ill conditioned, see Dong (2009). The second possible solution is utilizing a VARMAX model, for which the input-output description relaxes the need for a very large past window needed due to the formulation of a deadbeat predictor, see Section 2.4. The third and well-used solution to the ill-posed least squares problem is the inclusion of a regularization quantity. In the batchwise case, Tikhonov regularization from (Tikhonov and Arsenin, 1977) can be included by replacing the least squares estimation of the Markov parameters in Algorithm 2.1 with:

$$\hat{\Xi} = Y\Psi^T(\Psi\Psi^T + \mu I)^{-1}, \quad (2.22)$$

where  $\mu$  is the regularization parameter. The Tikhonov regularization parameter  $\mu$  can be determined from a batch of data obtained around an operation point using the L-curve or Generalized Cross Validation (GCV) criterion, see Hansen (1992).

**Algorithm 2.1** (The VARX-based PBSID<sub>opt</sub> algorithm).

**input**  $u, y, p, f, W$ .

**require**  $f \leq p$

**Step 1:** The estimation of the Markov parameters

Construct the matrices  $Y, U, \bar{U}_p$ , and  $\bar{Y}_p$ , see (2.5) and (2.6).

Solve the linear problem in (2.16) using least squares:

$$\hat{\Xi} = Y\Psi^+, \text{ where } \Psi = \begin{bmatrix} \bar{U}_p^T & U^T & \bar{Y}_p^T \end{bmatrix}^T.$$

**Step 2:** The estimation of the state sequence

Construct the matrices  $\widehat{\Gamma\mathcal{L}}$  and  $\widehat{\Gamma\mathcal{K}}$  in (2.20).

Calculate the sequence  $\widehat{\Gamma X} = W(\widehat{\Gamma\mathcal{L}}\bar{U}_p + \widehat{\Gamma\mathcal{K}}\bar{Y}_p)$ .

Solve the low-rank approximation problem using the SVD:

$$\widehat{\Gamma X} = \begin{bmatrix} U_n & U_\perp \end{bmatrix} \begin{bmatrix} \Sigma_n & 0 \\ 0 & \Sigma \end{bmatrix} \begin{bmatrix} V_n \\ V_\perp \end{bmatrix},$$

where the system order  $n$  is determined by detecting a gap that separates the  $n$  largest singular values.

Calculate the state sequence:  $\hat{X} = \Sigma_n^{1/2} V_n$

**Step 3:** The estimation of the system matrices

Solve linear problem using least squares<sup>a</sup>:

$$\begin{bmatrix} C & D \end{bmatrix} = Y(:, 1:N-p-1) \begin{bmatrix} \hat{X}(:, 1:N-p-1) \\ U(:, 1:N-p-1) \end{bmatrix}^+.$$

Compute the innovation sequence  $\hat{E} = Y - C\hat{X} - DU$ .

Solve linear problem using least squares<sup>a</sup>:

$$\begin{bmatrix} A & B & K \end{bmatrix} = \hat{X}(:, 2:N-p) \begin{bmatrix} \hat{X}(:, 1:N-p-1) \\ U(:, 1:N-p-1) \\ \hat{E}(:, 1:N-p-1) \end{bmatrix}^+.$$

**Step 3a:** Calculate the stabilizing Kalman gain

Compute the residuals<sup>a</sup>:

$$\begin{bmatrix} \hat{W} \\ \hat{V} \end{bmatrix} = \begin{bmatrix} \hat{X}(:, 2:N-p) \\ Y(:, 1:N-p-1) \end{bmatrix} - \begin{bmatrix} A & B \\ C & D \end{bmatrix} \begin{bmatrix} \hat{X}(:, 1:N-p-1) \\ U(:, 1:N-p-1) \end{bmatrix}.$$

Estimate the covariance matrices:

$$\begin{bmatrix} Q & S \\ S^T & R \end{bmatrix} = \frac{1}{N-p} \begin{bmatrix} \hat{W} \\ \hat{V} \end{bmatrix} \begin{bmatrix} \hat{W}^T & \hat{V}^T \end{bmatrix}.$$

Calculate the solution of the Riccati equation:

$$P = APA^T - (S + APC^T)(R + CPC^T)^{-1}(S + APC^T)^T + Q.$$

Estimate a stabilizing Kalman gain:

$$K = (S + APC^T)(R + CPC^T)^{-1}.$$

<sup>a</sup>For simplicity MATLAB<sup>TM</sup> notation is used.

## 2.4 Batchwise identification with VARMAX-based PBSID<sub>opt</sub>

In this section, the theoretical framework for the VARMAX-based predictor-based subspace identification method to identify LTI state-space models from measured data is presented. The estimation of a VARMAX model comes at the cost of a non-linear problem; however most of the times it can be estimated efficiently.

### 2.4.1 VARMAX predictor with finite past window

We define the one-step-ahead Vector Auto-Regressive Moving Average with eXogenous inputs (VARMAX) predictor as:

$$\hat{y}_{k|k-1} = \sum_{i=0}^p \tilde{\Xi}^{(u_{k-i})} u_{k-i} + \sum_{i=1}^p \tilde{\Xi}^{(y_{k-i})} y_{k-i} + \sum_{i=1}^p \tilde{\Xi}^{(e_{k-i})} \hat{e}_{k-i}, \quad (2.23)$$

where  $\hat{y}_{k|k-1}$  is the predicted output for time instant  $k$  using the inputs of time instants  $k, \dots, k-p$  and using the outputs and estimated innovation of time instants  $k-1, \dots, k-p$ . By substituting  $\hat{e}_{k-i} = y_{k-i} - \hat{y}_{k-i|k-i-1}$  in the former gives an expression which resembles the well-known ARMAX model structure as:

$$\sum_{i=0}^p \tilde{\Xi}^{(e_{k-i})} \hat{y}_{k-i|k-i-1} = \sum_{i=0}^p \tilde{\Xi}^{(u_{k-i})} u_{k-i} + \sum_{i=1}^p \left( \tilde{\Xi}^{(e_{k-i})} + \tilde{\Xi}^{(y_{k-i})} \right) y_{k-i}, \quad (2.24)$$

with  $\tilde{\Xi}^{(e_k)} = I$ , and  $\tilde{\Xi} \in \mathbb{R}^{\ell \times p(r+2\ell)+r}$  is the set of VARMAX parameters to be estimated:

$$\tilde{\Xi} \triangleq \begin{bmatrix} \tilde{\Xi}^{(u_{k-p})} & \dots & \tilde{\Xi}^{(u_k)} & \tilde{\Xi}^{(y_{k-p})} & \dots & \tilde{\Xi}^{(y_{k-1})} & \tilde{\Xi}^{(e_{k-p})} & \dots & \tilde{\Xi}^{(e_{k-1})} \end{bmatrix}. \quad (2.25)$$

Although the one-step-ahead predictor does not have the property that it is a linear function in the Markov parameters, the computation of the solution can still be done efficiently using the extended least squares recursion, see also Subsection 2.4.3.

In Subsection 2.3.1, an approximation for the state was introduced to get an implicit relation between the Markov parameters and the VARX predictor. We had to assume that the past window is chosen large enough such that the contribution of the initial state can be neglected, in other words the matrix  $\tilde{A}$  is nilpotent. In the model description in (2.4), an additional freedom in the form of a deadbeat observer matrix  $M$  was introduced on purpose to get the nilpotent or deadbeat property during the estimation VARMAX parameters. From pole-placement theory, for example by placing all the eigenvalues of the matrix  $\tilde{A}$  in the origin, a deadbeat matrix  $M$  of degree  $p \geq n$ , i.e.  $\tilde{A}^p = 0$ , is known to exist if the system is observable. As only an observable part of the system can be identified from the measurement data, this is expected to exist. With this reasoning, a very large past window is not required and a direct implicit relation between the Markov parameters and the VARMAX predictor can be proposed.

**Proposition 2.2.** *Under Assumptions 2.1, consider the linear estimation of (2.25) asymptotically with an infinite data sequence ( $N \rightarrow \infty$ ) and  $p \geq n$ , then the VARMAX parameters in (2.25) are related to the Markov parameters in (2.11), and described by:*

$$\tilde{\Xi}^{(u_{k-i})} = \begin{cases} D, & \text{if } i = 0 \\ C\tilde{A}^{i-1}\tilde{B}, & \text{if } i > 0 \end{cases}, \quad \tilde{\Xi}^{(y_{k-i})} = C\tilde{A}^{i-1}M, \quad \tilde{\Xi}^{(e_{k-i})} = C\tilde{A}^{i-1}\tilde{K}.$$



*Proof.* See Appendix A.  $\square$

If the system  $\mathcal{S}$  in (2.1) has no process noise, i.e.  $w_k = 0$ , then the feedback gain  $K = 0$  in the system  $\mathcal{S}_{\mathcal{D}}$  in (2.4), and basically becomes the system  $\mathcal{S}_{\mathcal{P}}$  in (2.4) with  $K$  replaced by  $M$ , which can be related to a VARX predictor with finite past window.

### 2.4.2 The relation between the VARMAX predictor and the state

Almost similar as in Subsection 2.3.2, the predicted state sequence can be estimated using the estimated Markov parameters and the estimated innovation sequence. With  $p \geq n$ , the state  $x_k$  in (2.11) is given by:

$$\hat{x}_k = \tilde{L}\tilde{u}_{k-p,p} + \tilde{P}\tilde{y}_{k-p,p} + \tilde{K}\tilde{e}_{k-p,p}. \quad (2.26)$$

With the deadbeat property, we can also rewrite the system in (2.11) as:

$$\tilde{y}_{k,f} = \tilde{\Gamma}\tilde{L}\tilde{u}_{k-p,p} + \tilde{\Gamma}\tilde{P}\tilde{y}_{k-p,p} + \tilde{\Gamma}\tilde{K}\tilde{e}_{k-p,p} + \tilde{G}\tilde{u}_{k,f} + (I - \tilde{F})\tilde{y}_{k,f} + \tilde{H}\tilde{e}_{k,f}, \quad (2.27)$$

where  $\tilde{\Gamma}\tilde{L} \in \mathbb{R}^{f\ell \times pr}$ ,  $\tilde{\Gamma}\tilde{P} \in \mathbb{R}^{f\ell \times p\ell}$  and  $\tilde{\Gamma}\tilde{K} \in \mathbb{R}^{f\ell \times p\ell}$  are the products between the extended observability and the extended controllability matrices, and are given by:

$$\begin{aligned} \tilde{\Gamma}\tilde{L} &= \begin{bmatrix} \tilde{\Xi}^{(u_{k-p})} & \tilde{\Xi}^{(u_{k-p+1})} & \dots & \tilde{\Xi}^{(u_{k-p+f-1})} & \dots & \tilde{\Xi}^{(u_{k-1})} \\ 0 & \tilde{\Xi}^{(u_{k-p})} & \dots & \tilde{\Xi}^{(u_{k-p+f-2})} & \dots & \tilde{\Xi}^{(u_{k-2})} \\ \vdots & \ddots & \ddots & \vdots & \ddots & \vdots \\ 0 & \dots & 0 & \tilde{\Xi}^{(u_{k-p})} & \dots & \tilde{\Xi}^{(u_{k-f})} \end{bmatrix}, \\ \tilde{\Gamma}\tilde{P} &= \begin{bmatrix} \tilde{\Xi}^{(y_{k-p})} & \tilde{\Xi}^{(y_{k-p+1})} & \dots & \tilde{\Xi}^{(y_{k-p+f-1})} & \dots & \tilde{\Xi}^{(y_{k-1})} \\ 0 & \tilde{\Xi}^{(y_{k-p})} & \dots & \tilde{\Xi}^{(y_{k-p+f-2})} & \dots & \tilde{\Xi}^{(y_{k-2})} \\ \vdots & \ddots & \ddots & \vdots & \ddots & \vdots \\ 0 & \dots & 0 & \tilde{\Xi}^{(y_{k-p})} & \dots & \tilde{\Xi}^{(y_{k-f})} \end{bmatrix}, \\ \tilde{\Gamma}\tilde{K} &= \begin{bmatrix} \tilde{\Xi}^{(e_{k-p})} & \tilde{\Xi}^{(e_{k-p+1})} & \dots & \tilde{\Xi}^{(e_{k-p+f-1})} & \dots & \tilde{\Xi}^{(e_{k-1})} \\ 0 & \tilde{\Xi}^{(e_{k-p})} & \dots & \tilde{\Xi}^{(e_{k-p+f-2})} & \dots & \tilde{\Xi}^{(e_{k-2})} \\ \vdots & \ddots & \ddots & \vdots & \ddots & \vdots \\ 0 & \dots & 0 & \tilde{\Xi}^{(e_{k-p})} & \dots & \tilde{\Xi}^{(e_{k-f})} \end{bmatrix}. \end{aligned} \quad (2.28)$$

These are upper block triangular matrices, because the introduced zeros follow from the deadbeat property. When  $f \leq p$ , this implies that the matrices  $\tilde{\Gamma}\tilde{L}$ ,  $\tilde{\Gamma}\tilde{P}$  and  $\tilde{\Gamma}\tilde{K}$  can be fully constructed by the Markov parameters  $\tilde{\Xi}$  obtained from the model estimation problem given in (2.23). Observe that the product between the state and the extended observability matrix is given by:

$$\tilde{\Gamma}\hat{x}_k = \tilde{\Gamma}\tilde{L}\tilde{u}_{k-p,p} + \tilde{\Gamma}\tilde{P}\tilde{y}_{k-p,p} + \tilde{\Gamma}\tilde{K}\tilde{e}_{k-p,p}. \quad (2.29)$$

To summarize, after the construction of the matrices  $\tilde{\Gamma}\tilde{L}$ ,  $\tilde{\Gamma}\tilde{P}$  and  $\tilde{\Gamma}\tilde{K}$ , we obtain a product of the extended observability matrix and the state sequence. Both of them can be estimated by solving a low-rank approximation problem. Generally, the relation in (2.29) can also be pre-multiplied with any pre-defined weight matrix  $W \in \mathbb{R}^{f\ell \times f\ell}$  from the left side.

### 2.4.3 The VARMAX-based PBSID<sub>opt</sub> algorithm

In this subsection the VARMAX-based predictor-based subspace identification method for LTI systems operating in open or closed loop is proposed. The batchwise system identification problem can be formulated similar as in Problem Definition 2.1 stated in Section 2.3.3. Solving the problem with the theory of the previous subsections results in the so-called VARMAX-based PBSID<sub>opt</sub> in Algorithm 2.3. The algorithm can again be divided in three steps, where in the first step a non-linear problem needs to be solved instead.

The VARMAX model estimation problem can not be solved directly using ordinary least squares, because the innovation elements in (2.23) are not known in advance. In fact this becomes a non-linear problem, which can be solved for example by applying the residual whitening iterations (Phan et al., 1995) or the recursive Extended Least Squares (ELS) method (Ljung, 1977, 1999), where the recursive scheme is preferred by experience due to its simplicity, computational complexity, and better convergence properties. The ELS algorithm is basically an extended version of the exponentially-weighted regularized RLS scheme, and given in Algorithm 2.2. It is recommended to employ such an exponentially-weighted and regularized cost function, because it ensures that during the initial iterations of the scheme a local solution exists and converges. It is possible to apply the exponentially weighting only during initial calculations and set it later close to one, which means no forgetting. For clarification, the iteration shows the conventional implementation of the recursive ELS scheme. For practical use it is recommended to use square-root or array-type implementations, for better numerical robustness against round-off errors, see for more discussion (Houtzager et al., 2011a).

**Algorithm 2.2** (The Extended Least Squares (ELS) algorithm).

**input**  $\bar{\Xi}_{-1}, \lambda, \mu$   
**require**  $0 \leq \lambda \leq 1, \mu > 0$   
**init**  $\bar{e}_{-p,p} = 0, P_{-1} = \frac{1}{\mu} I^{p(r+2\ell)+r}$   
**for**  $k = 0, 1, 2, \dots$   
     **input**  $u_k, y_k$   
  
     *Shift:*  

$$\bar{z}_{k-p,p} = [\bar{u}_{k-p,p}^T \quad u_k^T \quad \bar{y}_{k-p,p}^T \quad \bar{e}_{k-p,p}^T]^T.$$
  
  
     *Update RLS:*  

$$G_k = \bar{z}_{k-p,p}^T P_{k-1} \left( \lambda I + \bar{z}_{k-p,p}^T P_{k-1} \bar{z}_{k-p,p} \right)^{-1},$$

$$\bar{\Xi}_k = \bar{\Xi}_{k-1} + \left( y_k - \bar{\Xi}_{k-1} \bar{z}_{k-p,p} \right) G_k,$$

$$P_k = \lambda^{-1} \left( P_{k-1} - P_{k-1} \bar{z}_{k-p,p} G_k \right).$$
  
  
     *Estimate innovation:*  

$$\hat{e}_k = y_k - \bar{\Xi}_k \bar{z}_{k-p,p}.$$

**Algorithm 2.3** (The VARMAX-based PBSID<sub>opt</sub> algorithm).

**input**  $u, y, p, f, W$ .

**require**  $f \leq p$

**Step 1:** The estimation of the Markov parameters

Construct the matrices  $Y, U, \bar{U}_p$ , and  $\bar{Y}_p$ , see (2.5) and (2.6).

Solve the non-linear problem in (2.23) using ELS in Algorithm 2.2.

Construct the matrix  $\bar{E}_p$  (similar as in (2.6)) using the estimated innovation.

**Step 2:** The estimation of the state sequence

Construct the matrices  $\widehat{\Gamma\mathcal{L}}, \widehat{\Gamma\mathcal{P}}$  and  $\widehat{\Gamma\mathcal{K}}$  in (2.28).

Calculate the sequence  $\widehat{\Gamma X} = W(\widehat{\Gamma\mathcal{L}}\bar{U}_p + \widehat{\Gamma\mathcal{P}}\bar{Y}_p + \widehat{\Gamma\mathcal{K}}\bar{E}_p)$ .

Solve the low-rank approximation problem using the SVD:

$$\widehat{\Gamma X} = \begin{bmatrix} U_n & U_\perp \end{bmatrix} \begin{bmatrix} \Sigma_n & 0 \\ 0 & \Sigma \end{bmatrix} \begin{bmatrix} V_n \\ V_\perp \end{bmatrix},$$

where the system order  $n$  is determined by detecting a gap that separates the  $n$  largest singular values.

Calculate the state sequence:  $\hat{X} = \Sigma_n^{1/2} V_n$

**Step 3:** The estimation of the system matrices

Solve linear problem using least squares<sup>a</sup>:

$$\begin{bmatrix} C & D \end{bmatrix} = Y(:, 1:N-p-1) \begin{bmatrix} \hat{X}(:, 1:N-p-1) \\ U(:, 1:N-p-1) \end{bmatrix}^+.$$

Compute the innovation sequence  $\hat{E} = Y - C\hat{X} - DU$ .

Solve linear problem using least squares<sup>a</sup>:

$$\begin{bmatrix} A & B & K \end{bmatrix} = \hat{X}(:, 2:N-p) \begin{bmatrix} \hat{X}(:, 1:N-p-1) \\ U(:, 1:N-p-1) \\ \hat{E}(:, 1:N-p-1) \end{bmatrix}^+.$$

**Step 3a:** Calculate the stabilizing Kalman gain

Compute the residuals<sup>a</sup>:

$$\begin{bmatrix} \hat{W} \\ \hat{V} \end{bmatrix} = \begin{bmatrix} \hat{X}(:, 2:N-p) \\ Y(:, 1:N-p-1) \end{bmatrix} - \begin{bmatrix} A & B \\ C & D \end{bmatrix} \begin{bmatrix} \hat{X}(:, 1:N-p-1) \\ U(:, 1:N-p-1) \end{bmatrix}.$$

Estimate the covariance matrices:

$$\begin{bmatrix} Q & S \\ S^T & R \end{bmatrix} = \frac{1}{N-p} \begin{bmatrix} \hat{W} \\ \hat{V} \end{bmatrix} \begin{bmatrix} \hat{W}^T & \hat{V}^T \end{bmatrix}.$$

Calculate the solution of the Riccati equation:

$$P = APA^T - (S + APC^T)(R + CPC^T)^{-1}(S + APC^T)^T + Q.$$

Estimate a stabilizing Kalman gain:

$$K = (S + APC^T)(R + CPC^T)^{-1}.$$

<sup>a</sup>For simplicity MATLAB<sup>TM</sup> notation is used.

The practical experience with different models is that the global minimum is usually found without too much problem, especially if the initial  $\tilde{\Xi}_{-1}$  is close to the global minimum, for example by first estimating  $\tilde{\Xi}$  by linear techniques and substitute  $\tilde{\Xi}_{-1}^{(u)}$  and  $\tilde{\Xi}_{-1}^{(y)}$  by  $\tilde{\Xi}^{(u)}$  and  $\tilde{\Xi}^{(y)}$  and setting  $\tilde{\Xi}_{-1}^{(e)}$  to zero. If the solution still has not converged to the minimum yet, then the iteration can be repeated with as initial solution the previous. Although not common, the method is known to diverge if the following positive real condition:

$$\operatorname{Re} \left\{ I + C \left( e^{i\omega} I - \bar{A} \right)^{-1} \bar{K} \right\} > \frac{1}{2}, \quad \forall \omega \in \mathbb{R}, \quad (2.30)$$

is not fulfilled, see for proof and possible solutions (Ljung, 1977, 1999).

## 2.5 Simulation study

In this section, the effectiveness of the VARMAX-based PBSID<sub>opt</sub> method in comparison with the VARX-based PBSID<sub>opt</sub> method are shown by simulation studies on modern wind turbine models. Three modern wind turbine models are considered to be identified; namely, the model from the SUSCON project (van Engelen et al., 2007), a model in TURBU software (van Engelen, 2007), and a model in PHATAS software (Lindenberg, 2005). Since, the section focuses on the application of the PBSID<sub>opt</sub> algorithms to the provided benchmark data of Kanev et al. (2009), the full physical first-principle modelling concepts of the models are omitted, but a clear distinction can be made with respect to the increasing model complexity. The SUSCON wind turbine model has simplified LTI dynamics, and the noise and excitation signals fulfil the Assumption 2.1. The TURBU wind turbine model has approximated LTI dynamics of high-order and the Coleman transformation is applied to remove the first harmonic component of the periodic disturbances. The PHATAS wind turbine model has a full non-linear description, where the Coleman transformation is also used to linearise the periodic dynamics as far as possible (at least up to the second rotational frequency). In both cases, the process noise resembling the turbulence are created by a turbulence generator, and the excitation signals are designed, to conform to a real wind turbine identification scenario, therefore the assumptions in Assumption 2.1 of white Gaussian noise sequences and of persistent excitation are not perfectly fulfilled.

### 2.5.1 Identification results with the SUSCON wind turbine model

In this subsection, the SUSCON wind turbine model is used to demonstrate the effectiveness of the algorithms with a small past window size. The simplified SUSCON wind turbine model is of seventh-order and is described in van Engelen et al. (2007). Despite the small order of this model, which is useful for fast calculations, it still has the relevant features of individually pitch-controlled blades, main rotation, fore-aft and sideways tower bending mode, shaft torsion mode, and controllable generator torque. The blades are considered to be rigid. The model describes thus the most relevant rotational dynamics of a modern wind turbine around a particular operating point and has a constant state matrix while the input and output matrices strongly

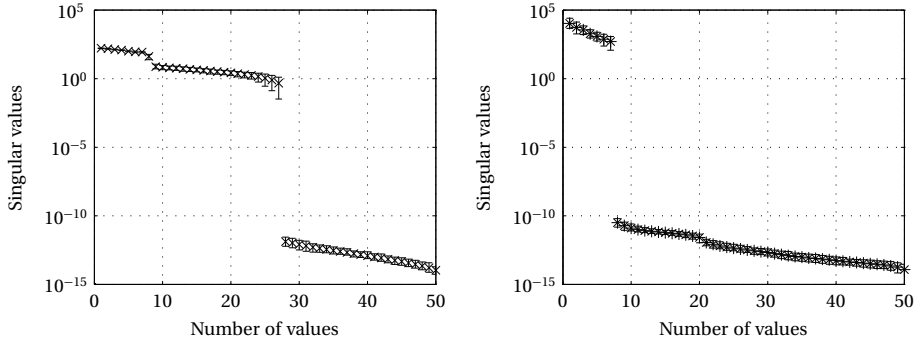
depend on the azimuth angle. For this reason, this model has been used before in van Wingerden et al. (2009) to demonstrate the closed-loop subspace LPV system identification algorithm. Similarly as in Selvam et al. (2009); van Engelen et al. (2007), the Coleman transformation can be used to transform the wind turbine model to LTI. The full description of the LTI state-space model is given in Wei and Verhaegen (2009).

The LTI SUSCON wind model is used to obtain the input and output sequences for the identification experiments. For this purpose, the equations are converted to discrete time using a zero-order hold discretization method with a sample time of 0.1 s. The wind turbine system is not asymptotically stable, as it has an pure integrator. Therefore, controllers are added in feedback loops to the system for stabilization. The descriptions of these controllers and additional filters used can be found in Selvam et al. (2009); van Engelen et al. (2007). For consistent closed-loop estimation, it is important that there is sufficient excitation from an external excitation signal or a controller of sufficiently high order, see Ljung (1999); Van den Hof and Schrama (1995). As the controllers are not sufficiently high order, we take an additional zero-mean white noise with  $\text{var}(\theta_{i,k}) = 1 \text{ deg}$  with  $i = 1, 2, 3$ , which is added to the control signal of the collective pitch controller. As additional excitation input for the generator torque we take also a zero-mean white noise signal with  $\text{var}(T_{\text{ge},k}) = 1 \cdot 10^6 \text{ Nm}$ . The wind disturbance signal is also zero-mean white noise with  $\text{var}(V_{i,k}) = 1 \text{ m/s}$ , but this signal is assumed to be unknown.

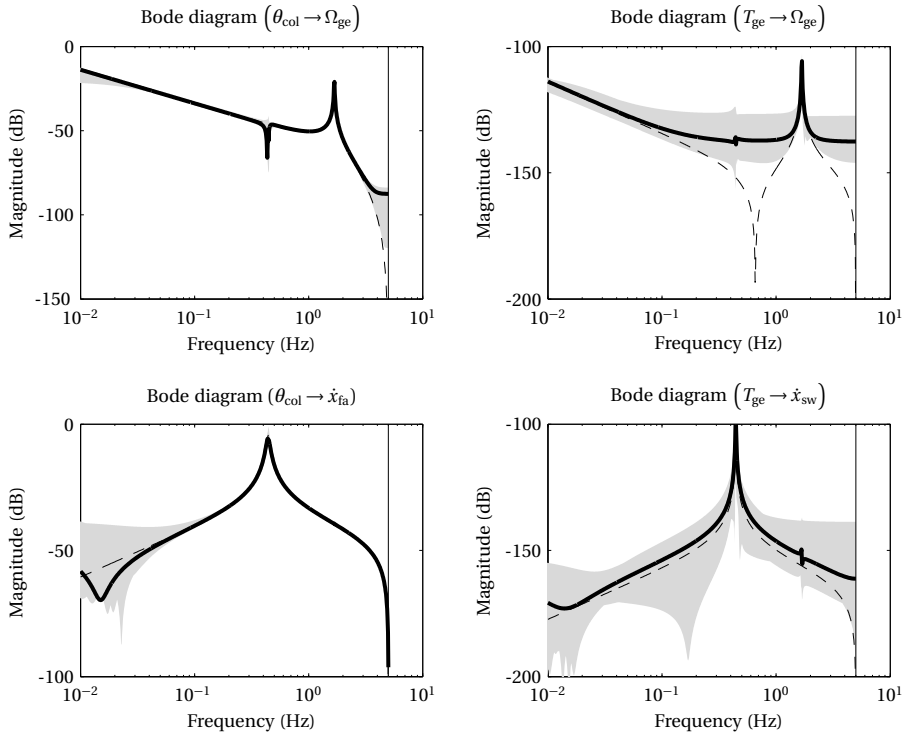
The collected data of  $u_k$  and  $y_k$  from the simulations are used in the identification experiments. For the identification experiments we used  $N = 10000$ ,  $p = f = 10$ , and  $\lambda = 0.99$  for the forgetting in the ELS algorithm. To emphasize the difference with small windows, the VARX model estimation is also carried out with the same small window size. To investigate the sensitivity of the identification algorithm with respect to the wind disturbances, a Monte-Carlo Simulation (MCS) with 100 runs was carried out. For each of the 100 MCS, a different realization of the input  $u_k$  and wind disturbance  $V_k$  is used. The performance of the identified system is evaluated by looking at the singular values of matrix  $\widehat{\Gamma X}$ , and comparing the frequency response functions of the identified models to the real model.

In Figure 2.2 the singular values of matrix  $\widehat{\Gamma X}$ , including the error bounds for 100 MCS are given using VARX/VARMAX-based model identification. As expected, the singular values of VARMAX-based identification show a large gap after the first seven largest singular values, which equals the order of the system. This is not the case with the VARX-based method, because the past window is too short to successfully model the system. Note, that 19 singular values are added to the tail of the 7 singular values of the system; this corresponds precisely to the 19 states of the controller. Figure 2.3 and 2.4 show the Bode diagrams of a selected number of inputs and outputs. The VARMAX-based method gives more consistent identifications results for the wind turbine system. For both methods, the identified natural frequencies are very close to the true natural frequencies, although a zero, which is clearly visible in the bode diagram of  $(T_{\text{ge}} \rightarrow \Omega_{\text{ge}})$ , is not estimated with the VARX-based method.

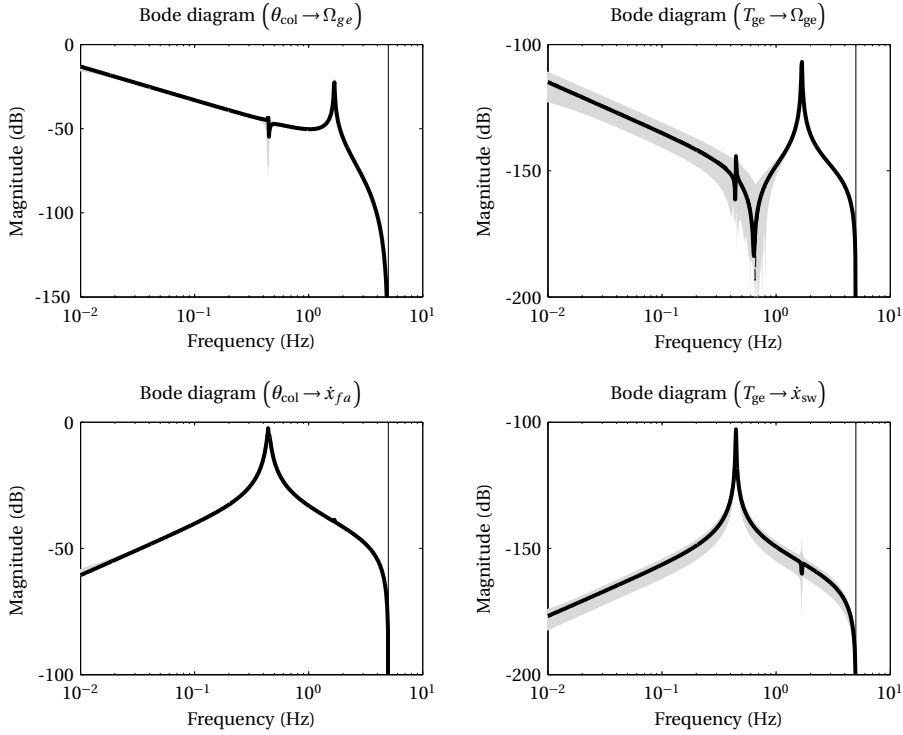
Further simulations showed that increasing the value of the past window size, which was possible due to the additional white noise excitation signal, improves the model obtained from VARX-based identification considerably, and hardly any improvement was visible in the model obtained from VARMAX-based identification.



**Figure 2.2:** The singular values of the matrix  $\widehat{\Gamma X}$  with error bounds for 100 MCS using VARX (left) and VARMAX (right).



**Figure 2.3:** Bode diagrams of the original transfer functions (dashed) and the identified transfer functions of the experiment with the best fit (bold) using VARX-based identification. The other 99 MCS are within the grey region.



**Figure 2.4:** Bode diagrams of the original transfer functions (dashed) and the identified transfer functions of the experiment with the best fit (bold) using VARMAX-based identification. The other 99 MCS are within the grey region.

As expected from the theory in Section 2.3, the VARX-based identification method gives for large past window sizes ( $p > 30$ ) almost similar results as the VARMAX-based identification results with past window size  $p = 10$ . In the next subsection, the VARX-based and VARMAX-based PBSID<sub>opt</sub> algorithm will be applied on data from a more realistic wind turbine model, because it is expected that under these conditions the VARX-based method can perform better than the VARMAX-based method.

### 2.5.2 Identification results with the TURBU wind turbine model

In this subsection, an aero-elastic wind turbine model created with the ECN software TURBU (van Engelen, 2007) is used to demonstrate the effectiveness of the closed-loop subspace LTI system identification algorithms. The model describes the rotational dynamics of a wind turbine around a particular operating point with mean wind speed  $V = 18$  m/s. The multi-body model contains around 100 states, representing the degrees of freedom in the tower, drivetrain, blades and the pitch servo actuators. The input signals to the model are three reference blade pitch angles  $\theta_i$  and the reference generator torque  $T_{ge}$ . The outputs are defined as the generator speed  $\Omega_{ge}$ , the tower top fore-aft velocity  $\dot{x}_{fa}$  and tower top sideways velocity  $\dot{x}_{sw}$ . The disturbance signals are the three blade effective wind signals  $V_i$ . The wind turbine

system is not asymptotically stable, as it contains a pure integrator. Therefore, a collective pitch controller and a generator torque controller are added in the feedback loop of the system for stabilization. The collective pitch controller, used to regulate the rotational speed, is based on a PI compensator, and the torque controller on a P compensator. Due to the restrictions from a third commercial party, most of the results in this chapter are normalized. This means that the frequencies, amplitudes, and other values illustrated in the figures are scaled.

Following the benchmark description in Kanev et al. (2009), the TURBU simulator is used to generate input and output data sequences for the identification algorithms. In Kanev et al. (2009), a proper excitation signal was selected by carefully inspecting the effect of additional loads on the drivetrain and on the tower in the fore-aft and sideways directions. The excitation signals for the collective pitch and generator torque inputs were created by filtering a Pseudo Random Binary Signal (PRBS). In Figure 2.5, the power spectral densities of a realization for both excitation signals are given. It is clearly visible that filtering was applied at the higher frequencies (for a smoother signal) and also in the case of the collective pitch around the first natural frequency of the fore-aft displacement of the tower. In Kanev et al. (2009) the data length was chosen to be  $N = 5000$  with a sampling time of  $h = 0.4s$ , which corresponds to 33 min. Longer data lengths will normally give better results, however in reality the wind turbine stays only in an operation region for a limited amount of time, depending on the wind. Two different turbulence realizations have been used; one for identification and one for validation. The turbulence was generated by the ECN software SWIFT and is of the Von Karman type (Kanev et al., 2009). Figure 2.5(c) shows the frequency content of a turbulence realization in the rotational frame. The turbulence spectrum has a low-pass nature with a peak at the rotational frequency.

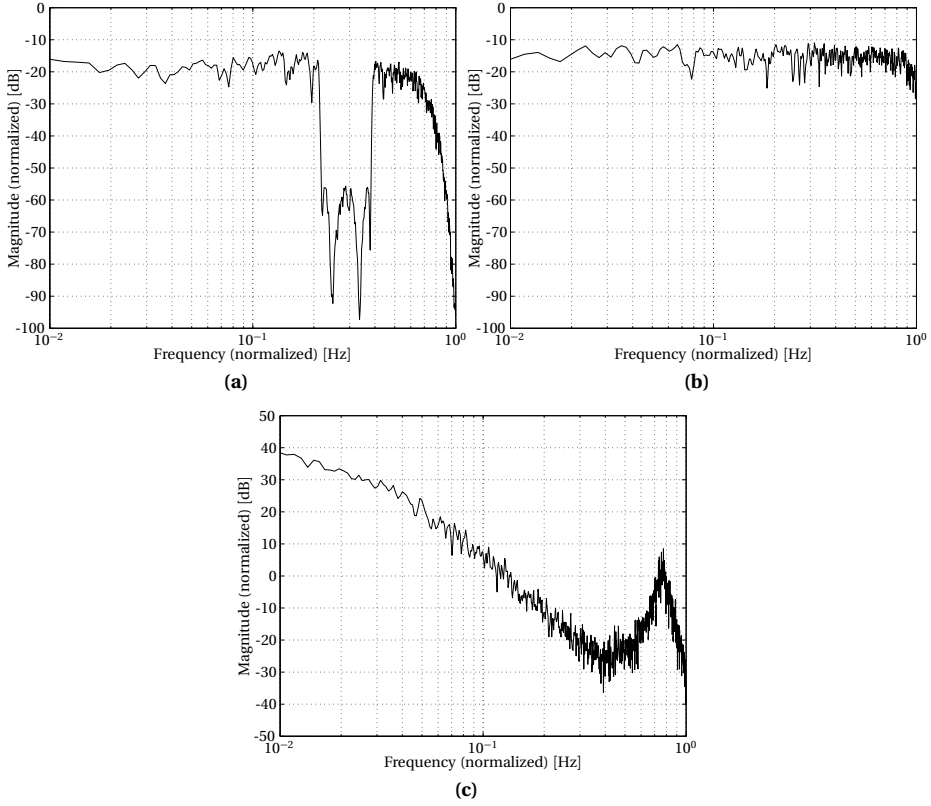
To investigate the sensitivity of the identification algorithm with respect to excitation, Monte-Carlo Simulations (MCS) with 100 runs are carried out. For each simulation a different realization of the filtered PRBS excitation signal is used. Before the identification is applied, all the input and output signals are individually scaled to an equal variance, to improve the conditioning of the identification problem. Afterwards, the scaling is reversed on the identified state-space model. The identified models are analysed both in the time and frequency domains. Time-domain analysis consists of the computation of the Variance-Accounted-For (VAF) on a data set different from the data set used for determining the model. The VAF resembles the percentage of the output variation that is explained by the model. The VAF is defined as:

$$\text{vaf}(y_k, \hat{y}_k) = \max \left\{ 1 - \frac{\text{var}(y_k - \hat{y}_k)}{\text{var}(y_k)}, 0 \right\} \cdot 100\%,$$

where  $\hat{y}$  denotes the output signal obtained by simulating the identified model,  $y$  is the measured output signal, and  $\text{var}$  denotes the variance of a quasi-stationary signal. Frequency-domain validation of the identified models covers the comparative evaluation of the most relevant channel-wise frequency functions: two from the collective pitch ( $\theta_{\text{col}} \rightarrow \Omega_{\text{ge}}, \dot{x}_{\text{fa}}$ ), and two transfers from the generator torque ( $T_{\text{ge}} \rightarrow \Omega_{\text{ge}}, \dot{x}_{\text{sw}}$ ).

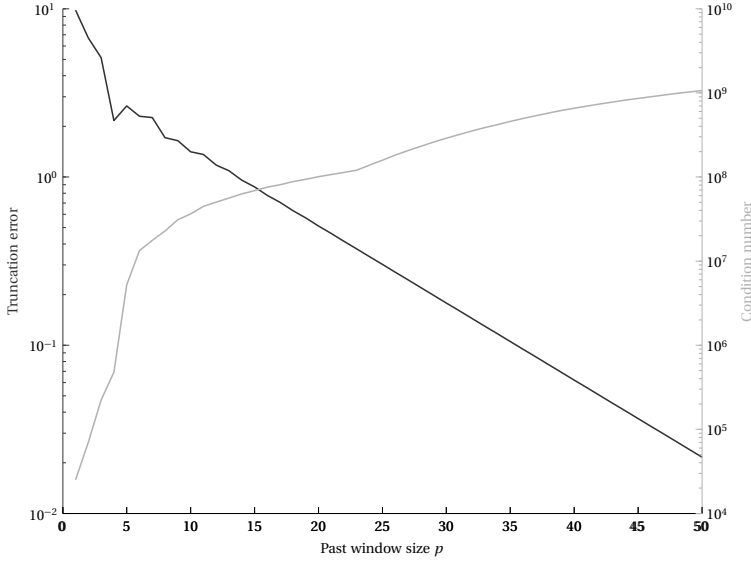
When the “optimized” predictor-based subspace identification methods are applied, the state dimension  $n$  and the past and future window size  $p$  and  $f$  need to be tuned. Theoretically, the model behind the TURBU simulator has almost 100





**Figure 2.5:** Spectral densities of the excitation signals on the pitch (a) and the generator torque (b) inputs, and of the turbulence in the rotational frame (c).

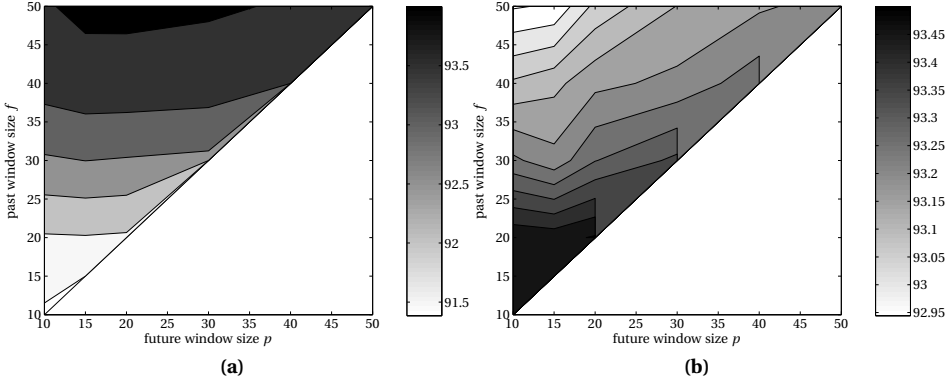
states. Not all of these states are relevant or visible during identification, because their contribution are small compared with the noise or their dynamics are too fast and therefore they are beyond the Nyquist frequency. Overall, a reasonable gap in the singular values is detected at  $n = 16$ , so a considerable lower-order model is identified. It is hard to quantify the effect for a finite past window size  $p$ . First, there is the effect of the truncation error for the VARX predictor, which equals to the sum of the remaining tail of Markov parameters. Second, is that the matrix  $\Psi$  becomes ill conditioned for large past windows, because normally the designed excitation signals do not persistently excite the system. To quantify this effect, we use the so-called condition number, which equals the ratio between the largest and smallest non-zero singular value. In Figure 2.6, the trade-off between the truncation error and condition number is illustrated. Tikhonov regularization is used for the VARX-based method in order to overcome possible numerical problems, such as singularity of the regression problem. The selection of the future window size  $f$  is even more difficult. It heavily depends on the input spectrum and system properties. It is suggested in Chiuso (2010) that a large future window is often better for identification experiments when the Signal-to-Noise Ratio (SNR) is low (more averaging effect). Otherwise for higher SNR, the optimal future window size has normally a smaller finite value.



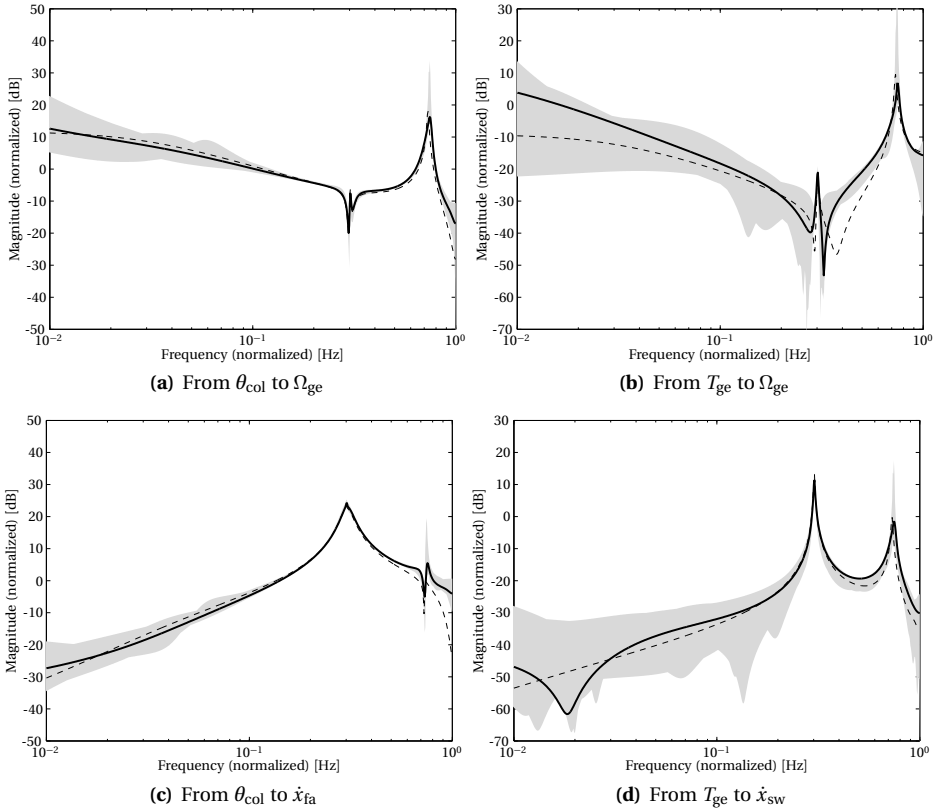
**Figure 2.6:** The truncation error of the VARX predictor and the condition number of  $\Psi\Psi^T$  as a function of the past window size  $p$ .

With the definition of the VAF, the influence of the past and future window sizes  $p$  and  $f$ , which are indispensable for the PBSID<sub>opt</sub> method, can be investigated. Computationally, the only constraint on the window sizes is to consider  $p \geq f$ . Figure 2.7 gives the mean VAF percentage for different past and future window sizes. This shows clearly the nature of the VARX-based and VARMAX-based model structures. For the VARX-based methods, more accurate mean VAF values are provided when the past window size is large. For the VARMAX-based method, more accurate mean VAF values are provided with relatively small past window sizes. In both experiments, the future horizon can be relatively small. It is noted that Tikhonov regularization must be used for the VARX-based method to obtain consistent VAF results to reflect the characteristics of the truncation error. These facts motivate the choice of the following windows for the TURBU model identification:  $p = 50$  and  $f = 15$  for the VARX-based with Tikhonov regularization, and  $p = 15$  and  $f = 15$  for the VARMAX-based method.

Using the two identification techniques, the four relevant input-output frequency responses are verified. Figures 2.8 and 2.9 show the comparative analysis of the frequency responses based on the pre-selected model order and past and future window sizes. Note that MIMO system identification is performed and Bode magnitude diagrams are plotted. As a generic conclusion, despite the short data length and the low SNR due to the turbulence intensity, one can see accurate identification of the TURBU model around the natural frequencies. The transfer functions from the pitch angle are more accurate for the TURBU model than those from the generator torque, because more excitation was allowed on this channel resulting in an higher SNR. The lower accuracy in the low frequencies due to the restricted number of repetitions of these frequencies in the short data sequence. In contrast to the previous subsection, the VARX-based method with the pre-selected variables performs better in terms of

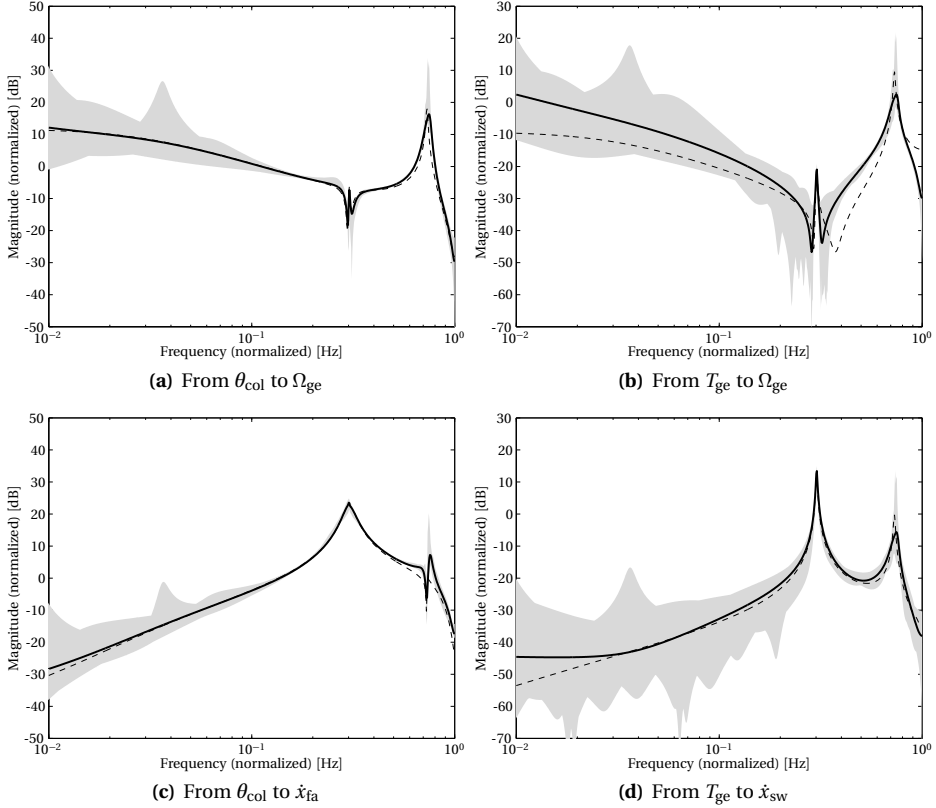


**Figure 2.7:** The mean VAF as a function of the past window and future window sizes for the identified TURBU model, where (a) is VARX-based with Tikhonov regularization and (b) is VARMAX-based. Note that the plots have different scales. ( $n = 16$ )



**Figure 2.8:** Bode diagrams of the TURBU model (dashed) and obtained using VARX-based PBSID<sub>opt</sub> with the highest VAF value (bold). The other 99 MCS are within the grey region. ( $n = 16$ ,  $p = 50$ ,  $f = 15$ )

the mean VAF in Figure 2.7 and the error bounds in Figures 2.8 and 2.9 than the VARMAX-based method. Although the extended least squares problem had enough persistent excitation due to the smaller past window size, it turns out that for the VARMAX-based method it is more difficult to accurately estimate a low-order model, especially the damping of the drivetrain; see the peak of the second natural frequency.



**Figure 2.9:** Bode diagrams of the TURBU model (dashed) and obtained using VARMAX-based PBSID<sub>opt</sub> with the highest VAF value (bold). The other 99 MCS are within the grey region. ( $n = 16$ ,  $p = 15$ ,  $f = 15$ )

### 2.5.3 Identification results with the PHATAS wind turbine model

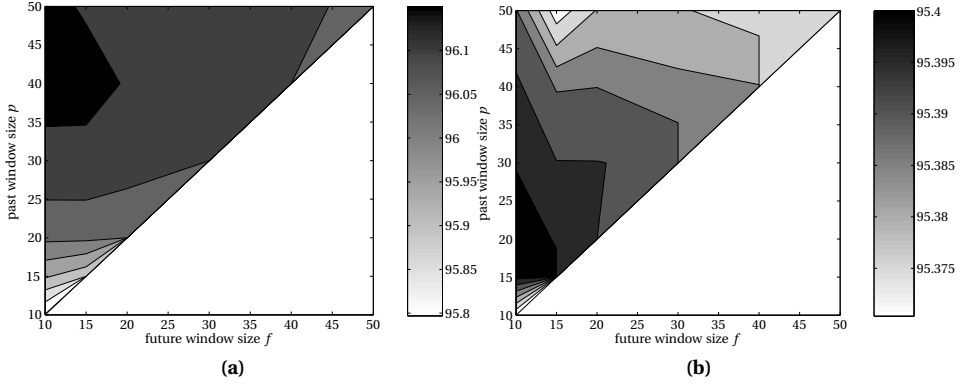
In this section, the same wind turbine as in the previous section is going to be modelled in the ECN software PHATAS instead. The non-linear aero-elastic wind turbine model is used to demonstrate the closed-loop subspace LTI system identification algorithms. The model describes the rotational dynamics of a wind turbine with mean wind speed  $V = 17\text{m/s}$ . Due to the introduction of a wind shear correction profile, the mean wind speed is lower than in the previous subsection. The model contains the degrees of freedom in the foundation, tower, drive train, blades and

the pitch servo actuators. The input signals to the model are three reference blade pitch velocities  $\dot{\theta}_i$  (not angles as before) and the reference generator torque  $T_{ge}$ . The outputs are again defined as the generator speed  $\Omega_{ge}$ , the tower top fore-aft velocity  $\dot{x}_{fa}$  and tower top sideways velocity  $\dot{x}_{sw}$ . The disturbance signals are the three blade effective wind signals  $V_i$ . A non-linear identification is not considered in the sequel, but a dedicated Coleman transformation to fixed-frame coordinates is used to provide linear-like behaviour of the wind turbine dynamics.

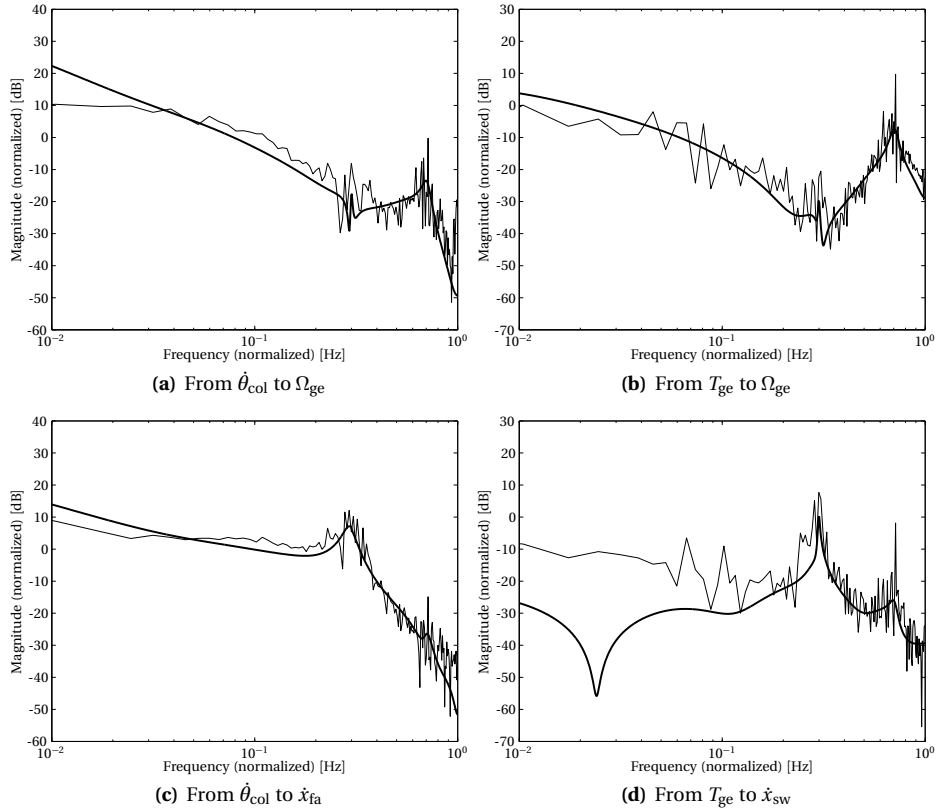
The identification of the PHATAS model is performed under almost the same conditions as with the identification with the TURBU model, see also the benchmark description in Kanev et al. (2009). The first difference is that the PHATAS simulator uses the pitch rate command signal instead, therefore the additional excitation signal for the pitch angle is differenced to obtain the pitch rate. The second difference is that due to the costly (in terms of time and availability) PHATAS simulations the identification and its verification can only be performed on one data sequence, therefore no Monte Carlo experiments can be applied. At last, we do not possess a linearised model of the non-linear PHATAS model. Consequently, the frequency response verification is done using the spectral analysis method (see Appendix B), as the true frequency response is not exactly unknown. Let us emphasize again that the validation/verification results are based on a single data realization.

The influence of the past and future window sizes  $p$  and  $f$ , which are indispensable for PBSID<sub>opt</sub> method, is again investigated using the VAF. Computationally, the only constraint on the window sizes is to consider  $p \geq f$ . Figure 2.10 gives the VAF percentage for different past and future window sizes. This shows again clearly the nature of the VARX-based and VARMAX-based model structure. For the VARX-based methods, more accurate mean VAF values are provided when the past window size is large. For the VARMAX-based method, more accurate mean VAF values are provided with relatively small past window sizes. In both experiments, the future horizon can be relatively small. It is noted again, that Tikhonov regularization must be used for the VARX-based method to obtain consistent VAF results to reflect the characteristics of the truncation error. These facts motivate the choice of the following windows for the PHATAS model identification:  $p = 50$  and  $f = 15$  for the VARX-based with Tikhonov regularization, and  $p = 15$  and  $f = 15$  for the VARMAX-based method.

Using the two identification techniques, the four relevant input-output frequency responses are verified using spectral analysis. Figures 2.11 and 2.12 show the comparative analysis of the frequency responses based on the pre-selected model order and past and future window sizes. Note that MIMO system identification is performed and Bode magnitude diagrams are plotted. Similar to the TURBU identifications, one can see accurate identification of the PHATAS model around the natural frequencies. The transfer functions from the pitch rate are more accurate for the PHATAS model than those from the generator torque, because more excitation was allowed in this pitch channel resulting in a higher SNR. The lower accuracy in the low frequencies is due to the restricted number of repetitions of these frequencies in the short data sequence. Similar to the previous subsections, the performance of VARX-based method have again been increased relative to the VARMAX-based method in terms of the VAF in Figure 2.10 and the results in Figures 2.11 and 2.12. With the increasing complexity of the system to be modelled, it turns out that for the VARMAX-based

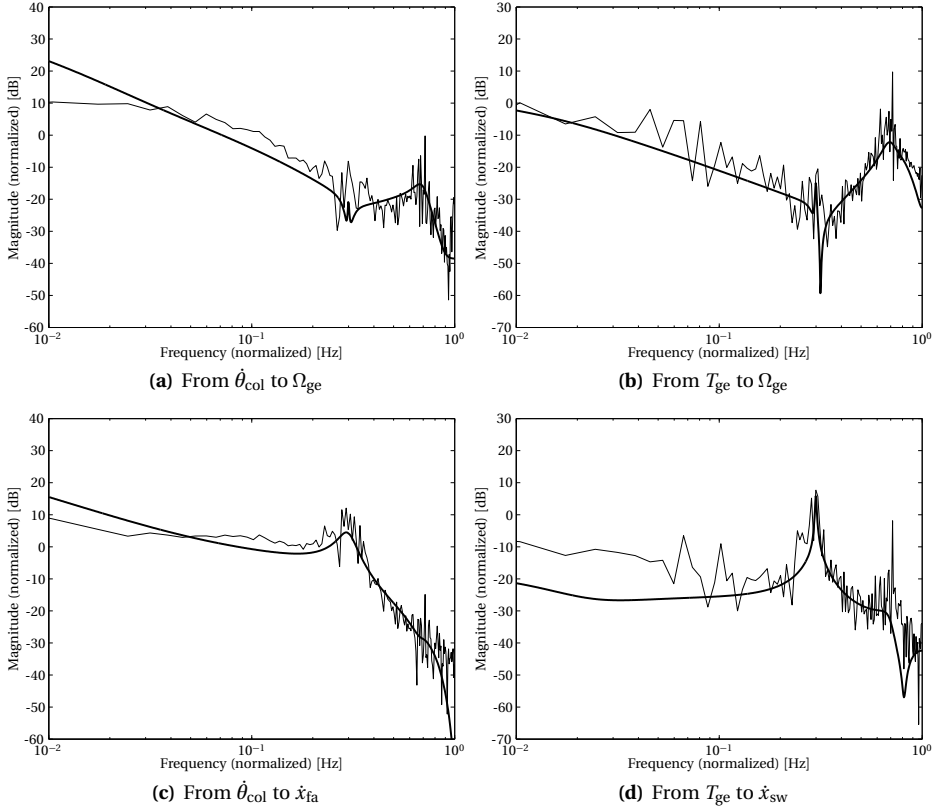


**Figure 2.10:** The mean VAF as a function of the past window and future window sizes of the identified PHATAS model, where (a) is VARX-based with Tikhonov regularization and (b) is VARMAX-based. Note that the plots have different scales. ( $n = 16$ )



**Figure 2.11:** Bode diagrams of the PHATAS model obtained using spectral analysis (grey line) and obtained using VARX-based PBSID<sub>opt</sub> (black line). ( $n = 16$ ,  $p = 50$ ,  $f = 15$ )

method it is more difficult to accurately estimate a low-order model, especially at the natural frequency of the drivetrain which is in turn very important to determine the stability of the collective pitch controller.



**Figure 2.12:** Bode diagrams of the PHATAS model obtained using spectral analysis (grey line) and obtained using VARMAX-based PBSID<sub>opt</sub> (black line). ( $n = 16$ ,  $p = 15$ ,  $f = 15$ )

It can be concluded from these identification experiments (and also with other models and experiments), that the VARMAX-based PBSID<sub>opt</sub> is less robust to small violations in the zero-mean white Gaussian assumption on the noise sequences and to the dynamics that are not represented in the identified state-space model of lower order. These violations will normally cause the estimated innovation sequence to be not perfectly white and/or Gaussian. This heavily impacts the course of the following steps in the VARMAX-based algorithm and therefore will influence the final quality of the model estimate more than in the case of the VARX-based algorithm.

Finally, it must be emphasized that the identification of the TURBU and PHATAS models are performed under extreme (worst case) turbulence conditions. With normal turbulence conditions, it is expected that the accuracy of the identified models will improve considerably. If still a more accurate model is required, a possible

additional step is to further optimize the individual state-space parameters using the Prediction-Error Method (PEM) (Ljung, 1999). Even for the parameter estimation of LTI state-space models, it normally involves solving a non-convex optimization problem. It may therefore be difficult to guarantee that the global optimum will be found. It is considered to be very likely that an identified model from subspace identification lies in the region of attraction of the global optimum (Lyzell et al., 2009). If the predictor form of the identified model is stable, and the parametrization of the state-space model is carefully selected to avoid numerical problems (Gevers and Li, 1993), the prediction-error estimate can then be obtained by a local search starting at this initial estimate.

## 2.6 Conclusion

In this chapter, the optimized predictor-based subspace identification method is presented and its effectiveness for the estimation of a wind turbine model in closed loop is shown. This identification technique does not require any controller related information, consequently the identified model becomes consistent no matter whether the wind turbine operates with or without a controller in the loop. Since the wind turbine is not asymptotically stable in open loop, only closed-loop experiments are supported in reality. This fact makes the proposed method very attractive for the wind power community. Further, a novel extension to the optimized predictor-based subspace identification method is developed that relaxes the requirement that the past window size has to be large for asymptotically consistent estimates. For this purpose, a finite description of the input-output relation is formulated, and is solved efficiently using the extended least squares recursion. The effectiveness and robustness of the proposed methods are discussed in a simulation study, where the algorithms are used on a data source based on controller-in-the-loop simulations of a typical modern wind turbine. It can be concluded that the novel method performs very well with a small past window under perfect conditions, however the novel method is less robust than the original method to the increasing violations that occur in the noise assumptions or the model order in practice. Despite the short data length due to the limited amount of time in an operation region, and the low signal-to-noise ratio due to the high turbulence intensity, an accurate identification of the SUSCON/TURBU/PHATAS model is obtained for both methods, especially around the natural frequencies of the wind turbine.



## Recursive Identification with Application to the Real-Time Closed-Loop Tracking of Flutter

---

A novel recursive predictor-based subspace identification method is presented to identify linear time-invariant systems with multiple inputs and multiple outputs. The method is implemented in real time and is able to operate in open loop or closed loop. The recursive identification is performed via the subsequent solution of only three linear problems, which are solved using recursive least squares. The recursive implementation of the method is not only able to identify linear time-invariant models from measured data, but can also be used to track slowly time-varying dynamics if adaptive filters are used. The computational complexity is reduced by exploiting the structure in the data equations and by using array algorithms to solve the main linear problem. This results in a fast recursive predictor-based subspace identification method suited for real-time implementation. The real-time implementation and the ability to work with multi-input and multi-output systems operating in closed loop makes this approach suitable for online estimation of unstable dynamics. The ability to do so is demonstrated by the detection of flutter on an experimental 2D-airfoil system.

---

### 3.1 Introduction

THE literature on recursive identification is almost entirely based on parametrized transfer functions. Since many control design methods have been derived for state-space models, a state-space approach is a useful extension to the existing results on recursive identification for systems operating in both open loop and closed loop. The PBSID<sub>opt</sub> method from the previous chapter is becoming a well-established technique to identify LTI state-space models from real systems in practice, see Balini

et al. (2010); van Wingerden et al. (2011). However, extending this approach to deal with time-varying dynamics is still a general problem. One possible extension to  $\text{PBSID}_{\text{opt}}$  is the identification of affine Linear Parameter-Varying (LPV) or bilinear models (van Wingerden and Verhaegen, 2009; van Wingerden et al., 2009). The time-varying dynamics in these linear model descriptions are considered linear dependent on known scheduling signals. The challenge is to make this extension suitable for practical applications, which is difficult because of the very large memory requirements due to the so-called curse of dimensionality (van Wingerden, 2008). Recursive Subspace IDentification (RSID) methods have much lower memory requirements than their batchwise counterparts. In this chapter we will not include the extension with scheduling signals, but we will introduce and improve the recursive implementation of the  $\text{PBSID}_{\text{opt}}$  method. If formulated in the right way and by using adaptive filters, the recursive implementation can be used to track slowly varying dynamics by introducing forgetting.

RSID methods are intended for real-time implementation, such that the system can be identified online. When dealing with high sampling rates and high dimensional systems the limiting computational complexity is an additional challenge. It was suggested in Chiuso et al. (2010); Wu et al. (2008) that  $\text{PBSID}_{\text{opt}}$  can be adapted to a recursive implementation with the algorithms used in a number of successful RSID methods (Gustafsson, 1998; Lovera, 2003; Lovera et al., 2000; Oku and Kimura, 2002). However, these algorithms are based on the Projection Approximation Subspace Tracking (PAST) method (Yang, 1995). As remarked in Mercère et al. (2004), the disadvantage of PAST is that the identified state-space model does not converge to a constant state basis; the basis slowly changes over time. This observation presents a problem for the recursive  $\text{PBSID}_{\text{opt}}$  implementation, especially if it is used for change detection, like flutter detection. Instead, the use of the Propagator Method (PM) is proposed in Mercère and Lovera (2007); Mercère et al. (2008). Together with the inclusion of a weight to the algorithm, it results in the novel recursive  $\text{PBSID}_{\text{opt}}$  implementation. The main advantage of the weighted PM over the previous concept is that if the dynamics are time-invariant, the recursively estimated state-space realization converges after a short transient period to a constant state-basis.

The real-time implementation and the ability to work with systems operating in closed loop make this approach suitable for online estimation of unstable dynamics, like flutter. Flutter is the most feared aerodynamic instability, where wing mode oscillations extract energy from the airstream and introduce excessive aerodynamic forces that can lead to structural failure (Mukhopadhyay, 2003). The 2D-airfoil system is commonly used for research in flutter suppression (Rivera et al., 1992). The main issue is that the 2D-airfoil dynamics are rather complex and strongly dependent on the free-stream wind speed. A common way to describe the aeroelastic response in a mathematical model is to optimize a large number of parameters of a first-principles model, obtained for example from the theory in Theodorsen (1935). Due to the size and the non-linearity of the parameter identification problem, the suggested optimisation methods in Bieniawski (2005); Cavagna et al. (2009); De Gaspari et al. (2009) require hours of offline computation, and for a satisfactory result a good initial estimate is needed. Faster identification methods based on the Eigensystem Realization Algorithm (ERA) have been successfully applied offline in De Marqui et al.

(2006); McEver et al. (2007). These methods do not require a priori knowledge of the LTI model structure and work in closed loop, but are computationally difficult to implement online and made be adaptive. The use of adaptive RSID methods has been explored with open-loop simulations in De Cock et al. (2006); Goethals et al. (2004). Although they have the potential to be implemented in real time, these algorithms still have problems with identifying systems operating in closed loop.

The main contributions of this chapter are fourfold. The first contribution is the recursive implementation of the  $\text{PBSID}_{\text{opt}}$  method using PM (the so-called  $\text{RPBSID}_{\text{pm}}$  method) to identify LTI models from data measured in open or closed loop. The  $\text{RPBSID}_{\text{pm}}$  method can also be used to track slowly time-varying dynamics if adaptive filters are used. The second contribution is that we show that a certain weight is required to minimize variations of the state basis. The third contribution is that computational complexity is reduced by exploiting the structure in the data equations and by using array algorithms to solve the main linear problem, such that a real-time implementation of the proposed method can be obtained. The final contribution is that we show that we can detect the onset of flutter. Moreover, we can track the natural frequencies and modal damping. We will provide guidelines on how to pick the free parameters in the proposed algorithm.

The outline of this chapter is as follows. In Section 3.2 the recursive implementation of the VARX-based predictor-based subspace identification method is presented and discussed. In Section 3.3 the practical implementation issues are discussed and some modifications to the algorithm are presented. In Section 3.4 the closed-loop performance of the proposed RSID method to existing methods are evaluated by a simulation study. In Section 3.5 the effectiveness of the proposed algorithm is emphasized with an experimental study on flutter detection of a 2D-airfoil system operating in closed loop and some tuning guidelines are provided. In the final section we present the conclusions in this chapter.

## 3.2 Recursive identification with $\text{RPBSID}_{\text{pm}}$

In this section, the recursive identification and tracking with predictor-based subspace methods of systems with slowly time-varying dynamics operating in either open loop or in closed loop is presented. The recursive system identification problem can be formulated as:

**Problem Description 3.1** (Recursive system identification). *Given the system matrices  $A_{k-1}$ ,  $B_{k-1}$ ,  $C_{k-1}$ ,  $D_{k-1}$ , and  $K_{k-1}$  up to a unknown similarity transformation  $T$ , and given the new input sample  $u_k$ , and output sample  $y_k$  find, if they exist, system matrices  $A_k$ ,  $B_k$ ,  $C_k$ ,  $D_k$ , and  $K_k$  in a similar state basis.*

### 3.2.1 Step 1: Adaptive filtering of the Markov parameters

Adaptive filters are used to estimate the Markov parameters, because of their ability to track slow variations in the underlying signal statistics. This is handled in a natural

way by assigning less weight to older measurements that are no longer representative for the system. Overviews of the properties of different adaptive filters can be found in Diniz (2008); Ljung and Söderström (1983); Sayed (2008).

Let's rewrite (2.16) to the following linear regression:

$$y_k = \tilde{\Xi}_k \varphi_k + e_k, \quad (3.1)$$

where  $\varphi_k = [\tilde{u}_{k-p,p}^T \quad u_k^T \quad \tilde{y}_{k-p,p}^T]^T$ . The adaptive filters that can be considered are described by the following structure:

$$\tilde{\Xi}_k = \tilde{\Xi}_{k-1} + (y_k - \tilde{\Xi}_{k-1} \varphi_k) \varphi_k^T P_k, \quad (3.2)$$

with different choices of the error covariance matrix  $P_k$ . For the Least Mean Squares (LMS) filter, the error covariance matrix  $P_k = \mu_1 I$  is used, where the scalar  $\mu_1 > 0$  is a step-size parameter that must be sufficiently small to prevent instability of the update process but large enough to adopt any changes in the dynamics. For the Recursive Least Squares (RLS) filter, the error covariance matrix is started with  $P_{-1} = \frac{1}{\rho_1} I$ , and iterated as:

$$P_k = \frac{1}{\lambda_1} (P_{k-1} - P_{k-1} \varphi_k (\lambda_1 I + \varphi_k^T P_{k-1} \varphi_k)^{-1} \varphi_k^T P_{k-1}),$$

where the scalar  $0 \ll \lambda_1 \leq 1$  is the forgetting factor to weight past data less (or equal if  $\lambda = 1$ ) than more recent data. The initial regularization value  $\rho_1 > 0$  makes sure the problem is well conditioned in the first iterations. For the Kalman filter, the error covariance matrix is started with  $P_{-1} = \frac{1}{\rho_1} I$ , and iterated as:

$$P_k = P_{k-1} - P_{k-1} \varphi_k (R + \varphi_k^T P_{k-1} \varphi_k)^{-1} \varphi_k^T P_{k-1} + Q,$$

where the covariance matrix  $Q > 0$  denotes the variations of the Markov parameters. The values of these parameters have to be chosen by insight of the user. In this chapter the focus is on the RLS filter. The parameter  $\lambda_1$  of the RLS filter is much easier in practice to tune than the parameters  $R$  and  $Q$  of the Kalman filter. In general the RLS filter has much better convergence properties of the estimate during changes in parameters compared with the LMS algorithm, at the cost of more computations for each update. However it is noted that there exist models for which the LMS filter performs better than the RLS filter, see Sayed (2008).

### 3.2.2 Step 2: The selection of the weight and state vector basis

From Section 2.3.2 it is given that after the estimation of the Markov parameters, the matrices  $\tilde{\Gamma}\tilde{\mathcal{L}}$  and  $\tilde{\Gamma}\tilde{\mathcal{K}}$  are constructed, and the predicted state  $x_k$  can be estimated from a low-rank approximation using the relation:

$$\tilde{\Gamma} \hat{x}_k = \tilde{\Gamma}\tilde{\mathcal{L}} \tilde{u}_{k-p,p} + \tilde{\Gamma}\tilde{\mathcal{K}} \tilde{y}_{k-p,p}. \quad (3.3)$$

The following modification becomes extremely relevant when we consider recursive identification. Using Lemma 2.1, it is possible to obtain a product between the state and the extended observability matrix, given by:

$$\begin{aligned} \tilde{\mathcal{H}}^{-1} \tilde{\Gamma} \hat{x}_k &= \tilde{\mathcal{H}}^{-1} \tilde{\Gamma}\tilde{\mathcal{L}} \tilde{u}_{k-p,p} + \tilde{\mathcal{H}}^{-1} \tilde{\Gamma}\tilde{\mathcal{K}} \tilde{y}_{k-p,p} \\ \Gamma \hat{x}_k &= \Gamma\tilde{\mathcal{L}} \tilde{u}_{k-p,p} + \Gamma\tilde{\mathcal{K}} \tilde{y}_{k-p,p}. \end{aligned} \quad (3.4)$$

Generally the relation in (3.3) can be pre-multiplied with any pre-defined weight matrix  $W \in \mathbb{R}^{f\ell \times f\ell}$  from the left side. In the batchwise, the SVD method gives the best low-rank approximation. In online identification, it is important to update the model during one sampling period. In spite of the existence of updating methods for the SVD, it is still difficult to make them adaptive and to implement these algorithms online due to the computational load and numerical stability. The main difficulty with adaptive SVD methods is to keep the left and right subspace orthogonal, the property which makes the SVD unique. Consequently, researchers try to find alternative methods to apply the subspace concept in an adaptive framework.

The transformation in Lemma 2.1 leads also to an alternative way to obtain the system matrices. Like in most traditional SID methods, the matrices  $A$  and  $C$  are obtained from the observability matrix  $\Gamma_k$  as<sup>1</sup>:

$$A = \Gamma(1:(f-1)\ell, :)^+ \Gamma(\ell+1:f\ell, :), \quad C = \Gamma(1:\ell, :),$$

where  $(\cdot)^+$  is the pseudo inverse. This alternative method plus the recursive estimation of  $B$  and  $K$  can be quite costly to compute for large window sizes, therefore the calculation of the system matrices through the state sequence is preferred.

Most RSID methods use alternative methods to obtain a low-rank approximation. If the order  $n$  is known, it is possible to design a selection matrix  $S_k \in \mathbb{R}^{n \times \ell f}$ , such that:

$$\hat{x}_k = S_k z_k = S_k W_k (\widetilde{\Gamma} \tilde{\mathcal{L}}_k \bar{u}_{k-p,p} + \widetilde{\Gamma} \tilde{\mathcal{K}}_k \bar{y}_{k-p,p}), \quad (3.5)$$

where with an SVD update, the selection matrix becomes  $S_k = \Sigma_{n,k}^{-1/2} U_{n,k}^T$ . At the cost of the addition of another linear problem, it is also possible to estimate the selection matrix  $S_k$  using subspace trackers, like the Projection Approximation Subspace Tracking (PAST) method in Yang (1995) or the gradient type method in Oku and Kimura (2002). The update of the PAST method is given by:

$$S_k^T = S_{k-1}^T + (z_k - S_{k-1}^T S_{k-1} z_k) z_k^T P_{S,k}, \quad (3.6)$$

which is very similar to the filter update in (3.2) and also has similar choices for the error covariance matrix  $P_{S,k}$ . The disadvantage of subspace trackers is that the estimated state vector  $\hat{x}_k = T_k x_k = S_k \tilde{\Gamma} x_k$  does not converge to a constant basis, because the estimated  $S_k$  varies slowly over time due to forgetting during the updating of  $P_{S,k}$  (Mercère et al., 2004).

More recently, the Propagator Method (PM) has been put forward in Mercère et al. (2008). As the system to be modelled  $\mathcal{S}$  in (2.1) is observable, the observability matrix  $\Gamma$  has at least  $n$  linearly independent rows. If the order  $n$  is known, it is possible to design a stationary selection matrix  $S \in \mathbb{R}^{n \times \ell f}$  such that:

$$\hat{x}_k = S(\Gamma \tilde{\mathcal{L}}_k \bar{u}_{k-p,p} + \Gamma \tilde{\mathcal{K}}_k \bar{y}_{k-p,p}). \quad (3.7)$$

The main advantage of this Propagator Method (PM) over the previous concept is that as long as the linear dynamics are time invariant, the recursively estimated state vector converges after a short transient period to a constant state basis:  $T_k \rightarrow T = S\Gamma$ .

<sup>1</sup>For simplicity MATLAB<sup>TM</sup> notation is used.

However this advantage is not yet completely valid for the proposed method with the observability matrix in the predictor form, because the Kalman gain in  $\tilde{\Gamma}$  has the tendency to vary over time when forgetting is introduced even when the deterministic system dynamics are constant.

In recursive identification with a moving horizon, for example using a sliding window or by the inclusion of forgetting, it is well known and expected that the Kalman gain  $K$  becomes non-stationary, and varies when the Kalman predictor is updated with new data (and downdated with data which falls out of the window). As a result, the estimated state vector  $\hat{x}_k = T_k x_k = S\tilde{\Gamma}_k x_k$  does not converge to a constant basis, because  $\tilde{\Gamma}_k$  is perturbed over time by the non-stationary Kalman gain. Fortunately the transformation in Lemma 2.1 can transform  $\tilde{\Gamma}_k$  into  $\Gamma$ . By selecting the weight matrix in (3.5) to be  $W_k = \tilde{\mathcal{H}}_k^{-1}$ , the propagator method can be applied. The matrix  $\tilde{\mathcal{H}}_k$  can be constructed from the estimated Markov parameters  $\tilde{\Xi}_k^{(y)}$ .

### 3.2.3 Step 3: Adaptive filtering of the system matrices

For clarification, the last two linear problems for the estimation of the system matrices are written in short as:

$$y_k = \Theta_k^{(y)} \phi_k + e_k, \quad (3.8)$$

$$\hat{x}_{k+1} = \Theta_k^{(x)} \psi_k, \quad (3.9)$$

where  $\phi_k = [\hat{x}_k^T \ u_k^T]^T$ ,  $\psi_k = [\hat{x}_k^T \ u_k^T \ \hat{e}_k^T]^T$ , and  $\Theta_k$  denotes the set of state-space parameters with  $\Theta_k^{(y)} = [C_k \ D_k]$  and  $\Theta_k^{(x)} = [A_k \ B_k \ K_k]$  of the corresponding state-space equations. The innovation noise sequence, which is needed for the second linear problem in (3.8), can be obtained from the solution of the first linear problem (3.8) by  $\hat{e}_k = y_k - \Theta_k^{(y)} \phi_k$ . For recursive solvers of these two linear problems, similar adaptive filters can be used as with the estimation of the Markov parameters in Subsection 3.2.1. For a recursive implementation of the calculation of stabilizing Kalman gain (Step 3a in 2.1) the forward Riccati recursion can be considered, but due to the larger computational cost this is not applied in the recursive implementation.

### 3.2.4 The RPBSID<sub>pm</sub> algorithm

Using the recursive methods mentioned in the previous subsections, the PBSID<sub>opt</sub> method in Algorithm 2.1 can be converted to the recursive method in Algorithm 3.1, the so-called RPBSID<sub>pm</sub> method. By comparing the recursive and batchwise method, two clear differences can be remarked. The first difference is that for the recursive method the three steps are calculated in a sample cascaded manner, i.e., after new available input-output data the three steps are updated directly, and not in a batchwise cascaded manner. Thus the last two adaptive filters are updating their estimate with data containing possible errors from the preceding adaptive filter. Therefore the information, especially in the initial transient period, is somewhat misused. The second difference is that the system order is not estimated online and therefore needs to be given as an input to the algorithm.

An advantage of the recursive predictor-based subspace method over most other RSID methods is that the estimated state vector (and also the system matrices) directly becomes available and not after  $f$  steps of delay. For example, with MOESP-PO (Verhaegen and Verdult, 2007) based methods the data  $u_{k-f+1} \dots u_k$  and  $y_{k-f+1} \dots y_k$  are used as instrumental variables to eliminate the influence of noise. This results that at time step  $k$  the state  $x_{k-f}$  is estimated instead, see Lovera et al. (2000); Mercère et al. (2008).

**Algorithm 3.1** (The RPBSID<sub>pm</sub> algorithm).

**input**  $p, f, n, S, \Theta_{-1}, \lambda_{1,2,3}, \rho_{1,2,3}$

**require**  $n > 0, n/\ell \leq f \leq p, 0 \ll \lambda_{1,2,3} \leq 1, \rho_{1,2,3} > 0$

**init**  $P_{-1} = \frac{1}{\rho_1} I, M_{-1} = \frac{1}{\rho_2} I, N_{-1} = \frac{1}{\rho_3} I, \tilde{\Xi}_{-1}$  from  $\Theta_{-1}$

**for**  $k = 0, 1, 2, \dots$

**input**  $u_k, y_k$

**Step 1:** Update the Markov parameters

Update RLS:

$$P_k = \frac{1}{\lambda_1} P_{k-1} - \frac{1}{\lambda_1} P_{k-1} \varphi_k (\lambda_1 I + \varphi_k^T P_{k-1} \varphi_k)^{-1} \varphi_k^T P_{k-1},$$

$$\tilde{\Xi}_k = \tilde{\Xi}_{k-1} + (y_k - \tilde{\Xi}_{k-1} \varphi_k) \varphi_k^T P_k.$$

**Step 2:** Estimate the state vector

Construct the matrices  $\tilde{\Gamma} \tilde{\mathcal{L}}_k$  and  $\tilde{\Gamma} \tilde{\mathcal{K}}_k$  in (2.20).

Construct the matrix  $\tilde{\mathcal{H}}_k$  in (2.9).

Compute:  $\hat{x}_k = S \tilde{\mathcal{H}}_k^{-1} (\tilde{\Gamma} \tilde{\mathcal{L}}_k \bar{u}_{k-p,p} + \tilde{\Gamma} \tilde{\mathcal{L}}_k \bar{y}_{k-p,p})$ .

**Step 3:** Update the system matrices

Update RLS:

$$M_k = \frac{1}{\lambda_2} M_{k-1} - \frac{1}{\lambda_2} M_{k-1} \phi_{k-1} (\lambda_2 I + \phi_{k-1}^T M_{k-1} \phi_{k-1})^{-1} \phi_{k-1}^T M_{k-1},$$

$$\Theta_k^{(y)} = \Theta_{k-1}^{(y)} + (y_{k-1} - \Theta_{k-1}^{(y)} \phi_{k-1}) \phi_{k-1}^T M_k.$$

Compute  $e_{k-1} = y_{k-1} - \Theta_{k-1}^{(y)} \phi_{k-1}$ .

Update RLS:

$$N_k = \frac{1}{\lambda_3} N_{k-1} - \frac{1}{\lambda_3} N_{k-1} \psi_{k-1} (\lambda_3 I + \psi_{k-1}^T N_{k-1} \psi_{k-1})^{-1} \psi_{k-1}^T N_{k-1},$$

$$\Theta_k^{(x)} = \Theta_{k-1}^{(x)} + (x_k - \Theta_{k-1}^{(x)} \psi_{k-1}) \psi_{k-1}^T N_k.$$

### 3.3 Practical implementation in real time

In this section, practical implementation issues are discussed and solutions are presented. In the first subsection, regularization is introduced to overcome the typical ill-conditioned Markov estimation problem. In the second subsection, faster and more numerical stable algorithms are presented for real-time implementation.

### 3.3.1 Regularization of the Markov estimation problem

In Assumption 2.2 an approximation is made which requires the past window to be chosen as large as possible. However, in practical experiments it is most of the time needed to design the excitation signal such that it does not exceed the load specifications and ensures that the system to be identified operates around a particular operation point. In this case, especially with a low-order controller in the feedback loop, the matrix  $P_k$  becomes ill conditioned, because the input signals do not persistently excite the system. Apart from just selecting a lower past window size, a number of other solutions exist to get a well-conditioned matrix  $P_k$  for the least squares update in (3.2). The first possible solution is generating more persistently exciting control signals by changing the design properties of the controller. For example automatically generated constraints can be added to the control design problem when the matrix  $P_k$  becomes ill conditioned, see Dong (2009). The second possible solution is utilizing a VARMAX model, for which the input-output description relaxes the need for a very large past window due to the formulation of a deadbeat predictor, see (Houtzager et al., 2009b). However, a non-linear problem is formulated which can be solved recursively by the Extended Least Squares (ELS) method. The third and well-used solution to the ill-posed least squares problem is the inclusion of a regularization quantity in the cost to be minimized. This solution has been used for the experimental study in Section 3.5.

In the batchwise case, Tikhonov regularization from (Tikhonov and Arsenin, 1977) can be included by replacing the least squares estimation of the Markov parameters in Algorithm 2.1 with:

$$\hat{\Xi} = Y\Psi^T (\Psi\Psi^T + \mu I)^{-1}, \quad (3.10)$$

where  $\mu$  is the regularization parameter. The Tikhonov regularization of the information matrix corresponds in the recursive case to the normalization of the covariance matrix  $P_k$ . When we let the RLS filter iterate for  $k = 0 \dots N - 1$ , the covariance matrix at time step  $N$  is described by:

$$P_{N-1}^{-1} = \lambda_1^N \rho_1 I + \sum_{k=0}^{N-1} \lambda_1^{N-k+1} \varphi_k \varphi_k^T, \quad (3.11)$$

where the affect of the initial regularization  $\rho_1$  fades away when  $N$  increases. To keep the covariance matrix normalized over time, we want a non-fading regularization as (Tsakiris et al., 2010):

$$\bar{P}_{N-1}^{-1} = \mu I + \sum_{k=0}^{N-1} \lambda_1^{N-k+1} \varphi_k \varphi_k^T. \quad (3.12)$$

When the effect of the initial regularization is neglected, the normalized covariance matrix is related to the non-normalized covariance matrix by:

$$\bar{P}_k = P_k (I + \mu P_k)^{-1}. \quad (3.13)$$

So Tikhonov regularization can be included by replacing  $P_k$  with  $\bar{P}_k$  in (3.2). However the computation is very costly for each iteration and still needs the ill-conditioned matrix  $P_k$  to be built. Although faster calculations are possible by



approximations, modification of the covariance matrix update equation is preferred instead. Consider the following update of the information matrix:

$$\bar{P}_k^{-1} = \lambda_1 \bar{P}_{k-1}^{-1} + \varphi_k \varphi_k^T + (1 - \lambda_1) \mu I. \quad (3.14)$$

By taking the inverse of (3.14) and by starting with  $\bar{P}_{-1} = \frac{1}{\mu} I$ , we can iterate the covariance matrix as (Gunnarsson, 1994):

$$\bar{P}_k = \frac{1}{\lambda_1} (\bar{P}_{k-1} - \bar{P}_{k-1} \varphi_k (\lambda_1 I + \varphi_k^T \bar{P}_{k-1} \varphi_k)^{-1} \varphi_k^T \bar{P}_{k-1}) + Q_1,$$

where

$$Q_1 = -(1 - \lambda_1) \mu \bar{P}_k (I - (1 - \lambda_1) \mu \bar{P}_k)^{-1} \bar{P}_k.$$

From this iteration it is obvious that the addition of  $Q_1$  directly affects the variation of the estimated Markov parameters. The introduction of Tikhonov regularization not only effects the conditioning of the covariance matrix but also the tracking ability of the algorithm, see also Gunnarsson (1996). It is however still a problem that  $Q_1$  is given by an implicit relation.

To have the computational benefits of low-rank updates, we can update the equation in (3.14) by adding  $(1 - \lambda_1) \mu$  in  $m = p(\ell + r) + r$  low-rank updates (Gay, 1995). If we consider only one additional low-rank update at each iteration, we get an approximated covariance matrix update as follows:

$$\bar{P}_k^{-1} = \lambda_1 \bar{P}_{k-1}^{-1} + \varphi_k \varphi_k^T + \xi_k \xi_k^T. \quad (3.15)$$

where

$$\xi_k = [\cdots \quad 0 \quad \underbrace{\sqrt{(1 - \lambda_1^m) \mu}}_{(k \bmod m) + 1 \text{th column}} \quad 0 \quad \cdots]^T.$$

By taking the inverse of (3.15) and by starting with  $\bar{P}_{-1} = \frac{1}{\mu} I$ , we can iterate the covariance matrix as:

$$\begin{aligned} \hat{P}_k &= \frac{1}{\lambda_1} (\bar{P}_{k-1} - \bar{P}_{k-1} \varphi_k (\lambda_1 I + \varphi_k^T \bar{P}_{k-1} \varphi_k)^{-1} \varphi_k^T \bar{P}_{k-1}), \\ \bar{P}_k &= \hat{P}_k - \hat{P}_k \xi_k (I + \xi_k^T \hat{P}_k \xi_k)^{-1} \xi_k^T \hat{P}_k. \end{aligned}$$

Using only one low-rank update at each iteration causes the regularization value to vary periodically with period  $m$ . The shape of the variation becomes a sawtooth wave with an amplitude of  $(1 - \lambda_1^m) \mu$ . If the forgetting factor  $\lambda < 1$  is chosen such that  $m \ll 2/(1 - \lambda_1)$ , one can obtain a reasonably steady amount of regularization.

### 3.3.2 Computational cost

For simplicity, Algorithm 3.1 shows the conventional implementation of the RLS filter. In Verhaegen (1989) it is recommended for practical use to implement a square-root type of implementation instead of the referred conventional RLS filter implementation due to better numerical robustness against round-off errors. The array methods are

powerful variants of square-root RLS that perform the computations in a reliable manner using a sequence of elementary Givens and/or hyperbolic rotations, see for stable implementations (Chandrasekaran and Sayed, 1996; Sayed, 2008). Another advantage is that array methods can be used to exploit any shifting structure in the data. This can become very beneficial if you have to select a large past window size. The fast-array RLS scheme in Algorithm 3.2 exploits the shifting structure of the data vector  $\bar{u}_{k-p}$  and  $\bar{y}_{k-p}$  in (2.19). This results for the main linear problem in a computational load in the order of  $\mathcal{O}((r+\ell)^2 p)$ , instead of  $\mathcal{O}((r+\ell)^2 p^2)$ . As Algorithm 3.2 shows, it is possible to implement  $r+\ell$  low-rank regularization updates in a similar shifted structure at the cost that the computational load becomes  $\mathcal{O}((r+\ell)^3 p)$ . Further, to fit the Tikhonov regularization in the shifted structure, the iteration starts with the regularization matrix in the form  $\bar{P}_{-1} = \frac{1}{\mu} \text{diag}(\lambda^{p-1} I, \lambda^{p-2} I, \dots, I)$  instead of  $\bar{P}_{-1} = \frac{1}{\mu} I$ .

**Algorithm 3.2** (The fast-array RLS filter with regularization).

**input**  $\Xi_{-1}, \lambda_1, \mu$

**require**  $0 \ll \lambda_1 \leq 1, \mu > 0$

**init**  $\gamma_{-1} = 1, \chi_{j,-1} = 1, \bar{L}_{-1} = \frac{1}{\sqrt{\mu}} \begin{bmatrix} 0 & \cdots & 0 & I \\ \sqrt{\lambda_1^p} & 0 & \cdots & 0 \end{bmatrix}^T$

**for**  $k = 0, 1, 2, \dots$

**input**  $u_k, y_k$

*Shift:*  $\varphi_k = [\varphi_{k-1}^T(r+\ell+1:(r+\ell)p, :) \quad y_{k-1}^T \quad u_k^T]^T$ .

*Find a  $J$ -orthogonal matrix  $\Phi$  using (hyperbolic) Givens rotations, with  $J = \text{diag}(1, -I, I)$ , such that:*

$$\begin{bmatrix} \gamma_{k-1}^{-1/2} & \frac{1}{\sqrt{\lambda_1}} \varphi_k^T \bar{L}_{k-1} \\ \begin{bmatrix} g_{k-1} \gamma_{k-1}^{-1/2} \\ 0 \end{bmatrix} & \frac{1}{\sqrt{\lambda_1}} \bar{L}_{k-1} \end{bmatrix} \Phi = \begin{bmatrix} \gamma_k^{-1/2} & 0 \\ \begin{bmatrix} 0 \\ g_k \gamma_k^{-1/2} \end{bmatrix} & \bar{L}_k \end{bmatrix}.$$

$$\bar{\Xi}_k = \bar{\Xi}_{k-1} + (y_k - \bar{\Xi}_{k-1} \varphi_k) ((g_k \gamma_k^{-1/2}) \gamma_k^{1/2})^T.$$

**if**  $\lambda_1 < 1$

**for**  $j = 1, 2, \dots, r+\ell$

**if**  $(k \bmod p) + 1 = p$

$$\varepsilon_{j,k} = [\cdots \quad 0 \quad \underbrace{\sqrt{(1-\lambda_1^p)\mu}}_{j\text{th column}} \quad 0 \quad \cdots]$$

**else**

$$\varepsilon_{j,k} = [0 \quad \cdots \quad 0]$$

*Shift:*  $\xi_{j,k} = [\xi_{j,k-1}^T(r+\ell+1:(r+\ell)p, :) \quad \varepsilon_{j,k}]^T$ .

*Find a  $J$ -orthogonal matrix  $\Phi$  using (hyperbolic) Givens rotations, with  $J = \text{diag}(1, -I, I)$ , such that:*

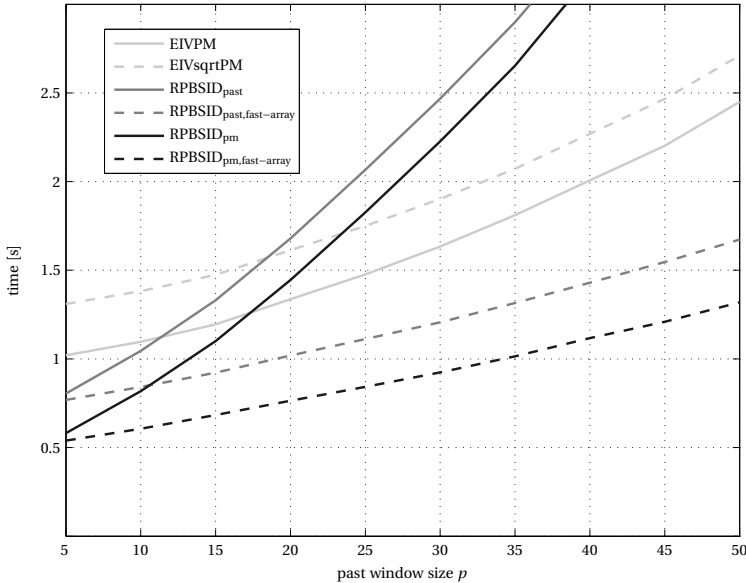
$$\begin{bmatrix} \chi_{j,k-1}^{-1/2} & \xi_{j,k}^T \bar{L}_k \\ \begin{bmatrix} h_{j,k-1} \chi_{j,k-1}^{-1/2} \\ 0 \end{bmatrix} & \bar{L}_k \end{bmatrix} \Phi = \begin{bmatrix} \chi_{j,k}^{-1/2} & 0 \\ \begin{bmatrix} 0 \\ h_{j,k} \chi_{j,k}^{-1/2} \end{bmatrix} & \bar{L}_k \end{bmatrix}.$$

The recommended inclusion of the transformation in Lemma 2.1 increases the computational cost of the method considerably, especially when the future window size  $f$  is large. Using Gauss-Jordan elimination for calculating the inverse of the matrix  $\tilde{\mathcal{H}}$ , the computational complexity of the multiplication becomes  $\mathcal{O}(\ell^3 f^2 p)$ . Fortunately, a faster algorithm can be obtained by exploiting the structure of the Markov parameters in the lifted matrices. In Dong et al. (2008), a fast algorithm is described to apply this transformation on (3.3) which is given by:

$$[\Gamma \tilde{\mathcal{L}} \quad \Gamma \tilde{\mathcal{K}}]^{(i)} = [\widetilde{\Gamma \mathcal{L}} \quad \widetilde{\Gamma \mathcal{K}}]^{(i)} + \sum_{j=1}^{i-1} (\tilde{\Xi}^{(y_{k-i+j})}) [\Gamma \tilde{\mathcal{L}} \quad \Gamma \tilde{\mathcal{K}}]^{(j)},$$

where  $[\cdot]^{(i)}$  with  $i = 1, \dots, f$  denotes the  $i$ th block row. The computational load is then only  $\mathcal{O}(\ell^2 f p)$ , which is a reduction compared with Gauss-Jordan elimination.

A comparison of the computational time needed for six RSID schemes to compute the system matrices with different past (and future) window sizes is given in Figure 3.1. The methods EIVPM and EIVsqrtPM (square-root version of the previous) are found in Mercère et al. (2008). The proposed method is denoted by  $\text{RPBSID}_{\text{pm}}$  where the solid line is the square-root implementation and the dashed line is the implementation with the fast-array method. Further we denote  $\text{RPBSID}_{\text{past}}$  as the method where the PM method is replaced with the PAST method to estimate the weight. From the figure, it is clearly visible that the proposed  $\text{RPBSID}_{\text{pm}}$  method with fast-array RLS has the smallest computation time, especially for large past window sizes. The  $\text{RPBSID}_{\text{past}}$  method is a bit slower due to the implementation of an additional adaptive filter.



**Figure 3.1:** Computational time of EIVPM, EIVsqrtPM,  $\text{RPBSID}_{\text{past}}$ , and  $\text{RPBSID}_{\text{pm}}$  (dashed line is with fast-array RLS) for different past window sizes.  $N = 1000$

### 3.4 Simulation study

In order to evaluate and compare the closed-loop performance of the proposed RSID method with the existing methods EIVPM (Mercère et al., 2008) and VPC (Wu et al., 2008), the state-space system from Mercère et al. (2008) is used:

$$\begin{aligned} x_{k+1} &= \begin{bmatrix} 0.8 & -0.4 & 0.2 \\ 0 & 0.3 & -0.5 \\ 0 & 0 & 0.5 \end{bmatrix} x_k + \begin{bmatrix} 0 & 0 \\ 0 & -0.6 \\ 0.5 & 0 \end{bmatrix} u_k + \begin{bmatrix} 0.055 & 0 & 0 \\ 0 & 0.05 & 0 \\ 0 & 0 & 0.045 \end{bmatrix} w_k, \\ y_k &= \begin{bmatrix} 0.5 & 0.5 & 0 \\ 0 & 0 & 1 \end{bmatrix} x_k + \begin{bmatrix} 0.025 & 0 \\ 0 & 0.03 \end{bmatrix} v_k, \end{aligned} \quad (3.16)$$

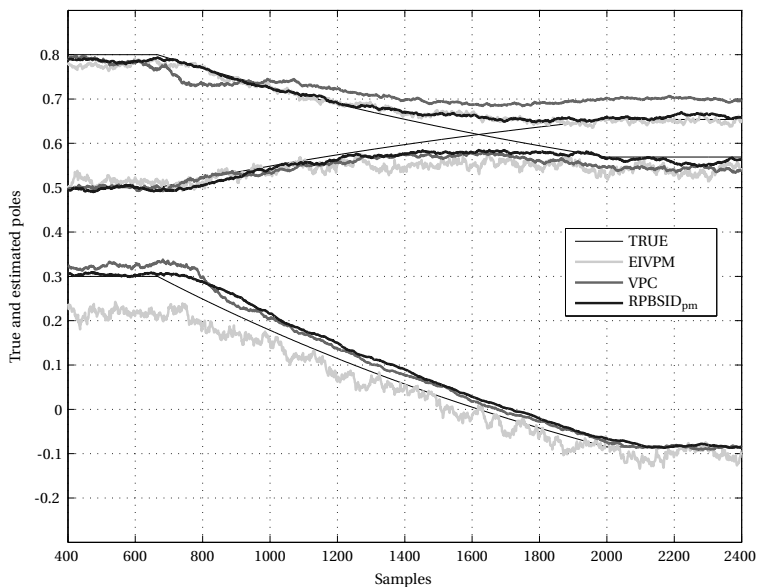
where  $v_k$  and  $w_k$  are white noise sequences of variance 1. To create a closed-loop system, a time-varying state-feedback control law is applied as described in Ogata (1994) that stabilizes the above system over the whole trajectory. For sufficient excitation, a white noise sequences of variance 1 is added to the input signal, therefore no regularization is required. The EIVPM (Mercère et al., 2008) method is a fast recursive version of the MOESP (Verhaegen and Verdult, 2007) method, where the QR factorization of the data matrix is updated by low-rank updates, the A and C matrix are obtained through an estimated observability matrix using the PM algorithm, and the B and D matrix can be obtained using an additional RLS algorithm. The VPC (Wu et al., 2008) method is a recursive version of the SSARX (Jansson, 2003) method, where the VARX parameters are obtained by the RLS algorithm, and the A, B, C, and D matrix are obtained through an estimated state vector using the PAST algorithm. Two practical situations are considered to evaluate and compare the tracking performance: a slowly time-varying case and an abrupt-change case.

#### 3.4.1 Slowly time-varying case

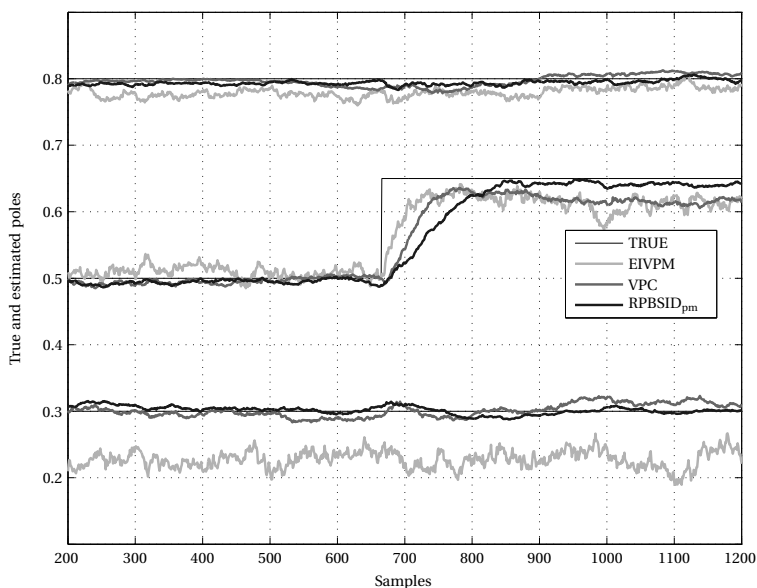
After a time-invariant phase of 665 samples, the following state matrix is used:

$$A + \text{diag}(-0.3, -0.5, 0.2) \frac{\exp(-(k-665)/2000) - 1}{\exp(-1) - 1}. \quad (3.17)$$

Thus the poles of the system drift from  $\{0.3, 0.5, 0.8\}$  during the next 1335 samples and then becomes time invariant again. The estimated pole trajectories averaged over the 100 MCS are displayed in Figure 3.2. As expected the open-loop EIVPM method gives biased results, because the system to be modelled operates in closed-loop. The EIVPM method cannot handle the problem that the future inputs are correlated with past noises. The VPC and the presented RPBSID<sub>pm</sub> method follow the eigenvalue trajectories much better. From the averaged responses, it visible that VPC has more difficulties and sometimes gives some biased results after the change, because the PAST algorithm changes the state basis. Further, it was observed from the simulations that the variance around the mean trajectories is much larger for the VPC method compared with the RPBSID<sub>pm</sub> method.



**Figure 3.2:** Trajectories of the estimated poles using EIVPM, VPC, and  $\text{RPBSID}_{\text{pm}}$  in a slowly changing environment. The following parameters are used:  $\lambda_1 = \lambda_2 = 0.98$ ,  $f = p = 5$ ,  $\text{SNR} = 25\text{dB}$ , and  $\text{MCS} = 100$ .



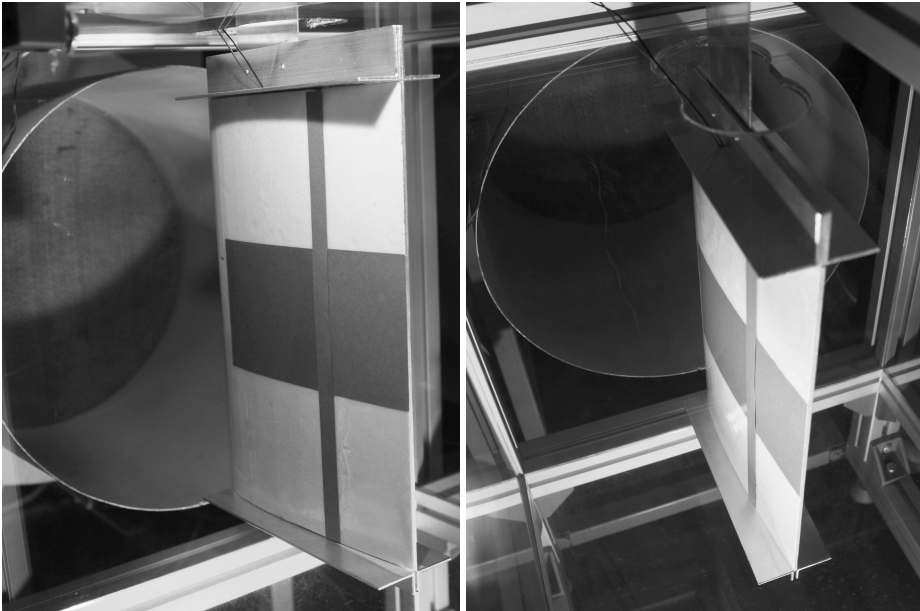
**Figure 3.3:** Trajectories of the estimated poles using EIVPM, VPC, and  $\text{RPBSID}_{\text{pm}}$  in an abrupt time-varying environment. The following parameters are used:  $\lambda_1 = \lambda_2 = 0.98$ ,  $f = p = 5$ ,  $\text{SNR} = 25\text{dB}$ , and  $\text{MCS} = 100$ .

### 3.4.2 Abrupt-change case

After a time-invariant phase of size 665 samples, the following abrupt change  $A(3,3) = 0.65$  in the state matrix is applied. Thus the pole 0.5 shifts to 0.65. The estimated pole trajectories averaged over the 100 Monte-Carlo Simulations (MCS) are displayed in Figure 3.3. Similar as in Subsection 3.4.1, the open-loop EIVPM method gives biased results. In the figure, VPC shows a faster convergence than  $\text{RPBSID}_{\text{pm}}$  in spite of some bias after the change. The PAST algorithm makes this method more adaptive directly after the abrupt change. This comes at the cost that the change in the state basis causes afterwards some loss in performance. Further, it is shown in simulation that the individual responses of  $\text{RPBSID}_{\text{pm}}$  are again much smoother.

## 3.5 Experimental study on 2D-airfoil system

In this section we compare the accuracy of the models from the proposed  $\text{RPBSID}$  method with different settings on a 2D-airfoil system operating in closed loop. The 2D-airfoil system can become unstable when the wind speed reaches a certain wind speed; the so-called flutter limit. First we present the experimental setup used to show the feasibility of flutter detection with the advanced identification method. Second, we describe the batchwise identification experiment to obtain an accurate model for the verification and control synthesis. Third, the proposed  $\text{RPBSID}$  method is applied on the 2D-airfoil system where the wind speed reaches the flutter limit.



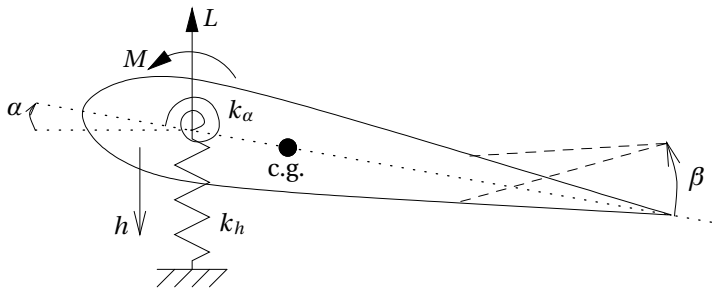
**Figure 3.4:** Photo of the 2D airfoil in front of the wind generator

### 3.5.1 Description of the experiment

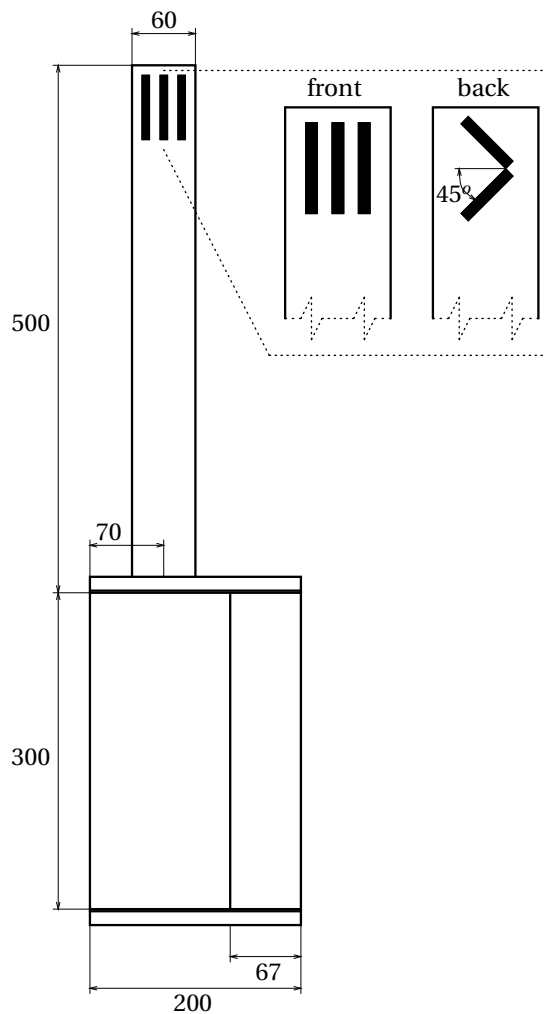
The experimental setup mainly consists of the following components: wind generator, blade, actuator and sensors, and real-time environment. The wind generator blows air with considerable turbulence into a tunnel section with the 2D airfoil hanging inside. The diameter of the outlet of the wind generator is 350mm and the tunnel section is 400mm by 400mm. The maximum attainable air speed is around 15m/s.

The 2D airfoil that we use for our experimental validation is an airfoil with at the trailing edge a control surface, the so-called trailing-edge flap. An illustration of the blade with trailing-edge flap is given in Figure 3.4. The blade is at the top connected to a half meter long aluminium plate with the other end fixed to a rigid frame. As stated in the name “2D airfoil” the blade has two degrees of freedom. The plate allows the blade only to move in the flapwise and torsional direction. The flutter limit of the 2D airfoil is around 12.5m/s. The exact flutter limit varies from day to day with 0.5m/s depending on the temperature and the density of the air. In Figure 3.5 and 3.6, schematic representations of the 2D-airfoil cross section and sideview are provided. The airfoil is shaped from soft foam and cut in two halves, where each halve is attached to a side of an aluminium plate and covered with a plastic skin to provide a smooth surface. The shape is based on the NACA 0012 airfoil with a chord of 200mm and a span of 300mm. The NACA 0012 airfoil is a symmetrical, low-drag airfoil, and 12% thick (thickness divided by chord). The total mass of the blade is 0.7kg. The dimensions of the aluminum plate are  $500 \times 60 \times 1$ mm. The flap also spans the 300mm and covers one third of the chord length (67mm). The active part of the trailing-edge flap consists of a THUNDER™ TH-7R actuator (FACE International). These are piezo-electric based benders which deflect several millimeters under the application of a voltage ranging from -300V to 700V. However, for safety reasons we limit the maximum voltage from -250V to 250V. The actual defection depends on the structure around the piezo-electric based bender and the aerodynamic loading.

For control purposes, the 2D airfoil is equipped with sensors which measure the dynamic behaviour of the blade. Since the final goal for this experiment is to detect flutter without destabilizing the 2D-airfoil system, two Macro Fiber Composite (MFC) (Smart Material) patches are adhered to the root at the backside of the half meter long aluminium plate under an angle of 45 degrees to measure the high strains associated with the first flapwise and torsional bending mode. Thus two strain sensors will measure approximately the same strain under flapwise bending, but will have an opposite sign under torsional bending. The main advantage of an MFC is that no amplification is required to have a good signal-to-noise ratio. However, with the MFC it is not possible to do static measurements due to the capacitance behaviour of the MFC. This high-pass behaviour is desirable for this experiment, as we want to control the dynamic behaviour of the system, rather than the static deformations. The trailing-edge flap does not stabilize the 2D-airfoil system when there is no control enabled. The controller intelligence and data acquisition capability are added with a dSPACE™ system. The controller and data acquisition scheme are fully developed in the MATLAB™ (Mathworks, a) and SIMULINK™ (Mathworks, b) environment and then compiled to the dSPACE™ (dSPACE) chip. On a separate computer, all the signals are monitored using CONTROL DESK™ (dSPACE) and also the control and identification parameters can be adjusted in real-time.



**Figure 3.5:** Schematic representation of the 2D airfoil cross section



**Figure 3.6:** Schematic representation of the 2D airfoil sideview. The dimensions are given in millimetres.



### 3.5.2 Experimental modelling and control

Due the instability of the 2D-airfoil system, we have to apply feedback control. The main issue is that the 2D-airfoil dynamics (the state matrix  $A$  and input matrix  $B$ ) are rather complex and strongly dependent on the free-stream wind speed and will necessitate model-based controller design. A model for modern model-based controller design is a mathematical model normally governed by (preferably linear) differential equations. For controller synthesis this model should only contain the relevant dynamics between the input and the outputs, and should be accurate over the bandwidth of the controller. Analytical modelling has been performed in the design stage using the theory in Theodorsen (1935), which strongly depends on a large number of parameters that determine the dynamic behaviour of the system. Most of these parameters can be roughly estimated or calculated. Still, a large amount of uncertainty is present; this makes it difficult to design a stable feedback controller based on such models. This motivates that predictor-based subspace identification can become a necessary building block for high bandwidth controller design.

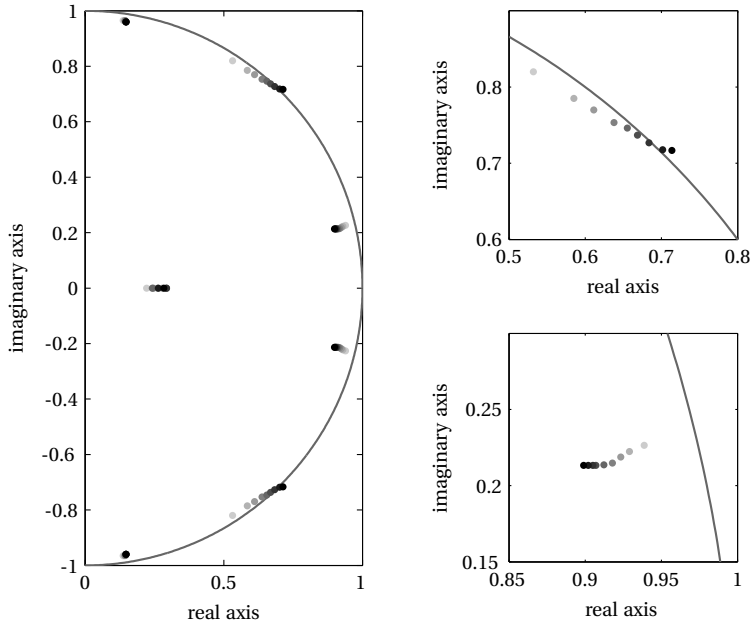
The control loops in the prototyped 2D-airfoil system are the transfer functions between the trailing-edge flap ( $u$ ), and the two backside MFC sensors ( $y$ ). Identification can be done for different operating points. First, we performed open-loop batchwise identification experiments for six stable operating conditions from 6m/s to 12.5m/s. The identified model with the operating point nearest to the flutter limit of 12.5m/s is used as an initial model for the model-based controller design in the unstable region. For this purpose, this state-space model of seventh order is converted to the so-called modal form, where the pole locations of the torsion mode are manually adjusted (by changing its frequency and damping value) to the expected unstable pole locations just outside the unit circle. Second, we performed the batchwise identification experiments in closed loop, including the unstable operating conditions 12.7m/s and 13m/s. With these newly identified and unstable models the model-based controller design is redone to obtain better stability and performance properties. The identified pole locations and Bode diagrams in the discrete time for a wind speed ranging from 9m/s to 13m/s are given in Figure 3.7 and Figure 3.8, respectively. The first resonance frequency is located between 0.7–0.8Hz and is related to the 1st flapwise bending mode. The second resonance frequency is located between 2.4–3.3Hz and is related to the 1st torsional bending mode. This is the mode that becomes unstable above 12.5m/s. The third resonance frequency is located at 4.5Hz and is related to the 2nd flapwise bending mode.

For the controller synthesis we use the so-called  $\mathcal{H}_2$  method which is illustrated in the well-known generalized plant setting in Figure 3.9. The goal of the feedback controller is to suppress the unknown disturbances as much as possible with the requirement that the system should remain stable and that the control input signal is bounded between  $-250V$  and  $+250V$ . The method does this by minimizing the  $\mathcal{H}_2$  norm between  $e_k$  and  $\{z_k, y_k\}$  and is mathematically given by:

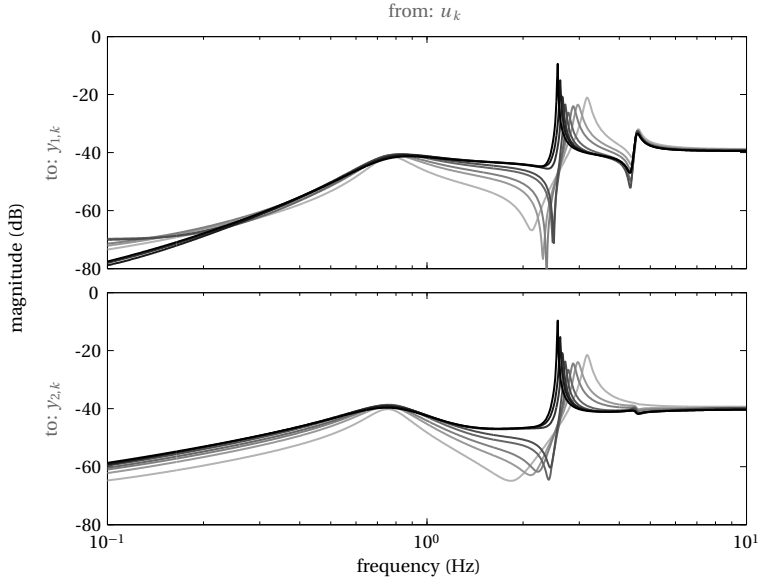
$$\min_c \left\| \begin{array}{c} W_U(I + CGW_L)^{-1}H \\ (I + GW_LC)^{-1}H \end{array} \right\|_2^2, \quad (3.18)$$

where

$$G(z) = D + C(zI - A)^{-1}B, \quad H(z) = R^{1/2} + C(zI - A)^{-1}KR^{1/2},$$

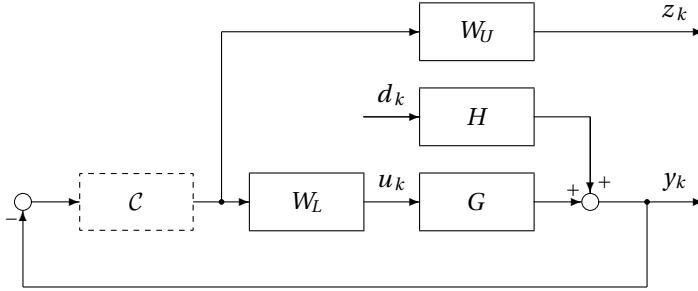


**Figure 3.7:** Identified pole locations, for a wind speed ranging between 9m/s (light grey) and 13m/s (black). The boxes to the right show a magnification of two pole locations. For  $\text{PBSID}_{\text{opt}}$ , closed loop,  $N = 15000$ .

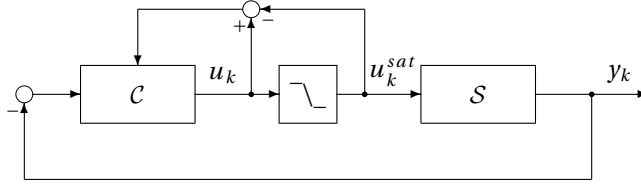


**Figure 3.8:** Identified Bode diagrams, for a wind speed ranging between 9m/s (light grey) and 13m/s (black). For  $\text{PBSID}_{\text{opt}}$ , closed loop,  $N = 15000$ .

where the covariance and system matrices of the innovation form in (2.2) can be obtained using the Algorithm 2.1. In these objective functions we can embed manually the loop shaping ideas in the weighting filters  $W_L$  and  $W_U$ . We choose them in such a way that the resulting controller will not act on high frequencies in the error signal by weighting the controller sensitivity by a filter with high gains at high frequencies. The  $\mathcal{H}_2$  controller is very effective at stabilizing, but it frequently encounters the actuator saturation when operating in the unstable region. During input saturation the closed-loop system can even become unstable. To overcome this problem, a so-called anti-windup feedback loop is designed, see Åström and Wittenmark (1997) and Figure 3.10, which takes action when the control signal saturates.



**Figure 3.9:** Block diagram of generalized plant to synthesize the  $\mathcal{H}_2$  controller



**Figure 3.10:** Block diagram of closed-loop system with anti-windup loop

### 3.5.3 Experimental results

In this subsection, the main results of the experiments on the 2D-airfoil system are presented. We present five cases that will discuss and show some features of the proposed recursive predictor-based subspace identification method. When the recursive predictor-based subspace identification method is applied on a system in real-time, a large number of parameters have to be tuned. We have the past and future window  $p$  and  $f$ , the selection matrix  $S$ , the forgetting factors  $\lambda_1$ ,  $\lambda_2$ , and  $\lambda_3$ , and the regularization parameter  $\mu$ . In the following five cases some tuning guidelines are provided by analysing and discussing the effects of these tuning parameters.

#### Effect of the past window size $p$

It is difficult to quantify the effect for the finite past window size  $p$ . First, there is the effect of the truncation error for the VARX predictor, which equals to the sum of

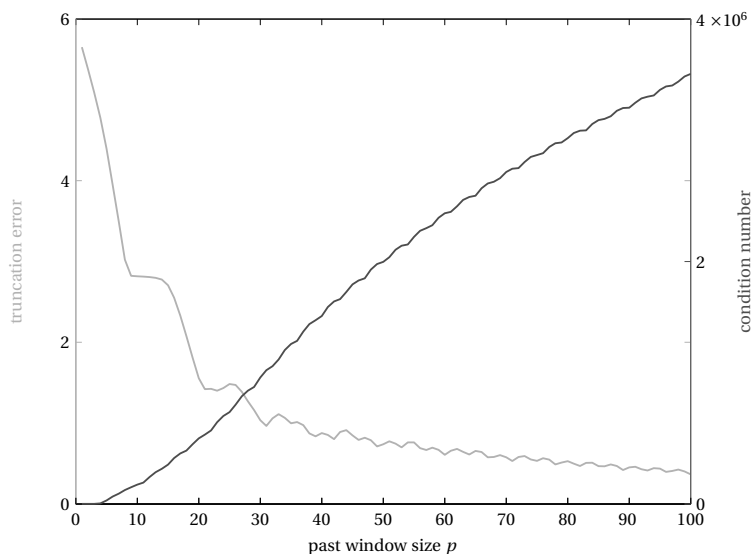
the remaining tail of Markov parameters. Second, the matrix becomes ill conditioned for large past window sizes, because normally in practice the input signals do not persistently excite the system. To quantify this effect, we use the so-called condition number of matrix  $\Psi\Psi^T$ , which equals the distance between the largest and smallest nonzero singular value. In Figure 3.11, the tradeoff between the truncation error and condition number is illustrated. Third, the number of Markov parameters to be estimated grows linearly with the past window size. It is well known that an increasing number of parameters to be estimated will increase the variance of the solution. Lastly, the computational time of the Markov estimation increases considerably when the past window becomes larger.

### Effect of the future window size $f$ and the weight matrix

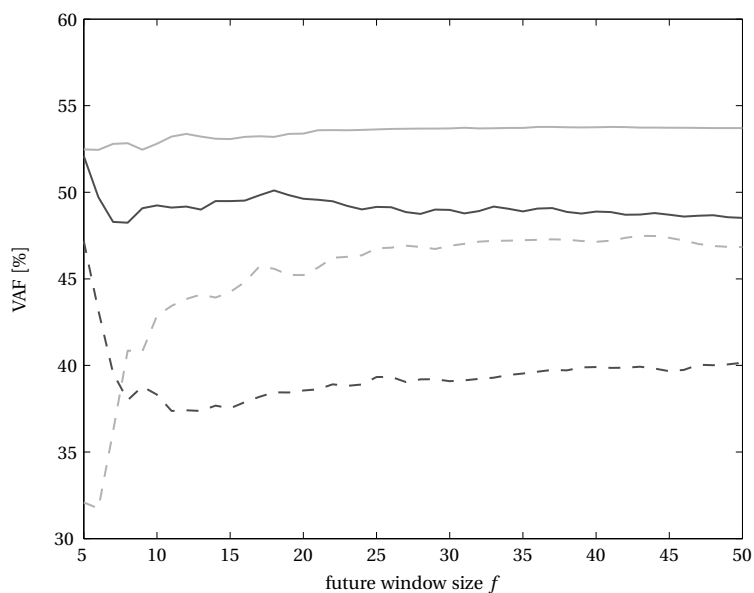
The selection of the future window size  $f$  is even more difficult. The effect of this parameter on the identification is known to heavily depend on the input spectrum and system properties. In Chiuso (2010), it is suggested through simulation studies that a large future window is often better for identification experiments when the Signal-to-Noise Ratio (SNR) is low, due to the so-called averaging effect. Otherwise for large SNR, the optimal future window size is normally the smallest value. However, from our experiments it is observed that the future window size does not only depend on the above properties, but also heavily on whether the transformation in Lemma 2.1 is applied.

From practical experience with different systems it is found that applying the transformation first with the weight matrix  $W_k = \tilde{H}_k^{-1}$  gives two advantages. First, applying the weight allows the user to select a smaller future window, which has a positive effect on the computational time. Second, the estimation of the system matrices behaves more smoothly over time, as the system matrices are estimated in a more constant state basis, especially when a lot of forgetting is applied. The construction of a predefined selection matrix  $S$  can be a challenge if the matrix  $\Gamma$  is not known in advance. As discussed in Mercère et al. (2008), for simple MISO systems the matrix  $S$  can easily be chosen as the identity matrix, such that the first  $n$  rows of  $S\Gamma$  are linearly independent, as the one output observes all the states. For MIMO systems this is generally not the case, because it is possible that one of the outputs does not observe a particular state. A solution is given in Mercère et al. (2008), where the selection matrix  $S$  is constructed in such a way that it replaces for each future step one of the outputs by a combination of the outputs. As described in Mercère and Bako (2011), this solution will bring the estimated state-space model close to canonical form. It is well known that the canonical form is very sensitive to numerical errors, therefore this solution was not very reliable for the system in practice. For the experiment, we construct the matrix  $\Gamma\tilde{L}\tilde{U}_p + \Gamma\tilde{K}\tilde{Y}_p$  from a batch of data and applied a rank-revealing QR decomposition. The row permutations show which rows are the most important. An alternative is to apply, as suggested in Chiuso et al. (2010), an SVD and take the first  $n$  columns of the left nullspace as:  $S = U_n$ .

In Figure 3.12 the effect of the future window size on the accuracy of the identification is illustrated for both the batchwise and the recursive case. The accuracy of the identification is evaluated by looking at the Variance-Accounted-For (VAF) on a data



**Figure 3.11:** The truncation error of a VARX predictor and the condition number of matrix  $\Psi\Psi^T$  against the past window size.  $N = 5000$ ,  $V = 11.7$



**Figure 3.12:** The VAF against the future window size. Solid lines are for  $\text{PBSID}_{\text{opt}}$  and dashed lines are for  $\text{RPBSID}_{\text{pm}}$ . The light grey is without and dark grey with transformation  $\tilde{\mathcal{H}}^{-1}$  applied.  $N = 5000$ ,  $V = 11.7$

set different from the data set used for determining the model. The VAF is defined as:

$$\text{vaf}(y_k, \hat{y}_k) = \max \left\{ 1 - \frac{\text{var}(y_k - \hat{y}_k)}{\text{var}(y_k)}, 0 \right\} \cdot 100\%,$$

where  $\hat{y}$  denotes the output signal obtained by simulating the identified model,  $y$  is the measured output signal, and  $\text{var}$  denotes the variance of a quasi-stationary signal. It is observed that when the transformation is not applied, the best accuracy for both the batchwise and the recursive case is obtained when the future window size is chosen close to the past window size. If the transformation is applied, then the optimal future window size is normally the smallest value. Moreover, in the recursive case the transformation with the smallest future window gives the best accuracy in most cases. For the batchwise case it is the other way around; the largest future window without transformation gives the best accuracy in most cases.

### Effect of the forgetting and the regularization

The choice of the forgetting factor is conceptually simple. Select it so that the covariance matrix contains more recent data that are relevant for the current dynamical representation of the system. The data within a window length of  $N = \frac{2}{(1-\lambda)}$  accounts for approximately 90% of the data contained in the covariance matrix, and therefore this relation can be used as an initial guess for the forgetting factor. The trade-off is between the tracking capability, which benefits from more forgetting, and the noise sensitivity, which improves with less forgetting. It can be noticed that the last two adaptive filters are updating their estimate with data containing possible errors from the adaptive filter before them. It is well known that least squares estimates are very sensitive to possible errors in the variables. Therefore the data is somewhat misused, especially in the initial transient period, and should therefore be less weighted than later measurements of which the information contents are more accurate. Further, we consider a constant forgetting factor for each adaptive filter. An adaptive choice of  $\lambda$  could be implemented, although it should also take into account that the changes in  $\lambda$  directly influence the size of the regularization.

If regularization is necessary, as in our experimental study, the Tikhonov regularization parameter  $\mu$  can be determined from a batch of data obtained around an operation point using the L-curve or Generalized Cross Validation (GCV) criterion, see Hansen (1992). When applying regularization it is strongly advised to scale the input and output signals, especially with MIMO systems. We have scaled for example the signals before the algorithm such that  $\text{var } u_k = 1$  and  $\text{var } y_k = 1$ . For recursive estimation, especially when the SNR changes considerably over time, as is the case in flutter experiment, the optimal regularization parameter changes considerably over time and should preferably be made adaptive. In Figure 3.13, the trend of the regularization parameter  $\mu$  using the GCV criterion is illustrated during the flutter detection experiment. Further, the regularization parameter can be lowered during estimation to increase the sensitivity to any changes, therefore it can also be used to alter the tracking capability of the algorithm.

### Batch vs. recursive identification

We compare the recursive identification scheme with the original batchwise identification scheme by looking at the accuracy of the estimated models. The identification is done with a Generalized Binary Noise (GBN) signal with an amplitude of 200V on the flap actuator. The data of 5000 samples is acquired with a sampling rate of 20Hz. The accuracy of the identification is evaluated by looking at the Variance-Accounted-For (VAF) on a data set different from the data set used for determining the model. The VAF values are lower for the higher wind speeds. This is caused by the fact that with similar excitation the contribution of the turbulence is larger in the measured signals and consequently the SNR is lower, which has a direct effect on the VAF.

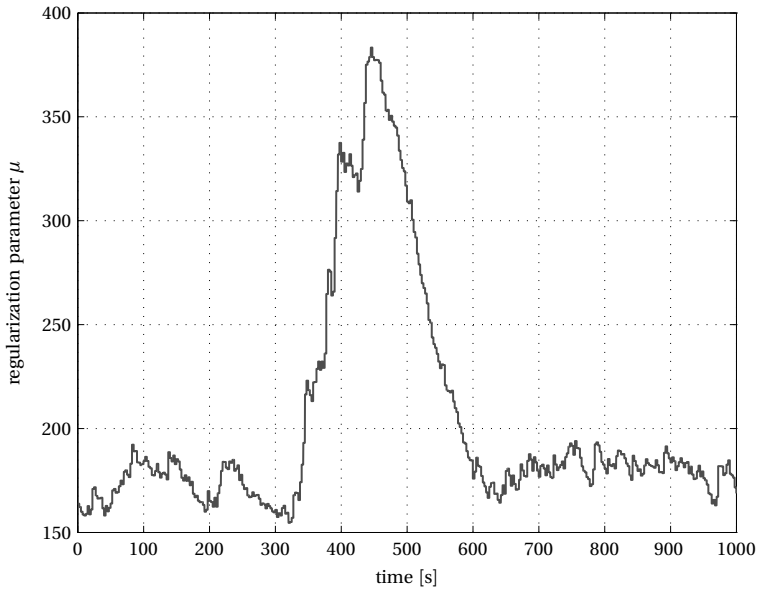
The results of batchwise and recursive identification experiments in the stable operating region can be found in Table 3.1. We see, as expected, that the batchwise identification scheme is superior to the recursive identification scheme for all the wind velocities. By comparing the experiments with and without transformation, we can see that applying the weight gives better results in the recursive case. In the batchwise case it is the opposite, without transformation gives the best VAF results. Thus the benefits of applying the transformation of Lemma 2.1 are only observed for the recursive predictor-based subspace identification method. The explanation is that applying the weight in the recursive algorithm causes the system matrices to be estimated in a constant basis, therefore giving a positive effect on the accuracy. This effect is not applicable for the batchwise algorithm.

**Table 3.1:** Variance-Accounted-For (VAF) for the stable operating points. N=5000

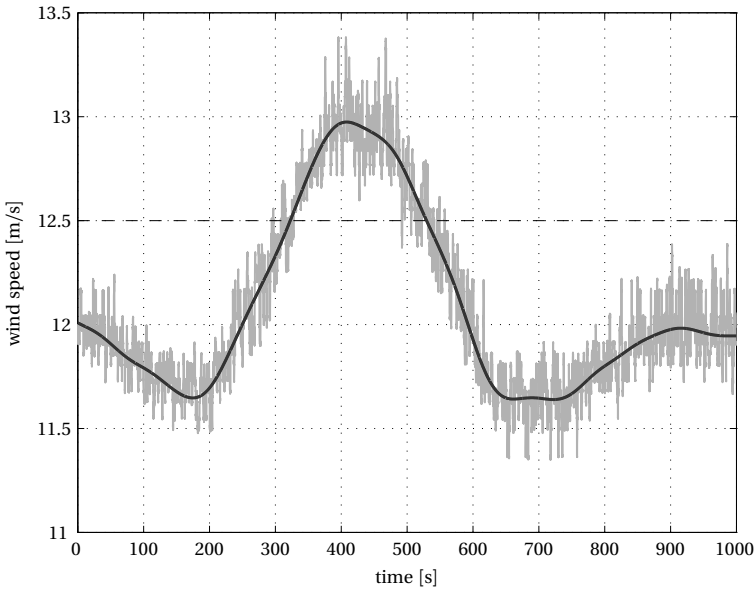
| VAF[%]  |      | PBSID <sub>opt</sub>      |        | RPBSID <sub>pm</sub>      |        |
|---------|------|---------------------------|--------|---------------------------|--------|
| V [m/s] | SNR  | $f=5, W=\mathcal{H}^{-1}$ | $f=50$ | $f=5, W=\mathcal{H}^{-1}$ | $f=50$ |
| 9.1     | 19.6 | 88.5                      | 88.7   | 87.9                      | 87.3   |
| 10.6    | 12.8 | 76.4                      | 76.6   | 76.1                      | 75.1   |
| 11.2    | 10.4 | 66.0                      | 66.2   | 65.8                      | 65.2   |
| 11.7    | 5.6  | 52.9                      | 53.7   | 47.8                      | 46.9   |
| 12.2    | 3.6  | 30.3                      | 35.0   | 30.1                      | 29.3   |
| 12.5    | 2.1  | 17.4                      | 17.5   | 11.5                      | 9.4    |

### Tracking of the flutter dynamics

Introducing forgetting in the recursive predictor-based subspace identification problem makes the recursive method suitable for detecting small and slow changes in the flutter dynamics. We show the effectiveness of the proposed algorithm to this application. With the wind generator a gust of wind is created according to the wind-speed profile with period of  $T = 900$ s and specified in IEC (2004), see also the measured air velocity given in Figure 3.14. As the wind speed crosses the flutter limit of 12.5m/s, the identification experiment is done in closed-loop. Using the modern model-based controller synthesis presented in Subsection 3.5.2, a robustly stabilizing controller is designed for the operating region between 11m/s and 13m/s. The identification experiment is done with an additional external GBN signal  $r_k$  with an amplitude of 100V, see Figure 2.1. The data is resampled to a sampling rate of 20Hz.



**Figure 3.13:** Regularization parameter  $\mu$  over time using the GCV criterion. The value of the regularization parameter is updated every 2.5s (50 samples).



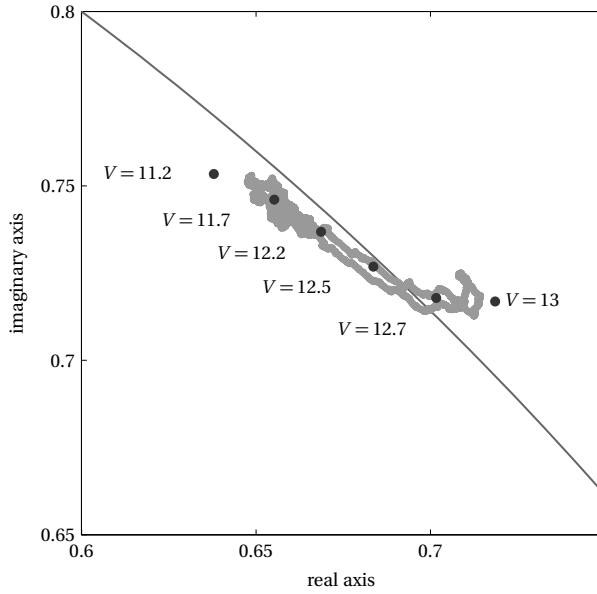
**Figure 3.14:** Measured wind-speed profile (light grey) and filtered wind-speed profile (dark grey) of the applied wind gust. The dashed line resembles the flutter limit. Wind speeds higher than the limit will bring the airfoil to flutter.



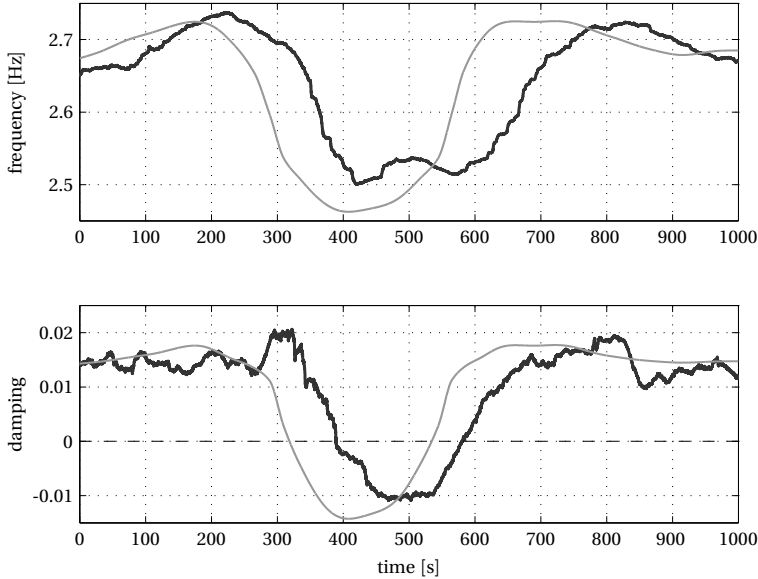
The torsional bending is the mode of interest, because this mode becomes unstable during flutter. In Figure 3.15, we can see pole locations of the 1st torsional bending mode during the wind gust using the  $\text{RPBSID}_{\text{pm}}$  algorithm compared with the pole locations obtained from batchwise identification experiments around multiple operating points. From this figure we can see that the recursively estimated pole locations follow the expected pole trajectory quite closely. In Figure 3.16, we can see the estimated natural frequencies and modal damping values during the wind gust. The torsional bending frequency is smoothly estimated with a delay between 50–100s compared with the expected frequencies values. The  $\text{RPBSID}_{\text{pm}}$  method estimates the damping values less smoothly, but the damping values take as expected a negative value, again with a delay between 50–100s, when the wind speed crosses the flutter limit. Thus after some delay, the  $\text{RPBSID}_{\text{pm}}$  algorithm successfully detects the instability when the 2D airfoil is operating in flutter. The large delay time was expected, because of the low SNR a high value for the forgetting factor was selected to overcome the sensitivity to the noise. It is also noted that the delay time builds up in threefold, because the three RLS filters are in cascade. The delay time can be reduced considerably by selecting a lower forgetting factor, or in smaller amount by selecting a lower regularization value, at the cost of a less accurate estimation. To lower the delay time, one could also investigate the use of flutter detection algorithms where the damping values are not directly calculated from state space, but rather a statistical threshold is derived to determine if the system is in flutter, for example as done in Mevel et al. (2003). As described in Basseville et al. (2007), these flutter detection algorithms have considerable smaller delay time compared with online subspace identification, however less accuracy and robustness to turbulent excitation.

### 3.6 Conclusion

In this chapter we presented a novel recursive predictor-based subspace identification method which is a recursive version of the "optimized" predictor-based subspace identification method for batch data. The recursive identification is performed via the subsequent solution of only three linear problems, which are solved using recursive least squares. The recursive implementation of the method is not only able to identify linear time-invariant models from measured data in open or closed loop, but can also be used to track slowly time-varying dynamics if adaptive filters are used. The computational complexity is reduced by exploiting the structure in the data equations and by using array algorithms to solve the main linear problem, such that the proposed method can be implemented in real-time. We have shown in the experiments that a certain weight with transformation is required to minimize variations of the particular state used and provided some guidelines on how to pick the free parameters in the algorithm. Moreover, the chapter suggests to use Tikhonov regularization in order to overcome possible numerical problems in practice (and therefore give better results in some sense), such as the singularity of the regression problem. The real-time implementation and the ability to work with multi-input and multi-output systems operating in closed loop make this approach suitable for online estimation of unstable dynamics. Flutter detection is one of the applications of this approach and we have demonstrated this on an experimental 2D-airfoil system. We can track after some delay the natural frequencies and modal damping values.



**Figure 3.15:** The light-grey dots are the pole locations of the 1st torsional bending mode during the wind gust with  $\text{RPBSID}_{\text{pm}}$ ,  $W = \tilde{\mathcal{H}}^{-1}$ ,  $f = 5$ ,  $\lambda_1 = 0.9993$ ,  $\lambda_{2,3} = 0.9992$ . The dark-grey dots are the pole locations from the  $\text{PBSID}_{\text{opt}}$  experiments around multiple operating points in Figure 3.7.



**Figure 3.16:** The dark-grey lines are the frequencies (top) and damping values (bottom) of the 1st torsional bending mode during the wind gust with  $\text{RPBSID}_{\text{pm}}$ ,  $W = \tilde{\mathcal{H}}^{-1}$ ,  $f = 5$ ,  $\lambda_1 = 0.9993$ ,  $\lambda_{2,3} = 0.9992$ . The light-grey lines are the interpolated frequencies (top) and damping values (bottom) calculated from the pole locations from the  $\text{PBSID}_{\text{opt}}$  experiments in Figure 3.7.

## Rejection of Periodic Wind Disturbances on an Experimental Smart Rotor Test Section

---

A repetitive control method is presented that is implemented in real time for periodic wind disturbance rejection for linear systems with multiple inputs and multiple outputs and with both repetitive and non-repetitive disturbance components. The novel repetitive controller can reject the periodic wind disturbances for fixed-speed wind turbines and variable-speed wind turbines operating above-rated and we will demonstrate this on an experimental “smart” rotor test section. The “smart” rotor is a rotor where the blades are equipped with a number of control devices that locally change the lift profile on the blade, combined with appropriate sensors and controllers. The rotational speed of wind turbines operating above-rated will vary around a defined reference speed, therefore methods are given to robustify the repetitive controllers for a mismatch in the period. The design of the repetitive controller is formulated as a lifted linear stochastic output-feedback problem on which the mature techniques of discrete linear control may be applied. For real-time implementation, the computational complexity can be reduced by exploiting the structure in the lifted state-space matrices. With relatively slow changing periodic disturbances it is shown that this repetitive control method can significantly reduce the structural vibrations of the “smart” rotor test section. The cost of additional wear and tear of the “smart” actuators are kept small, because a smooth control action is generated as the controller mainly focuses on the reduction of periodic disturbances.

---

### 4.1 Introduction

THE load disturbances acting on an individual wind turbine blade are to a large extent deterministic; for instance, tower shadow, wind shear, yawed error, unbalance, and gravity, and they depend on the rotation angle and speed. The contribution

of these periodic disturbances is becoming even larger, due to the increasing length and mass of the rotor blades. Repetitive controllers consist of a periodic signal generator, enabling rejection of only these periodic disturbances. Repetitive Control (RC) methods for discrete Multiple-Input and Multiple-Output (MIMO) systems that can potentially expand the application to wind turbines greatly have been developed in the time-domain, like in de Roover et al. (2000), and in the trial domain using lifted models, see Lee et al. (2001). A recent overview of past literature in RC can be found in Wang et al. (2009). RC methods can achieve perfect reduction in the deterministic case, when excluding non-repetitive disturbances. In Rice and Verhaegen (2010b), a lifted RC method is presented for MIMO linear systems with both repetitive and non-repetitive disturbance components. This design method is formulated as a lifted linear stochastic output-feedback problem on which the mature techniques of discrete linear control may be applied. The computational complexity is even considerably reduced by exploiting the structure in the lifted state-space matrices.

An important drawback of RC is that a small mismatch between the controller period and the actual period of the disturbance signal can decrease the performance of RC substantially. Several approaches have been proposed in the literature to improve robustness and are summarized in Gupta and Lee (2006). These approaches can be divided into two groups, depending if the true rotor speed is accurately measured or not known. In this paper, we focus on the latter as this is the case with the “smart” rotor test section. Although it is expected that RC performs better with the approaches that require the rotor speed to be measured. In Rice and Verhaegen (2010b), a weight is included in the formulated lifted output-feedback problem to specify the amount of variation in the periodic wind disturbances. In Gupta and Lee (2006); Pipeleers et al. (2008); Steinbuch (2002); Steinbuch et al. (2007) the RC is extended with more memory loops. This so-called high-order RC can be made more robust for small changes in period time, but this comes at the cost of the amplification of noise at the non-harmonic frequencies. Using these developments, the application of RC to wind turbines with small variations in rotational speed, such as fixed-speed and variable-speed operating above-rated with a pitch controller that will keep the rotational speed of the wind turbine close to the reference speed, becomes more practical.

The contributions of this chapter are as follows. First, we propose a RC method which can deal with periodic wind disturbances, and leave the non-periodic disturbances (turbulence) mostly unaffected. This will generate a smooth control action that will keep the additional wear and tear on the “smart” actuators small. Second, we implement and study two extensions (random walk weight, multiple memory loops) in the lifted output-feedback problem to robustify the performance against period mismatch. Third is the experimental verification of RC on a “smart” rotor test section with periodic and turbulent wind disturbances generated by a wind generator. In addition, we verify the capability of RC to operate with an additional feedback controller, which is present in modern wind turbines for stabilization. To the author’s knowledge, this is the first experiment with an implemented lifted repetitive controller to reduce the periodic wind disturbances (with small variations in the period time) in a turbulent wind field, although repetitive controllers in the time-domain for disturbance rejection have been implemented in real time before in Pipeleers et al. (2009); Steinbuch et al. (2007); Tammi et al. (2007).

The outline of this chapter is as follows. In Section 4.2 the problem formulation, the assumptions made, and some notations are presented. In Section 4.3 the theoretical framework is presented for the RC problem and an algorithm for MIMO Linear Time-Invariant (LTI) systems is given. In Section 4.4 the effectiveness of the proposed algorithm is shown with an experimental study of disturbance rejection on a “smart” rotor test section. Moreover, some tuning guidelines are derived. In the final section we present the conclusions of this chapter.

## 4.2 System description, assumptions and notations

In this section the system description, the assumptions made, and some notation is presented.

### 4.2.1 System description and assumptions

The dynamics of the discrete-time system  $S$  to be controlled can be written as the following state-space model:

$$S \begin{cases} x_{k+1} = Ax_k + Bu_k + Fd_k + w_k, \\ y_k = Cx_k + Du_k + v_k, \end{cases} \quad (4.1)$$

where  $x_k \in \mathbb{R}^n$ ,  $u_k \in \mathbb{R}^r$ ,  $y_k \in \mathbb{R}^\ell$ ,  $d_k \in \mathbb{R}^f$ ,  $w_k \in \mathbb{R}^n$ , and  $v_k \in \mathbb{R}^\ell$ , are the state, input, output, periodic disturbance, process noise, and measurement noise vectors. The state-space matrices  $A \in \mathbb{R}^{n \times n}$ ,  $B \in \mathbb{R}^{n \times r}$ ,  $F \in \mathbb{R}^{n \times f}$ ,  $C \in \mathbb{R}^{\ell \times n}$ , and  $D \in \mathbb{R}^{\ell \times r}$  are also called the state, input, periodic input, output, and direct feedthrough matrices, respectively. For simplification and clarification, we consider an LTI system. However the lifted repetitive control method described in this chapter can be extended to periodic time-varying systems.

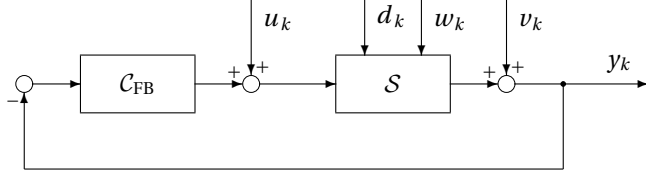
The process and measurement noise disturbances are considered Gaussian noise sequences with  $E[v_k v_k^T] = R_v$  and  $E[w_k w_k^T] = R_w$ , respectively. In the case of coloured Gaussian noise sequences, the state-space system  $S$  can be augmented with the noise filter dynamics. The periodic disturbance may consist of an integrated “random walk” component as (Chin et al., 2004; Cho et al., 2005; Lee et al., 2001):

$$d_k = d_{k-p} + n_k, \quad (4.2)$$

where  $n_k$  is a Gaussian noise sequence with  $E[n_k n_k^T] = R_n$ , and  $p \in \mathbb{N}^+$  is the trial window size (or period). This is a quite versatile noise formulation; the disturbance contains not only a stochastic and a periodic component, but also a random walk nature, so the “periodic disturbance” can vary between trials. In addition, the noise and periodic disturbances can be correlated in time.

We consider also the possibility that the system  $S$  already operates in closed loop with a feedback controller for stabilization or load reduction due to turbulence. A condition is that the feedback controller does not have poles on the unit disk, thus the feedback controller should let the repetitive controller deal with the repetitive

disturbances. In this chapter, we consider the repetitive control input  $u_k$  as an additional input signal fed to the system, see Figure 4.1. As discussed in De Cuyper (2006), the repetitive control input  $u_k$  could also be fed as a reference signal to the feedback controller, but this can require a redesign of the feedback controller to include reference tracking capabilities. For a successful RC design, the state-space description in (4.1) should in both cases be replaced by its closed-loop state-space description. Thus the system  $S$  should also be augmented with a state-space description of the feedback controller  $C_{FB}$ .



**Figure 4.1:** Block diagram of the system  $S$  operating in closed loop

#### 4.2.2 Lifted system description and assumptions

With the trial window size  $p$ , the following stacked vector can be defined as:

$$\bar{y}_{k,p} = [y_k^T \quad y_{k+1}^T \quad \cdots \quad y_{k+p-1}^T]^T. \quad (4.3)$$

The stacked vectors  $\bar{u}_{k,p}$ ,  $\bar{d}_{k,p}$ ,  $\bar{n}_{k,p}$ ,  $\bar{w}_{k,p}$ , and  $\bar{v}_{k,p}$  are defined in a similar way. Using the definition of the stacked vectors, we can “lift” the system in (4.1) to the following lifted description:

$$\bar{S} \begin{cases} x_{k+p} &= A^p x_k + \mathcal{L} \bar{u}_{k,p} + \mathcal{F} \bar{d}_{k,p} + \mathcal{P} \bar{w}_{k,p}, \\ \bar{y}_{k,p} &= \Gamma x_k + \mathcal{G} \bar{u}_{k,p} + \mathcal{J} \bar{d}_{k,p} + \mathcal{H} \bar{w}_{k,p} + \bar{v}_{k,p}, \end{cases} \quad (4.4)$$

where  $\mathcal{L} \in \mathbb{R}^{n \times pr}$ ,  $\mathcal{F} \in \mathbb{R}^{n \times pf}$ , and  $\mathcal{P} \in \mathbb{R}^{n \times pn}$  are the extended controllability matrices, and are given by:

$$\mathcal{L} = [A^{p-1}B \quad \cdots \quad AB \quad B], \quad \mathcal{F} = [A^{p-1}F \quad \cdots \quad AF \quad F], \quad \mathcal{P} = [A^{p-1} \quad \cdots \quad A \quad I],$$

and  $\mathcal{G} \in \mathbb{R}^{p\ell \times pr}$ ,  $\mathcal{J} \in \mathbb{R}^{p\ell \times pf}$ , and  $\mathcal{H} \in \mathbb{R}^{p\ell \times pn}$  are the impulse matrices with a lower block triangular structure,  $\Gamma \in \mathbb{R}^{p\ell \times n}$  is the extended observability matrix, and are given by:

$$\Gamma = \begin{bmatrix} C \\ CA \\ \vdots \\ CA^{p-1} \end{bmatrix}, \quad \mathcal{H} = \begin{bmatrix} 0 & 0 & \cdots & 0 \\ C & 0 & \ddots & \vdots \\ \vdots & \ddots & \ddots & 0 \\ CA^{p-2} & CA^{p-3} & \cdots & 0 \end{bmatrix},$$

$$\mathcal{G} = \begin{bmatrix} D & 0 & \cdots & 0 \\ CB & D & \ddots & \vdots \\ \vdots & \ddots & \ddots & 0 \\ CA^{p-2}B & CA^{p-3}B & \cdots & D \end{bmatrix}, \quad \mathcal{J} = \begin{bmatrix} 0 & 0 & \cdots & 0 \\ CF & 0 & \ddots & \vdots \\ \vdots & \ddots & \ddots & 0 \\ CA^{p-2}F & CA^{p-3}F & \cdots & 0 \end{bmatrix}.$$

These lifted system matrices are considered trial invariant. Note that  $I$  is used to represent an identity matrix of appropriate dimensions.

In Assumption 4.1, we adopt the mild conditions commonly used for the lifted repetitive control problem (Rice and Verhaegen, 2010b).

**Assumption 4.1.**

- The system  $\tilde{S}$  is asymptotically stable (if not, stabilize with feedback control), controllable on  $(A^p, \mathcal{L})$ , observable on  $(A^p, \Gamma)$ , and does not contain any zeros at  $+1$ .
- The process noise  $w_k$  and measurement noise  $v_k$  are zero-mean white or coloured (if so, augment dynamics to system) Gaussian sequences.
- The “random walk” noise  $n_k$  is a zero-mean white Gaussian sequence.

When the number of inputs is smaller than the number of outputs ( $r < \ell$ ), the controllability/observability condition in Assumption 4.1 is normally violated. In this case the matrix  $\mathcal{G} + \Gamma(I - A^p)^{-1} \mathcal{L}$  does not have full row rank. It is proposed to only reduce the error that lies in the column space of that matrix, thus we try not to achieve perfect rejection for all disturbance signals (or all the harmonics). To achieve this, we use the Singular Value Decomposition (SVD) as:

$$\mathcal{G} + \Gamma(I - A^p)^{-1} \mathcal{L} = \begin{bmatrix} U_s & U_\perp \end{bmatrix} \begin{bmatrix} \Sigma_s & 0 \\ 0 & \Sigma \end{bmatrix} \begin{bmatrix} V_s \\ V_\perp \end{bmatrix}, \quad (4.5)$$

where the order  $s$  is determined by detecting a gap that separates the  $s$  largest singular values from the remaining ones. Now, we can replace the output equation of the lifted system in (4.4) by a projected output equation given by:

$$U_s^T \tilde{y}_{k,p} = U_s^T \Gamma x_k + U_s^T \mathcal{G} \tilde{u}_{k,p} + U_s^T \mathcal{J} \tilde{d}_{k,p} + U_s^T \mathcal{H} \tilde{w}_{k,p} + U_s^T \tilde{v}_{k,p}. \quad (4.6)$$

Furthermore, we can select the number of singular values  $s$  not only on the controllability/observability condition, but also to further reduce the size of the RC output-feedback problem and let the lifted repetitive controller focus on the main harmonic components and therefore robustify the controller performance in some sense; see also Subsection 4.4.3.

## 4.3 Repetitive control with lifted LQG design

In this section the theoretical framework for the repetitive control method to reject periodic disturbances is presented. The goal is to formulate the RC design problem into an output-feedback problem. After the formulation, the state-space matrices of the lifted repetitive controller can be calculated by solving two Riccati equations.

### 4.3.1 Output-feedback formulation

To let the repetitive controller respond to periodic disturbances, a “non-stochastic” output from (4.4) is defined as:

$$\tilde{y}_{k,p} = \bar{y}_{k,p} - \mathcal{H}\bar{w}_{k,p} - \bar{v}_{k,p} = \Gamma x_k + \mathcal{G}\bar{u}_{k,p} + \mathcal{J}\bar{d}_{k,p}, \quad (4.7)$$

and a “non-stochastic” error is defined, which is independent of the noise sequence from trial  $k$  ( $\bar{w}_{k,p}$ ,  $\bar{v}_{k,p}$ ) as:

$$\tilde{e}_{k,p} = \bar{r}_{k,p} - \tilde{y}_{k,p}, \quad (4.8)$$

where  $\bar{r}_{k,p}$  is considered trial invariant. The change between two trials is denoted by the  $\Delta$  operator, such that for example the difference between the vector of the input from time  $k$  and the vector of the input from time  $k-p$  is defined as  $\Delta\bar{u}_{k,p} = \bar{u}_{k,p} - \bar{u}_{k-p,p}$ . Using this definition, it follows that the “non-stochastic” tracking error is defined as:

$$\tilde{e}_{k,p} = \bar{e}_{k-p,p} - \Gamma\Delta x_k - \mathcal{G}\Delta\bar{u}_{k,p} - \mathcal{J}\bar{n}_{k,p}. \quad (4.9)$$

An expression for the current “stochastic” tracking error based on the previous “non-stochastic” error can be obtained and is given by:

$$\bar{e}_{k,p} = \tilde{e}_{k,p} - \mathcal{H}\bar{w}_{k,p} - \bar{v}_{k,p}. \quad (4.10)$$

By using the  $\Delta$  operator again on the state as  $\Delta x_k = x_k - x_{k-p}$ , the difference in the initial conditions of  $\tilde{S}$  between trials is derived from (4.7), and (4.2) as:

$$\Delta x_{k+p} = A^p \Delta x_k + \mathcal{L}\Delta\bar{u}_{k,p} + \mathcal{F}\bar{n}_{k,p} + \mathcal{P}\Delta\bar{w}_{k,p}. \quad (4.11)$$

Now we can combine (4.9), (4.10), and (4.11) into a stochastic linear system description as:

$$\tilde{S}'_{RC} \begin{cases} \xi'_{k+p} &= \mathcal{A}'\xi'_k + \mathcal{B}'\Delta\bar{u}_{k,p} + \mathcal{B}'_w\bar{v}_{k,p}, \\ \bar{e}_{k,p} &= \mathcal{C}'\xi'_k + \mathcal{D}\Delta\bar{u}_{k,p} + \mathcal{D}_w\bar{v}_{k,p}, \end{cases} \quad (4.12)$$

where  $'$  denotes the system is differenced once, the vectors  $\xi'_k \in \mathbb{R}^{p(\ell+n)+n}$ , and  $\bar{v}_{k,p} \in \mathbb{R}^{p(f+n+\ell)}$  are given by:

$$\xi'_k = [\Delta x_k^T \quad \bar{e}_{k-p,p}^T \quad \bar{w}_{k-p,p}^T]^T, \quad \bar{v}_{k,p} = [\bar{n}_{k,p}^T \quad \bar{w}_{k,p}^T \quad \bar{v}_{k,p}^T]^T,$$

and the system matrices  $\mathcal{A}' \in \mathbb{R}^{p(\ell+n)+n \times p(\ell+n)+n}$ ,  $\mathcal{B}' \in \mathbb{R}^{p(\ell+n)+n \times pr}$ ,  $\mathcal{C}' \in \mathbb{R}^{p\ell \times p(\ell+n)+n}$ ,  $\mathcal{D} \in \mathbb{R}^{p\ell \times pr}$ ,  $\mathcal{B}'_w \in \mathbb{R}^{p(\ell+n)+n \times p(f+n+\ell)}$ , and  $\mathcal{D}_w \in \mathbb{R}^{p\ell \times p(f+n+\ell)}$  are given by:

$$\mathcal{A}' = \begin{bmatrix} A^p & 0 & -\mathcal{P} \\ -\Gamma & I & 0 \\ 0 & 0 & 0 \end{bmatrix}, \quad \mathcal{B}' = \begin{bmatrix} \mathcal{L} \\ -\mathcal{G} \\ 0 \end{bmatrix}, \quad \mathcal{C}' = [-\Gamma \quad I \quad 0], \quad \mathcal{D} = -\mathcal{G},$$

$$\mathcal{B}'_w = \begin{bmatrix} \mathcal{F} & \mathcal{P} & 0 \\ -\mathcal{J} & 0 & 0 \\ 0 & I & 0 \end{bmatrix}, \quad \mathcal{D}_w = -[\mathcal{J} \quad \mathcal{H} \quad I].$$



### 4.3.2 Multiple memory loops

As described in Pipeleers et al. (2008); Steinbuch (2002); Steinbuch et al. (2007), additional memory loops can be introduced to make the RC more robust for small changes in period time or less sensitive to disturbances at non-periodic frequencies. This high-order RC output-feedback problem can be formulated in an almost similar state-space system as in (4.12). In Gupta and Lee (2006) it was shown that the multiplication of the multiple memory loops with the repetitive controller is commutative in the lifted setting. This means that an additional memory loop can be introduced in the output-feedback problem by double-differencing the input. Let's denote the double-differenced input as:

$$\Delta^2 \bar{u}_{k,p} = \bar{u}_{k,p} - \theta_1 \bar{u}_{k-p,p} - \theta_2 \bar{u}_{k-2p,p},$$

where  $\theta_1 \in \mathbb{R}$  and  $\theta_2 \in \mathbb{R}$  with  $\theta_1 + \theta_2 = 1$ . For example, in Steinbuch et al. (2007) it was shown that  $\theta_1 = 2$  and  $\theta_2 = -1$  gives the best performance against period mismatch, and that  $\theta_1 = 2/3$  and  $\theta_2 = 1/3$  gives the lowest sensitivity to disturbances at non-periodic frequencies. The equations in (4.9), and (4.11) are differenced again as:

$$\bar{e}_{k,p} = \theta_1 \bar{e}_{k-p,p} + \theta_2 \bar{e}_{k-2p,p} - \Gamma \Delta^2 x_k - \mathcal{G} \Delta^2 \bar{u}_{k,p} - \mathcal{J} \bar{n}_{k,p} - \theta_2 \mathcal{J} \bar{n}_{k-p,p}, \quad (4.13)$$

and

$$\Delta^2 x_{k+p} = A^p \Delta^2 x_k + \mathcal{L} \Delta^2 \bar{u}_{k,p} + \mathcal{F} \bar{n}_{k,p} + \theta_2 \mathcal{F} \bar{n}_{k-p,p} + \mathcal{P} \Delta^2 \bar{w}_{k,p}. \quad (4.14)$$

Similarly as the system in (4.12) with single-differenced input, the system with double-differenced input can also be combined into two matrix equations. This results in the following double-differenced (or second-order) RC output-feedback problem:

$$\bar{\mathcal{S}}_{\text{RC}}'' \begin{cases} \xi_{k+p}'' &= \mathcal{A}'' \xi_k'' + \mathcal{B}'' \Delta^2 \bar{u}_{k,p} + \mathcal{B}_w'' \bar{v}_{k,p}, \\ \bar{e}_{k,p} &= \mathcal{C}'' \xi_k'' + \mathcal{D} \Delta^2 \bar{u}_{k,p} + \mathcal{D}_w \bar{v}_{k,p}, \end{cases} \quad (4.15)$$

where  $''$  denotes the system with double-differenced input, the vector  $\xi_k'' \in \mathbb{R}^{p(2\ell+2n)+n}$  is given by:

$$\xi_k'' = \begin{bmatrix} \Delta^2 x_k^T & \bar{e}_{k-p,p}^T & \bar{e}_{k-2p,p}^T & \bar{n}_{k-p,p}^T & \bar{w}_{k-p,p}^T & \bar{w}_{k-2p,p}^T \end{bmatrix}^T,$$

and the system matrices  $\mathcal{A}'' \in \mathbb{R}^{p(2\ell+2n+f)+n \times p(2\ell+2n+f)+n}$ ,  $\mathcal{B}'' \in \mathbb{R}^{p(2\ell+2n+f)+n \times pr}$ ,  $\mathcal{B}_w'' \in \mathbb{R}^{p(2\ell+2n+f)+n \times p(f+n+\ell)}$ , and  $\mathcal{C}'' \in \mathbb{R}^{p\ell \times p(2\ell+2n+f)+n}$  are given by:

$$\mathcal{A}'' = \begin{bmatrix} A^p & 0 & 0 & \theta_2 \mathcal{F} & -\theta_1 \mathcal{P} & -\theta_2 \mathcal{P} \\ -\Gamma & \theta_1 I & \theta_2 I & -\theta_2 \mathcal{J} & 0 & 0 \\ 0 & I & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & I & 0 \end{bmatrix}, \quad \mathcal{B}'' = \begin{bmatrix} \mathcal{L} \\ -\mathcal{G} \\ 0 \\ 0 \\ 0 \\ 0 \end{bmatrix}, \quad \mathcal{B}_w'' = \begin{bmatrix} \mathcal{F} & \mathcal{P} & 0 \\ -\mathcal{J} & 0 & 0 \\ 0 & 0 & 0 \\ I & 0 & 0 \\ 0 & I & 0 \\ 0 & 0 & 0 \end{bmatrix},$$

$$\mathcal{C}'' = \begin{bmatrix} -\Gamma & \theta_1 I & \theta_2 I & -\theta_2 \mathcal{J} & 0 & 0 \end{bmatrix}.$$

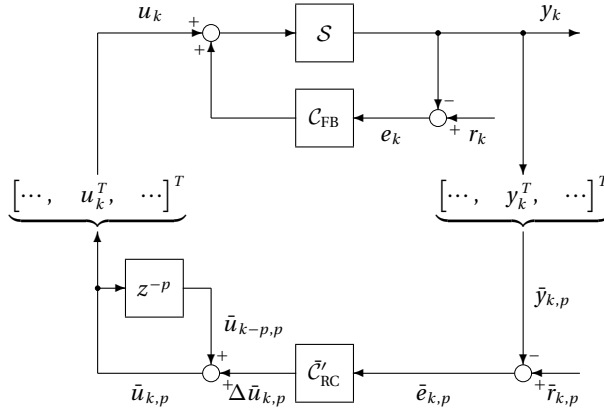
The above computations can be repeated to design a period-robust structure based on a higher number of memory loops. For example, the high-order RC output-feedback problem for  $m \in \mathbb{N}^+$  memory loops would have a  $m$ -times differenced input as:

$$\Delta^m \bar{u}_{k,p} = \bar{u}_{k,p} - \sum_{i=1}^m \theta_i \bar{u}_{k-ip,p}, \quad \sum_{i=1}^m \theta_i = 1. \quad (4.16)$$

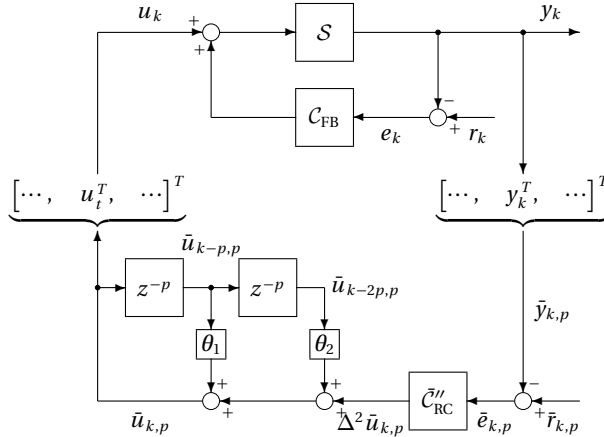
### 4.3.3 The lifted LQG solution

In this subsection the repetitive control method for LTI systems with both repetitive and non-repetitive disturbances is presented. Let's replace ' and '' by a more general notation of \* to denote the system is differenced with an arbitrary number. Now, the high-order RC problem can be formulated as:

**Problem Description 4.1** (Lifted repetitive control). *Given the system in the form of  $\bar{S}_{RC}^*$ , like in (4.12) or (4.15), and  $\theta$  in (4.16), design a strictly-proper dynamic repetitive controller  $\bar{C}_{RC}^*$ .*



**Figure 4.2:** Block diagram of the closed-loop system with the RC loop with one memory loop.



**Figure 4.3:** Block diagram of the closed-loop system with the RC loop with two memory loops.

The controller is strictly proper, because current trial measurements are assumed unavailable for feedback. The block diagrams in Figure 4.2 and 4.3 illustrate how

the lifted repetitive controllers can be implemented with one memory loop and two memory loops, respectively.

Given Assumption 4.1, the infinite-horizon discrete-time LQG solution is well known, see Åström and Wittenmark (1997); Ogata (1994), and it is desirable in our situation, as it stabilizes the system  $\tilde{S}_{\text{RC}}^*$  and minimizes the expectation of the loss function:

$$J = \mathbb{E} \left[ \lim_{N \rightarrow \infty} \sum_{i=0}^N \|\bar{e}_{k+i,p}\|_2^2 + \rho \|\Delta^* \bar{u}_{k+i,p}\|_2^2 \right].$$

Compared with the minimum-variance criterion in Rice and Verhaegen (2010b), we introduced in the LQG criterion a weight to penalize the differenced input, and therefore we are able to make a trade-off between the variance of the differenced input and output signals. By requiring that  $\tilde{C}_{\text{RC}}^*$  stabilizes  $\tilde{S}_{\text{RC}}^*$ , it can be shown to get exponential convergence of the RC algorithm. This does not imply that the RC algorithm has monotonic decay ( $\|\bar{e}_{k+p,p}\|_2 \leq \|\bar{e}_{k,p}\|_2$ ) as recommended in Longman (2000), but the experience is that the introduction of a weight on the differenced input could give the engineer some means to prevent for the large overshoots conventionally seen with some RC laws with non-monotonic decay in the initial transient periods, see also Subsection 4.4.3. Using a steady-state approximation of the optimal LQG solution, the lifted controller  $\tilde{C}_{\text{RC}}^*$  is given by:

$$\tilde{C}_{\text{RC}}^* \begin{cases} \hat{\xi}_{k+p}^* &= \mathcal{A}_C \hat{\xi}_k^* + \mathcal{L}_C \bar{e}_{k,p}, \\ \Delta^* \bar{u}_{k,p} &= -\mathcal{K}_C \hat{\xi}_k^*, \end{cases} \quad (4.17)$$

where

$$\begin{aligned} \mathcal{A}_C &= \mathcal{A}^* - \mathcal{B}^* \mathcal{K}_C - \mathcal{L}_C \mathcal{C}^* + \mathcal{L}_C \mathcal{D}^* \mathcal{K}_C, \\ \mathcal{L}_C &= (\mathcal{S}_W^* + \mathcal{A}^* \mathcal{Z} \mathcal{C}^{*T}) (\mathcal{R}_W^* + \mathcal{C}^* \mathcal{Z} \mathcal{C}^{*T})^{-1}, \\ \mathcal{K}_C &= (\mathcal{R} + \mathcal{B}^{*T} \mathcal{X} \mathcal{B}^*)^{-1} (\mathcal{B}^{*T} \mathcal{X} \mathcal{A}^* + \mathcal{S}^T). \end{aligned}$$

The matrices  $\mathcal{X}$  and  $\mathcal{Z}$  are the positive-definite stabilizing solutions to the Discrete Algebraic Riccati Equations (DARE):

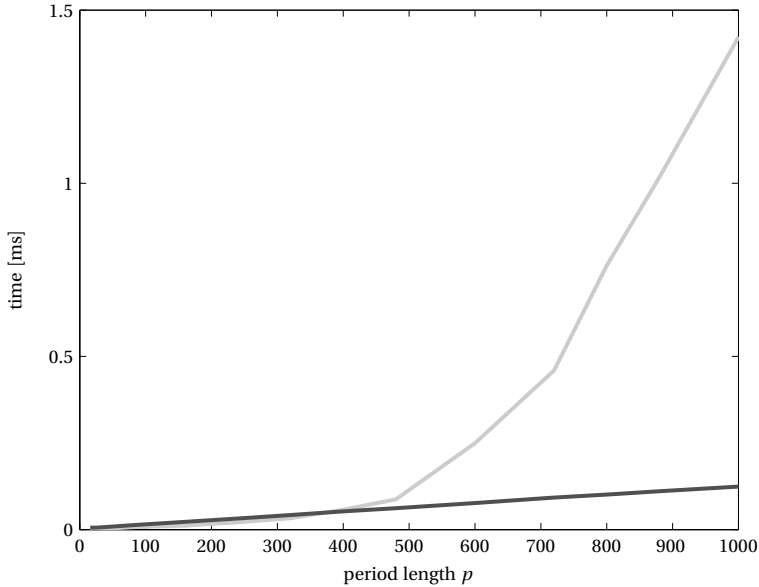
$$\begin{aligned} \mathcal{X} &= \mathcal{Q} - (\mathcal{A}^{*T} \mathcal{X} \mathcal{B}^* + \mathcal{S}) \mathcal{K}_C + \mathcal{A}^{*T} \mathcal{X} \mathcal{A}^*, \\ \mathcal{Z} &= \mathcal{Q}_W^* - \mathcal{L}_C (\mathcal{C}^* \mathcal{Z} \mathcal{A}^{*T} + \mathcal{S}_W^*) + \mathcal{A}^* \mathcal{Z} \mathcal{A}^{*T}, \end{aligned} \quad (4.18)$$

where  $\mathcal{R} = \rho I + \mathcal{D}^{*T} \mathcal{D}^*$ ,  $\mathcal{Q} = \mathcal{C}^{*T} \mathcal{C}^*$ ,  $\mathcal{S} = \mathcal{C}^{*T} \mathcal{D}^*$ , and the disturbance covariance matrices are calculated as:

$$\begin{bmatrix} \mathcal{R}_W^* & \mathcal{S}_W^{*T} \\ \mathcal{S}_W^* & \mathcal{Q}_W^* \end{bmatrix} = \begin{bmatrix} \mathcal{D}_W \\ \mathcal{B}_W^* \end{bmatrix} \mathbb{E} \begin{bmatrix} \tilde{v}_{k,p} \tilde{v}_{k,p}^T \end{bmatrix} \begin{bmatrix} \mathcal{D}_W \\ \mathcal{B}_W^* \end{bmatrix}^T. \quad (4.19)$$

In many cases, the solvers for the Riccati equations in (4.18) have difficulties to determine the solution numerically. For example the QZ-based DARE solver in MATLAB™ will fail, because the Hamiltonian or symplectic matrix has eigenvalues on and/or close to the unit circle. To overcome this problem, the learning rate should be lowered by adding forgetting or leakage. With the inclusion of forgetting, the weights of the  $m$  times differenced input in (4.16) should be constrained with  $\sum_{i=1}^m \theta_i = \lambda$ , where the scalar  $0 \ll \lambda < 1$  is the so-called forgetting factor.

The biggest challenge of implementing this RC description of the output-feedback problem is that the lifted matrices are enormous, and it requires large memory storage and will take a long time to solve the DARE. These issues can be resolved by exploiting the matrix structure in (4.18). For matrices in dense form, the computational complexity for DARE solvers using the Schur method is  $\mathcal{O}(\ell^3 p^3)$ . In Rice and Verhaegen (2010b), efficient methods for solving Riccati equations using the so-called Sequentially Semi-Separable (SSS) matrix structure (Chandrasekaran et al., 2003) and its properties are discussed. An SSS matrix toolbox for MATLAB™ has been developed that includes iterative algorithms to check matrix stability, to solve Lyapunov and Riccati equations, and to compute LQG controllers in  $\mathcal{O}(\ell^3 p)$  computational complexity. The matrices  $\mathcal{A}^*$ ,  $\mathcal{B}^*$ ,  $\mathcal{C}^*$ ,  $\mathcal{D}$ ,  $R_w^*$ ,  $Q_w^*$ , and  $S_w^*$  can be put into this structure for fast computation of the similar structured LQG solutions  $K$  and  $L$  to the RC output-feedback problem. Further, The lifted repetitive controller  $\bar{C}_{RC}^*$  in Figures 4.2 and 4.3 can be implemented real time using three SSS structured matrix-vector computations in  $\mathcal{O}(\ell^2 p)$  instead of  $\mathcal{O}(\ell^2 p^2)$ . In Figure 4.4, the m-function of the SSS structured matrix-vector product is compared with the inbuilt dense matrix-vector product in computation time. Even in m-code the SSS matrix-vector product is faster for  $p > 400$ . Although not directly beneficial for the “smart” rotor section experiment (with  $p = 60$ ), the SSS matrix-vector algorithms are used to verify their capability for real-time computation. For controller synthesis, which requires the offline computation of two Riccati matrix equations, the SSS Riccati solver using sign iterations is also considerably faster for  $p > 250$ , see Rice and Verhaegen (2010b). It is noted that the SSS Riccati solver requires after some iterations a model reduction step, therefore the result will be an approximation of the true solution up to a given tolerance. Increasing the tolerance value will considerably speed up the computation of the DARE solution.



**Figure 4.4:** Computation time averaged over 10 runs of matrix-vector product. Light grey line is with dense matrix structure and dark grey line with SSS matrix structure.

## 4.4 Experimental study on “smart” rotor section

In this section we compare the periodic wind disturbance rejection of the proposed RC method with different settings on a “smart” airfoil section with trailing-edge flap actuator. First, we present the experimental setup used to show the feasibility of periodic wind disturbance rejection with the repetitive control method. Second, we describe the identification experiment to obtain an accurate model for the RC design. Third, the proposed RC method is applied on the “smart” rotor test section, where the periodic and non-periodic disturbances are created using a wind generator.

### 4.4.1 Description of the experiment

As in Chapter 3, the experimental setup mainly consists of the following components: wind generator, blade, actuator and sensors, and real-time environment. The wind generator blows air with considerable turbulence into a tunnel section with a blade section hanging inside. The diameter of the outlet of the wind generator is 350mm and the tunnel section is 400mm by 400mm. The blade section that we use for our experimental verification is an airfoil with at the trailing edge a control surface, the so-called trailing-edge flap. The blade section is at the top connected to a half meter long aluminium plate with the other end fixed to a rigid frame and has two degrees of freedom. The plate allows the blade only to move in the flapwise and torsional direction. A full description and schematic illustrations of the experimental setup are given in Chapter 3, where it has already successfully been used to study the recursive predictor-based subspace identification method.

For control purposes, the “smart” rotor section is equipped with sensors which measure the dynamic behaviour of the blade. Since the final goal for this experiment is to reduce vibrations of the structure, one of the three Macro Fiber Composite (MFC) (Smart Material) patches that are adhered to the root at the frontside of the half meter long aluminium plate is used to measure the high strains associated with the first flapwise bending mode. The main advantage of an MFC is that no amplification is required to have a good Signal-to-Noise Ratio (SNR) ratio. However, with the MFC it is not possible to do static measurements due to the capacitance behaviour of the MFC. This high-pass behaviour is desirable for this experiment, as we want to control the dynamic behaviour of the system, rather than the static deformations.

The trailing-edge flap can reduce the structural vibrations of the “smart” rotor section when this repetitive controller is enabled. The controller intelligence and data acquisition capability are added with a dSPACE™ (dSPACE) system. The controller and data acquisition scheme are fully developed in the MATLAB™ (Mathworks, a) and SIMULINK (Mathworks, b) environment and then compiled to the dSPACE™ (dSPACE) chip. On a separate computer, all the signals are monitored using CONTROL DESK™ (dSPACE) and also the control parameters can be adjusted in real time. The input to the wind generator is a reference wind speed profile with a given period and is generated by a separate system. For this reason, the actual period of the periodic wind disturbances is not directly available to the repetitive controller.

#### 4.4.2 Experimental modelling and control

A model for modern model-based controller design is a mathematical model normally governed by (preferably linear) differential equations. For controller synthesis this model should only contain the relevant dynamics between the input and the outputs and should be accurate over the bandwidth of the controller. Analytical modelling has been performed in the design stage using the theory in Theodorsen (1935), which strongly depends on a large number of parameters that determine the (aero) dynamic behaviour of the system. Most of these parameters can be roughly estimated or calculated. Still, a large amount of uncertainty is present; this makes it difficult to design a stable controller based on such models. This motivates that identification can become a necessary building block for model-based controller design.

The control loops in the prototyped “smart” rotor section are the transfer functions between the trailing-edge flap ( $u$ ), and the first frontside MFC sensor ( $y$ ). Using the Predictor-Based Subspace IDentification (PBSID) method for systems with periodic disturbances as described in van Wingerden et al. (2011), we identify the system to be controlled in the innovation form as:

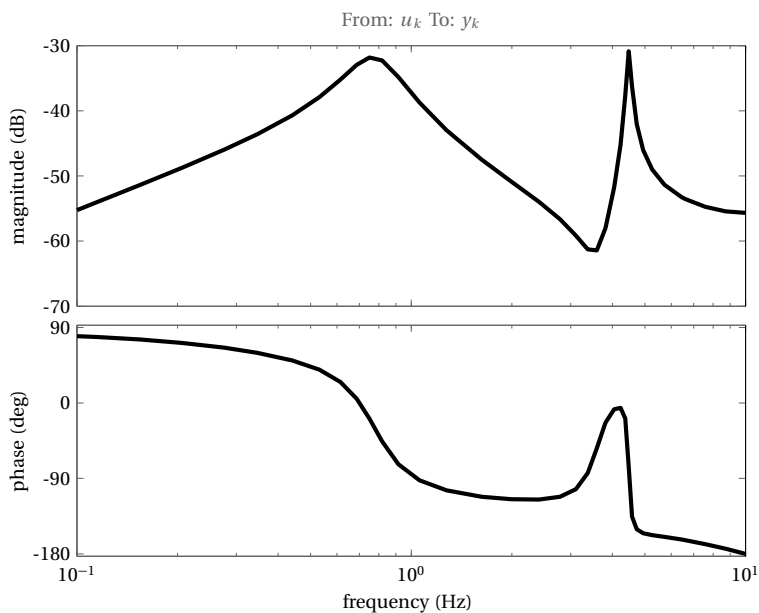
$$\hat{\mathcal{S}} \begin{cases} \hat{x}_{k+1} &= A\hat{x}_k + Bu_k + F\tilde{d}_k + Ke_k, \\ y_k &= C\hat{x}_k + Du_k + e_k, \end{cases} \quad (4.20)$$

where  $\hat{x}_k \in \mathbb{R}^n$  denotes the predicted state vector with  $E[x_k - \hat{x}_k] = 0$  and  $E[(x_k - \hat{x}_k)(x_k - \hat{x}_k)^T]$  minimal,  $\tilde{d}_k \in \mathbb{R}^f$  contains the basis functions needed to express the periodic disturbances, and  $e_k \in \mathbb{R}^\ell$  denotes the innovation sequence with  $E[e_k e_k^T] = R$ , and  $K \in \mathbb{R}^{n \times \ell}$  denotes the Kalman gain matrix. The identified Bode diagram of the fifth-order SISO model is given in Figure 4.5. The first resonance frequency is located at 0.75Hz and is related to the 1st flapwise bending mode. The second resonance frequency is located at 4.5Hz and is related to the 2nd flapwise bending mode. The resonance frequency located at 3Hz related to the 1st torsional bending mode is hardly observable due to the sensor position and therefore is not identified, however in Figure 4.6 a small resonance peak is still visible at that location.

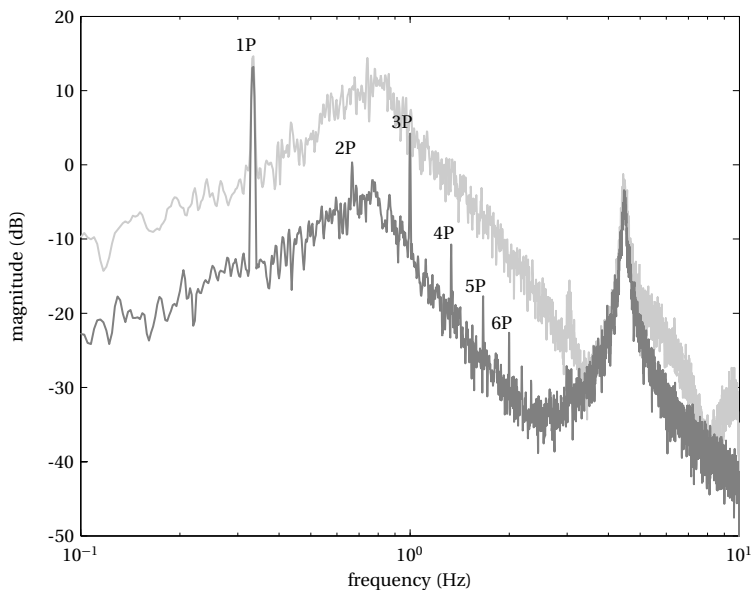
The subspace identification is done with a pink (1/f) Generalized Binary Noise (GBN) signal with a bandwidth of 50Hz and with an amplitude of 200V on the flap actuator. The data is filtered and resampled to a sampling rate of 20Hz ( $h = 0.05s$ ). The periodic wind disturbance is created by changing the speed of the fan in the wind generator. The reference signal given to the wind generator fan is based on the bilinear windshear model and is described by:

$$V_{\text{ref},k} = 10.5 + \begin{cases} \sin(kh \frac{2\pi}{ph}), & \text{if } V_{\text{ref},k} \geq 10.5, \\ 1.5 \sin(kh \frac{2\pi}{ph}), & \text{if } V_{\text{ref},k} < 10.5. \end{cases}$$

As the wind disturbance profile is not purely a sinusoid, multiple harmonics are expected to be excited. In Figure 4.6, the square-root of the power spectral density with and without GBN excitation is illustrated. With no excitation, we clearly see peaks at the 1P–6P (Per revolution) frequencies due to the wind disturbance with a velocity ranging between 9 and 11.5 m/s, period of  $ph = 3s$  ( $p = 60, h = 0.05s$ ), and no variation on the period time:  $\text{std}(ph) = 0$ . When we excite the system, we see that



**Figure 4.5:** Identified Bode diagram. For  $N = 15000$ .



**Figure 4.6:** Square-root of the PSD of the output signal. The light grey line is with GBN excitation and the dark grey line is without excitation. For  $V = 9 - 11.5 \text{ m/s}$ ,  $p = 60$  and  $\text{std}(ph) = 0$ .

especially at lower frequencies a good SNR is obtained. However, the 1P is dominantly present and that is the main reason we use an identification scheme that can handle these periodic disturbances. Further, we can conclude from Figure 4.6 that there is a good control authority at the 1P-6P frequencies, because these harmonics are in the output range in terms of amplitude and bandwidth of the actuator.

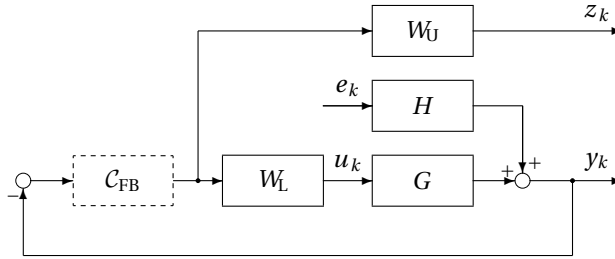
For the feedback control synthesis we use the so-called  $\mathcal{H}_2$  controller synthesis method which is illustrated in the well-known generalized plant setting in Figure 4.7. The goal of the feedback controller is to suppress the unknown disturbances as much as possible with the requirement that the system should remain stable and that the control input signal is bounded between  $-250\text{V}$  and  $+250\text{V}$ . The method does this by minimizing the  $\mathcal{H}_2$  norm between  $e_k$  and  $\{z_k, y_k\}$  and is mathematically given by:

$$\min_{C_{\text{FB}}} \left\| \begin{bmatrix} W_{\text{U}}(I + C_{\text{FB}}G W_{\text{L}})^{-1} H \\ (I + G W_{\text{L}} C_{\text{FB}})^{-1} H \end{bmatrix} \right\|_2^2, \quad (4.21)$$

where

$$G(z) = D + C(zI - A)^{-1} B, \quad H(z) = R^{1/2} + C(zI - A)^{-1} K R^{1/2}.$$

In these objective functions we can embed manually the loop shaping ideas in the weighting filters  $W_{\text{L}}$ ,  $W_{\text{U}}$ , and  $H$ . The wind turbulence spectrum  $H$  is obtained from identification. To let the controller not act on the 1st torsion and the 2nd flapwise bending mode frequencies in the error signal, the input sensitivity function is weighted by a first-order filter  $W_{\text{U}}$  with high gains at those high frequencies. A second-order low-pass filter  $W_{\text{L}}$  is added in series with the controller, such that the final feedback controller will roll-off with increasing frequency.



**Figure 4.7:** Block diagram of generalized plant to synthesize the  $\mathcal{H}_2$  controller.

#### 4.4.3 Experimental results

In this subsection, the main results of the experiments on the “smart” rotor test section are presented. We present four cases that will show the features of the proposed lifted RC algorithm. During the design of the proposed lifted RC method, a large number of parameters have to be tuned. We have the singular value order  $s$ , the disturbance covariance matrices  $R_w$ ,  $R_v$ , and  $R_n$ , and the weight  $\rho$  on the differenced input, and the number of memory loops  $m$  and its weighting factors  $\theta$  (and  $\lambda$ ). In the following four cases some tuning guidelines are provided by analysing the effects of these tuning parameters.



### Effect of the singular value order $s$

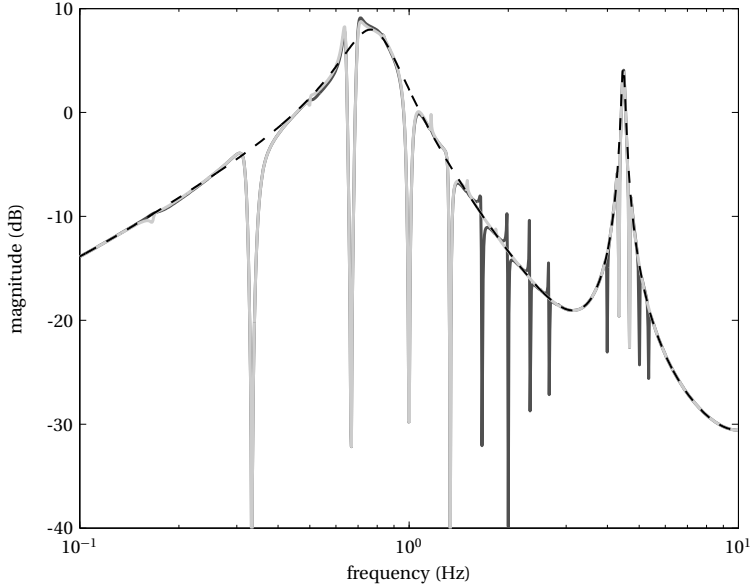
In Subsection 4.2.2, we proposed to select the number of singular values  $s$  in (4.5) not only on the controllability/observability condition, but also to further reduce the size of the RC output-feedback problem. It turns out that for even lower values of  $s$ , such that the error to be reduced is in a smaller column space, the lifted repetitive controller will focus more on the main harmonic components and therefore robustify the controller performance in some sense. To analyse the effect, we consider the frequency response of the lifted repetitive controller  $C_{RC}$ . From Figure 4.2 and Figure 4.3, we can derive the following frequency-domain representation of the complete repetitive controller as:

$$\begin{aligned} C_{RC}(e^{j\omega}, k) = & - \begin{bmatrix} e^{(1-k)j\omega} I & \dots & e^{(p-k)j\omega} I \end{bmatrix} \\ & \times \frac{1}{1 - \sum_{i=1}^m \theta_i e^{-ipj\omega}} \mathcal{K}_C (e^{pj\omega} I - \mathcal{A}_C)^{-1} \mathcal{L}_C \\ & \times \begin{bmatrix} e^{(-p-k+1)j\omega} I & \dots & e^{-kj\omega} I \end{bmatrix}^T. \end{aligned}$$

The lifted repetitive controller  $C_{RC}$  is in fact a periodic linear time-varying system, thus this transformation to the frequency domain does not generally hold. However during the design is observed that for LTI systems, the lifted controller  $C_{RC}$  behaves almost as a very high-order LTI system. This is observed from the frequency responses  $C_{RC}(e^{j\omega}, k)$  that are very similar to each other for every step  $k$  within the period. Plotting one of the Bode diagrams of the sensitivity function is then also a useful engineering tool for lifted RC design.

In Figure 4.8, the Bode diagrams of two noise sensitivity functions with  $s = 12$  and  $s = 26$  are given. The two noise sensitivity functions are almost similar for the main (1P–4P) harmonic components. Increasing the number of singular values to  $s = 26$  introduces also notches at the (5P–8P) harmonic components. Basically, reducing the number of singular values lets the lifted repetitive controller focus on the main harmonic components. In our case we selected the number  $s = 12$ , because first the repetitiveness of the higher harmonics (5P–8P) are questioned (especially if the period time  $ph$  varies over time) and second the system should not be excited around 2–3Hz, due to the badly damped (unobservable) torsion mode.

For the SISO case, an almost similar Bode diagram of the noise sensitivity function can be created using  $H_\infty$  control design synthesis with inverse notches as weights. The resulting time-domain controller will probably be also of very high-order, because the weights should be of quite a high order to represent these steep notches. If the frequency response closely matches each other, the asymptotic behaviour will be almost similar. However, the transient behaviour will be very different. The feedback controller based on inverse notches will immediately react on any harmonic variations in the error signal. Instead with lifted RC, the input signals are averaged over previous periods and are not directly influenced by the current error signal, therefore the input signal will be considerably smoother, but it will take longer to converge.



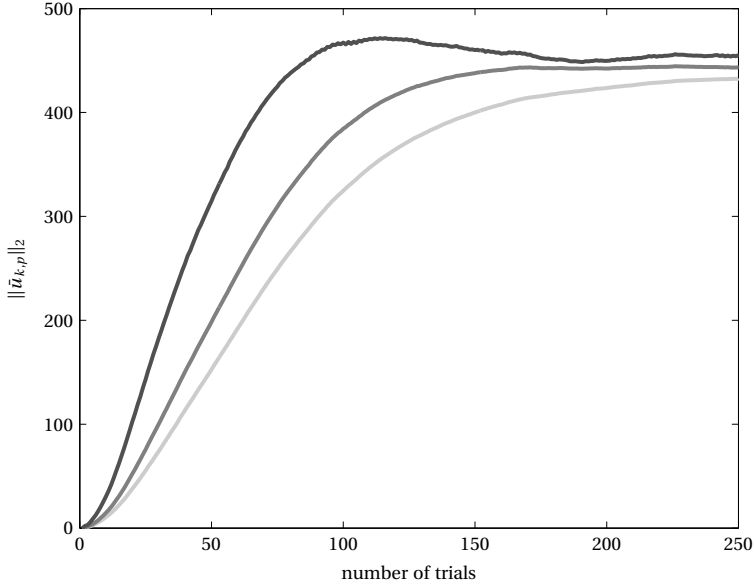
**Figure 4.8:** Bode diagram of the two noise sensitivity functions  $S = (I + GC_{RC})^{-1} H$ . Dark grey line (partly under light grey) is designed with  $s = 26$  and light grey line is designed with  $s = 12$ . Dotted line is the noise sensitivity function  $H$ . For  $\rho = 0.05$ ,  $R_n = 0.001$ ,  $m = 1$ , and  $\lambda = 0.99$ .

### Effect of parameter $\rho$

As the identified state-space model  $\hat{S}$  is considered to be a close estimate of the system  $S$  to be controlled, the disturbance covariance matrices can be estimated from the identification as:  $\hat{R}_w = KRK^T$ ,  $\hat{R}_v = R$ . The strategy to pick the parameter  $\rho$  is very similar to the time-domain LQG problem, see Åström and Wittenmark (1997). It is common practice to pick the parameter  $\rho$  such that a smooth control signal is obtained and a large overshoot is avoided. In Figure 4.9, the initial responses of the input  $\|\bar{u}_{k,p}\|_2$  for three different weights are given. In Longman (2000) it is even recommended to have a monotonic decrease of  $\|\bar{e}_{k,p}\|_2$ , which is almost (apart from the initial differenced state) directly related to the monotonic increase of  $\|\bar{u}_{k,p}\|_2$ . By increasing the weight  $\rho$ , the overshoot over the trials can be lowered in most cases and improve the smoothness of the generated control signal at the cost that it takes longer to converge to steady state. As shown in Figure 4.9, an overshoot during the initial response of  $\|\bar{u}_{k,p}\|_2$  is clearly visible with  $\rho = 0.01$ , and monotonic increase is visible with the higher values of  $\rho = 0.05$  and  $\rho = 0.1$ .

### RC with variations in period time

In this case we show the performance of solely the repetitive controller using the results of the experiments on the “smart” rotor test section. We have stochastic



**Figure 4.9:** Initial responses of the input  $\|\tilde{u}_{k,p}\|_2$ . Dark grey line is designed with  $\rho = 0.01$ , grey line is designed with  $\rho = 0.05$ , and light grey line is designed with  $\rho = 0.1$ . For  $s = 12$ ,  $R_n = 0.001$ ,  $m = 1$ , and  $\lambda = 0.99$ .

disturbances coming from the wind (turbulence) and periodic disturbances due to generated wind speed profile. Experimentally the implications of variations in period time to the performance with respect to the periodic disturbance rejection and the smoothness of control signal are investigated. For this reason, the experiments have been repeated with different settings for the deviation of the period time. The considered maximum deviation  $\text{std}(ph) = 0.03\text{s}$  (1%) is based on 10 min simulations with the UPWIND 5MW wind turbine (see Bossanyi (2009); Jonkman et al. (2008)) in above-rated conditions and with a turbulence intensity of 12% (IEC, 2004). The maximum difference in rotational speed was about 3%.

As suggested in the introduction, two ways (or better: a combination of the two) are proposed to improve the performance of the repetitive controller in the case there is a small mismatch between the controller period and the actual period of the periodic disturbance signal. If the period is not measured it is common practice to formulate a high-order repetitive control problem with multiple memory loops. The selection of the number of memory loops  $m$  and its weighting factors  $\theta$  (and  $\lambda$ ) is not an easy task. In Pipeleers et al. (2008); Steinbuch et al. (2007) it is shown that there is a clear trade-off between improved suppression of periodic disturbances and amplification of non-periodic noise. It was shown that  $\theta = (2, -1)$  and  $\theta = (3, -3, 1)$  gives the best performance against period mismatch and that  $\theta = (2/3, 1/3)$  and  $\theta = (3/6, 2/6, 1/3)$  gives the lowest sensitivity to non-periodic noise for two and three memory loops respectively. The design of these parameters highly depends on the periodic and non-periodic disturbance characteristics. In Pipeleers et al. (2008); Steinbuch et al. (2007) periodic and non-periodic performance criterion

are used for designing time-domain high-order repetitive controllers. As shown in Subsection 4.3.2, multiple memory loops can be introduced in the lifted RC problem at the cost of larger matrices. However the design of the weighting factors  $\theta$  becomes even more difficult and in this chapter we have even tuned manually, because they are implicitly related with the output-feedback problem formulation in (4.15).

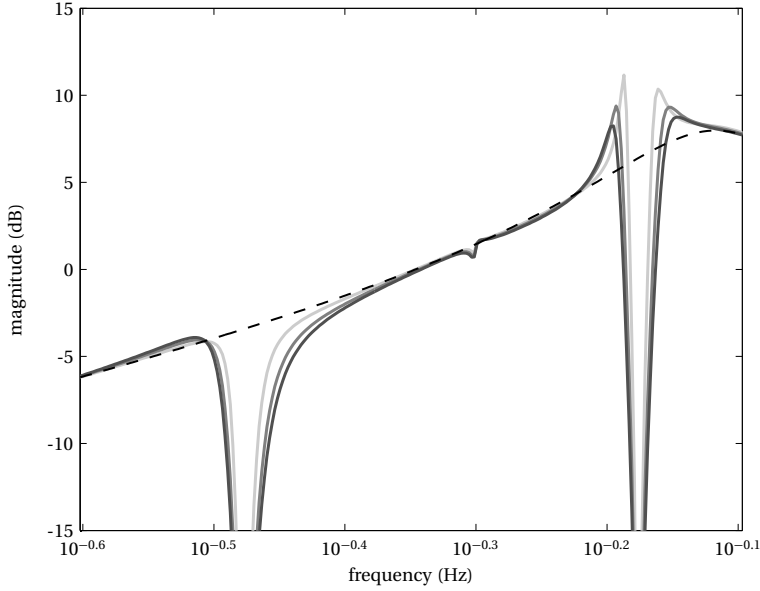
Compared with the time-domain RC methods in Pipeleers et al. (2008); Steinbuch et al. (2007), the lifted RC problem allows a more versatile disturbance formulation. It includes a periodic component with a random walk nature  $n_k$ , such that the periodic disturbance can vary between trials. In addition, the noise and periodic disturbances can be even correlated and varying in time. In Figure 4.10, the Bode diagram of the noise sensitivity function for three different weights  $R_n$  are illustrated. With an increasing value  $R_n$  the inverted notches at 1P and 2P frequencies become wider, but less deep. It can be concluded that the increasing weight provides better disturbance rejection at frequencies near the harmonics at the cost of performance degradation at the exact repetitive frequency and its harmonics. When the system is identified (without exact period time) in the innovation form in (4.20), where basis functions in  $\tilde{d}_k$  are used to express the periodic disturbances, it is even possible to relate a lifted covariance matrix  $\tilde{R}_n$  with the variance of the period time  $ph$  by a first-order approximation as:

$$\hat{\tilde{R}}_n = \frac{\partial \tilde{d}_{k,p}(ph)}{\partial ph} \text{var}(ph) \frac{\partial \tilde{d}_{k,p}^T(ph)}{\partial ph},$$

where  $\tilde{d}_{k,p}$  is the stacked vector of  $\tilde{d}_k$ . As the lifted covariance matrix  $\tilde{R}_n$  has a time-varying structure, the start of the repetitive controller should be matched with the phases of the basis functions.

In Table 4.1, the manually tuned values for the parameters of  $R_n$  and  $\theta$  are given with an increasing value for the deviation of the period time and an increasing value for the number of memory loops  $m$ . These values have been used during the experiments on the “smart” rotor test section; see the results in the next subsections. The parameters have been tuned by looking at the cost  $J = \text{var}(y_k) + \text{var}(\Delta^2 u_k / \Delta h^2) \cdot 10^{-4}$  after the control signal has converged to steady state. By including an approximation for the acceleration of the generated input signal to the cost, we try to achieve a smooth input signal which should reduce the wear and tear of the actuator. From the table it is observed that with increasing deviation of the period-time the weight  $R_n$  should be increased, although at a slower rate if more memory loops are used. The memory loop weights  $\theta$  are moving clearly in the direction of the values  $\theta = (2, -1)$  and  $\theta = (3, -3, 1)$  which should give the best performance against period mismatch for two and three memory loops respectively. It is noted that in most wind turbines and in the rotating “smart” rotor in van Wingerden et al. (2011), the period time could be estimated from the measured rotational speed or from the measured azimuth angles of the blades. For future research it would be interesting to formulate a real-time RC output feedback problem, where we can make the parameters  $R_n$  and/or  $\theta$  adaptive to changes in the measured data.

Table 4.2 summarizes the variance values of the output in percentage with respect to the output without any control  $J_y = 100\% \cdot (1 - \text{var}(y_k) / \text{var}(y_{0,k}))$  and the approximated input acceleration  $J_u = \text{var}(\Delta^2 u_k / \Delta h^2)$  after the control signal has converged



**Figure 4.10:** Bode diagram of the noise sensitivity function  $S = (I + GC_{RC})^{-1} H$  at the 1P and 2P frequencies. Dark grey line is designed with  $R_n = 0.01$ , grey line is designed with  $R_n = 0.005$ , and light grey line is designed with  $R_n = 0.001$ . For  $s = 12$ ,  $\rho = 0.05$ ,  $m = 1$ , and  $\lambda = 0.99$ .

**Table 4.1:** Values for  $R_n$  and  $\theta$  for different deviations of period time. For  $s = 12$ , and  $\lambda = 0.99$  (multiply  $\theta$  with  $\lambda$ ).

|              | $m = 1$ |       |            |            | $m = 2$ |            |            |            | $m = 3$ |            |            |            |
|--------------|---------|-------|------------|------------|---------|------------|------------|------------|---------|------------|------------|------------|
| std ( $ph$ ) | $R_n$   | $R_n$ | $\theta_1$ | $\theta_2$ | $R_n$   | $\theta_1$ | $\theta_2$ | $\theta_3$ | $R_n$   | $\theta_1$ | $\theta_2$ | $\theta_3$ |
| 0%           | 0.001   | 0.001 | 1.25       | -0.25      | 0.001   | 1.3        | -0.4       | 0.1        | 0.001   | 1.3        | -0.4       | 0.1        |
| 1/3%         | 0.003   | 0.002 | 1.54       | -0.54      | 0.002   | 1.45       | -0.6       | 0.15       | 0.002   | 1.45       | -0.6       | 0.15       |
| 1/2%         | 0.006   | 0.003 | 1.55       | -0.55      | 0.003   | 1.55       | -0.75      | 0.2        | 0.003   | 1.55       | -0.75      | 0.2        |
| 2/3%         | 0.016   | 0.008 | 1.63       | -0.63      | 0.006   | 1.75       | -1         | 0.25       | 0.006   | 1.75       | -1         | 0.25       |
| 1%           | 0.022   | 0.015 | 1.7        | -0.7       | 0.014   | 2.25       | -2         | 0.25       | 0.014   | 2.25       | -2         | 0.25       |

**Table 4.2:** Reduction of variance for output and approximated input acceleration for different deviations of period time. For  $N = 6000$ ,  $s = 12$ , and  $\lambda = 0.99$ .

|              | $m = 1$ |       | $m = 2$ |       | $m = 3$ |       |
|--------------|---------|-------|---------|-------|---------|-------|
| std ( $ph$ ) | $J_y$   | $J_u$ | $J_y$   | $J_u$ | $J_y$   | $J_u$ |
| 0%           | 41.8%   | 48    | 42.2%   | 44    | 42.7%   | 42    |
| 1/3%         | 40.9%   | 57    | 41.2%   | 54    | 41.5%   | 52    |
| 1/2%         | 38.2%   | 82    | 38.7%   | 80    | 39.5%   | 79    |
| 2/3%         | 36.3%   | 110   | 37.9%   | 102   | 38.2%   | 98    |
| 1%           | 34.0%   | 116   | 34.9%   | 105   | 35.5%   | 101   |

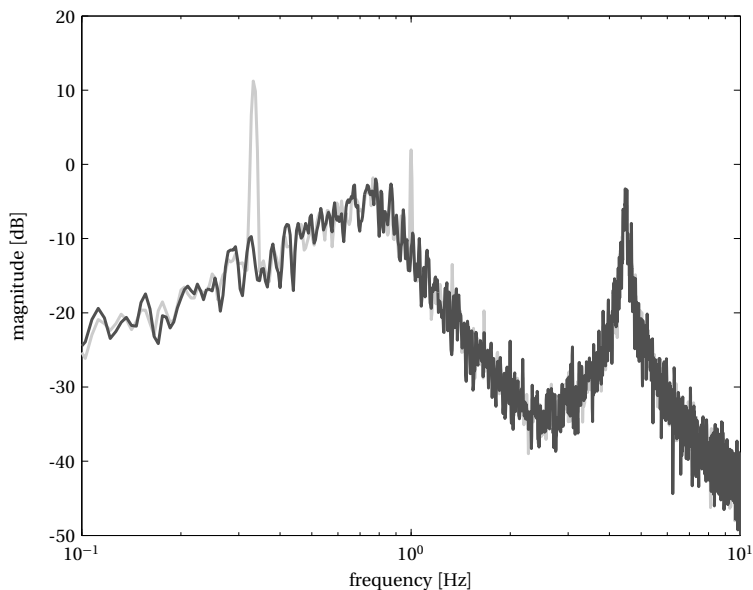
to steady state. The variance of the corresponding signals can be seen as a rough measure for fatigue. We see that we have a reduction in variance of at least 34% if only repetitive disturbance signals are compensated. This increases to a variance reduction of around 42% when the actual period of the wind profile matches the period of the controller. The use of more memory loops gives an increase between 1–2%, which is a small increase compared with the improvements made in earlier experiments with high-order time-domain repetitive controllers in Pipeleers et al. (2009); Steinbuch et al. (2007). The tuning of the weight  $R_n$  already considerably improves the robust performance for variations in period time, such that incorporating more memory loops will not make that much difference anymore. However with more memory loops, the smoothness of the input signal is most of the time much better, probably due to the averaging effect over multiple periods, at the cost that the convergence is considerably slower.

The steady-state performance of any repetitive controller is bounded by an inherent trade-off between the suppression of periodic disturbances and the amplification of non-periodic inputs (Pipeleers et al., 2009). In Figures 4.11 and 4.12 the square-root of the Power-Spectral Density (PSD) is given for the system output without any control and with the proposed repetitive control for both 0% and 1% deviation in period time, respectively. In both figures, we clearly see an improvement with respect to the uncontrolled case. For the 0% period variation case, the spikes at the 1P–4P harmonic frequencies are almost completely removed, and the noise amplifications at some lower and higher intermediate frequencies are small. For the 1% period variation case, the spikes at the 1P–4P harmonic frequencies are only partly reduced, and this reduction comes even at the cost that some amplifications are visible at some intermediate frequencies and higher harmonic frequencies.

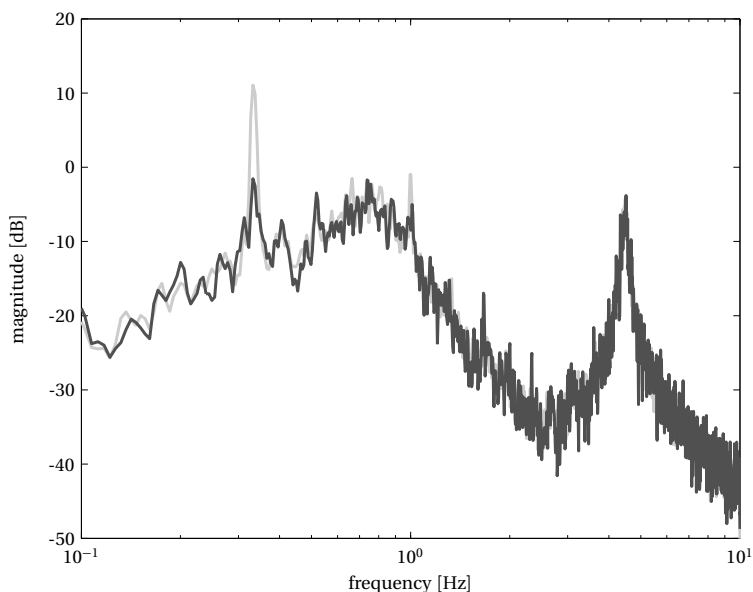
### RC with feedback control

In this case we show the performance of the repetitive controller and feedback controller together using the results of the experiments on the “smart” rotor test section. For this case, the repetitive controller has been redesigned with the model of the closed loop system in Figure 4.1. Table 4.3 again summarizes the deviation values of the output and approximated input acceleration after the control signal has converged to steady state. We see that we have an improved reduction in deviation of at least 64% on the sensor channel if both the feedback and the repetitive control is active. This increases to a reduction of around 72% when the period of the wind profile matches the period of the controller. The use of more memory loops gives again an increase between 1–2%. The introduction of feedback makes the variance of the input acceleration 4–8 times larger than with RC only. Thus the input signal is considerably less smooth and can cause in practice a considerable reduction in actuator lifetime. Most of this increase is caused by the feedback controller.

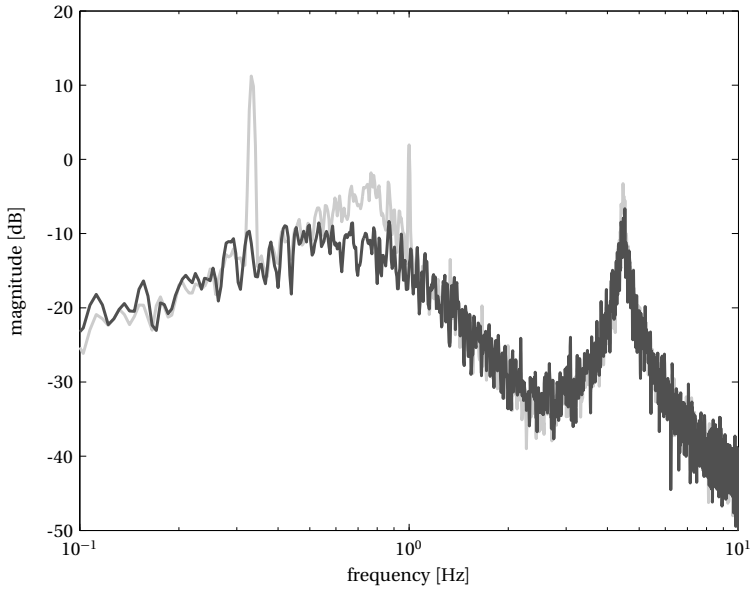
In Figures 4.13 and 4.14 the square-root of the Power-Spectral Density (PSD) are given for the system output without any control and with the feedback and repetitive control together for both 0% and 1% deviation in period time. In both figures, we again see a clear improvement with respect to the uncontrolled case. For both cases, there is an additional 10dB reduction at the first natural frequency caused by



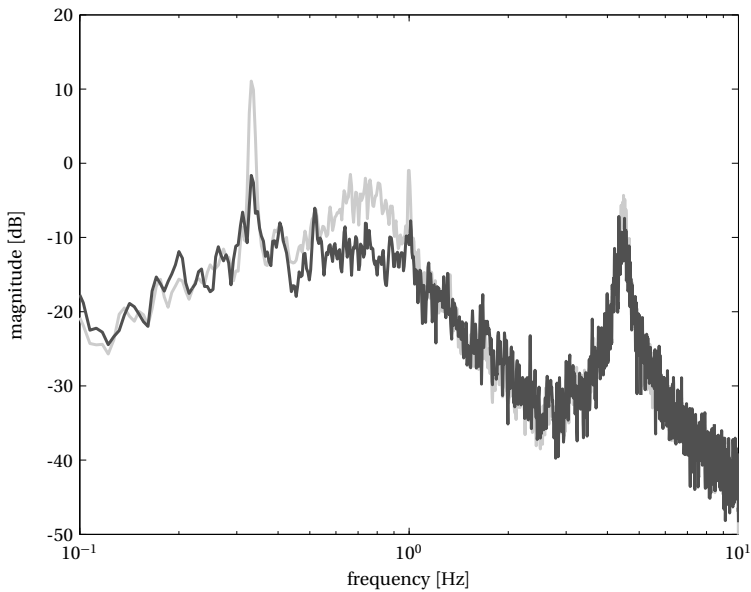
**Figure 4.11:** Square-root of the PSD of the output signal. Dark grey line is the output spectrum with RC and light grey line is without. For  $V = 9 - 11.5\text{m/s}$ ,  $m = 3$ ,  $p = 60$  and  $\text{std}(ph) = 0\text{s}$ .



**Figure 4.12:** Square-root of the PSD of the output signal. Dark grey line is the output spectrum with RC and light grey line is without. For  $V = 9 - 11.5\text{m/s}$ ,  $m = 3$ ,  $p = 60$  and  $\text{std}(ph) = 0.03\text{s}$ .



**Figure 4.13:** Square-root of the PSD of the output signal. Dark grey line is the output spectrum with RC+FB and light grey line is without. For  $V = 9\text{--}11.5\text{m/s}$ ,  $m = 3$ ,  $p = 60$  and  $\text{std}(ph) = 0$ .



**Figure 4.14:** Square-root of the PSD of the output signal. Dark grey line is the output spectrum with RC+FB and light grey line is without. For  $V = 9\text{--}11.5\text{m/s}$ ,  $m = 3$ ,  $p = 60$  and  $\text{std}(ph) = 0.03\text{s}$ .



the feedback control action. This comes at the cost by some small amplification of the lower frequencies and the frequencies between the two natural frequencies. Compared with RC only, the spikes at the 2P–3P frequencies are reduced further to the same level of the surrounding frequencies. Thus feedback and repetitive control could be used together for the reduction of both harmonic and natural frequencies, if the dynamics of the feedback controller are taken into the RC design.

**Table 4.3:** Reduction of variance for output and input acceleration (and RC only) for different deviations of period time described in percentage of period time. For  $N = 6000$ ,  $s = 14$ , and  $\lambda = 0.99$ .

|             | $m = 1$ |          | $m = 2$ |          | $m = 3$ |          |
|-------------|---------|----------|---------|----------|---------|----------|
| std( $ph$ ) | $J_y$   | $J_u$    | $J_y$   | $J_u$    | $J_y$   | $J_u$    |
| 0%          | 71.8%   | 378(26)  | 71.9%   | 375(22)  | 72.0%   | 375(21)  |
| 1/3%        | 70.0%   | 402(92)  | 70.4%   | 401(87)  | 70.8%   | 398(81)  |
| 1/2%        | 67.9%   | 463(102) | 68.4%   | 446(94)  | 69.2%   | 442(90)  |
| 2/3%        | 66.5%   | 475(106) | 67.1%   | 461(102) | 67.6%   | 455(98)  |
| 1%          | 64.1%   | 484(150) | 65.2%   | 475(140) | 65.6%   | 470(139) |

## 4.5 Conclusion

In this chapter we presented a novel repetitive control method that is implemented in real time for periodic wind disturbance rejection in linear systems with multiple inputs and multiple outputs and with both repetitive and non-repetitive disturbance components. The design of the repetitive controller is formulated as a lifted linear stochastic output-feedback problem on which the mature techniques of discrete-time linear control may be applied. The formulation of the lifted repetitive control problem can be made more robust to small changes in period time by using multiple memory loops, but also by including a periodic component with a random walk nature, so that the periodic disturbance can vary between trials. Efficient algorithms exist for controller synthesis and for real-time implementation to reduce the computational complexity and memory usage by exploiting the structure in the lifted state-space matrices. Moreover, the chapter provides some guidelines on how to pick the free parameters in the algorithm. The novel repetitive controller could learn the periodic wind disturbances for fixed-speed wind turbines and variable-speed wind turbines operating above-rated and we have demonstrated this on an experimental “smart” rotor test section. For relatively slow changing periodic and turbulent wind disturbances created by a wind generator it was shown that this repetitive control method could reduce the variance of the load signals of the “smart” rotor test section up to 42%. The cost of additional wear and tear of the “smart” actuators are kept small, because a smooth control action is generated as the controller mainly focuses on the reduction of periodic disturbances.



## Wind Turbine Load Reduction by Rejecting the Periodic Load Disturbances

---

To decrease the cost per kWh, the trend in offshore wind turbines is to increase the rotor diameter as much as possible. The increasing dimensions have led to a relative increase of the loads on the wind turbine structure, thus it is necessary to react to disturbances in a more detailed way; for example, each blade separately. The disturbances acting on an individual wind turbine blade are to a large extent deterministic; for instance, tower shadow, wind shear, yawed error, and gravity are depending on the rotational speed and azimuth angle, and will change slowly over time.

This chapter aims to contribute to the development of individually pitch controlled blades by proposing a lifted repetitive controller that can reject these periodic load disturbances for modern fixed-speed wind turbines and modern variable-speed wind turbines operating above-rated. The performance of the repetitive control method is evaluated on the UPWIND 5MW wind turbine model and compared with typical individual pitch control. Simulation results indicate that for relatively slow changing wind disturbances this lifted repetitive control method can significantly reduce the vibrations in the wind turbine structure with considerable less high-frequent control action.

---

### 5.1 Introduction

THIS chapter investigates the load reduction capabilities of Repetitive Control (RC) with individual pitch actuators, and in the operation of the UPWIND 5MW wind turbine in the aerolastic modelling software GH BLADED 3.85 (Garrað Hassan). A lifted RC method is presented for MIMO linear systems with both repetitive and non-repetitive disturbance components, which already have been verified experimentally

in previous chapter for its load reduction capabilities on a “smart” rotor test section with periodic disturbances generated by a wind generator. The advantage of using RC instead of a typical Individual Pitch Control (IPC) is that the Coleman transformation is not needed. Therefore, the RC method is more generic, for example to extend to rotors with more actuators and sensors, or to other rotor designs; like, two-bladed rotors. We also address an important drawback of RC, which is a small mismatch between the controller period and the actual period of the disturbance signal that can decrease the performance of the repetitive controller substantially. However, the application of RC to wind turbines with small variations in rotational speed, such as fixed-speed and variable-speed operating above-rated with a collective pitch controller that will keep the rotational speed of the wind turbine close to the given reference speed, is still considered competitive. Therefore, the relative performance in terms of load reduction and control action of RC is compared with the typically-used IPC method with Coleman transformation and with different turbulence intensities and consequently different variations in the rotational rotor speed.

The outline of this chapter is as follows. In Section 5.2, the simulation model of the UPWIND 5MW wind turbine is briefly described including the baseline controllers, and the wind conditions. In Section 5.3, the model linearisation and the theoretical framework is presented for the lifted RC design problem and an algorithm for MIMO periodic Linear Time-Varying (LTV) systems is given. In Section 5.4, the effectiveness of the proposed RC algorithm is evaluated and compared with typical IPC in an simulation study on the UPWIND 5MW wind turbine operating in a periodic wind field and some tuning guidelines are provided. In the final section, we present the conclusions of this chapter.

## 5.2 Description of the simulations

This section describes the specifications of the reference wind turbine, the simulation environment which simulates the operation of the wind turbine equipped with the load reduction control, the baseline controllers for above-rated operation, a typical IPC for comparison, and the wind conditions during the simulation runs.

### 5.2.1 Specifications of the reference wind turbine

The reference wind turbine used for this simulation is the UPWIND 5MW (also referred to as the NREL 5MW) wind turbine described in Jonkman et al. (2008). The UPWIND 5MW wind turbine is fairly representative for typical commercial wind turbines in its class, although it does not represent any particular commercial wind turbine (Bossanyi, 2009). For this reason, this wind turbine is commonly used as a baseline model for the research in state-of-the-art controller design, for example with IPC in Bossanyi (2009) and SRC in Lackner and van Kuik (2010a). The wind turbine under consideration is onshore, upwind, variable-speed, three-bladed, and pitch-controlled. Its design specifications are given in Table 5.1, and a schematic representation is given in Figure 1.4.

**Table 5.1:** UPWIND 5 MW wind turbine specifications

| description                  | symbol                        | value             |
|------------------------------|-------------------------------|-------------------|
| rated power                  | $P_{\text{rated}}$            | 5 MW              |
| rotor diameter               | $d_{\text{ro}}$               | 126 m             |
| hub height                   | $h_{\text{hub}}$              | 90 m              |
| cut in wind speed            | $v_{\text{cutin}}$            | 4 m/s             |
| rated wind speed             | $v_{\text{rated}}$            | 11.3 m/s          |
| cut out wind speed           | $v_{\text{cutout}}$           | 25 m/s            |
| rated rotational rotor speed | $\Omega_{\text{ro}}$          | $2\pi/4.95$ rad/s |
| gearbox ratio                | $\nu$                         | 97                |
| pitch rate limit             | $\dot{\theta}_{\text{limit}}$ | 8 deg/s           |

### 5.2.2 Simulation environment

There is a wide variety of different model simulation codes that can be used to model the dynamic behaviour of wind turbines, however only a small part is verified and even a smaller part is validated (Molenaar, 2003). For this case, the simulations of the UPWIND 5MW wind turbine are carried out in the proven aerolastic modelling software GH BLADED™ 3.85 (Garrad Hassan). The GH BLADED™ software provides the following important features in their model representation (Lackner and van Kuik, 2010a):

- The aerodynamics are calculated using the well-known blade element momentum approach. Dynamic inflow and dynamic stall models are incorporated to represent the turbine wake and deal with unsteady aerodynamic conditions.
- The structural dynamics of the turbine model are calculated using a limited degree of freedom modal model. Note, that models with limited order are normally preferred for model-based control design.
- The dynamics of the power train (shaft, gearbox and generator) are modelled.
- The external wind conditions can be generated, including three-dimensional (3-D) turbulent wind fields, wind shear, tower shadow effects and prescribed gusts.
- Control of the turbine can be accomplished using either internal controllers provided by GH BLADED™, or by external controllers created by the user.
- The loads on the various components of the turbine and the turbine performance can be calculated, analysed and exported.

The capability to use external controllers using the shared libraries is a very necessary requirement, because the proposed controllers can not be represented by the standard controllers in GH BLADED™. The building of these so-called Dynamically Linked Libraries (DLL) is not an easy task. First, the controllers have to be written in the code Fortran or C. Second, the code has to be compiled correctly using a suitable compiler (with Microsoft VISUAL STUDIO™ C++ (Microsoft)) to a DLL. To automate

the process of DLL building and to ease the algorithm writing in an environment already familiar for most control engineers, the Mathworks Real-Time Workshop (RTW) toolbox for Simulink™ (Mathworks, b) has been used to automatically generate executable code from implemented controllers in Simulink diagrams. For this purpose the generic Real-Time Target in the RTW toolbox is adjusted following the specifications in (GH, 2010b). The specially adjusted Real-Time Target files have been made available to the readers through our website.<sup>1</sup>

### 5.2.3 Baseline control for above-rated operation

The UPWIND 5MW wind turbine operates at variable speeds, which has the advantage that at lower wind speeds than the rated wind speed, called below rated, the rotor speed can vary to maintain an optimal power production. At low wind speeds the reduced rotor speed also reduces the aerodynamically generated acoustic noise and the stress on the mechanical structure. For wind speeds higher than the rated wind speed, called above rated, the rotational speed is regulated at rated rotational speed to keep the rotational speed (and indirectly the power) within the design specifications of the power train, and also to limit the stresses on the mechanical structure.

The baseline controllers for the UPWIND 5MW wind turbine including the logic for switching between below-rated and above-rated operation using only the measurements of the generator torque  $T_{ge}$  and the generator speed  $\Omega_{ge}$  (and not directly from the wind speed) are described in Bossanyi (2009). Because the simulations with lifted repetitive control are intended for above-rated operation only, we will describe the following implemented baseline controllers for above-rated operation: Collective Pitch Controller (CPC), additional Non-linear Pitch Controller (NPC) to overcome large wind gusts, and the Torsional Vibration Damping (TVD) filter for torsional drive train vibrations. Note, that the filter for damping the tower In Bossanyi (2009) is not implemented to better assess the load reduction capabilities of the proposed controller. For comparison, a typical IPC with the Coleman transformation is implemented. Because the IPC was not fully specified in Bossanyi (2009), we describe in the next subsection our interpretation/implementation of the IPC following the recipe in Bossanyi (2009). All the baseline controllers are described in a continuous-time representation. For implementation, each filter is discretized separately with the Tustin approximation, where the feedback control loop runs at the sample rate of 100Hz.

In below-rated operation, it is expected that the larger variations in rotational speed will reduce the load reduction performance of the repetitive controller considerably and therefore it is not considered. In above-rated operation, the CPC will keep the rotational speed of the wind turbine close to the rated rotational speed. Thus a well performing CPC can increase the load reduction performance of the RC. For an honest comparison, we will use in the GH BLADED™ simulations the following CPC as

---

<sup>1</sup><http://www.dsc.tudelft.nl/~datadriven/discon/>

specified in Bossanyi (2009):

$$C_{CPC}(s) = \underbrace{K_s(\theta) \left( K_p + \frac{K_i}{s} \right)}_{\text{gain-scheduled PI controller}} \times \underbrace{\frac{\omega_L^2}{s^2 + 2\zeta_L \omega_L s + \omega_L^2}}_{\text{low-pass filter } \mathcal{L}} \times \underbrace{\frac{s^2 + 2\zeta_{A,1} \omega_A s + \omega_A^2}{s^2 + 2\zeta_{A,2} \omega_A s + \omega_A^2}}_{\text{notch filter } \mathcal{A}} \times \underbrace{\frac{s^2 + 2\zeta_{B,1} \omega_B s + \omega_B^2}{s^2 + 2\zeta_{B,2} \omega_B s + \omega_B^2}}_{\text{notch filter } \mathcal{B}},$$

where the parameters are specified in Table 5.2. This PI controller with additional filters was designed to generate a pitch position demand  $\theta$  from the generator speed error  $\Omega_{ge}$  without causing too much excitation of the fore-aft tower vibrations, because adjusting the pitch influences not only the aerodynamic torque but also the rotor thrust. The pitch rate was limited to 8 deg/s. Furthermore, the control response at 3P and 6P (Per revolution) was reduced. For these reasons, two notch filters at around 0.6Hz (3P) and 1.3Hz (6P) were added to the feedback loop. An low-pass filter with a bandwidth of 1.6Hz was also added is to prevent unnecessary high frequency pitch action. Close to rated, the sensitivity of aerodynamic torque to pitch angle is very small. Instead at higher wind speeds, a small change in pitch angle can have a large effect on the aerodynamic torque. To compensate for this non-linear aerodynamic characteristic, gain scheduling was added to the PI controller by a factor which varies linearly from 1 at fine pitch (0 deg) to 3.5 at 25 deg.

**Table 5.2:** Parameters of collective pitch control

| description                   | symbol        | value                          |
|-------------------------------|---------------|--------------------------------|
| gain-scheduled PI controller  |               |                                |
| proportional gain             | $K_p$         | 0.0135                         |
| integral gain                 | $K_i$         | 0.00453                        |
| scheduling gain               | $K_s$         | $1 + \frac{180}{10\pi} \theta$ |
| low-pass filter $\mathcal{L}$ |               |                                |
| frequency                     | $\omega_L$    | 10 rad/s ( $\approx 1.6$ Hz)   |
| damping                       | $\zeta_L$     | 1                              |
| notch filter $\mathcal{A}$    |               |                                |
| frequency                     | $\omega_A$    | 3.8 rad/s ( $\approx 0.6$ Hz)  |
| damping 1                     | $\zeta_{A,1}$ | 0.01                           |
| damping 2                     | $\zeta_{A,2}$ | 0.15                           |
| notch filter $\mathcal{B}$    |               |                                |
| frequency                     | $\omega_B$    | 8.2 rad/s ( $\approx 1.3$ Hz)  |
| damping 1                     | $\zeta_{B,1}$ | 0.01                           |
| damping 2                     | $\zeta_{B,2}$ | 0.2                            |

In addition to the CPC, a further contribution to the pitch position demand was recommended in Bossanyi (2009) to increase the response to sudden gusts. This additional Non-linear Pitch Control (NPC) term depends on both the generator speed error and its rate of change, which was obtained by differentiation preceded by a first-order low-pass filter with time constant of  $\tau = 0.05$  to prevent overreaction to noise.

Both the generator speed error and its rate of change was normalised by a scale factor  $c_1 = 25$  and  $c_2 = 10$ , respectively. When the sum of scaled signals is over 1, then any excess over 1 was multiplied by a gain factor of  $K_n = 0.15$  to give a contribution to the pitch rate demand as follows:

$$C_{NPC}(s) = K_n \begin{cases} 0, & \text{if } \beta \leq 1, \\ \beta - 1, & \text{if } \beta > 1. \end{cases}$$

where  $\beta = c_1 + c_2 s \frac{1}{\tau_s + 1}$ . In the discretized case, multiplying this pitch rate demand by the sample time  $h = 0.01$ s gives the increase in pitch position demand. In normal turbulent circumstances, this additional pitch demand remains zero. The logic of the NPC will only give additional pitch action in an extreme case; when the generator speed error is large, positive, and increasing.

In variable-speed wind turbines the torsional vibration modes of the drive train are usually very lightly damped, because the rotor blades provide very little aerodynamic damping in the in-plane direction, therefore the structural damping and any mechanical damping from the gearbox or generator will be small. In Bossanyi (2009), it was recommended to provide very significant damping by modifying the constant generator torque demand with an additional term which adds a small ripple at the appropriate frequencies (1st drivetrain torsion mode and a sideways tower bending mode). This was obtained by passing the measured generator speed through the following Torsional Vibration Damping (TVD) filter:

$$C_{TVD}(s) = K_1 \underbrace{\frac{2\zeta_1 \omega_1 s}{s^2 + 2\zeta_1 \omega_1 s + \omega_1^2}}_{\text{inverse notch filter 1}} + K_2 \underbrace{\frac{2\zeta_2 \omega_2 (1 + \tau_2) s}{s^2 + 2\zeta_2 \omega_2 s + \omega_2^2}}_{\text{inverse notch filter 2}},$$

where the parameters are specified in Table 5.3.

**Table 5.3:** Parameters of torsional vibration damping filter

| description            | symbol     | value                           |
|------------------------|------------|---------------------------------|
| inverse notch filter 1 |            |                                 |
| gain                   | $K_1$      | 1560                            |
| frequency              | $\omega_1$ | 24.2 rad/s ( $\approx 3.85$ Hz) |
| damping                | $\zeta_1$  | 0.132                           |
| inverse notch filter 2 |            |                                 |
| gain                   | $K_2$      | 1625                            |
| frequency              | $\omega_2$ | 8.998 rad/s ( $\approx 1.4$ Hz) |
| damping                | $\zeta_2$  | 0.5041                          |
| offset                 | $\tau_2$   | 0.0138                          |

#### 5.2.4 Individual pitch control for load reduction

Wind turbine simulations with IPC using the Coleman transformation are considered for comparison with the RC simulations in load reduction performance and control



action. The goal of this IPC is to reduce the out-of-plane bending vibrations in the blade root of each of the three blades ( $M_{y,1}$ ,  $M_{y,2}$  and  $M_{y,3}$ ), by adjusting the blade pitch angles  $\theta_1$ ,  $\theta_2$  and  $\theta_3$ . The major challenge of implementing load reduction control is that the blades are rotating, and so the equations of motion that relate  $M_{y,1}$ ,  $M_{y,2}$  and  $M_{y,3}$  and  $\theta_1$ ,  $\theta_2$  and  $\theta_3$  are dependent on the azimuth angle  $\varphi$  and the rotational speed  $\Omega_{ro}$ . The result is that a model obtained through linearisation is normally a periodic LTV system, and it is much more challenging to design controllers for periodic LTV systems than LTI systems.

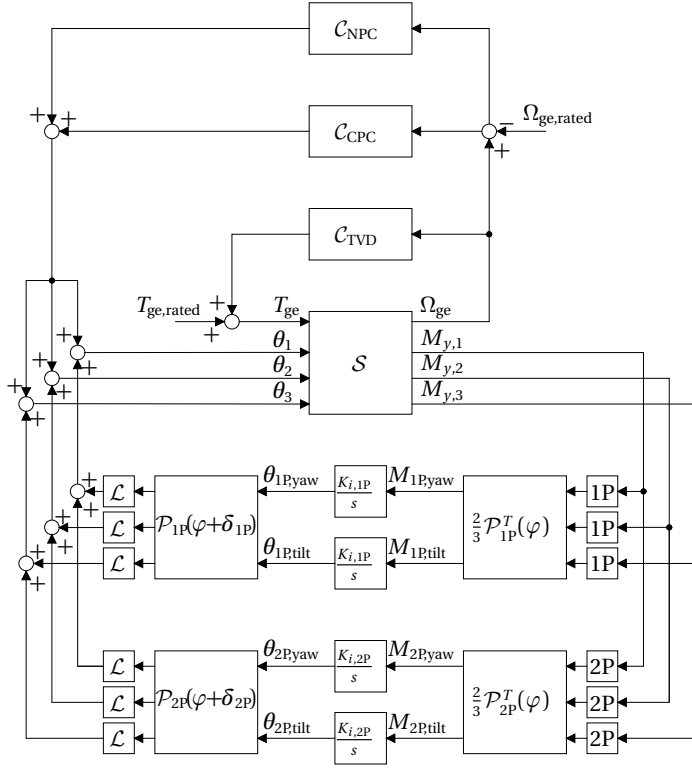
The typical solution is the use of the Coleman transformation (also referred to as the Park or the Multiple Blade Coordinate (MBC) transformation), which maps the variables of each blade in the rotating reference frame to a fixed reference frame of the nacelle-tower. For the application of the Coleman transformation to variable-speed wind turbines, it is assumed that the azimuth angle of the rotor blades are measured accurately. It is not assumed that the rotor blades have to be identical structurally or aerodynamically, however they have to be equally spaced around the rotor azimuth (Bir, 2008). Applying the Coleman transformation on complex models of wind turbines, such as the non-linear models used in GH BLADED, does not result in a completely decoupled LTI system. However, a Coleman transformation designed for the 1P frequency, will at least decouple the periodic signal components at that 1P frequency, such that SISO LTI feedback control techniques like loop-shaping can be used to reduce the 1P loads. Loads at higher harmonic frequencies can be reduced using the solutions in Bossanyi (2009); van Engelen (2006), where parallel IPC feedback loops are placed to the existing 1P-harmonic individual pitch controller, see Figure 5.1. In practice, IPC with Coleman transformation are primarily designed for reducing the 1P loads only, and sometimes 2P is considered. The remaining periodic components are ignored, because current solutions require at these higher harmonics considerable more pitch control action, due to the larger compensation needed for the larger amount of phase lag (delay) in the input-output dynamics, and from the more complicated MIMO controllers to overcome the increasing coupling between the transformed measurement channels.

The IPC was not fully specified in Bossanyi (2009), but by following the recipe in the same reference we have obtained a competitive implementation with almost similar performance results. Before the Coleman transformation, the out-of-plane bending moments measured in the blade root are scaled and filtered with a notch filter tuned to the  $jP$  frequency. The Coleman transformation is applied to decouple the loads in the direction of two axes, that for the 1P case may be thought as the asymmetrical load components representing the horizontal (yaw) and vertical (tilt) directions. The Coleman transformation for the  $jP$  harmonic frequency can be written as:

$$\begin{bmatrix} M_{jP_{yaw}} \\ M_{jP_{tilt}} \end{bmatrix} = \frac{2}{3} \mathcal{P}_{jP}^T(\varphi) \begin{bmatrix} M_{y,1} \\ M_{y,2} \\ M_{y,3} \end{bmatrix},$$

with

$$\mathcal{P}_{jP}(\varphi) = \begin{bmatrix} \cos(j\varphi) & \sin(j\varphi) \\ \cos\left(j\left(\varphi + \frac{2\pi}{3}\right)\right) & \sin\left(j\left(\varphi + \frac{2\pi}{3}\right)\right) \\ \cos\left(j\left(\varphi + \frac{4\pi}{3}\right)\right) & \sin\left(j\left(\varphi + \frac{4\pi}{3}\right)\right) \end{bmatrix},$$



**Figure 5.1:** Closed-loop wind turbine system  $\mathcal{S}$  for above-rated operation with the baseline controllers (CPC, NPC, TVD) and 1P and 2P IPC feedback loops in detail.

where  $\varphi$  is the azimuth angle of the first rotor blade, and  $M_{jP, \text{tilt}}$  and  $M_{jP, \text{yaw}}$  are the transformed loads. After the loads are transformed, each channel is decoupled and is integrated using I controllers, as recommended in Kanev (2009). The gains for the I controllers are tuned to be same for both yaw and tilt directions, as the case in Bossanyi (2009). However, for each rotational frequency different gains are applied. All the I controllers are subjected to limits of 5 deg on their outputs, which means that after reverse transformation the maximum amplitude of the near-sinusoidal individual pitch action is limited. For controllers with only integral action, this saturation limit is easily implemented, but for complicated controller designs the difficulties with saturation should be taken care of by using anti-windup loops as described in Kanev (2009). The corresponding reverse Coleman transformation is given by:

$$\begin{bmatrix} \theta_1 \\ \theta_2 \\ \theta_3 \end{bmatrix} = \mathcal{P}_{jP}(\varphi) \begin{bmatrix} \theta_{jP, \text{yaw}} \\ \theta_{jP, \text{tilt}} \end{bmatrix}.$$

After the reverse transformation, the signals are filtered again by a low-pass filter with a bandwidth of 1.6Hz (similar low-pass filter is used with CPC) to obtain a smooth control signals. The complete IPC controller for the 1P frequency can be represented

as:

$$C_{\text{IPC},1\text{P}}(s) = \underbrace{\frac{\omega_L^2}{s^2 + 2\zeta_L \omega_L s + \omega_L^2}}_{\text{low-pass filter } \mathcal{L}} I_{3 \times 3} P_{1\text{P}}(\varphi + \delta_{1\text{P}}) \begin{bmatrix} \frac{K_{i,1\text{P,yaw}}}{s} & 0 \\ 0 & \frac{K_{i,1\text{P,tilt}}}{s} \end{bmatrix} \frac{2}{3} P_{1\text{P}}^T(\varphi) \\ \times K_{1\text{P}} \underbrace{\frac{2\zeta_{1\text{P}} \omega_{1\text{P}} s}{s^2 + 2\zeta_{1\text{P}} \omega_{1\text{P}} s + \omega_{1\text{P}}^2}}_{\text{inverse notch filter 1P}} I_{3 \times 3},$$

where the parameters, including the parameters for the 2P harmonic frequency, are specified in Table 5.4.

The performance of these IPC controllers can be reduced considerably by having a large phase lag (delay) between the controller and the pitch actuator. It was suggested in Bossanyi (2009), that a small offset added to the azimuth angle in the reverse Coleman transformation can be used to compensate the delay. Moreover, as the control signals from the 1P and 2P harmonics are near sinusoidal it is also possible to compensate for the complete phase lag of the open-loop dynamic system. In our case, by looking at the Bode diagram of the linearised plant with filters, this is 25 deg and 33 deg for the 1P and 2P harmonic frequencies, respectively. With this additional phase compensation and the non-linear transformation using the azimuth angle, one could say that the IPC with Coleman transformation becomes and behaves more as an adaptive feedforward controller than a feedback controller. Although, this becomes considerable more challenging for higher harmonics.

**Table 5.4:** Parameters of individual pitch control

| description                    | symbol                 | value                                 |
|--------------------------------|------------------------|---------------------------------------|
| I controller                   |                        |                                       |
| integral gain                  | $K_{i,1\text{P,yaw}}$  | 0.01                                  |
| integral gain                  | $K_{i,1\text{P,tilt}}$ | 0.01                                  |
| integral gain                  | $K_{i,2\text{P,yaw}}$  | 0.002                                 |
| integral gain                  | $K_{i,2\text{P,tilt}}$ | 0.002                                 |
| low-pass filter $\mathcal{L}$  |                        |                                       |
| frequency                      | $\omega_L$             | 10 rad/s ( $\approx 0.16$ Hz)         |
| damping                        | $\zeta_L$              | 1                                     |
| inverse notch filter 1P        |                        |                                       |
| scaling gain                   | $K_{1\text{P}}$        | $10^{-6}$                             |
| frequency                      | $\omega_{1\text{P}}$   | $2\pi/4.95$ rad/s ( $\approx 0.2$ Hz) |
| damping                        | $\zeta_{1\text{P}}$    | 0.5                                   |
| inverse notch filter 2P        |                        |                                       |
| scaling gain                   | $K_{2\text{P}}$        | $10^{-6}$                             |
| frequency                      | $\omega_{2\text{P}}$   | $4\pi/4.95$ rad/s ( $\approx 0.4$ Hz) |
| damping                        | $\zeta_{2\text{P}}$    | 0.3                                   |
| reverse Coleman transformation |                        |                                       |
| phase offset                   | $\delta_{1\text{P}}$   | $25\pi/180$ rad                       |
| phase offset                   | $\delta_{2\text{P}}$   | $33\pi/180$ rad                       |

### 5.2.5 Wind conditions during the simulations

A variety of simulations are performed in GH BLADED in order to evaluate the effectiveness of the load reduction controllers; the baseline controllers CPC, NPC, and TVD with either IPC-1P, IPC-2P, or RC. It has been shown in Kühn (2001), that the wind phenomenon can be separated in a low and high frequent part, where the low frequent part is the undisturbed mean wind speed and the high frequent part is the turbulence. The external wind conditions used in the simulations, or load cases, are derived from the International Electrotechnical Commission (IEC) standards, specified in IEC (2004). Following the standards, the simulations are performed with at least of a length of 600 s, the mean wind speed values of 16 and 20 m/s, and with 3D turbulent data generated using the Von Karman spectrum model in GH BLADED. For each mean wind speed, eight load cases are generated in ascending order of turbulence intensity, up to the extreme turbulence intensity of  $I_T = 0.16$ . The details of the eight load cases in the turbulence intensities for all three the directions are summarized in Table 5.5. For the purpose of fatigue damage calculation, the IEC standard considers that the turbulence intensity of  $I_T = 0.16$  is “high”,  $I_T = 0.14$  is “medium”, and  $I_T = 0.12$  is “normal”. However, in Nino and Eecen (2001) it was concluded from measurements at multiple offshore locations that the common longitudinal turbulence intensities for a wind speed interval from 9 m/s to 18 m/s at 50 m height is in the range between 8% and 10%.

**Table 5.5:** Turbulence intensities for a reference wind speed of 15 m/s generated by the Von Karman model, and the related standard deviations of the period time.

| turb. intensity | longitudinal | lateral | vertical | std ( $ph$ ) |
|-----------------|--------------|---------|----------|--------------|
| $I_T = 0$       | —            | —       | —        | —            |
| $I_T = 0.04$    | 4.05 %       | 3.27 %  | 2.46 %   | 0.024 s      |
| $I_T = 0.06$    | 6.01 %       | 4.79 %  | 3.52 %   | 0.030 s      |
| $I_T = 0.08$    | 8.01 %       | 6.35 %  | 4.62 %   | 0.036 s      |
| $I_T = 0.10$    | 10.01 %      | 7.90 %  | 5.70 %   | 0.041 s      |
| $I_T = 0.12$    | 12.04 %      | 9.48 %  | 6.81 %   | 0.047 s      |
| $I_T = 0.14$    | 14.01 %      | 11.01 % | 7.88 %   | 0.056 s      |
| $I_T = 0.16$    | 16.01 %      | 12.57 % | 8.97 %   | 0.060 s      |

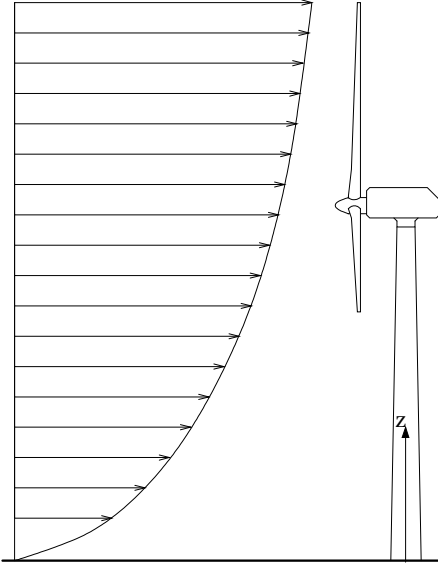
In the lower layers of the atmosphere, the mean wind speed is influenced by the friction with the surface of the earth. The roughness of the landscape, like buildings and trees, could reduce the wind speed considerably and increase turbulence. Two important periodic wind effects created by these obstructions in the airflow near the wind turbine are considered in the simulations: wind shear, and tower shadow. Wind shear is the variation in height of the average undisturbed wind velocity, see Figure 5.2. The selected exponential model relates the local wind speed disturbed by the wind shear effect at height  $z$  above the ground level to the wind speed at hub height  $v(z_0)$  as (GH , 2010a):

$$v_{ws}(z) = v(z_0) \left( \frac{z}{z_0} \right)^\alpha,$$

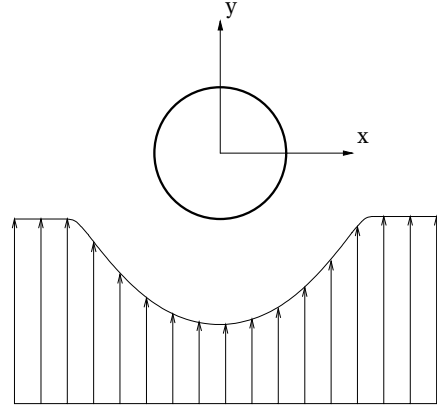
where  $\alpha = 0.2$  is the wind shear exponent. Tower shadow is the distortion of wind speed caused by the presence of a support structure in the wind flow, see Figure 5.3. The potential flow model relates modifies the local wind speed using the assumption of incompressible laminar flow around a cylinder of diameter  $D$  as (GH , 2010a):

$$v_{ts}(x, y) = \left( 1 - \frac{y^2 - x^2}{(y^2 + x^2)^2} \frac{(\kappa D)^2}{4} \right) v_{ws}(x, y),$$

where  $\kappa = 1.2$  is the tower diameter correction factor.



**Figure 5.2:** The left sideview of a wind turbine, showing the wind shear effect.



**Figure 5.3:** The topview of a cross section from the wind turbine tower, showing the tower shadow effect.

### 5.3 Lifted Repetitive Control Design

In this previous section the baseline controllers for above-rated operation, and a typical IPC, were described for comparison. Now the theoretical framework for the repetitive control method to reject periodic disturbances is presented. First, the system descriptions, the assumptions made, and some notations are presented. Then the goal is to formulate the RC design problem into an output-feedback problem. After the formulation, the state-space matrices of the lifted repetitive controller can be calculated by solving two large Riccati equations. For this purpose, the linearisation procedure for obtaining the state-space matrices are described. At last, the extensions to RC for the uncertain period time are discussed and some tuning guidelines are provided.

### 5.3.1 System descriptions and assumptions

The dynamics of the linearised system  $\mathcal{S}$  to be controlled can be written by the following state-space model:

$$\mathcal{S} \begin{cases} x_{k+1} = A_k x_k + B_k u_k + F_k d_k + w_k, \\ y_k = C_k x_k + D_k u_k + v_k, \end{cases} \quad (5.1)$$

where  $x_k \in \mathbb{R}^n$ ,  $u_k \in \mathbb{R}^r$ ,  $y_k \in \mathbb{R}^\ell$ ,  $d_k \in \mathbb{R}^f$ ,  $w_k \in \mathbb{R}^n$ , and  $v_k \in \mathbb{R}^\ell$ , are the state, input, output, periodic disturbance, process noise, and measurement noise vectors. The state-space matrices  $A_k \in \mathbb{R}^{n \times n}$ ,  $B_k \in \mathbb{R}^{n \times r}$ ,  $F_k \in \mathbb{R}^{n \times f}$ ,  $C_k \in \mathbb{R}^{\ell \times n}$ , and  $D_k \in \mathbb{R}^{\ell \times r}$  are also called the state, input, periodic input, output, and direct feedthrough matrices, respectively. In our case, the input vector contains the individual pitch angle demands, the output vector contains the individual blade-root bending moments, and the disturbance vector contains the wind disturbances as:

$$u_k = [\theta_{1,k} \quad \theta_{2,k} \quad \theta_{3,k}]^T, \quad y_k = [M_{y1,k} \quad M_{y2,k} \quad M_{y3,k}]^T, \quad d_k = [v_{1,k} \quad v_{2,k} \quad v_{3,k}]^T.$$

For simplification and clarification, we consider an LTV system of fixed order. However, the lifted repetitive control method described in this chapter can be extended to LTV systems of varying order. The linear time-varying dynamics are also considered to be periodic, thus  $A_k = A_{k-p}$ ,  $B_k = B_{k-p}$ ,  $F_k = F_{k-p}$ ,  $C_k = C_{k-p}$ , and  $D_k = D_{k-p}$ . The process and measurement noise disturbances are considered Gaussian noise sequences with  $E[v_k v_k^T] = R_v$  and  $E[w_k w_k^T] = R_w$ , respectively. In the case of coloured Gaussian noise sequences, for example with a random walk model to represent the wind turbulence dynamics as described in Selvam et al. (2009), the state-space system  $\mathcal{S}$  should be augmented with the noise filter dynamics.

With the trial window size  $p \in \mathbb{N}^+$ , the following stacked vector can be defined as:

$$\bar{y}_{k,p} = [y_k^T \quad y_{k+1}^T \quad \cdots \quad y_{k+p-1}^T]^T.$$

The stacked vectors  $\bar{u}_{k,p}$ ,  $\bar{d}_{k,p}$ ,  $\bar{n}_{k,p}$ ,  $\bar{w}_{k,p}$ , and  $\bar{v}_{k,p}$  are defined in a similar way. Using the definition of the stacked vectors, we can “lift” the system in (5.1) to the following lifted description:

$$\bar{\mathcal{S}} \begin{cases} x_{k+p} = \Phi_{k,p} x_k + \mathcal{L} \bar{u}_{k,p} + \mathcal{F} \bar{d}_{k,p} + \mathcal{P} \bar{w}_{k,p}, \\ \bar{y}_{k,p} = \Gamma x_k + \mathcal{G} \bar{u}_{k,p} + \mathcal{J} \bar{d}_{k,p} + \mathcal{H} \bar{w}_{k,p} + \bar{v}_{k,p}, \end{cases} \quad (5.2)$$

where  $\Phi_{k,p} = A_{k+p-1} \cdots A_{k+1} A_k$  is the state transition matrix, and  $\mathcal{L} \in \mathbb{R}^{n \times pr}$ ,  $\mathcal{F} \in \mathbb{R}^{n \times pf}$ , and  $\mathcal{P} \in \mathbb{R}^{n \times pn}$  are the extended controllability matrices, and are given by:

$$\mathcal{L} = [\Phi_{k,p-1} B_{k+p-1} \quad \cdots \quad A_k B_{k+1} \quad B_k], \quad \mathcal{F} = [\Phi_{k,p-1} F_{k+p-1} \quad \cdots \quad A_k F_{k+1} \quad F_k], \\ \mathcal{P} = [\Phi_{k,p-1} \quad \cdots \quad A_k \quad I],$$

and  $\mathcal{G} \in \mathbb{R}^{p\ell \times pr}$ ,  $\mathcal{J} \in \mathbb{R}^{p\ell \times pf}$ , and  $\mathcal{H} \in \mathbb{R}^{p\ell \times pn}$  are the impulse matrices with a lower block triangular structure,  $\Gamma \in \mathbb{R}^{p\ell \times n}$  is the extended observability matrix, and are given in (5.3). These lifted system matrices are trial invariant, because of the

assumption that the time-domain dynamics are periodic. Note, that  $I$  is used to represent an identity matrix of appropriate dimensions.

$$\begin{aligned}
 \Gamma &= \begin{bmatrix} C_k \\ C_{k+1}A_k \\ \vdots \\ C_{k+p-1}\Phi_{k+1,p-1} \end{bmatrix}, \quad \mathcal{G} = \begin{bmatrix} D_k & 0 & \cdots & 0 \\ C_{k+1}B_k & D_{k+1} & \ddots & \vdots \\ \vdots & \ddots & \ddots & 0 \\ C_{k+p-1}\Phi_{k+1,p-1}B_k & C_{k+p-1}\Phi_{k+2,p-1}B_{k-1} & \cdots & D_{k+p-1} \end{bmatrix}, \\
 \mathcal{H} &= \begin{bmatrix} 0 & 0 & \cdots & 0 \\ C_{k+1} & 0 & \ddots & \vdots \\ \vdots & \ddots & \ddots & 0 \\ C_{k+p-1}\Phi_{k+1,p-1} & C_{k+p-1}\Phi_{k+2,p-1} & \cdots & 0 \end{bmatrix}, \\
 \mathcal{J} &= \begin{bmatrix} 0 & 0 & \cdots & 0 \\ C_{k+1}F_k & 0 & \ddots & \vdots \\ \vdots & \ddots & \ddots & 0 \\ C_{k+p-1}\Phi_{k+1,p-1}F_k & C_{k+p-1}\Phi_{k+2,p-1}F_{k-2} & \cdots & 0 \end{bmatrix}.
 \end{aligned} \tag{5.3}$$

The stacked periodic disturbance may consist of an integrated “random walk” component as (Chin et al., 2004; Cho et al., 2005; Lee et al., 2001):

$$\bar{d}_{k,p} = \bar{d}_{k-p,p} + \bar{n}_{k,p}, \tag{5.4}$$

where  $\bar{n}_{k,p}$  is a Gaussian noise sequence with  $E[\bar{n}_{k,p}\bar{n}_{k,p}^T] = \bar{R}_n$ . This is a quite versatile noise formulation; the process disturbance contains not only a stochastic and a periodic component, but also a random walk nature, so the “periodic disturbance” can vary between trials. In addition, the noise and periodic disturbances can be correlated in time.

In Assumption 5.1, we adopt the mild conditions commonly used for the lifted repetitive control problem (Rice and Verhaegen, 2010b).

**Assumption 5.1.**

- The system  $\bar{S}$  is asymptotically stable (if not, stabilize with feedback control), controllable on  $(\Phi_{k,p}, \mathcal{L})$ , observable on  $(\Phi_{k,p}, \Gamma)$ , and does not contain any zeros at +1.
- The process noise  $w_k$  and measurement noise  $v_k$  are zero-mean white or coloured (if so, augment dynamics to system) Gaussian sequences.
- The “random walk” noise  $\bar{n}_{k,p}$  is a zero-mean white Gaussian sequence.

When the number of inputs is smaller than the number of outputs ( $r < \ell$ ), the controllability/observability condition in Assumption 5.1 is normally violated. In this case the matrix  $\mathcal{G} + \Gamma(I - \Phi_{k,p})^{-1}\mathcal{L}$  does not have full row rank. It is proposed to only reduce the error that lies in the column space of that matrix, thus we try not to achieve

perfect rejection for all disturbance signals (or all the harmonics). To achieve this, we use the Singular Value Decomposition (SVD) as:

$$\mathcal{G} + \Gamma(I - \Phi_{k,p})^{-1} \mathcal{L} = \begin{bmatrix} U_s & U_\perp \end{bmatrix} \begin{bmatrix} \Sigma_s & 0 \\ 0 & \Sigma \end{bmatrix} \begin{bmatrix} V_s \\ V_\perp \end{bmatrix}, \quad (5.5)$$

where the order  $s$  is determined by detecting a gap that separates the  $s$  largest singular values from the remaining ones. Now, we can replace the output equation of the lifted system in (5.2) by a projected output equation given by:

$$U_s^T \tilde{y}_{k,p} = U_s^T \Gamma x_k + U_s^T \mathcal{G} \tilde{u}_{k,p} + U_s^T \mathcal{J} \tilde{d}_{k,p} + U_s^T \mathcal{H} \tilde{w}_{k,p} + U_s^T \tilde{v}_{k,p}. \quad (5.6)$$

Furthermore, we can select the number of singular values  $s$  not only on the controllability/observability condition, but also to further reduce the size of the RC output-feedback problem and let the lifted repetitive controller focus on the main harmonic components and therefore robustify the controller performance in some sense.

### 5.3.2 Lifted repetitive control by LQG design

In this subsection the repetitive control method for periodic LTV systems with both repetitive and non-repetitive disturbances is presented. To let the repetitive controller respond to periodic disturbances, a “non-stochastic” output from (5.2) is defined as:

$$\tilde{y}_{k,p} = \bar{y}_{k,p} - \mathcal{H} \tilde{w}_{k,p} - \tilde{v}_{k,p} = \Gamma x_k + \mathcal{G} \tilde{u}_{k,p} + \mathcal{J} \tilde{d}_{k,p}, \quad (5.7)$$

and a “non-stochastic” error independent of the noise disturbances from trial  $k$  ( $\tilde{w}_{k,p}$ ,  $\tilde{v}_{k,p}$ ) as:

$$\tilde{e}_{k,p} = \bar{r}_{k,p} - \tilde{y}_{k,p}, \quad (5.8)$$

where  $\bar{r}_{k,p}$  is considered trial invariant. The change between two trials (or periods) is denoted by the  $\Delta$  operator, such that for example the difference between the vector of the input from time  $k$  and the vector of the input from time  $k-p$  is defined as  $\Delta \tilde{u}_{k,p} = \tilde{u}_{k,p} - \tilde{u}_{k-p,p}$ . Using this definition, it follows that the “non-stochastic” tracking error is defined as:

$$\tilde{e}_{k,p} = \tilde{e}_{k-p,p} - \Gamma \Delta x_k - \mathcal{G} \Delta \tilde{u}_{k,p} - \mathcal{J} \tilde{n}_{k,p}. \quad (5.9)$$

An expression for the current “stochastic” tracking error based on the previous “non-stochastic” error can be obtained and is given by:

$$\bar{e}_{k,p} = \tilde{e}_{k,p} - \mathcal{H} \tilde{w}_{k,p} - \tilde{v}_{k,p}. \quad (5.10)$$

By using the  $\Delta$  operator again on the state as  $\Delta x_k = x_k - x_{k-p}$ , the difference in the initial conditions of  $\tilde{S}$  between trials is derived from (5.7), and (5.4) as:

$$\Delta x_{k+p} = \Phi_{k,p} \Delta x_k + \mathcal{L} \Delta \tilde{u}_{k,p} + \mathcal{F} \tilde{n}_{k,p} + \mathcal{P} \Delta \tilde{w}_{k,p}. \quad (5.11)$$

Now we can combine (5.9), (5.10), and (5.11) into an stochastic linear system description as:

$$\tilde{S}_{\text{RC}} \begin{cases} \xi_{k+p} &= \mathcal{A} \xi_k + \mathcal{B} \Delta \tilde{u}_{k,p} + \mathcal{B}_w \tilde{v}_{k,p}, \\ \bar{e}_{k,p} &= \mathcal{C} \xi_k + \mathcal{D} \Delta \tilde{u}_{k,p} + \mathcal{D}_w \tilde{v}_{k,p}, \end{cases} \quad (5.12)$$



where the vectors  $\xi_k \in \mathbb{R}^{p(\ell+n)+n}$ , and  $\bar{v}_{k,p} \in \mathbb{R}^{p(f+n+\ell)}$  are given by:

$$\xi_k = [\Delta x_k^T \quad \bar{e}_{k-p,p}^T \quad \bar{w}_{k-p,p}^T]^T, \quad \bar{v}_{k,p} = [\bar{n}_{k,p}^T \quad \bar{w}_{k,p}^T \quad \bar{v}_{k,p}^T]^T,$$

and the system matrices  $A \in \mathbb{R}^{p(\ell+n)+n \times p(\ell+n)+n}$ ,  $B \in \mathbb{R}^{p(\ell+n)+n \times pr}$ ,  $C \in \mathbb{R}^{p \times p(\ell+n)+n}$ ,  $D \in \mathbb{R}^{p \times pr}$ ,  $B_w \in \mathbb{R}^{p(\ell+n)+n \times p(f+n+\ell)}$ , and  $D_w \in \mathbb{R}^{p \times p(f+n+\ell)}$  are given by:

$$A = \begin{bmatrix} \Phi^{k,p} & 0 & -\mathcal{P} \\ -\Gamma & I & 0 \\ 0 & 0 & 0 \end{bmatrix}, \quad B = \begin{bmatrix} \mathcal{L} \\ -\mathcal{G} \\ 0 \end{bmatrix}, \quad C = [-\Gamma \quad I \quad 0], \quad D = -\mathcal{G},$$

$$B_w = \begin{bmatrix} \mathcal{F} & \mathcal{P} & 0 \\ -\mathcal{J} & 0 & 0 \\ 0 & I & 0 \end{bmatrix}, \quad D_w = -[\mathcal{J} \quad \mathcal{H} \quad I].$$

The lifted RC problem can now be formulated as:

**Problem Description 5.1** (Lifted repetitive control). *Given the system in the form of  $\tilde{S}_{RC}$  in (5.12), design a strictly-proper dynamic lifted repetitive controller  $\tilde{C}_{RC}$ .*

Given Assumption 5.1, the infinite-horizon discrete-time LQG solution is well known, see Åström and Wittenmark (1997), and it is desirable in our situation, as it stabilizes the system  $\tilde{S}_{RC}$  and minimizes the expectation of the cost function:

$$J = E \left[ \lim_{N \rightarrow \infty} \sum_{i=0}^N \|\bar{e}_{k+ip,p}\|_2^2 + \rho \|\Delta \bar{u}_{k+ip,p}\|_2^2 \right]. \quad (5.13)$$

Compared with the minimum-variance criterion in Rice and Verhaegen (2010b), we introduced in the LQG criterion a weight to penalize the differenced input, and therefore we are able to make a trade-off between the variance of the differenced input and output signals. By requiring that  $\tilde{C}_{RC}$  stabilizes  $\tilde{S}_{RC}$ , it can be shown to get exponential convergence of the RC algorithm. This does not imply that the RC algorithm has monotonic decay ( $\|\bar{e}_{k+p,p}\|_2 \leq \|\bar{e}_{k,p}\|_2$ ) as recommended in Longman (2000), but the experience is that the introduction of a weight on the differenced input could give the engineer some means to prevent for the large overshoots conventionally seen with some RC laws with non-monotonic decay in the initial transient periods. Using a steady-state approximation of the optimal LQG solution, the lifted controller  $\tilde{C}_{RC}$  is given by:

$$\tilde{C}_{RC} \begin{cases} \hat{\xi}_{k+p} &= \mathcal{A}_C \hat{\xi}_k + \mathcal{L}_C \bar{e}_{k,p}, \\ \Delta \bar{u}_{k,p} &= -\mathcal{K}_C \hat{\xi}_k, \end{cases} \quad (5.14)$$

where

$$\begin{aligned} \mathcal{A}_C &= A - BK_C - \mathcal{L}_C C + \mathcal{L}_C D K_C, \\ \mathcal{L}_C &= (\mathcal{S}_W + A Z C^T) (\mathcal{R}_W + C Z C^T)^{-1}, \\ \mathcal{K}_C &= (\mathcal{R} + B^T \mathcal{X} B)^{-1} (B^T \mathcal{X} A + S^T). \end{aligned}$$

The matrices  $\mathcal{X}$  and  $\mathcal{Z}$  are the positive-definite stabilizing solutions to the Discrete Algebraic Riccati Equations (DARE):

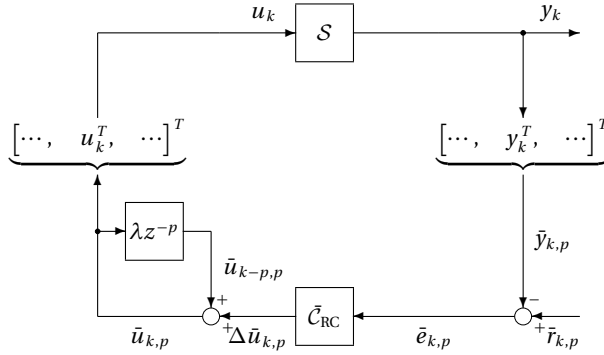
$$\begin{aligned} \mathcal{X} &= Q - (A^T X B + S) \mathcal{K}_C + A^T X A, \\ \mathcal{Z} &= Q_W - \mathcal{L}_C (C Z A^T + S_W) + A Z A^T, \end{aligned} \quad (5.15)$$

where  $\mathcal{R} = \rho I + \mathcal{D}^T \mathcal{D}$ ,  $\mathcal{Q} = \mathcal{C}^T \mathcal{C}$ ,  $\mathcal{S} = \mathcal{C}^T \mathcal{D}$ , and the disturbance covariance matrices are calculated as:

$$\begin{bmatrix} \mathcal{R}_W & \mathcal{S}_W^T \\ \mathcal{S}_W & \mathcal{Q}_W \end{bmatrix} = \begin{bmatrix} \mathcal{D}_W \\ \mathcal{B}_W \end{bmatrix} \mathbb{E} \begin{bmatrix} \tilde{v}_{k,p} \tilde{v}_{k,p}^T \end{bmatrix} \begin{bmatrix} \mathcal{D}_W \\ \mathcal{B}_W \end{bmatrix}^T. \quad (5.16)$$

In many cases, the solvers for the DARE in (5.15) have difficulties to determine the solution numerically. For example the QZ-based DARE solver in MATLAB<sup>TM</sup> will fail, because the Hamiltonian or symplectic matrix has eigenvalues on and/or close to the unit circle. To overcome this problem, the learning gain should be lowered by adding forgetting or leakage. With the inclusion of forgetting, the input difference is defined as  $\Delta \tilde{u}_{k,p} = \tilde{u}_{k,p} - \lambda \tilde{u}_{k-p,p}$ , where the scalar  $0 \ll \lambda < 1$  is the forgetting factor.

The block diagram in Figure 5.4 illustrates how the lifted repetitive controller including the memory loop can be implemented. The output measurements are  $p$ -times sampled during the previous trial and buffered into a stacked vector. For the start of the next trial, the lifted repetitive controller is evaluated and an updated stacked input vector is returned. According to the sample time, the elements of the calculated stacked input vector are unbuffered during the current trial and given as input to the pitch actuator of the wind turbine system. Note, that the lifted repetitive controller is strictly proper, because the current trial output measurements are assumed unavailable for feedback.



**Figure 5.4:** Block diagram of the wind turbine system  $S$  (CPC and TVD included in block) with the RC loop with one memory loop.

The biggest challenge for the implementation of this RC description is that the lifted matrices are enormous, and it requires large memory storage and can take a long time to calculate the solution. A possible solution that can be used to resolve the computational complexity issue is by exploiting the matrix structure. In Appendix C, efficient linear algebra computations using the so-called Sequentially Semi-Separable (SSS) matrix structure are proposed. The lifted repetitive controller could be implemented real time using three SSS structured matrix-vector computations such that the complexity is  $\mathcal{O}(\ell^2 p)$  instead of  $\mathcal{O}(\ell^2 p^2)$ .

Another challenge for lifted RC is that in practice the inputs are normally constrained to certain limits. All smart actuators currently in development have operational limits on the input signal; for example, piezo-electric flaps have always a

certain limitation in deflection, and MEM-tabs can only be switched on or off. If not taken care of correctly, it is well known that RC will destabilize the closed-loop system if the inputs are saturated. Since Model Predictive Control (MPC) can in principle handle the above features, our formulation can be converted to a repetitive MPC problem as have been done in Gupta and Lee (2006); Lee et al. (2001). The difficulty is that normally an optimization have to be performed online to solve the quadratic problem with inequality constraints. The problem with input saturation can occur for example when the wind turbine leaves the above-rated region. In the current RC implementation, the learning capability of RC is simply set to a hold when the collective pitch angle reaches 2 deg or lower. This means that the stacked input vector is not updated any more in the below-rated region. Moreover, the stacked input values gradually converge to zero due to the forgetting. If the wind turbine enters the above-rated operation region after a long leave, it would take several periods to arrive at steady-state.

### 5.3.3 Model linearisation for the lifted RC design

The use of models has a long tradition in control engineering. Most modern control techniques, including the proposed RC method, are model based and require a mathematical representation of the system under consideration. Thus a model describing the relevant number of properties of the wind turbine in a mathematical language is needed to tune and even design the lifted repetitive controller. The linearisation tool in GH BLADED is used for this purpose, to obtain a linear model in a state-space system form suited for the proposed RC design. During the linearisation, the following inputs are selected: the variations in the generator torque  $T_{ge}$ , the pitch angles  $\theta_i$ , and the wind speed disturbance  $v_i$  on each of the three rotor blades ( $i = 1, 2, 3$ ). Further, the following outputs are selected: the variations in generator speed  $\Omega_{ge}$ , the out-of-plane blade-root bending moments  $M_{y,i}$  of each rotor blade ( $i = 1, 2, 3$ ). In Figure 1.4, a schematic representation of the wind turbine with the selected inputs, disturbances and outputs are given.

The equations of motion that relate  $M_{y,1}$ ,  $M_{y,2}$  and  $M_{y,3}$  and the blade pitch angles  $\theta_1$ ,  $\theta_2$  and  $\theta_3$  are dependent on the azimuth angle  $\varphi$  and the rotational rotor speed  $\Omega_{ro}$ , therefore a periodic LTV system through linearisation is acquired. In our case, the azimuth is divided in 99 sections, thus for each increment of 3.636 deg(or 0.05s with rated rotational rotor speed) an LTI state-space system is obtained. The lifted RC loop is intended to run at the sample rate of 20Hz, i.e sample time of  $h = 0.05s$ , and the LTI system is discretized accordingly using the zero-order-hold method. The obtained linear models from the linearisation tool in GH BLADED are still in open-loop form and do not contain the dynamics of the baseline controllers. These controller dynamics have been included afterwards by closing the feedback loops, for example from the generator speed  $\Omega_{ge}$  to the the pitch angle  $\theta_i$  of each rotor blade with the CPC transfer description. Furthermore, we apply a second-order high-pass filter with bandwidth from 0.1Hz in series with the output of the blade-root bending moments  $M_{y,i}$  of each rotor blade. The dynamic transfer function of the high-pass filter is given

by:

$$\mathcal{H}(s) = \frac{s^2 + 2\zeta_{H,1}\omega_H s}{s^2 + 2\zeta_{H,2}\omega_H s + \omega_H^2},$$

where the frequency is  $\omega_H = 0.63$  rad/s and damping values are  $\zeta_{H,1} = 0.7$  and  $\zeta_{H,2} = 1$ . This high-pass behaviour is desirable for this simulation experiment, as we want to control the dynamic behaviour of the system, rather than the static deformations. This behaviour is normally present in the open-loop system, but then in the form of sensors dynamics, like with Macro Fiber Composites (MFC) sensors from Smart Material have typical high-pass behaviour. At last, a low-pass filter similar to the one in CPC and IPC is used to smooth the control signals from the RC and before they are sent to the pitch actuator.

For the model-based control design, the model should only contain the relevant dynamics between the inputs, and the outputs and should be accurate over the bandwidth of the controller; for RC design, these are the main harmonic frequencies. Model reduction could be applied (is not used in this chapter) to obtain a model of sufficiently small order with a relevant number of properties, which is normally preferred for model-based control design (Molenaar, 2003). In this chapter, we consider models based on the analytical first-principles modelling through the simulation environment of GH BLADED. The equations behind the software strongly depends on a large number of parameters that determine the (aero) dynamic behaviour of the wind turbine system. Most of these parameters can be roughly estimated or calculated. Still, a large amount of uncertainty can be present and this makes it difficult in practice to design a stable controller based on such models, because it is known from Doyle (1978) that the LQG design is sensitive to uncertainties. More accurate models could be devised from measured data, for example using the periodic LPV system identification technique in van Wingerden et al. (2009). It is suggested that identification becomes a necessary building block for the model-based RC design in practice.

### 5.3.4 Extensions to RC for uncertain period time

An important drawback of RC is that a small mismatch between the controller period and the actual period of the disturbance signals can decrease the performance of RC substantially. This drawback makes the application of RC less practical for wind turbines with small variations in the rotational rotor speed, in spite of a pitch controller that will keep (during above-rated operation) the rotational speed of the wind turbine close to the defined reference rotational speed. Several approaches have been proposed in the literature to improve robustness of RC in the presence of possible variations in the period-time of  $ph = 4.95$  s. For the proposed lifted RC, these possible approaches can be divided into two groups, depending if the true period time or the true rotational rotor speed is accurately measured.

An important issue for RC designed with lifted periodic LTV model compared with lifted LTI models is that the true azimuth angle of the wind turbine should correspond to the index in the sequence of LTV state-space matrices during lifting, otherwise the linearised LTV dynamics will not match with the true dynamics of the wind

turbine system. If not taken care of correctly, the variations in the period time will cause an increasing mismatch that can eventually lead to a considerable reduction in performance and even instability. To keep the period mismatch as minimal as possible during the operation, the buffering of the input and output samples over the period will be done according an increment of 3.636 deg in the azimuth angle of the first blade, which approximately equals the sampling time of  $h = 0.05$  s when the rotational rotor speed is rated. The azimuth angle of the first blade is assumed to be measured accurately, as is the case with IPC. In other words, we allow the sample rate to vary instead, because the sampler is designed and implemented using the angular position instead of time, so that the disturbance period is always exactly one rotation regardless of speed variations. On the other hand, sample time adaptation of a lifted repetitive controller where the sample rate of the underlying model is unchanged, may also result in a decrease in performance and even instability. The experience of the authors, and also the experience in Tsao et al. (2000); Wen and Longman (2001), is that the use of this approach affects the robustness of the closed-loop system in more favourable way, but a robustness analysis as given in Costa-Castelló et al. (2011); Olm et al. (2011) will be very challenging, and is therefore not explored.

If it is not possible to measure the period time accurately, it is common practice to formulate a high-order RC problem with multiple memory loops. in Pipeleers et al. (2008); Steinbuch (2002); Steinbuch et al. (2007) the RC problem is extended with more memory loops. This so-called high-order RC can be made more robust for small changes in period time, but this comes at the cost of the amplification of noise at the non-harmonic frequencies. As shown in Gupta and Lee (2006), multiple memory loops can be introduced in the lifted RC problem at the cost of larger matrices. Compared with the time-domain RC methods in Pipeleers et al. (2008); Steinbuch et al. (2007), the lifted RC problem in Rice and Verhaegen (2010b) allows a more versatile disturbance formulation to specify the amount of variation in the periodic wind disturbances. With the introduction of a random walk nature  $n_k$  in (5.4), the RC formulation allows the periodic disturbance to vary between trials. When the periodicity of the disturbances are known, for example with the identification techniques in van Baars et al. (1993); van Wingerden et al. (2011), it is even possible to relate the lifted covariance matrix  $\hat{R}_n$  with the variance of the period time  $ph$  by a first-order approximation as:

$$\hat{R}_n = \frac{\partial \bar{d}_{k,p}(ph)}{\partial ph} \text{var}(ph) \frac{\partial \bar{d}_{k,p}^T(ph)}{\partial ph}.$$

where the basis functions in  $\bar{d}_k$  are used to express the periodic disturbances. Both approaches perform for small variations in period time, however the latter approach is preferred in this case. First, the addition of more memory loops comes at the cost of a considerable increased computational burden. Second, the selection of the number of memory loops and its weighting factors is not an easy task. At last, it is noted that the combination of the approaches will probably only give 1–2% additional performance, as was experimentally shown in Chapter 4.

### 5.3.5 Guidelines for selection of the parameters

In this subsection, some guidelines are given on how to pick some initial values for the free parameters in the algorithm. During the design of the proposed lifted RC method, a large number of parameters have to be selected. We have the singular value order  $s$ , the disturbance covariance matrices  $R_w$ ,  $R_v$ , and  $\bar{R}_n$ , and the weight  $\rho$  on the differenced input, and the forgetting factor  $\lambda$ .

An initial guess for the number singular values  $s$  that have to be taken into account can be determined by detecting gaps that separates groups of singular values in (5.5). It turns out that for even lower values of  $s$ , the lifted repetitive controller will focus more on the main harmonic components and therefore robustify the controller performance in some sense. In our case, we selected the values  $s = 18$ ,  $s = 24$ , and  $s = 30$  for further investigation.

The values of the process and measurement noise covariance matrices  $R_w$ ,  $R_v$  must be chosen according to the level of measurement and process noise present in the controlled system. A close estimate of these covariance matrices could be found from an identification of the system to be controlled. In our simulations, apart of the turbulence, no additional process or measurement noise is considered, therefore we selected these values to be small:  $R_w = 0.01I$ , and  $R_v = 0.01I$ .

The choice of the non-periodic disturbance covariance matrix  $\bar{R}_n$  can be very challenging. First, the value depends on the level of non-periodic turbulence included in the wind disturbance. Second, the value can be increased even more to make the lifted repetitive controller more robust against period-time mismatch. In our case, the covariance matrix is chosen to be  $\bar{R}_n = \eta I$ , where  $\eta$  is a scalar to be selected. If the contributions of the weights  $R_w$ ,  $R_v$  to the cost function in (5.13) can be considered small, such that it can be neglected compared with the contribution of the weight  $\bar{R}_n$ , then the expected performance of the RC will only be a trade-off between  $\eta$  and  $\rho$ . So let  $\eta = 10$ , then we can only adjust  $\rho$  until a satisfactory response.

The strategy to pick the parameter  $\rho$  is very similar to the time-domain LQG problem, see Åström and Wittenmark (1997). It is common practice to pick the parameter  $\rho$  such that a smooth control signal is obtained and a large overshoot is avoided. in Longman (2000) it is even recommended to have a monotonic decrease of  $\|\tilde{e}_{k,p}\|_2$ , which is almost (apart from the initial differenced state) directly related to the monotonic increase of  $\|\tilde{u}_{k,p}\|_2$ . By increasing the weight  $\rho$ , the overshoot over the trials can be lowered in most cases and improve the smoothness of the generated control signal at the cost that it takes longer to converge to steady state and is less adaptive to any changes in the periodic disturbance.

The updated stacked input vector for the next period is a combination of the previous input vector times a forgetting factor and a small correction obtained from the lifted repetitive controller. Thus, the updated stacked input vector consists for large part by a component which is averaged over the past trials (or periods). Forgetting will introduce less weight on the past trials than more recent trials, such that the more recent trials are more relevant for the current periodic disturbance representation. With  $\lambda = 0.99$ , the stacked input vectors up to  $\frac{2}{(1-\lambda)} = 200$  periods in the past, will approximately contribute for 90% to the averaged stacked input vector,

therefore this relation can be helpful for an initial selection. Further, the value  $\lambda = 0.99$  for the forgetting factor is normally more than sufficient to overcome the problem with DARE solvers.

## 5.4 Simulation results and analysis

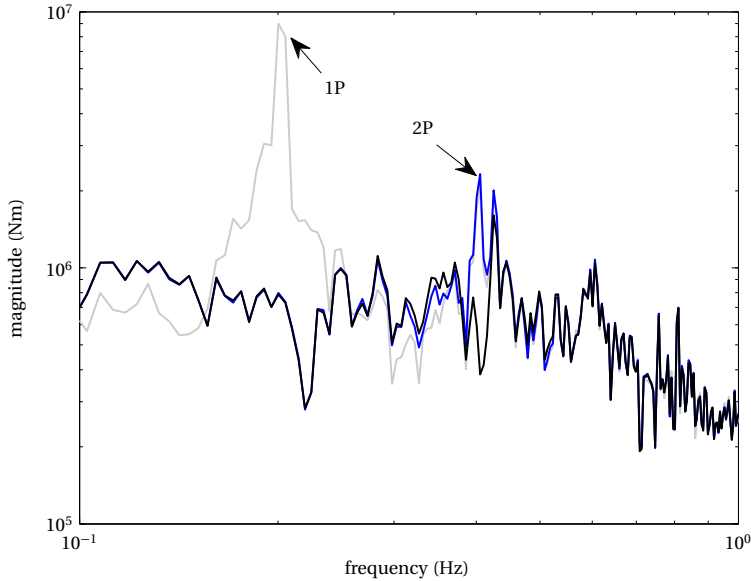
In this section, the main results of the different simulation studies on the UPWIND 5MW wind turbine operating in a periodic wind field are presented. The effectiveness of the proposed RC algorithm is analysed and compared with the IPC algorithms with Coleman transformation. First, we will show the results of the IPC with Coleman transformation. Second, we will compare them with the results of the proposed lifted RC. It should be emphasized that the relative performance of RC and IPC detailed here depends strongly on both the selection of the free parameters, for example less aggressive integrator gains would lower the pitch rates considerably and would likely lead to lower load reductions. A truly equal comparison is difficult to be pursued, however some important differences can be concluded from evaluating the trends.

The simulations are performed for two different mean wind speed values of 16 and 20 m/s, and for eight turbulent cases given in Table 5.5. To evaluate the performance in load reduction, the results of the blade-root bending moment  $M_y$ , the hub tilt moment  $M_{\text{tilt}}$ , and the hub yaw moment  $M_{\text{yaw}}$  are calculated in the form of the standard deviation (std) or in percentage of the baseline control results. The 1Hz damage equivalent loads were also calculated, however these results are more uncertain than the standard deviation, and therefore not considered. With the damage equivalent loads, it was much more difficult to compare the load reduction results between the different load cases, although similar trends can be shown in the results of equal turbulent cases. The cost of additional wear in the pitch actuators is quantified by an increase in the standard deviation of the pitch rate  $\text{std}(\dot{\theta})$ . The output of the more representative pitch forces or the related pitch accelerations are currently not supported in GH BLADED. It is expected that by examining the pitch rate the high-frequent pitch actions would be less dominant, therefore the results are less favourable than when the pitch acceleration would be used.

### 5.4.1 Performance of individual pitch controller

The simulation results of the percent load reduction and pitch action of IPC compared with the baseline control case are summarized in Tables 5.6 and 5.7. Four different IPC cases are considered; namely IPC-1P, IPC-1P-25deg, IPC-2P, and IPC-2P-33deg. These names represent the IPC approaches as discussed in Subsection 5.2.4. IPC-1P contains only the 1P feedback loop with a phase offset of  $\delta_1 = 0$  deg. IPC-1P-25deg contains only the 1P feedback loop with a phase offset of  $\delta_1 = 25$  deg. IPC-2P contains both 1P and 2P feedback loops with  $\delta_{1,2} = 0$  deg. IPC-2P-33deg contains both 1P and 2P feedback loops with  $\delta_1 = 25$  deg and  $\delta_2 = 33$  deg. All the IPC cases produce sizeable reductions in standard deviation of the loads in all turbulent cases and the two mean wind speeds. This comes at the cost that the pitch rates are considerably larger; for the higher turbulent cases, it is between two and four times higher.

Several observations can be made regarding the different IPC cases. The introduction of the 2P feedback loop is only beneficial in load reduction, if the phase offset of the reverse Coleman transformation is tuned correctly. For the high turbulent cases, this will give 1-2% extra load reduction at the cost of an increase 5% in pitch rate. In Figure 5.5, the square-root Power Spectral Density (PSD) of the blade-root bending moments are given for the baseline (light grey), IPC-1P-25deg (blue), and IPC-2P-33deg (black) controlled system. The PSD plots are very similar for both the 1P and 2P cases, except that the 2P frequency is unaffected in the 1P case.

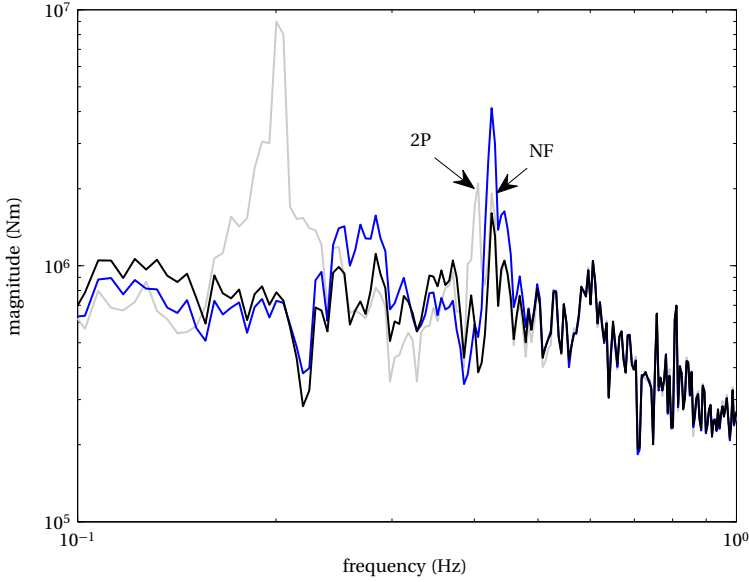


**Figure 5.5:** Square-root PSD of the blade-root bending moment  $M_{y,1}$ . Black line is with IPC-1P-25deg, blue line is with IPC-2P-33deg, and light grey line is with baseline control. For  $I_T = 0.12$ ,  $v = 16\text{m/s}$ .

Tuning the phase offset of the reverse Coleman transformation is clearly beneficial for both the 1P and 2P case for high turbulent cases. Especially, in the form of pitch rate needed to obtain similar or better load reduction results. For the 1P case, a reduction of 5% in pitch rate is observed, and for the 2P case even 10%. In the low turbulent cases, the results are very similar. In the cases without turbulence, the individual control signals for the tilt and yaw directions are almost a perfect sine and cosine wave, respectively. It is observed that the combined pitch signal is also a sine wave, but with a compensated phase shift constructed by a combination of the individual sine and cosine waves. With increasing turbulence intensity, the individual control signals become less sinusoidal, and a combination will not lead to perfectly corrected phase and will create signals with larger amplitudes. In Figure 5.6, the square-root PSD of the first blade-root bending moments are given for the baseline (light grey), IPC-2P (blue), and IPC-2P-33deg (black) controlled system. By comparing the PSD plots, the 0 deg case shows, compared with 25-33 deg case, an amplification at frequencies close to right of the 1P and 2P frequencies. This causes an additional



amplification of a natural frequency of an in-plane mode, which explains the worse load reduction results obtained with IPC-2P.

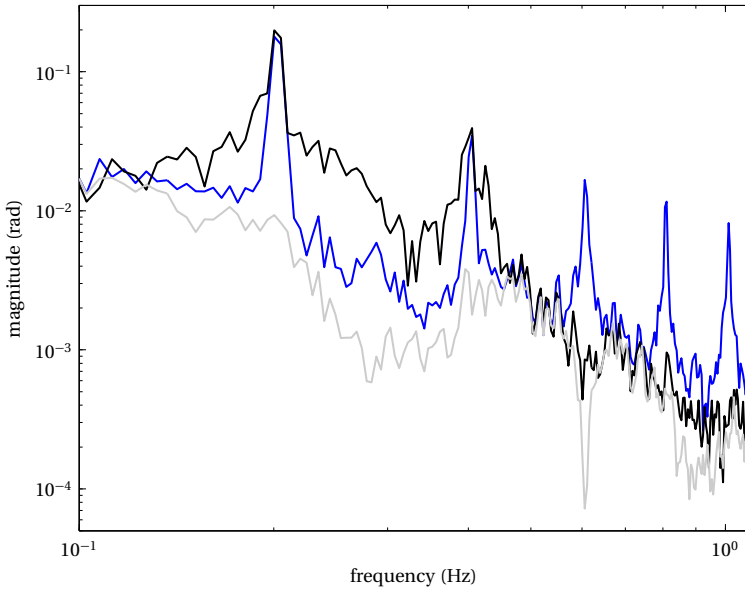


**Figure 5.6:** Square-root PSD of the blade-root bending moment  $M_{y,1}$ . Black line is with IPC-2P-33deg, blue line is with IPC-2P, and light grey line is with baseline control. For  $I_T = 0.12$ ,  $v = 16$  m/s.

#### 5.4.2 Performance of lifted repetitive controller

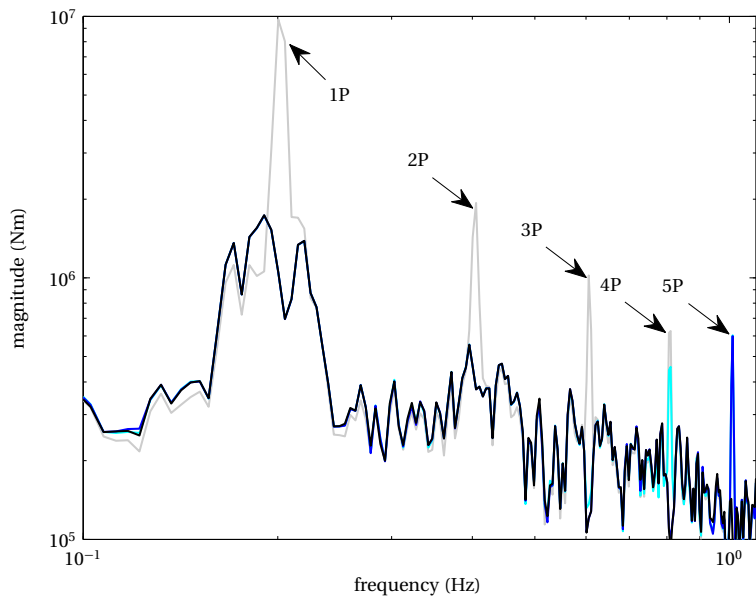
The simulation results of the percent load reduction and pitch action of RC compared with the baseline control case are summarized in Tables 5.6 and 5.7. Three different RC cases are considered; namely RC-18s, RC-24s, and RC-30s. These names represent to the RC approaches as discussed in Section 5.3. RC-18s, RC-24s, and RC-30s reduces the error that lies in a column space of the largest 18, 24, and 30 singular values. The simulation results in the tables are obtained by improving the load reduction results by tuning the parameter  $\rho$ , and by keeping the parameter  $\rho$  is constant for the value tuned for  $I_T = 0.12$ . All the RC cases produces sizeable reductions in standard deviation of the loads in all turbulent cases and the two mean wind speeds. Compared with IPC, RC performs better in the low turbulent cases; like,  $I_T = 0$  and  $I_T = 0.04$ . From  $I_T = 0.10$  and larger, IPC will outperform RC. This was expected, because the relative performance of RC depends heavily on the ratio between amount of periodic and stochastic disturbances. The reason is that RC focus mainly on the rejection periodic disturbances, where IPC also acts on turbulence by feedback. Further, more turbulence will generate more variations in rotational speed that can affect the performance of RC. That RC mainly focuses on the periodic components, is visible in the values for the pitch rate. For the increasing turbulence intensity, these pitch-rate values hardly increase when compared relative with the baseline results. We have 15%

to 30% less pitch rate action than for IPC at high turbulent cases. In Figure 5.7, the square-root of the PSD of the pitch angles of the first blade are given for the baseline (light grey), RC-30s (blue), and IPC-2P-33deg (black) controlled system. The PSD plots shows that compensation of the pitch actuator for RC is mainly focused on the 1P, 2P, 3P, 4P, and 5P frequencies, where IPC compensates the 1P, 2P and intermediate frequencies instead.

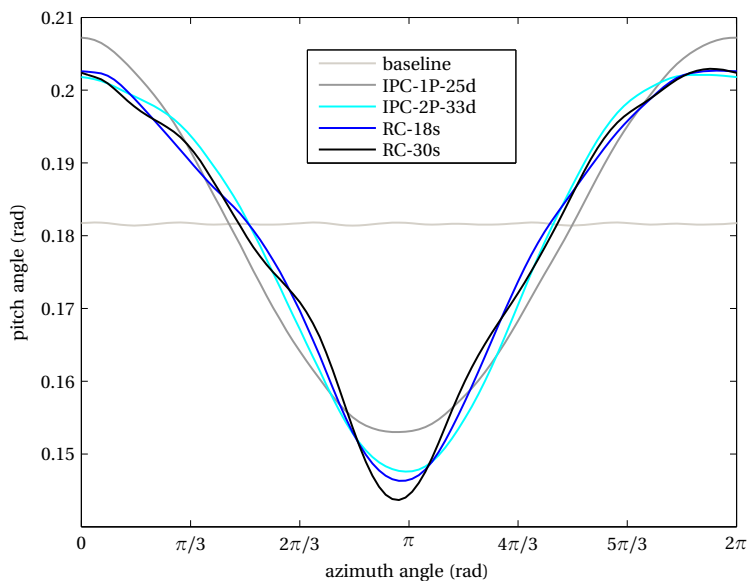


**Figure 5.7:** Square-root PSD of the pitch angle  $\theta_1$ . Black line is with IPC-2P-33deg, blue line is with RC-30s with  $\rho = 3 \cdot 10^5$ , and light grey line is with baseline control. For  $I_T = 0.12$ ,  $v = 16$  m/s.

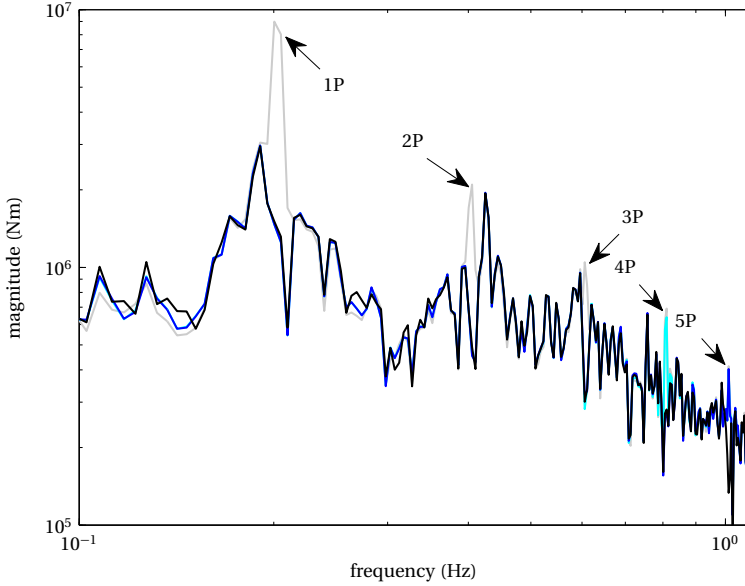
Several observations can be made regarding the different RC cases. The introduction of larger column space, such that RC compensates more harmonic components, does not give much performance improvements in terms of the standard deviation. The dominance of the 1P and 2P peak makes it difficult to investigate the load reduction capabilities of RC at higher harmonic frequencies. In Figure 5.8, the square-root PSD of the first blade-root bending moments are given for the baseline (light grey), RC-18s (light blue), RC-24s (blue), and RC-30s (black) controlled system. The reduction of the 4P and 5P peaks can be viewed in the spectrum, when the number singular values are increased: the RC-24s case rejects the 4P frequency, and the RC-30s case also the 5P frequency. In Figure 5.9, the variation in pitch angle  $\theta_1$  during one period is given. It shows that the addition of the 4P and 5P frequencies are translated to the time domain as a better fit to compensate the tower shadow effect. In Figure 5.10, the square-root PSD of the first blade-root bending moments are given when the turbulence intensity is increased from  $I_T = 0.04$  to  $I_T = 0.12$ . The contribution of the 4P and 5P frequencies are now almost neglectable compared with the surrounding noise level, therefore the additional harmonics are probably not worthwhile to compensate for.



**Figure 5.8:** Square-root PSD of the blade-root bending moment  $M_{y,1}$ . Black line is with RC-30s, blue line is with RC-24s, light blue line is with RC-18s, and light grey line is with baseline control. For  $I_T = 0.04$ ,  $v = 16$  m/s.

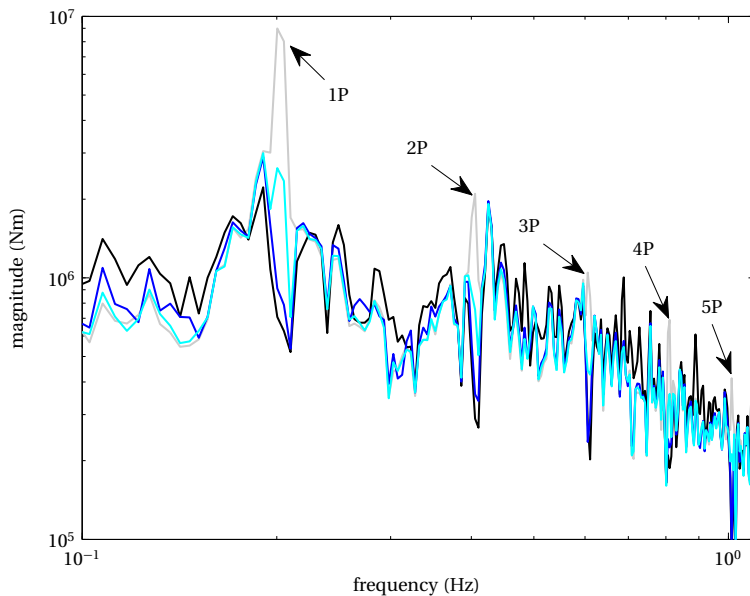


**Figure 5.9:** The variation in pitch angle  $\theta_1$  during one period. Averaged over 600 s. For  $I_T = 0$ ,  $v = 16$  m/s.

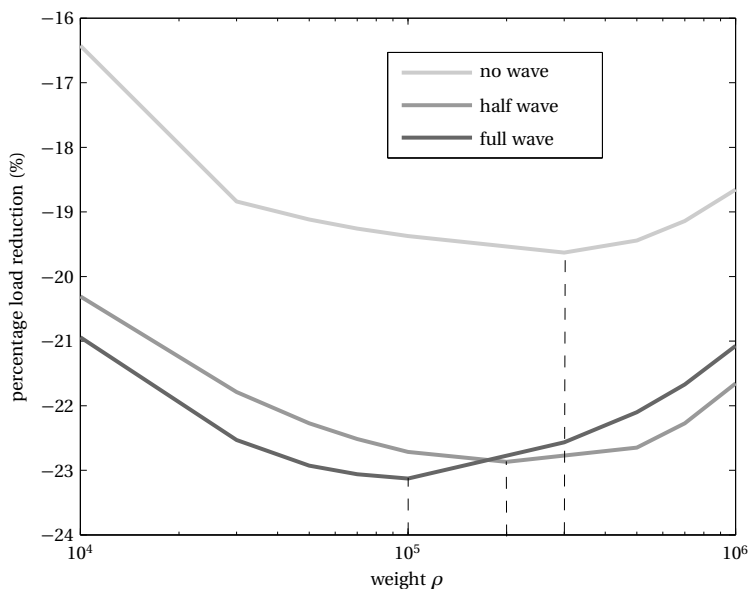


**Figure 5.10:** Square-root PSD of the blade-root bending moment  $M_{y,1}$ . Black line is with RC-30s, blue line is with RC-24s, light blue line is with RC-18s, and light grey line is with baseline control. For  $I_T = 0.12$ ,  $v = 16$  m/s.

Other observations can be made regarding the tuning of the parameter  $\rho$ . From the Tables 5.6 and 5.7, it can be determined that increasing the weight  $\rho$  improves the smoothness of the generated pitch signal at the cost that it is less adaptive to any changes in the periodic disturbance and takes longer to converge. Selecting a low-valued weight  $\rho$  can be beneficial for the load reduction in the low turbulent cases; like,  $I_T = 0$  and  $I_T = 0.04$ . It is not surprising, that for larger turbulence intensities the weight  $\rho$  should be increased to force the RC to design the pitch angle vector over more periods. When selecting a constant  $\rho$  (tuned for  $I_T = 0.12$ ), the results in Table 5.6 show only small reduction in load reduction performance. In Figure 5.11, the square-root PSD of the first blade-root bending moments are given baseline (light grey),  $\rho = 10^6$  (light blue),  $\rho = 10^5$  (blue), and  $\rho = 10^4$  (black) weights. It is visible in the figure, that small weights cause more unwanted amplifications at the intermediate frequencies, which explains the decreased load reduction at large turbulence intensities. The selection of the parameter  $\rho$  should not only be done on the basis of constant load cases. The RC can adapt the pitch action to the changing periodic loads, and probably a lower weight  $\rho$  is better to faster adapt to any changes. For this purpose, a sinusoidal wind direction (yawed error) transient with amplitude of 10deg is added to the simulations. In Figure 5.12, the percentage reduction in the blade-root bending moment is given for the cases of no transient (no wave), a slow transient (half wave), and a fast transient (full wave). It is shown that a better load reduction can be obtained by selecting a smaller weight to follow the adaptive direction of the wind.



**Figure 5.11:** Square-root of the PSD of the blade-root bending moment  $M_{y,1}$ . Black line is with  $\rho = 10^4$ , blue line is with  $\rho = 10^5$ , light blue line is with  $\rho = 10^6$ , and light grey line with baseline control. For  $I_T = 0.04$ ,  $v = 16$  m/s.



**Figure 5.12:** The weight  $\rho$  to the percentage reduction in the blade-root bending moment mean ( $\text{std}(M_{y,(1,2,3)})$ ). For RC-18s,  $I_T = 0.12$ ,  $v = 16$  m/s.

## 5.5 Conclusion

This chapter contribute to the development of wind turbines with individually pitch controlled blades or/and with the smart rotor control by proposing a novel lifted repetitive controller that can learn to reject periodic wind disturbances. The main advantage of using lifted repetitive control instead of typical individual pitch control is that the Coleman transformation is not needed. Instead a lifting technique is used, for which a periodic linear-time-varying model can be used with multiple inputs and multiple outputs and with both repetitive and non-repetitive disturbance components. Therefore, the lifted repetitive control method is more generic, for example to extend to rotors with more actuators and sensors, or to other rotor designs; like, two-bladed rotors. An important drawback of lifted repetitive control is also addressed, which is that a small mismatch between the controller period and the actual period of the disturbance signal can decrease the performance substantially. The formulation of the lifted repetitive control problem can be made more robust for small changes in period time by designing a sampler based on the angular position instead of time, and by including a periodic component with a random walk nature, so that the periodic disturbance can vary between trials. With these additions, the lifted repetitive controller is still considered competitive for fixed-speed and variable-speed wind turbines operating above-rated with a pitch controller, that will keep the rotational speed of the wind turbine close to the given rated rotor speed. Moreover, the chapter provides some guidelines on how to pick the free parameters in the algorithm. The load reduction capabilities of the lifted repetitive controller are investigated on the UPWIND 5MW wind turbine in the aerolastic modelling software GH BLADED. Therefore, the relative performance in terms of load reduction and control action is compared with individual pitch control with Coleman transformation and with different turbulence intensities and consequently different variations in the rotational rotor speed. Simulation results indicate that for relatively slow changing periodic wind disturbances this repetitive control method can significantly reduce the vibrations in the wind turbine structure with significant less high-frequent control action.

**Table 5.6:** IPC and RC (with variable and constant  $\mu$ ) results in standard deviation or in percentage of the baseline control results.

| $v_0 =$             | 16                                                            |       |       |       |       |       |       |       | 20    |       |       |       |       |       |       |       |
|---------------------|---------------------------------------------------------------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|
| $I_T =$             | 0                                                             | 0.04  | 0.06  | 0.08  | 0.10  | 0.12  | 0.14  | 0.16  | 0     | 0.04  | 0.06  | 0.08  | 0.10  | 0.12  | 0.14  | 0.16  |
|                     | mean (std ( $M_{y,(1,2,3)} \times 10^{-6}$ )) (Nm)            |       |       |       |       |       |       |       |       |       |       |       |       |       |       |       |
| baseline            | 0.900                                                         | 1.057 | 1.065 | 1.181 | 1.278 | 1.437 | 1.597 | 1.780 | 1.099 | 1.226 | 1.288 | 1.379 | 1.460 | 1.554 | 1.648 | 1.786 |
| IPC-1P [%]          | 74.4                                                          | 52.9  | 49.0  | 36.0  | 29.6  | 21.6  | 17.8  | 14.1  | 73.2  | 51.8  | 49.3  | 41.5  | 35.0  | 28.8  | 24.2  | 20.2  |
| IPC-1P-25deg [%]    | 74.7                                                          | 53.7  | 50.8  | 37.1  | 30.8  | 22.9  | 18.7  | 14.8  | 73.5  | 52.7  | 50.2  | 42.9  | 36.0  | 30.4  | 25.5  | 21.3  |
| IPC-2P [%]          | 86.4                                                          | 56.2  | 50.8  | 36.1  | 27.2  | 18.5  | 15.0  | 11.6  | 85.6  | 55.5  | 51.8  | 42.3  | 32.8  | 25.6  | 20.4  | 16.2  |
| IPC-2P-33deg [%]    | 86.4                                                          | 58.0  | 53.6  | 39.6  | 32.8  | 24.3  | 19.6  | 15.7  | 85.6  | 57.1  | 54.1  | 45.6  | 38.3  | 32.0  | 26.8  | 22.1  |
| $\mu = \times 10^4$ | 1                                                             | 1     | 3     | 7     | 10    | 30    | 70    | 100   | 1     | 1     | 3     | 7     | 10    | 30    | 70    | 100   |
| RC-18s [%]          | 90.7                                                          | 54.9  | 48.5  | 31.6  | 26.4  | 19.6  | 14.7  | 10.6  | 90.9  | 56.4  | 49.5  | 38.7  | 32.0  | 27.3  | 22.4  | 18.2  |
| RC-24s [%]          | 91.5                                                          | 55.0  | 48.7  | 31.5  | 26.4  | 19.6  | 14.7  | 10.3  | 91.6  | 56.5  | 49.6  | 38.7  | 32.1  | 27.4  | 22.4  | 18.2  |
| RC-30s [%]          | 93.7                                                          | 55.2  | 49.9  | 31.7  | 26.5  | 19.7  | 14.8  | 10.3  | 93.9  | 56.7  | 49.7  | 38.8  | 32.2  | 27.5  | 22.5  | 18.3  |
| $\mu = \times 10^4$ | 30                                                            | 30    | 30    | 30    | 30    | 30    | 30    | 30    | 30    | 30    | 30    | 30    | 30    | 30    | 30    | 30    |
| RC-18s [%]          | 85.2                                                          | 53.7  | 46.7  | 31.4  | 26.4  | 19.6  | 14.7  | 10.4  | 85.3  | 55.2  | 48.3  | 38.5  | 32.0  | 27.3  | 22.3  | 18.1  |
| RC-24s [%]          | 85.9                                                          | 54.0  | 46.9  | 31.4  | 26.4  | 19.6  | 14.7  | 10.3  | 86.2  | 55.5  | 48.5  | 38.6  | 32.1  | 27.4  | 22.4  | 18.1  |
| RC-30s [%]          | 86.5                                                          | 54.2  | 47.1  | 31.5  | 26.5  | 19.7  | 14.7  | 10.3  | 86.8  | 55.7  | 48.7  | 38.7  | 32.2  | 27.5  | 22.5  | 18.2  |
|                     | mean (std ( $\dot{\theta}_{y,(1,2,3)} \times 10^2$ )) (rad/s) |       |       |       |       |       |       |       |       |       |       |       |       |       |       |       |
| baseline            | 0.244                                                         | 1.208 | 1.436 | 1.544 | 1.827 | 2.136 | 2.387 | 2.698 | 0.395 | 1.856 | 2.093 | 2.294 | 2.776 | 3.094 | 3.472 | 3.938 |
| IPC-1P              | 5.162                                                         | 5.627 | 5.830 | 6.095 | 6.258 | 6.684 | 7.348 | 7.768 | 6.135 | 6.756 | 6.983 | 7.213 | 7.310 | 7.802 | 7.896 | 8.135 |
| IPC-1P-25deg        | 5.161                                                         | 5.586 | 5.759 | 5.829 | 6.031 | 6.480 | 7.044 | 7.550 | 6.135 | 6.683 | 6.823 | 6.895 | 7.159 | 7.346 | 7.449 | 7.768 |
| IPC-2P              | 5.526                                                         | 6.102 | 6.430 | 6.898 | 7.463 | 8.088 | 8.793 | 9.166 | 6.593 | 7.337 | 7.702 | 8.007 | 8.590 | 9.064 | 9.609 | 9.920 |
| IPC-2P-33deg        | 5.526                                                         | 5.989 | 6.213 | 6.414 | 6.632 | 7.150 | 7.594 | 8.099 | 6.593 | 7.206 | 7.419 | 7.495 | 7.817 | 8.043 | 8.230 | 8.569 |
| $\mu = \times 10^4$ | 1                                                             | 1     | 3     | 7     | 10    | 30    | 70    | 100   | 1     | 1     | 3     | 7     | 10    | 30    | 70    | 100   |
| RC-18s              | 5.660                                                         | 6.150 | 6.005 | 5.862 | 5.788 | 5.419 | 5.048 | 4.878 | 6.783 | 7.438 | 7.306 | 7.055 | 7.047 | 6.748 | 6.399 | 6.401 |
| RC-24s              | 5.765                                                         | 6.317 | 6.180 | 6.036 | 5.939 | 5.526 | 5.107 | 4.946 | 6.917 | 7.733 | 7.512 | 7.230 | 7.240 | 6.887 | 6.499 | 6.487 |
| RC-30s              | 5.916                                                         | 6.487 | 6.343 | 6.145 | 6.058 | 5.616 | 5.158 | 4.956 | 7.066 | 7.837 | 7.688 | 7.421 | 7.395 | 7.004 | 6.554 | 6.531 |
| $\mu = \times 10^4$ | 30                                                            | 30    | 30    | 30    | 30    | 30    | 30    | 30    | 30    | 30    | 30    | 30    | 30    | 30    | 30    | 30    |
| RC-18s              | 5.106                                                         | 5.154 | 5.173 | 5.243 | 5.305 | 5.419 | 5.528 | 5.591 | 6.148 | 6.284 | 6.315 | 6.394 | 6.533 | 6.748 | 6.967 | 7.255 |
| RC-24s              | 5.215                                                         | 5.269 | 5.279 | 5.355 | 5.414 | 5.526 | 5.651 | 5.706 | 6.283 | 6.418 | 6.450 | 6.529 | 6.669 | 6.887 | 7.113 | 7.407 |
| RC-30s              | 5.336                                                         | 5.382 | 5.395 | 5.457 | 5.516 | 5.616 | 5.696 | 5.739 | 6.419 | 6.550 | 6.580 | 6.659 | 6.787 | 7.004 | 7.155 | 7.423 |

**Table 5.7:** IPC and RC (with variable and constant  $\mu$ ) results in standard deviation or in percentage of the baseline control results.

| $v_0 =$             | 16                                            |       |       |       |       |       |       |       | 20    |       |       |       |       |       |       |       |
|---------------------|-----------------------------------------------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|
| $I_T =$             | 0                                             | 0.04  | 0.06  | 0.08  | 0.10  | 0.12  | 0.14  | 0.16  | 0     | 0.04  | 0.06  | 0.08  | 0.10  | 0.12  | 0.14  | 0.16  |
|                     | std ( $M_{\text{tilt}} \times 10^{-6}$ ) (Nm) |       |       |       |       |       |       |       |       |       |       |       |       |       |       |       |
| baseline            | 1.257                                         | 1.344 | 1.432 | 1.481 | 1.555 | 1.597 | 1.687 | 1.779 | 1.552 | 1.676 | 1.794 | 1.864 | 1.934 | 2.006 | 2.096 | 2.198 |
| IPC-1P [%]          | 71.6                                          | 60.3  | 55.3  | 49.4  | 44.1  | 38.6  | 35.6  | 30.0  | 71.5  | 58.1  | 53.6  | 48.2  | 41.7  | 38.3  | 32.1  | 28.5  |
| IPC-1P-25d [%]      | 71.9                                          | 61.7  | 56.8  | 51.9  | 46.4  | 41.6  | 38.9  | 32.4  | 71.8  | 59.4  | 54.8  | 50.6  | 43.4  | 40.7  | 34.9  | 30.9  |
| IPC-2P [%]          | 84.4                                          | 65.7  | 57.9  | 49.3  | 39.0  | 30.1  | 27.5  | 22.1  | 85.1  | 63.9  | 57.0  | 49.3  | 38.8  | 33.2  | 26.5  | 22.3  |
| IPC-2P-33d [%]      | 84.4                                          | 69.0  | 62.5  | 56.7  | 50.7  | 44.6  | 41.2  | 34.7  | 85.1  | 66.5  | 60.3  | 54.7  | 47.1  | 43.5  | 37.5  | 32.9  |
| $\mu = \times 10^4$ | 1                                             | 1     | 3     | 7     | 10    | 30    | 70    | 100   | 1     | 1     | 3     | 7     | 10    | 30    | 70    | 100   |
| RC-18s [%]          | 89.1                                          | 66.0  | 55.2  | 42.6  | 38.6  | 34.9  | 27.8  | 21.7  | 88.3  | 64.5  | 53.9  | 44.7  | 37.6  | 35.2  | 30.6  | 25.7  |
| RC-24s [%]          | 90.1                                          | 66.3  | 55.5  | 42.9  | 38.7  | 35.0  | 27.9  | 21.6  | 89.4  | 65.0  | 54.2  | 44.8  | 37.6  | 35.3  | 30.6  | 25.8  |
| RC-30s [%]          | 92.0                                          | 66.7  | 55.8  | 43.0  | 38.8  | 34.7  | 27.9  | 20.5  | 91.5  | 65.3  | 54.4  | 45.1  | 38.0  | 35.4  | 30.7  | 25.9  |
| $\mu = \times 10^4$ | 30                                            | 30    | 30    | 30    | 30    | 30    | 30    | 30    | 30    | 30    | 30    | 30    | 30    | 30    | 30    | 30    |
| RC-18s [%]          | 87.4                                          | 65.3  | 54.1  | 41.2  | 38.5  | 34.9  | 27.8  | 20.0  | 86.6  | 63.8  | 53.3  | 44.8  | 38.0  | 35.2  | 30.1  | 25.4  |
| RC-24s [%]          | 89.2                                          | 65.8  | 54.5  | 41.4  | 38.7  | 35.0  | 27.9  | 19.9  | 89.6  | 64.5  | 53.7  | 45.1  | 38.2  | 35.3  | 30.1  | 25.5  |
| RC-30s [%]          | 90.4                                          | 66.1  | 54.7  | 41.5  | 38.7  | 34.7  | 27.8  | 19.7  | 89.5  | 64.8  | 53.9  | 45.2  | 38.3  | 35.4  | 30.2  | 25.5  |
|                     | std ( $M_{\text{yaw}} \times 10^{-6}$ ) (Nm)  |       |       |       |       |       |       |       |       |       |       |       |       |       |       |       |
| baseline            | 1.257                                         | 1.344 | 1.493 | 1.484 | 1.551 | 1.606 | 1.686 | 1.739 | 1.550 | 1.658 | 1.775 | 1.866 | 1.937 | 2.001 | 2.072 | 2.188 |
| IPC-1P [%]          | 71.2                                          | 60.2  | 55.2  | 49.5  | 43.4  | 39.6  | 36.5  | 32.0  | 70.8  | 57.1  | 51.7  | 47.5  | 42.7  | 35.3  | 33.0  | 30.0  |
| IPC-1P-25d [%]      | 71.5                                          | 61.5  | 56.5  | 52.2  | 45.9  | 42.9  | 39.1  | 34.0  | 71.1  | 58.6  | 53.3  | 49.7  | 44.5  | 38.9  | 35.6  | 31.9  |
| IPC-2P [%]          | 84.3                                          | 65.8  | 58.1  | 49.8  | 38.7  | 32.5  | 29.8  | 24.9  | 85.0  | 62.9  | 55.1  | 48.5  | 39.0  | 30.0  | 25.9  | 22.4  |
| IPC-2P-33d [%]      | 84.4                                          | 68.9  | 62.5  | 57.0  | 50.4  | 47.2  | 42.8  | 37.1  | 85.1  | 65.4  | 58.5  | 54.0  | 48.0  | 41.5  | 37.3  | 33.0  |
| $\mu = \times 10^4$ | 1                                             | 1     | 3     | 7     | 10    | 30    | 70    | 100   | 1     | 1     | 3     | 7     | 10    | 30    | 70    | 100   |
| RC-18s [%]          | 88.9                                          | 65.9  | 55.5  | 43.5  | 38.6  | 35.4  | 30.0  | 23.2  | 88.3  | 63.5  | 52.4  | 44.5  | 39.4  | 34.9  | 30.6  | 26.5  |
| RC-24s [%]          | 90.8                                          | 66.4  | 55.7  | 43.5  | 38.6  | 35.6  | 30.1  | 23.2  | 90.6  | 63.8  | 52.6  | 44.8  | 39.5  | 35.0  | 30.7  | 26.6  |
| RC-30s [%]          | 92.0                                          | 66.6  | 55.8  | 43.6  | 38.7  | 35.7  | 30.4  | 23.3  | 91.9  | 64.0  | 52.8  | 44.9  | 39.5  | 35.1  | 30.7  | 26.6  |
| $\mu = \times 10^4$ | 30                                            | 30    | 30    | 30    | 30    | 30    | 30    | 30    | 30    | 30    | 30    | 30    | 30    | 30    | 30    | 30    |
| RC-18s [%]          | 87.7                                          | 65.3  | 54.1  | 43.3  | 38.2  | 35.4  | 30.0  | 23.1  | 85.8  | 63.1  | 52.0  | 44.4  | 39.1  | 34.9  | 30.6  | 26.5  |
| RC-24s [%]          | 89.6                                          | 65.8  | 54.4  | 43.4  | 38.3  | 35.6  | 30.1  | 23.1  | 88.9  | 63.6  | 52.2  | 44.5  | 39.3  | 35.0  | 30.7  | 26.5  |
| RC-30s [%]          | 90.6                                          | 66.1  | 54.5  | 43.5  | 38.4  | 35.7  | 30.1  | 23.2  | 89.7  | 63.8  | 52.3  | 44.6  | 39.3  | 35.1  | 30.7  | 26.5  |



## Conclusions and Recommendations

---

In this thesis two research goals have been addressed: one that deals with the development of data-driven modelling methods using system identification and one that deals with the development of a repetitive controller to reject periodic disturbances. This final chapter has two sections. In the first section, the main conclusions that can be drawn from the research presented in this thesis are summarized. In the second section, a number of suggestions and recommendations for further research will be made.

---

### 6.1 Conclusions

To achieve the two research goals, identification and control algorithms are developed for the rejection of periodic disturbances in modern wind turbines. These developments for the load reduction of modern wind turbines can achieve an increased lifetime of components and make the scaling to larger rotor diameters possible, and therefore improve the cost effectiveness of modern wind turbines. Also the success of future rotor designs will heavily depend on these new developments in identification and control algorithms. The following main conclusions can be drawn from the research in the main chapters.

In Chapter 2, the optimized Predictor-Based Subspace IDentification (PBSID<sub>opt</sub>) algorithm is proposed for the closed-loop wind turbine model estimation. The PBSID method is very attractive for the wind power community, as it is able to identify LTI systems with multiple inputs and multiple outputs, and does not require any controller related information. Further, a novel VARMAX-based PBSID<sub>opt</sub> method is developed that relaxes the requirement that the past window has to be large for asymptotically consistent estimates. For this purpose, a finite description of the input-output relation is formulated at the cost of a non-linear problem, but this is solved efficiently using the extended least squares recursion. From simulations, it can be concluded that the novel VARMAX-based method performs very well with a small past window under perfect conditions, however the novel method is less robust than the

original method to small violations that can occur in the noise assumptions or the model order. Despite this, an accurate identification of the TURBU model is obtained for both methods, especially around the natural frequencies of the wind turbine.

In Chapter 3, a novel Recursive Predictor-Based Subspace IDentification (RPBSID<sub>pm</sub>) algorithm is developed to identify LTI systems with multiple inputs and multiple outputs. The algorithm is suited for real-time implementation and will provide consistent estimates from data gathered in open loop or closed loop. The computational complexity is reduced by exploiting the structure in the data equations and by using array algorithms to solve the main linear problem, such that the proposed method can be implemented in real time. The ability to do so is demonstrated by the detection of flutter on an experimental “smart” rotor test section. In the experiments it is shown that a certain weight with transformation is required to minimize variations of the particular state used. Moreover, the chapter suggests to use Tikhonov regularization in order to overcome possible numerical problems in practice (and therefore give better results in some sense), such as the singularity of the regression problem. With these updates, the algorithm is successfully able to track after some delay the natural frequencies and modal damping values.

In Chapter 4, a novel lifted Repetitive Control (RC) algorithm for periodic wind disturbance rejection of linear systems with multiple inputs and multiple outputs and with both repetitive and non-repetitive disturbance components. The design of the repetitive controller is formulated as a lifted linear stochastic output-feedback problem on which the mature techniques of discrete-time linear control may be applied. The formulation of the lifted repetitive control problem can be made more robust to small changes in period time by using multiple memory loops, but also by including a periodic component with a random walk nature, so that the periodic disturbance can vary between trials. Efficient algorithms are developed for controller synthesis and for real-time implementation to reduce the computational complexity and memory usage by exploiting the structure in the lifted state-space matrices. For relatively slow changing periodic and turbulent wind disturbances created by a wind generator it was shown that this repetitive control method could reduce the variance of the load signals of the “smart” rotor test section up to 42%. The cost of additional wear and tear of the “smart” actuators are kept small, because a smooth control action is generated as the controller mainly focuses on the reduction of periodic disturbances.

In Chapter 5, the novel lifted Repetitive Control (RC) algorithm is extended for the use on modern fixed-speed wind turbines and modern variable-speed wind turbines operating above-rated equipped with individually pitch controlled blades. The repetitive control method is very generic, and can be extended to rotors with more actuators and sensors, or to other rotor designs; like, two-bladed rotors. An important drawback of repetitive control is also addressed, which is that a small mismatch between the controller period and the actual period of the disturbance signal can decrease the performance substantially. The formulation of the lifted repetitive control problem can be made more robust for small changes in period time by designing a sampler based on the angular position instead of time, and by including a periodic component with a random walk nature, such that the periodic disturbance is allowed to vary between trials. With these additions, the repetitive

controller is competitive for modern wind turbines operating above-rated with a pitch controller that will keep the rotational speed of the wind turbine close to the given rated rotor speed. The relative performance in terms of load reduction and control action is compared with typical individual pitch control and with different turbulence intensities and consequently different variations in the rotational rotor speed. The simulation results indicate that for relatively slow changing disturbances this method can significantly contribute to the reduction of vibrations in the wind turbine structure.

As an additional conclusion, it is noted that all the developed algorithms are very general, therefore the developed algorithms can also be used outside the field of wind energy. Further, the developed algorithms have been implemented by the author in software and are made available in the form of toolboxes. Also examples have been included in the toolboxes to make the tools easily applicable. For these reasons, the toolboxes have been extensively used in the application of active magnetic bearings (Balini et al., 2010, 2011), adaptive optics (Doelman et al., 2009a,b; Song, 2011), biomechanics (Kind et al., 2010; van Eesbeek et al., 2011), and fault detection (Wei et al., 2009).

## 6.2 Recommendations

Even though the results presented in this thesis have clearly demonstrated the opportunities and advantages of the identification and repetitive control algorithms for modern wind turbines, the research within this area is far from finished. In view of experience and insights gained over the course of time, the following suggestions and recommendations are made for future research:

- The proposed VARX-based predictor-based subspace identification method requires harder excitation conditions than required for the identification of innovation state-space systems, see Aling (1990). The selection of the past window strongly affects the excitation requirement, and therefore indirectly the quality of the estimated model. It has been suggested many times to use  $\ell_1$ -regularized methods for solving the main linear problem. The idea is to select a very large past window and let the optimizer introduce zeros into the solution, which are most likely the smallest Markov parameters. As the optimizer requires the selection of the  $\ell_1$ -regularization parameter which in turn replaces the selection of the past window, basically nothing is gained. However, in the future these methods can become useful when the  $\ell_1$ -regularization parameter can automatically be determined, for example with a more robust version of the method in presented Rojas and Hjarmarsson (2011).
- From the identification experiments with the VARMAX-based predictor-based subspace identification method it was concluded that identification methods can be very sensitive to small violations that can occur in the noise assumptions or the model order. These robustness issues in the form of model reduction or linearisation effects of these identification methods on the system under study are hardly analysed or researched in literature. A number of suggestions can

be made to improve the robustness of the VARMAX-based method. First, it is suggested to remove the dependence on the innovation in the following steps by using Lemma 2.3. This transformation can be used on the VARX Markov parameters from the VARMAX Markov parameters, than the state sequence can be obtained through a multiplication with the input and output sequence, and not with the innovation sequence any more. Second, it has been suggested many times to use more robust methods to estimate the VARMAX model; like instrumental variable methods. However, the selection of the correct instrumental variables in closed-loop operation is not very trivial and requires multiple iterations (and therefore even more parameters need to be tuned).

- In Chapter 5, the linear period time-varying model was obtained by linearising the first-principle non-linear model of the wind turbine. It was not possible to obtain an accurate model using (quasi) periodic LTV/LPV system identification methods, for example with the method in van Wingerden et al. (2009). The problem is that current methods require too much memory, due to the curse of dimensionality in the computational complexity of the algorithm. Further research is required to exploit the special structures in the data matrix. Specific tensor computations seems to have very similar structures, and if the structure fit, these computations could reduce the computational complexity and memory storage. Another solution is the use of special computational hardware; like distributed or parallel computing.
- Apart from the system identification itself, the data-driven modelling process requires the design of an excitation sequence and the verification/validation of the identified models for controller design. The experience (gained with the experiments) is that these tasks are very challenging and time-consuming if the system is MIMO and operates in closed-loop. Apart from the VAE, the prediction error, and the spectral analysis, it currently lacks of available verification and validation methods especially in contrast of the MIMO controller to be designed. A first step that can be helpful is the determination of the first-order variance bounds of the state-space model parameters for the proposed method. Also the development of methods for the analysis of the excitation sequence during closed-loop identification would be very useful, for example a closed-loop version of the method in Hallouzi (2008).
- With the predictor-based subspace identification of MIMO systems, it regularly occurs that identified poles and zeros from one transfer channel are parasitically coupled with the other transfer channels. A first step to a solution could be to generalize the proposed predictor-based subspace identification methods to estimate more generalized state-space models (the so-called Box-Jenkins and ARARMAX structures in (Reynders and De Roeck, 2009)), such that any parasitic couplings between the stochastic and deterministic dynamics are removed. For this purpose, the Lemma 2.1 can be used to obtain  $\Gamma \tilde{L} \tilde{U}_p$  and  $\Gamma \tilde{K} \tilde{Y}_p$ , and with a Canonical Correlation Analysis (CCA) between these sequences the state transformation (or weight matrix) can be obtained that separates the state sequence in a stochastic only, a deterministic only, and a combined state sequence.

- Further research to reliable methods for selecting a weight matrix is recommended, especially in the recursive case. The additional freedom to choose a pre-defined weight matrix, can give opportunities to improve model estimate under certain conditions. The optimization of a pre-defined weight matrix such that the low-rank approximation error is minimized by considering a weighted low-rank approximation problem is very non-linear and very hard (NP) to compute. However, other measures such as the nuclear norm can be used to determine the weight matrix and improve the model estimate, see Gebraad et al. (2011).
- Another suggestion is the development of identification algorithms that directly identify lifted state-space models with SSS-structured system matrices. These models can then be directly and efficiently used for lifted repetitive control or distributed control; for example with the optimization of floating wind farms. A first step in the development has been made by Rice and Verhaegen (2011), but the non-injective (at both global and generator level) parametrization of the SSS system matrices in this method is known to give convergence issues. Further, the linear computational complexity (in  $N$ ) of the algorithm is not proven, but only observed in simulations.
- An important restriction of the proposed repetitive controller is that it can only be applied for above-rated operation. The following suggestion is a wild idea to cope with the large changes in the period time for lifted repetitive and also iterative learning controllers. It is known that for a lifted state-space system, the individual system matrices with SSS matrix structure can be represented by a sequence of discrete-time mixed-causal LTV state-space systems, or 1D heterogeneous distributed system (Rice, 2010). It is also observed that for a synthesized lifted state-space controller, the individual system matrices have a SSS matrix structure. According to a measured azimuth angle or measured rational speed the generator subsystems can be efficiently be splitted and merged with neighbouring subsystems, concatenated and separated with additional subsystems, or replaced with other subsystems. Off course the difficulty is to show that these applied changes in the sequence of generator systems improve the performance, robustness, and stability of the global closed-loop system.
- In the current research, the fatigue in mechanical parts have been expressed in terms of signal variances of the measured accelerations or moments. Although it is expected that signal variances give a reasonable initial impression of the load reduction, the accurate description of the fatigue damage can contain some important differences in the results. The computation of the damage equivalent loads itself is straightforward. It is not only an implementation of rainflow counting and applying Miner's rule. The problem is that the fatigue damage results are very sensitive to the selected parameters during its calculation; namely, the data length, the number of bins, the width of the bins, the material characteristics in the S/N curve (Veldkamp, 2006). The tuning of these parameters such that the estimated damage equivalent loads represents reality is very challenging. To get consistent and accurate fatigue damage results, and for honest comparison between "smart" rotor controllers and concept, it is suggested that a predefined and verified benchmark for the UPWIND (or similar) 5MW turbine is designed.

- The computations with (multiple level) SSS matrices can also be very useful for solving and simulating discretized PDE's, for example to speed up the simulations of distributed wind turbines in a wind farm. Note, that this purpose was even the intended use of the first developers. With finite-difference discretization the matrices have normally a lot of zeros, and sparse matrix computations are in this case much faster. However, with the use of spectral methods or Green functions for the discretization of the PDE's the matrices are normally much smaller and dense, in this case the use SSS matrices can be competitive to sparse matrices (Chandrasekaran et al., 2003). Also, the current `MATLAB` implementation has a lot of overhead in the form of the calculation of the sizes of the generators. This overhead can considerably be reduced if the functions are implemented in C (or better in object-oriented C++, see the SSS matrix-vector implementation in the toolbox).

## On the Consistency of VARX Model Estimation in Closed Loop

---

In Chapter 2 and 3, an implicit relation between the VARX model and the Markov parameters of the state-space model in predictor form was proposed. Following the work of Chiuso and Picci (2005), some additional details with respect to the consistency of the VARX model estimation and its relation to the closed-loop conditions are presented in this appendix.

---

### A.1 Consistency when the true system is the VARX model

CONSIDER that the dynamics of the system  $\mathcal{S}_V$  to be modelled can be represented by the following Vector Auto-Regressive with eXogenous inputs (VARX) model:

$$\mathcal{S}_V \left\{ y_k = \sum_{i=0}^n \tilde{\Xi}_0^{(u_{k-i})} u_{k-i} + \sum_{i=1}^n \tilde{\Xi}_0^{(y_{k-i})} y_{k-i} + e_k, \right. \quad (\text{A.1})$$

where  $e_k$  is a zero-mean white noise sequence,  $n$  is the order, and  $\tilde{\Xi}_0 \in \mathbb{R}^{\ell \times n(r+\ell)+r}$  is the set of true VARX parameters. The following consistency result is well known.

**Proposition A.1.** *Let  $\mathcal{S}_V$  with  $n \leq p$  be the data generating system with an infinite data sequence ( $N \rightarrow \infty$ ) to Assumptions 2.1, then the estimates of the VARX model parameters in (2.17) are consistent:  $\hat{\Xi} \rightarrow \tilde{\Xi}_0$ .*

*Proof.* See Ljung (1999); Van den Hof (2007). □

Thus an unique and consistent VARX model is obtained despite the presence of feedback. It is not required that the excitation sequence  $r_k$  is measured, only the presence of a signal  $r_k$  is sufficient. In certain cases, the persistently excitation condition of the sequence  $r_k$  can be weakened if the applied feedback controller has a significantly high order (or is non-linear).

## A.2 Consistency when the true system is the innovation state-space model

That the true system is of the VARX type is not common in practical situations, and is also not very useful for controller synthesis. Proposition 2.2 suggest that Assumption 2.2 is sufficient to relate the VARX parameters with the Markov parameters of the state-space model in predictor form  $\mathcal{S}_P$  in (2.3). To simplify and clarify the relation, we first introduce the controller description, and some assumptions are made.

### A.2.1 Controller description and assumptions

It is considered that the controller has LTI dynamics and can be represented by the following state-space model:

$$\mathcal{C} \begin{cases} s_{k+1} &= A_c s_k + B_c (r_{1,k} - y_k), \\ u_k &= C_c s_k + D_c (r_{1,k} - y_k) + r_{2,k}, \end{cases} \quad (\text{A.2})$$

where  $s_k \in \mathbb{R}^c$ ,  $r_{1,k} \in \mathbb{R}^\ell$ , and  $r_{2,k} \in \mathbb{R}^r$ , are the controller state, reference, and excitation vectors. For the closed-loop identification problem, the reference and excitation sequences are unknown and are considered as noise sequences. The state-space matrices  $A_c \in \mathbb{R}^{c \times c}$ ,  $B_c \in \mathbb{R}^{c \times \ell}$ ,  $C_c \in \mathbb{R}^{r \times c}$ , and  $D_c \in \mathbb{R}^{r \times \ell}$ , are also called the controller state, input, output, and feedthrough matrix, respectively. To fulfil the Assumptions 2.1, the noise sequences  $w_k$  and  $v_k$  are considered to be uncorrelated with the reference and excitation sequences  $r_{1,k}$  and  $r_{2,k}$ , and the direct feedthrough terms are considered to obey  $DD_c = 0$  and  $D_c D = 0$ .

The reference and excitation sequences  $r_{1,k}$  and  $r_{2,k}$  are considered to be zero-mean white Gaussian noise, such that the controller can be represented in innovation form  $\mathcal{C}_I$  and in the predictor form  $\mathcal{C}_P$ . These controller forms are defined in a similar way as for the system with the predicted controller state sequence  $\hat{s}_k \in \mathbb{R}^c$ , the innovation sequence  $r_k \in \mathbb{R}^r$ , the Kalman gain  $K_c \in \mathbb{R}^{s \times \ell}$ , the matrix  $\tilde{B}_c = B_c - K_c C_c$ , and matrix  $\tilde{A}_c = A_c - K_c C_c$  asymptotically stable. By joining the innovation system in (2.2) and the innovation controller together by feedback, the system  $\mathcal{J}$  for the joint process can be obtained in the joint innovation form as:

$$\mathcal{J} \begin{cases} \hat{q}_{k+1} &= \mathbb{A} \hat{q}_k + \mathbb{K} \xi_k, \\ \begin{bmatrix} y_k \\ u_k \end{bmatrix} &= \mathbb{C} \hat{q}_k + \xi_k, \end{cases} \quad (\text{A.3})$$

where the structure of the state and system matrices are denoted by:

$$\hat{q}_k = \begin{bmatrix} \hat{x}_k \\ \hat{s}_k \end{bmatrix}, \quad \mathbb{A} = \begin{bmatrix} A + BD_c C & BC_c \\ B_c C & A_c + B_c DC_c \end{bmatrix}, \quad \mathbb{C} = \begin{bmatrix} C & DC_c \\ D_c C & C_c \end{bmatrix}.$$

and the structure of the Kalman gain matrix and innovation is denoted by:

$$\xi_k = \begin{bmatrix} I & D \\ D_c & I \end{bmatrix} \begin{bmatrix} e_k \\ r_k \end{bmatrix}, \quad \mathbb{K} = \begin{bmatrix} K & \tilde{B} \\ \tilde{B}_c & K_c \end{bmatrix}.$$



### A.2.2 The relation between the VARX and Markov parameters

The predictor form (output only) of the innovation model  $\mathcal{J}_I$  in (A.3) is denoted as:

$$\mathcal{J}_P \begin{cases} \hat{q}_{k+1|k,p+1} &= \tilde{\mathbb{A}}\hat{q}_{k|k-1,p} + \tilde{\mathbb{B}}u_k + \tilde{\mathbb{K}}y_k, \\ \hat{y}_{k|k-1,p} &= \tilde{\mathbb{C}}\hat{q}_{k|k-1,p} + Du_k + e_k, \end{cases} \quad (\text{A.4})$$

where the structure of the system matrices are denoted by:

$$\tilde{\mathbb{A}} = \begin{bmatrix} \tilde{A} & 0 \\ 0 & \tilde{A}_c \end{bmatrix}, \quad \tilde{\mathbb{B}} = \begin{bmatrix} \tilde{B} \\ K_c \end{bmatrix}, \quad \tilde{\mathbb{K}} = \begin{bmatrix} K \\ \tilde{B}_c \end{bmatrix}, \quad \tilde{\mathbb{C}} = \begin{bmatrix} C & 0 \end{bmatrix}.$$

The key observation is that the controller states  $\hat{s}_k$  of the predictor  $\mathcal{J}_P$  in (A.4) are not observable, and that after calculating the minimal realization, a predictor form similar to the system  $\mathcal{J}_P = \mathcal{S}_P$  in (2.3) remains. As the Kalman predictor  $\mathcal{S}_P$  is the optimal (causal) predictor that minimizes the prediction error, its infinite power series description can in the asymptotic case ( $N \rightarrow \infty$ ) directly be related to the estimated VARX predictor with  $p \rightarrow \infty$ . The lack of observability depends solely on the Assumptions 2.2, therefore the additional assumptions in Subsection A.2.1, namely that the controller is LTI and the reference and excitation sequence is white Gaussian noise, are not necessary but made the observation much clearer. Note, that the Assumption 2.2 is not yet required, because  $p \rightarrow \infty$  is assumed.

### A.2.3 The asymptotic behaviour for a finite VARX predictor

In the case that the past window is finite, the Assumption 2.2 is required to obtain a steady-state Kalman gain  $\mathbb{K}$ . To see this, consider the availability of infinite data ( $N \rightarrow \infty$ ), such that the deterministic part of the system  $\mathcal{J}_I$  in (A.3) is consistently estimated, and the joint noise covariance matrix is defined as:

$$\begin{aligned} \begin{bmatrix} \mathbb{Q} & \mathbb{S} \\ \mathbb{S}^T & \mathbb{R} \end{bmatrix} &= \begin{bmatrix} I & BD_c & BD_c & B \\ 0 & B_c & B_c & B_c D \\ 0 & 0 & I & D \\ 0 & D_c & D_c & I \end{bmatrix} \text{E} \begin{bmatrix} w_k w_k^T & 0 & w_k v_k^T & 0 \\ 0 & r_{1,k} r_{1,k}^T & 0 & r_{2,k} r_{1,k}^T \\ v_k w_k^T & 0 & v_k v_k^T & 0 \\ 0 & r_{1,k} r_{2,k}^T & 0 & r_{2,k} r_{2,k}^T \end{bmatrix} \\ &\times \begin{bmatrix} I & BD_c & BD_c & B \\ 0 & B_c & B_c & B_c D \\ 0 & 0 & I & D \\ 0 & D_c & D_c & I \end{bmatrix}^T. \end{aligned} \quad (\text{A.5})$$

From filtering theory, it is given that the minimum variance prediction of the output can be computed using the steady-state Kalman filter. The steady-state Kalman gain  $\mathbb{K}$  can be computed by updating the matrix  $\hat{\mathbb{K}}_j$  for  $j = 1, \dots, p$  as:

$$\hat{\mathbb{K}}_j = (\mathbb{S} + \mathbb{A}\hat{P}_{j-1}\mathbb{C}^T) (\mathbb{R} + \mathbb{C}\hat{P}_{j-1}\mathbb{C}^T)^{-1},$$

where the update equation for the state-error covariance matrix  $\hat{P}_j$  is given by the forward Riccati difference equation:

$$\hat{P}_j = \hat{\mathbb{K}}_j (\mathbb{S} + \mathbb{A}\hat{P}_{j-1}\mathbb{C}^T)^T + \mathbb{A}\hat{P}_{j-1}\mathbb{A}^T + \mathbb{Q},$$

and the initial state-error covariance matrix is expected to be  $\hat{P}_0 = E[q_{k-p} q_{k-p}^T]$ , because of the finite past window and of the neglected initial state during the estimation.

It is well known that the rate of convergence of this Kalman filter gain iteration depends on the converge rate of  $\tilde{\mathbb{A}}^j$  for a increasing past window (Anderson and Moor, 2005; Verhaegen and Verdult, 2007). As the matrix  $\tilde{\mathbb{A}}$  is decoupled, and the Kalman filter is known to be asymptotically stable, the limit for  $p \rightarrow \infty$  will bring the steady-state solution  $\tilde{\mathbb{K}}_j \rightarrow \mathbb{K}$ , and will converge at the same rate as  $\tilde{A}^j \rightarrow 0$ . Note, that the matrix  $\tilde{A} = A - KC$  represents the transition matrix of the true Kalman filter and not the estimated. With Assumption 2.2, i.e., the matrix  $\tilde{A}^j = 0$  for all  $j \geq p$ , the solution converges to the steady-state solution within at least  $p$  steps. This means that the decoupling in Subsection A.2.2 still takes place, and the Proposition 2.2 with this additional assumption is valid for a finite past window.

When we consider the estimated Kalman gain  $\hat{\mathbb{K}}$  as a perturbed version of the steady-state Kalman gain  $\mathbb{K}$ , then the state matrix of the predictor  $\mathcal{J}_p$  is denoted by:

$$\begin{aligned} \hat{\mathbb{A}} &= \mathbb{A} - \hat{\mathbb{K}}\mathbb{C} = \begin{bmatrix} A + BD_cC & BC_c \\ B_cC & A_c + B_cDC_c \end{bmatrix} - \begin{bmatrix} \hat{K} & \hat{B} - \hat{K}\hat{C} \\ \hat{B}_c - \hat{K}_c\hat{C}_c & \hat{K}_c \end{bmatrix} \begin{bmatrix} C & DC_c \\ D_cC & C_c \end{bmatrix} \\ &= \begin{bmatrix} A - \hat{K}C + (B - \hat{B})D_cC & ((B - \hat{B}) - \hat{K}(D - \hat{D}))C_c \\ ((B_c - \hat{B}_c) - \hat{K}_c(D_c - \hat{D}_c))C & A_c - \hat{K}_cC_c + (B_c - \hat{B}_c)DC_c \end{bmatrix}. \end{aligned} \quad (\text{A.6})$$

Thus in reality with a finite past window, the estimated VARX predictor has a coupling with the feedback controller as given by:

$$\tilde{\Xi}^{(u_{k-i})} = \begin{cases} D, & \text{if } i = 0, \\ \hat{\mathbb{C}}\hat{\mathbb{A}}^{i-1}\hat{\mathbb{B}} & \text{if } i > 0 \end{cases}, \quad \tilde{\Xi}^{(y_{k-i})} = \hat{\mathbb{C}}\hat{\mathbb{A}}^{i-1}\hat{\mathbb{K}}.$$

If the past window is chosen large enough, a contribution of the controller in the estimated VARX predictor is still present but small. These remaining dynamics of the controller are normally removed in second step by a sort of model reduction. By selecting only the singular values to the number of states of the system, which are probably the most dominant, the parasitic dynamics of the controller can be removed.

*Remark A.1.* In open loop there is off course no coupling with the controller states, but with a finite past window the estimated Kalman gain  $\hat{K}_p$  is still expected to be a perturbed version of the steady-state Kalman gain  $K$ . Unless the system is of the VARX type or there is no process noise.

*Remark A.2.* In a different asymptotic case with infinite data ( $N \rightarrow \infty$ ) and  $E[w_k w_k^T] = 0$ , that is no process noise, also an implicit relation can be formed between the Markov parameters and the power series description of a deadbeat predictor, which is also nilpotent if  $p \geq n$  (Houtzager et al., 2009b).

*Remark A.3.* If the VARX predictor with  $p \rightarrow \infty$  is estimated in a moving horizon, for example using a sliding window or by forgetting, the expected Kalman filter gain  $\tilde{\mathbb{K}}_j$  is also non-stationary, and varies when the predictor is updated (and downdated) with data. This is easily visible when the true covariance matrix in (A.5) is replaced with a estimated time-varying covariance matrix based on a finite data sequence.

## Spectral Analysis for the Verification of Models in Closed Loop

---

The spectral analysis method can estimate from inputs and outputs measurements the frequency response of the system to be modelled. However, it should be performed with care, because the conventional frequency response estimate will give a biased estimate under closed-loop. When we do not possess any exact frequency response of the model, this frequency response estimate can be used to verify that the identified model is a successful and accurate approximation of the real system under investigation.

---

### B.1 Transfer function description and assumptions

LET the system to be modelled be LTI and denoted by the transfer functions:

$$y_k = G(z)u_k + H(z)e_k, \quad (\text{B.1})$$

with

$$\begin{aligned} G(z) &= D + C(zI - A)^{-1}B, \\ H(z) &= I + C(zI - A)^{-1}K, \end{aligned}$$

and  $e_k \in \mathbb{R}^\ell$  is a zero-mean white noise sequence with covariance  $R$ . Let the LTI system to be modelled be operating in closed-loop and denoted by:

$$y_k = S(z)G(z)r_k + S(z)H(z)e_k, \quad (\text{B.2})$$

where the sensitivity and input-sensitivity functions are defined by:

$$\begin{aligned} S(z) &= (I^\ell + G(z)C(z))^{-1}, \\ U(z) &= (I^r + C(z)G(z))^{-1}. \end{aligned}$$

The assumptions for the frequency response estimation are similar as given in Assumption 2.1.

## B.2 Consistency with conventional spectral analysis

When only  $u_k$  and  $y_k$  are available from measurements, then the following closed-loop descriptions of the input and output signals are obtained:

$$\begin{aligned} u_k &= U(z)(r_k - C(z)H(z)e_k), \\ y_k &= S(z)(G(z)r_k + H(z)e_k). \end{aligned}$$

Now, the spectral densities  $\Phi_{yu}(\omega)$  and  $\Phi_u(\omega)$  can be defined as:

$$\begin{aligned} \Phi_u(\omega) &= U(e^{j\omega}) \left( \Phi_r(\omega) + C(e^{j\omega})H(e^{j\omega})RH(e^{j\omega})^T C(e^{j\omega})^T \right) U(e^{j\omega})^T, \\ \Phi_{yu}(\omega) &= S(e^{j\omega}) \left( G(e^{j\omega})\Phi_r(\omega) - H(e^{j\omega})RH(e^{j\omega})^T C(e^{j\omega})^T \right) U(e^{j\omega})^T. \end{aligned}$$

The conventional SPectral Analysis (SPA) estimate is defined as:

$$\hat{G}(e^{j\omega}) = \Phi_{yu}(\omega)\Phi_u(\omega)^{-1}. \quad (\text{B.3})$$

In order to see that (B.3) is not consistent, notice that:

$$\begin{aligned} \hat{G}(e^{j\omega}) &= S(e^{j\omega}) \left( G(e^{j\omega})\Phi_r(\omega) - H_0(e^{j\omega})RH(e^{j\omega})^T C(e^{j\omega})^T \right) \\ &\quad \times \left( \Phi_r(\omega) + C(e^{j\omega})H(e^{j\omega})RH(e^{j\omega})^T C(e^{j\omega})^T \right)^{-1} U(e^{j\omega})^{-1}. \end{aligned}$$

It can be easily seen that the estimated frequency response coincides with the true dynamics only in case of  $R = 0$ , thus noise-free measurements, or for  $C(e^{j\omega}) = 0$ . In all of the other cases, the estimate of the transfer function becomes biased.

## B.3 Spectral analysis for closed-loop estimation

An unbiased alternative is to use the cross-spectral density between the input/output signals and the external excitation signal  $r_k$  (Akaike, 1967). Hence, the SPA closed-loop estimate is defined as:

$$\hat{G}(e^{j\omega}) = \Phi_{yr}(\omega)\Phi_{ur}(\omega)^{-1} \quad (\text{B.4})$$

In order to see that (B.4) is consistent, notice that:

$$\begin{aligned} \hat{G}(e^{j\omega}) &= S(e^{j\omega})G(e^{j\omega})\Phi_r(\omega)\Phi_r(\omega)^{-1}U(e^{j\omega})^{-1} \\ &= G(e^{j\omega})U(e^{j\omega})U(e^{j\omega})^{-1} \\ &= G(e^{j\omega}). \end{aligned}$$

Note, that the quality depends heavily on how the additional excitation signal  $r_k$  excite the system to be modelled over the frequency range. Additional excitation on natural frequencies is not always allowed due to load restrictions. To get a smoother spectral analysis estimate, averaging in the frequency domain is normally applied, see (Van den Hof, 2007). The spectrum is smoothed locally in the region of the target frequencies, as a weighted average of the values to the right and left of a target frequency.

## Sequentially Semi-Separable Matrix Descriptions and Computations

---

Computations with Sequentially Semi-Separable (SSS) matrices enables you to perform very fast calculations for systems of linear equations with a sequentially semi-separable structure. Operations are available for most matrix computations; for example, addition, multiplication, division, and most matrix decompositions. To get you started, this Appendix will explain the current available matrix operations, where the sequentially semi-separable structure is exploited. For more information, see the examples and the descriptions of the functions in the SSS matrix toolbox (Houtzager, 2011c).

---

### C.1 Sequentially semi-separable matrix descriptions

CONSIDER the following discrete-time mixed-causal LTV state-space system:

$$\mathcal{S} \begin{cases} x_{k+1} = R_k x_k + Q_k u_k, & \text{with } x_1 = 0, \\ z_{k-1} = W_k z_k + V_k u_k, & \text{with } z_N = 0, \\ y_k = P_k x_k + U_k z_k + D_k u_k, \end{cases} \quad (\text{C.1})$$

where  $k = 1 \dots N$ , and  $x_k \in \mathbb{C}^{n_k}$ , and  $z_k \in \mathbb{C}^{m_k}$ ,  $u_k \in \mathbb{C}^{r_k}$ ,  $y_k \in \mathbb{C}^{\ell_k}$ , are the time-varying causal state, anti-causal state, input, and output vectors. The state-space matrices  $R_k \in \mathbb{C}^{n_{k+1} \times n_k}$ ,  $W_k \in \mathbb{C}^{m_{k-1} \times m_k}$ ,  $Q_k \in \mathbb{C}^{n_{k+1} \times r_k}$ ,  $V_k \in \mathbb{C}^{m_{k-1} \times r_k}$ ,  $P_k \in \mathbb{C}^{\ell_k \times n_k}$ ,  $U_k \in \mathbb{C}^{\ell_k \times m_k}$ , and  $D_k \in \mathbb{C}^{\ell_k \times r_k}$  are also called the time-varying causal state, anti-causal state, causal input, anti-causal input, causal output, and anti-causal output matrices, respectively. Note, that the input size, output size, and system orders can be time-varying and can even be zero valued.

With the window  $N$ , the following stacked vector can be defined as:

$$\bar{y}_N = [y_1^T \quad y_2^T \quad \dots \quad y_N^T]^T, \quad (\text{C.2})$$

with  $\bar{y}_N \in \mathbb{C}^{\bar{\ell} \times 1}$  where  $\bar{n}_N = \sum_{k=1}^N n_k$ . The stacked vector  $\bar{u}_N \in \mathbb{C}^{\bar{r}_N \times 1}$ ,  $\bar{r}_N$ ,  $\bar{n}_N$ , and  $\bar{m}_N$  are defined in a similar way. Using the definition of the stacked vectors, we can “lift” the system in (C.1) to the following lifted description:

$$\bar{y}_N = \bar{S} \bar{u}_N, \quad (\text{C.3})$$

where the SSS matrix  $\bar{S} \in \mathbb{R}^{\bar{\ell}_N \times \bar{r}_N}$  is defined by:

$$\bar{S} = \begin{bmatrix} D_1 & U_1 V_2 & U_1 W_2 V_3 & U_1 W_2 W_3 V_4 & U_1 W_2 \dots W_{N-1} V_N \\ P_2 Q_1 & D_2 & U_2 V_3 & U_2 W_3 V_4 & U_2 W_3 \dots W_{N-1} V_N \\ P_3 R_2 Q_1 & P_3 Q_2 & \ddots & \ddots & \vdots \\ P_4 R_3 R_2 Q_1 & P_4 R_3 Q_2 & \ddots & D_{N-1} & U_{N-1} V_N \\ P_N R_{N-1} \dots R_2 Q_1 & P_N R_{N-1} \dots R_3 Q_3 & \dots & P_N Q_{N-1} & D_N \end{bmatrix}.$$

In the remainder of this appendix, an SSS matrix is denoted by the bar symbol  $\bar{\cdot}$ . The global SSS matrix is not stored in the dense matrix form, but by its local generators  $D_k$ ,  $U_k$ ,  $V_k$ ,  $W_k$ ,  $P_k$ ,  $Q_k$ , and  $R_k$  instead, or better known by state-space matrices. The SSS generator realization is not unique, a different SSS generator realization can be obtained by the similarity transformations  $T_k \in \mathbb{C}^{n_{k+1} \times n_k}$  and  $S_k \in \mathbb{C}^{m_{k-1} \times m_k}$ :  $U_k S_k$ ,  $S_{k-1} V_k$ ,  $S_{k-1} W_k S_k$ ,  $P_k T_k$ ,  $T_{k+1} Q_k$ , and  $T_{k+1} R_k T_k$ .

The individual blocks of the SSS matrix (for  $i = 1 \dots N$  and  $j = 1 \dots N$ ) can be reconstructed to a dense matrix as:

$$\bar{S}_{ij} = \begin{cases} D_i, & \text{if } i = j \\ U_i W_{i+1} \dots W_{j-1} V_j, & \text{if } j > i \\ P_i R_{i-1} \dots R_{j+1} Q_j, & \text{if } j < i \end{cases}. \quad (\text{C.4})$$

It is not recommended to convert the SSS matrix to the dense matrix form completely, as it can require a lot of memory usage for large  $N$ . The number of parameters to describe the SSS matrix is given by:

$$P_{\text{SSS}} = \sum_{k=1}^N \ell_k r_k + \sum_{k=1}^{N-1} \ell_k m_k + \sum_{k=2}^N m_{k-1} r_k + \sum_{k=2}^{N-1} m_{k-1} m_k + \sum_{k=2}^N \ell_k n_k + \sum_{k=1}^{N-1} n_{k+1} r_k + \sum_{k=2}^{N-1} n_{k+1} n_k,$$

and the number of parameters for the dense matrix form is described by:

$$P_{\text{dense}} = \sum_{k=1}^N \ell_k \times \sum_{k=1}^N r_k.$$

Note, when the sizes of the generators are considered constant, than for an increasing  $N$  the number of parameters of the SSS matrix increases linearly instead of quadratically for the dense matrix form. Further, if a lifted LTI state-space realization is considered (or large part of the sequence is LTI), then the SSS matrix becomes “almost Toeplitz” (Rice, 2010), and the computational complexity and memory storage can be decreased again up to a factor  $N$ .

## C.2 Sequentially semi-separable matrix computations

Numerous operations with sequentially semi-separable matrices (also called quasi-separable or low-rank Hankel block matrices) can be found in the literature, however most exist on paper only. A nice overview of the most important SSS matrix computations can be found in Chandrasekaran et al. (2003, 2005); Rice (2010). Further, the book Dewilde and van der Veen (1998) describes various operations that are possible on time-varying systems in great detail, including the efficient application of orthogonal transformations. As shown in the previous subsection, the SSS matrix computations are strongly related to operations with LTV state-space systems.

In Table C.1 the currently available SSS matrix operations are summarized. Operations are available for the most commonly applied matrix computations; for example, addition, multiplication, division, and most matrix decompositions (except Hessenberg and the ones that require pivoting). In addition, conversion operations between SSS and dense/sparse matrices are implemented in the SSS matrix toolbox (Houtzager, 2011c). The operations are divided into two levels: 1, and 2.

**Level 1** This level contains the efficient SSS matrix operations based on direct methods. These operations can make use of other level 1 operations, but do **not** use level 2 operations. If the SSS matrix is stable, most of these operations (except the inverse and the determinant) are backward stable to numerical errors.

**Level 2** This level contains SSS matrix operations based on iterative methods originally used for dense matrices. The iterative methods are adjusted to preserve the SSS matrix structure. For this purpose, these operations make extensive use of level 1 operations. The computational complexity, numerical stability, and convergence properties depends heavily on successful low-order approximations of the generators during the structure preserving iterations.

To avoid numerical errors in the computations, it is required that the SSS matrix is stable. An SSS matrix is stable if  $\|W_k\|_2 \leq 1$  and  $\|R_k\|_2 \leq 1$  (Chandrasekaran et al., 2003). Commonly with the manual construction of an SSS matrix it occurs that the SSS matrix representation is only weakly stable, that is if  $\|W_2 W_3 \dots W_{N-1}\|_2 \leq 1$  and  $\|R_{N-1} \dots R_1\|_2 \leq 1$ . However, they can be made stable by using the fast order reduction algorithm for the generators, which estimates a stable low-order approximation of the weakly stable SSS matrix.

Essential for efficient computations with SSS matrices is that the off-diagonal blocks have low numerical rank. With efficient computations it is meant that operations have linear computational complexity in the number of generators  $N$ . In other words, if  $n_k \ll N$ ,  $m_k \ll N$ ,  $r_k \ll N$ , and  $\ell_k \ll N$ , than the computational complexity and memory storage of SSS matrices is more efficient than with dense matrices. Compared with sparse matrices, the SSS matrices are normally more efficient if the number of zeros in the matrix is very low. Some level 1 operations can increase the numerical rank of the off-diagonal blocks. The specific operations are denoted with the symbol  $^\dagger$  in Table C.1. After calling the specific level 1 operation in a level 2 operation, it is required to check if the generators do not grow unbounded. If this is the case an low-order approximation of the generators is necessary.

**Table C.1:** Operations with SSS matrices

| description                                        | mathematical description (example)                                                                                                                                                                                                                                                                                                                                            | algorithm (reference)                                                                                              | comments                                                                                        |
|----------------------------------------------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|--------------------------------------------------------------------------------------------------------------------|-------------------------------------------------------------------------------------------------|
| <b>SSS-dense/sparse matrix conversion</b>          |                                                                                                                                                                                                                                                                                                                                                                               |                                                                                                                    |                                                                                                 |
| SSS matrix construction                            | $\tilde{A} = \text{construct}(A, \ell_k, r_k)$                                                                                                                                                                                                                                                                                                                                | dense matrix in Chandrasekaran et al. (2003), banded matrix in Rice (2010), sym. pos. definite in Gu et al. (2010) | complexity $\mathcal{O}(N^2)$ with $p$ non-zero bands $\mathcal{O}(Np^3)$ , see also remark C.1 |
| dense matrix reconstruction                        | $A = \text{full}(A) = \tilde{A}I$                                                                                                                                                                                                                                                                                                                                             | matrix-vector multiplication is used                                                                               | for exact complexity see ref.                                                                   |
| <b>Level 1 operations</b>                          |                                                                                                                                                                                                                                                                                                                                                                               |                                                                                                                    |                                                                                                 |
| transpose                                          | $\tilde{B} = \tilde{A}^T$                                                                                                                                                                                                                                                                                                                                                     |                                                                                                                    |                                                                                                 |
| complex conjugate                                  | $\tilde{B} = \text{conj}(\tilde{A})$                                                                                                                                                                                                                                                                                                                                          |                                                                                                                    |                                                                                                 |
| splitting and merging                              | $\tilde{S}(N-1) = \begin{bmatrix} D_1 & U_1 \begin{bmatrix} V_2 & W_2 V_3 \end{bmatrix} & U_1 (W_2 W_3) V_4 \\ \begin{bmatrix} P_2 \\ P_3 R_2 \end{bmatrix} Q_1 & \begin{bmatrix} D_2 & U_2 V_3 \\ P_3 Q_2 & D_3 \end{bmatrix} & \begin{bmatrix} U_2 W_3 \\ U_3 \end{bmatrix} V_4 \\ P_4 (R_3 R_2) Q_1 & P_4 \begin{bmatrix} R_3 Q_2 & Q_3 \end{bmatrix} & D_4 \end{bmatrix}$ | in Chandrasekaran et al. (2003)                                                                                    |                                                                                                 |
| shuffled concatenation <sup>†</sup> and separation | $\tilde{C}T = \begin{bmatrix} \tilde{A} & \tilde{B} \end{bmatrix} T = \begin{bmatrix} A & B \end{bmatrix}$                                                                                                                                                                                                                                                                    | in Rice (2010)                                                                                                     | shuffled by known permutation $T$ to keep SSS structure                                         |
| addition <sup>†</sup> and subtraction <sup>†</sup> | $\tilde{C} = \tilde{A} + \tilde{B}$ , and $\tilde{C} = \tilde{A} - \tilde{B}$                                                                                                                                                                                                                                                                                                 | in Chandrasekaran et al. (2003)                                                                                    |                                                                                                 |
| low-rank update of matrix <sup>†</sup>             | $\tilde{B} = \tilde{A} + u v^T$                                                                                                                                                                                                                                                                                                                                               | in Chandrasekaran et al. (2003)                                                                                    |                                                                                                 |
| Hadamard low-rank product                          | $\tilde{B} = \tilde{A} \odot u v^T = \text{diag}(u) \tilde{A} \text{diag}(v)$                                                                                                                                                                                                                                                                                                 | in Chandrasekaran et al. (2003)                                                                                    | also called the Schur product                                                                   |
| matrix-scalar multiplication                       | $\tilde{B} = \alpha \tilde{A}$                                                                                                                                                                                                                                                                                                                                                |                                                                                                                    | for exact complexity see ref.                                                                   |
| matrix-vector multiplication                       | $b = \tilde{A}x$                                                                                                                                                                                                                                                                                                                                                              | in Chandrasekaran et al. (2003)                                                                                    | for exact complexity see ref.                                                                   |
| matrix-matrix multiplication <sup>†</sup>          | $\tilde{C} = \tilde{A} \tilde{B}$                                                                                                                                                                                                                                                                                                                                             | in Chandrasekaran et al. (2003)                                                                                    |                                                                                                 |
| Kronecker identity product <sup>†</sup>            | $\tilde{B} = \tilde{A} \otimes \tilde{I}$ , and $\tilde{B} = \tilde{I} \otimes \tilde{A}$                                                                                                                                                                                                                                                                                     | in Chandrasekaran et al. (2003)                                                                                    | also called the Tensor product                                                                  |



| description                                  | mathematical description (example)                                                                                            | algorithm (reference)                                                                                                                                                                | comments                                   |
|----------------------------------------------|-------------------------------------------------------------------------------------------------------------------------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|--------------------------------------------|
| <b>Level 1 operations (continued)</b>        |                                                                                                                               |                                                                                                                                                                                      |                                            |
| matrix-vector solver                         | $x = \arg \min \ b - \tilde{A}x\ _2^2$                                                                                        | with the QR/URV decomp. Dewilde and van der Veen (2001); Eidelman and Gohberg (2001); Sun (1997), or the fast solver in Chandrasekaran et al. (2003)                                 | for exact complexity see ref.              |
| matrix-matrix solver                         | $\tilde{X} = \arg \min \ \tilde{B} - \tilde{A}\tilde{X}\ _F^2$                                                                | with the QR/URV and backward substitution in Chandrasekaran et al. (2003), or the superfast solver in Chandrasekaran et al. (2003)                                                   |                                            |
| determinant                                  | $\det(\tilde{A})$                                                                                                             | in Rice (2010)                                                                                                                                                                       |                                            |
| (pseudo) inverse                             | $\tilde{B} = \tilde{A}^{-1}$ , and $\tilde{B} = \tilde{A}^+$                                                                  | with the QR/URV and backward substitutions with identity, or directly with Riccati recursions in Rice (2010)                                                                         |                                            |
| Cholesky decomposition                       | $\tilde{H} = \tilde{R}\tilde{R}^T$ , where $\tilde{H}$ is a positive-definite Hermitian matrix                                |                                                                                                                                                                                      |                                            |
| LU decomposition                             | $\tilde{A} = \tilde{L}\tilde{U}$                                                                                              |                                                                                                                                                                                      |                                            |
| QR/RQ/LQ/QL decomp.                          | $\tilde{A} = \tilde{Q}\tilde{R}$ with $\tilde{Q}^T\tilde{Q} = \tilde{I}$ and $R$ is an upper triangular matrix                | see matrix-vector solver                                                                                                                                                             | very similar to the URV/ULV decomposition. |
| JQR/JRQ/JLQ/JQL decomp.                      | $\tilde{A} = \tilde{Q}\tilde{R}$ with $\tilde{Q}^T \tilde{J}_1 \tilde{Q} = \tilde{J}_2$ and $R$ is an upper triangular matrix | basically, replace QR with JQR locally                                                                                                                                               | also called the hyperbolic QR              |
| matrix trace                                 | $\text{tr}(\tilde{H})$ , where $\tilde{H}$ is a symmetric matrix                                                              | in Rice (2010)                                                                                                                                                                       |                                            |
| Frobenius matrix norm                        | $\ \tilde{A}\ _F$                                                                                                             | in Rice (2010)                                                                                                                                                                       |                                            |
| order reduction of the generator realization | $\ \tilde{A} - \tilde{\tilde{A}}\ _* \leq \text{tol}$                                                                         | with the fast model reduction method in Chandrasekaran et al. (2003), or the balanced truncation in Rice (2010), or the Hankel norm approximation in van der Veen and Dewilde (1994) |                                            |

| description                                | mathematical description (example)                                                                                                                                                                                                                                                                                                                                        | algorithm (reference)                                                                                                                               | comments                  |
|--------------------------------------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-----------------------------------------------------------------------------------------------------------------------------------------------------|---------------------------|
| <b>Level 2 operations</b>                  |                                                                                                                                                                                                                                                                                                                                                                           |                                                                                                                                                     |                           |
| matrix second norm                         | $\ \tilde{A}\ _2$                                                                                                                                                                                                                                                                                                                                                         | with bisection in Rice (2010)                                                                                                                       |                           |
| matrix square-root                         | $\tilde{B} = \tilde{A}^{1/2}$                                                                                                                                                                                                                                                                                                                                             | with the sign iteration in Rice (2010), or with the Denman-Beavers iteration                                                                        |                           |
| matrix balancing                           | $\tilde{B} = \tilde{D} \tilde{A} \tilde{D}^{-1}$ where $\tilde{D}$ is a diagonal scaling matrix                                                                                                                                                                                                                                                                           | with the EBE iteration (a modified Osborne's method) in Rice (2010)                                                                                 |                           |
| matrix sign decomposition                  | $\text{sign}(\tilde{X}) = \tilde{V} \begin{bmatrix} -\tilde{I}_L & 0 \\ 0 & \tilde{I}_R \end{bmatrix} \tilde{V}^{-1}$ , where $X = P \begin{bmatrix} L & 0 \\ 0 & R \end{bmatrix} P^{-1}$ has an Jordan decomposition with $\lambda(L) \in \mathbb{C}^-$ and $\lambda(R) \in \mathbb{C}^+$                                                                                | in Rice (2010)                                                                                                                                      | used for low-rank approx. |
| Schur/eigenvalue decomp.                   | $\tilde{A} = \tilde{Q} \tilde{U} \tilde{Q}^T$ with $\tilde{Q}^T \tilde{Q} = \tilde{I}$ and $\tilde{U}$ is an upper triangular matrix with <b>unsorted</b> eigenvalues on the diagonal                                                                                                                                                                                     | with the explicit QR iteration in Chandrasekaran et al. (2007); Eidelman et al. (2005), or the LR iteration in Bevilacqua et al. (2010)             | see also Remark C.2       |
| SVD decomposition                          | $\tilde{A} = \tilde{U} \tilde{S} \tilde{V}^T$ with $\tilde{U}^T \tilde{U} = \tilde{I}$ and $\tilde{V}^T \tilde{V} = \tilde{I}$ and $\tilde{S}$ is a diagonal matrix with the singular values <b>unsorted</b>                                                                                                                                                              | see also Remark C.3                                                                                                                                 | see also Remark C.2       |
| Lyapunov equation                          | $\tilde{A} \tilde{X} + \tilde{X} \tilde{A}^T + \tilde{Q} = 0$ , and $\tilde{X} = \tilde{A} \tilde{X} \tilde{A}^T + \tilde{Q} = 0$                                                                                                                                                                                                                                         | for continuous-time the sign iteration in Rice (2010), and for discrete-time the squared Smith iteration                                            |                           |
| square-root Lyapunov eq.                   | $\tilde{A} \tilde{R} \tilde{R}^T + \tilde{R} \tilde{R}^T \tilde{A}^T + \tilde{B} \tilde{B}^T = 0$ , and $\tilde{R} \tilde{R}^T = \tilde{A} \tilde{R} \tilde{R}^T \tilde{A}^T + \tilde{B} \tilde{B}^T$                                                                                                                                                                     | for continuous-time in Larin and Aliev (1993), and for discrete-time in Fernando (1984)                                                             |                           |
| CARE and DARE (algebraic Riccati equation) | $\tilde{A}^T \tilde{X} + \tilde{X} \tilde{A} - (\tilde{X} \tilde{B} + \tilde{S}) \tilde{R}^{-1} (\tilde{B}^T \tilde{X} + \tilde{S}^T) + \tilde{Q} = 0$ , and $\tilde{X} = \tilde{A}^T \tilde{X} \tilde{A} - (\tilde{A}^T \tilde{X} \tilde{B} + \tilde{S}) (\tilde{B}^T \tilde{X} \tilde{B} + \tilde{R})^{-1} (\tilde{A}^T \tilde{X} \tilde{B} + \tilde{S})^T + \tilde{Q}$ | for continuous-time the sign iteration in Rice (2010), for discrete-time the doubling iteration, or the Newton iteration, or the Kleinman iteration |                           |

*Remark C.1* (SSS matrix construction). The SSS matrix construction operation proceed from the top-left to the bottom-right blocks of the dense matrix. The methods always try to compress the first block row, and then merge it with the second-block row. This is normally a sub-optimal choice. For example, after compressing the first block-row, and before merging with the second block-row, it might be wise to compress the second-block row first. This has the advantage of compressing a block-row with potentially much smaller rank. Of course, one can extend this idea to consider all the block-rows. It is recommended to find the best ordering before we start the algorithm. The reordering is difficult to apply in practice, as automated ordering algorithms are not (yet) suitable with the sequentially semi-separable matrix representation.

*Remark C.2* (Schur/Eigenvalue decomposition). The iterative QR algorithm is the most used method to obtain the Schur/Eigenvalue decomposition. To let the QR algorithm successfully converge to the Schur/Eigenvalue decomposition the following well known extensions are required:

- the initial conversion to Hessenberg matrix form and its invariance during the QR iterations
- clever shifts strategies (for example double Wilkinson shifts) for complex eigenvalues of real matrices
- the implicit implementation of shifts

It is not fully theoretically proven that these extensions guarantee the convergence of the so-called implicit QR algorithm under all conditions. However, in practice (for example the dense matrix implementation in LAPACK) these extensions proved to be successful in almost all cases. The problem is that these extensions can currently only be applied on SSS matrices with  $m_k = n_k = l_k = r_k = 1$  (Bella et al., 2008). For the more general case, the basic explicit QR algorithm (with single or complex shifts) can be used at the cost of slower convergence or even in some cases divergence. Further, the eigenvalues are not ordered in any way, because a column pivotation is not possible in the sequentially semi-separable matrix representation. Also the calculation of the left- and right-orthogonal matrices requires a lot of multiplications and the related generator reductions. For these reasons, it is recommended to use the Schur/Eigenvalue decomposition only for the calculation of the eigenvalues, and not for solving the low-rank matrix approximation problem, as the calculation of the orthogonal matrices is very inefficient and the unsorted eigenvalues makes the separation difficult. When the eigenvalues are obtained, the sign decomposition or the hyperbolic QR decomposition can be used to obtain the low-rank approximation of the matrix under study, see Rice (2010).

*Remark C.3* (SVD decomposition). To obtain the singular values of a matrix, the well-known relation between the singular value decomposition (SVD) and the eigenvalues of the so-called symmetric embedding is used. Consider that  $A = USV^T$  is the SVD of matrix  $A$ , then for  $i = 1 : \text{rank}(A)$  the following eigenvalue decomposition exists:

$$\begin{bmatrix} 0 & A \\ A^T & 0 \end{bmatrix} = \begin{bmatrix} u_i & \pm v_i \end{bmatrix} \pm \sigma_i \begin{bmatrix} u_i^T \\ \pm v_i^T \end{bmatrix},$$

where  $u_i = U(:, i)$ ,  $v_i = V(:, i)$ , and  $\sigma_i = S(i, i)$ . In the case of SSS matrices, it is noted that the concatenation shuffles the rows and columns.

### C.3 SSS matrix construction for efficient repetitive control synthesis

The SSS matrix operations can not only be used to speed up the LTV controller synthesis, but also for the synthesis of lifted ILC (Dijkstra, 2004; Tousain and Bosgra, 2001), lifted RC (Rice and Verhaegen, 2010b), 1D heterogeneous/homogeneous distributed systems (Rice and Verhaegen, 2009), and rationally parametric LPV systems (Rice and Verhaegen, 2010c). Compared with the other systems, the lifted RC formulation in Chapter 4 and 5 requires for an efficient SSS matrix implementation a reordering of the state basis for the lifted state-space models in (4.12), (4.15), and (5.12). See also the Remark C.1 on the construction operation for the SSS matrices.

Basically, an efficient SSS representation can be obtained if the state basis and also the noise vector basis is shuffled as follows (as a deck of cards):

$$\begin{bmatrix} \Delta x_k \\ e_k \\ w_k \\ \vdots \\ e_{k+p-1} \\ w_{k+p-1} \end{bmatrix} = T_\xi \begin{bmatrix} \Delta x_k \\ \tilde{e}_{k-p,p} \\ \tilde{w}_{k-p,p} \end{bmatrix} = T_\xi \xi'_k, \quad \begin{bmatrix} n_k \\ w_k \\ v_k \\ \vdots \\ n_{k+p-1} \\ w_{k+p-1} \\ v_{k+p-1} \end{bmatrix} = \begin{bmatrix} \tilde{n}_{k,p} \\ \tilde{w}_{k,p} \\ \tilde{v}_{k,p} \end{bmatrix} T_v = \tilde{v}_{k,p} T_v.$$

An explicit calculation of the transformations  $T_\xi$  and  $T_v$  is not required. If the shuffled concatenation in Table C.1 is used to build the lifted state-space matrices in (4.12), (4.15), and (5.12) from the individual lifted blocks, the required shuffled state basis is automatically obtained in a SSS matrix representation with  $N = p + 1$ . Afterwards the individual blocks can even be splitted, such that for one memory loop a SSS matrix representation with  $N = 2p + 1$  is obtained. If the time-domain state-space model should be augmented with the noise model or/and the feedback controller description, it is also worthwhile to first lift these state-space systems separately, and then augment the systems in the lifted domain using concatenation and splitting operations. This will result (for one memory loop) in a SSS matrix representation with  $N = 4p + 3$ .

## Bibliography

- H. Akaike. Some problems in the application of the cross-spectral method. In B. Harris, editor, *Spectral analysis of time series*, pages 81–107. Wiley, 1967.
- H. Aling. *Identification of Closed-Loop Systems: Identifiability, Recursive Algorithms and Application to a Power Plant*. PhD thesis, Delft University of Technology, 1990.
- P. B. Andersen, L. Henriksen, M. Gaunaa, C. Bak, and T. Buhl. Deformable trailing edge flaps for modern megawatt wind turbine controllers using strain-gauge sensors. *Wind Energy*, 13:193–206, 2010.
- B. D. O. Anderson and J. B. Moor. *Optimal Filtering*. Dover Publications, 2005.
- K. Åström and B. Wittenmark. *Computer-Controlled Systems: Theory and Design*. Prentice Hall, 1997.
- C. Bak, M. Gaunaa, P. B. Andersen, T. Buhl, P. Hansen, K. Clemmensen, and R. Moeller. Wind tunnel test on wind turbine airfoil with adaptive trailing edge geometry. In *45th AIAA Aerospace Sciences Meeting and Exhibit*, Reno, Nevada, USA, 2007.
- H. M. N. K. Balini. *Advanced Systems Theory Applied to Active Magnetic Bearing Systems*. PhD thesis, Delft University of Technology, 2011.
- H. M. N. K. Balini, I. Houtzager, J. Witte, and C. W. Scherer. Subspace identification and robust control of an AMB system. In *American Control Conference*, Baltimore, MD, USA, 2010.
- H. M. N. K. Balini, I. Houtzager, J. Witte, and C. W. Scherer. Subspace and frequency domain methods for closed loop identification of AMB-spindle. *Submitted to IEEE Transactions on Control System Technology*, 2011.
- T. K. Barlas and G. A. M. van Kuik. Review of state of the art in smart rotor control research for wind turbines. *Progress in Aerospace Sciences*, 46(1):1–27, 2010.
- M. Basseville, A. Benveniste, M. Goursat, and L. Mevel. In-flight vibration monitoring of aeronautical structures: subspace-based online automated identification versus detection. *IEEE Control Systems Magazine, Special Issue on Applications of System Identification*, 27(5):27–42, 2007.
- T. Bella, Y. Eidelman, I. Gohberg, and V. Olshevsky. Computations with quasiseparable polynomials and matrices. *Theoretical Computer Science*, 409:158–179, 2008.
- A. Betz. *Introduction to the Theory of Flow Machines*. Oxford: Pergamon Press, 1966.
- R. Bevilacqua, E. Bozzo, and G. M. Del Corso. QD-type methods for quasiseparable matrices. Technical report, Università di Pisa, Dipartimento di Informatica, 2010.
- F. D. Bianchi, H. De Battista, and R. J. Mantz. *Wind Turbine Control Systems; Principles, Modelling and Gain Scheduling Design*. Springer-Verlag London, 2007.
- S. R. Bieniawski. *Distributed Optimization and Flight Control Using Collectives*. PhD thesis, Stanford University, 2005.

- G. Bir. Multi-blade coordinate transformation and its application to wind turbine analysis. In *ASME Wind Energy Symposium*, Reno, NV, USA, 2008.
- R. S. Blom. *Model-based Process Monitoring and Control of Micro-milling using Active Magnetic Bearings*. PhD thesis, Delft University of Technology, 2011.
- P. M. M. Bongers. *Modeling and identification of flexible wind turbines and a factorizational approach to robust control*. PhD thesis, Delft University of Technology, 1994.
- P. M. M. Bongers and G. E. van Baars. Experimental validation of a flexible wind turbine model. In *30th IEEE Conference on Decision and Control*, Brighton, England, 1991.
- H. Bonke, D. Vonk, J. Breeuwer, C. Paul, and W. Dobber. *Cornelis Corneliszoon van Uitgeest: uitvinder aan de basis van de Gouden Eeuw, 2e druk*. Walburg Pers, 2004.
- E. A. Bossanyi. The design of closed loop controllers for wind turbines. *Wind Energy*, 3:149–163, 2000.
- E. A. Bossanyi. Wind turbine control for load reduction. *Wind Energy*, 6:229–244, 2003a.
- E. A. Bossanyi. Individual blade pitch control for load reduction. *Wind Energy*, 6: 119–128, 2003b.
- E. A. Bossanyi. Further load reductions with individual pitch control. *Wind Energy*, 8: 481–485, 2005.
- E. A. Bossanyi. Controller for 5mw reference turbine. Technical Report 11593/BR/04, Garrad Hassan and Partners Limited, 2009.
- E. A. Bossanyi and A. Wright. Field testing of individual pitch control on the NREL CART-2 wind turbine. In *European Wind Energy Conference*, Marseille, France, 2009.
- E. A. Bossanyi, G. Ramtharan, and B. Savini. The importance of control in wind turbine design and loading. In *17th Mediterranean Conference on Control & Automation*, Thessaloniki, Greece, 2009.
- E. A. Bossanyi, A. Wright, and P. Fleming. Progress with field testing of individual pitch control. In *3rd Conference on the Science of Making Torque from Wind*, Heraklion, Crete, Greece, 2010.
- T. Buhl, M. Gaunaa, and C. Bak. Potential load reduction using airfoils with variable trailing edge geometry. *Journal of Solar Energy*, 127(4), 2005.
- T. Burton, D. Sharpe, N. Jenkins, and E. Bossanyi. *Handbook of Wind Energy, 2nd Edition*. John Wiley & Sons, 2011.
- L. Cavagna, S. Ricci, and A. Scotti. Active aeroelastic control over a four control surface wing model. *Aerospace Science and Technology*, 13(7):374–382, 2009.
- S. Chandrasekaran and A. H. Sayed. Stabilizing the generalized schur algorithm. *SIAM Journal on Matrix Analysis & Applications*, 17(4):950–983, 1996.

- S. Chandrasekaran, P. Dewilde, M. Gu, T. Pals, A.-J. van der Veen, and D. White. Fast stable solvers for sequentially semi-separable linear systems of equations. Technical report, Lawrence Livermore National Laboratory, 2003.
- S. Chandrasekaran, P. Dewilde, M. Gu, T. Pals, X. Sun, A.-J. van der Veen, and D. White. Some fast algorithms for sequentially semiseparable representations. *SIAM Journal on Matrix Analysis and Applications*, 27(2):341–364, 2005.
- S. Chandrasekaran, M. Gu, J. Xia, and J. Zhu. A fast QR algorithm for companion matrices. *Operator Theory: Advances and Applications*, 179:111–143, 2007.
- I. Chin, S. J. Qin, K. S. Lee, and M. Cho. A two-stage iterative learning control technique combined with real-time feedback for independent disturbance rejection. *Automatica*, 18:1913–1922, 2004.
- A. Chiuso. Asymptotic variance of closed-loop subspace identification methods. *IEEE Transactions on Automatic Control*, 51(8):1299–1341, 2006.
- A. Chiuso. The role of vector auto regressive modeling in predictor based subspace identification. *Automatica*, 43(6):1034–1048, 2007.
- A. Chiuso. On the asymptotic properties of closed loop CCA-type subspace algorithms: Equivalence results and role of the future horizon. *IEEE Transactions on Control Systems Technology*, 55(3):643–649, 2010.
- A. Chiuso and G. Picci. Consistency analysis of some closed-loop subspace identification methods. *Automatica*, 41(3):377–391, 2005.
- A. Chiuso, R. Muradore, and E. Marchetti. Dynamic calibration of adaptive optics systems: A system identification approach. *IEEE Transactions on Control Systems Technology*, 8(3):705–713, 2010.
- M. Cho, Y. Lee, S. Joo, and K. S. Lee. Semi-empirical model-based multivariable iterative learning control of an rtp system. *IEEE Transactions on Semiconductor Manufacturing*, 18:430–439, 2005.
- R. Costa-Castelló, J. M. Olm, and G. A. Ramos. Design and analysis for digital repetitive control systems with time-varying reference/disturbance period. *International Journal of Control*, 84(7):1–14, 2011.
- K. De Cock, G. Mercère, and B. De Moor. Recursive subspace identification for in-flight modal analysis of airplanes. In *International Conference on Noise and Vibration Engineering*, Leuven, Belgium, 2006.
- J. De Cuyper. *Linear Feedback Control for Durability Test Rigs in the Automotive Industry*. PhD thesis, Catholic University of Leuven, 2006.
- A. De Gaspari, S. Ricci, L. Riccobene, and A. Scotti. Active aeroelastic control over a multisurface wing: Modeling and wind-tunnel testing. *AIAA Journal*, 47(9):1995–2010, 2009.

- C. De Marqui, Jr., D. C. Rebolho, E. M. Belo, and F. D. Marques. Identification of flutter parameters for a wing model. *Journal of the Brazilian Society of Mechanical Sciences and Engineering*, 28(3):339–346, 2006.
- D. de Roover, O. H. Bosgra, and M. Steinbuch. Internal-model-based design of repetitive and iterative learning controllers for linear multivariable systems. *International Journal of Control*, 73(10):914–929, 2000.
- P. Dewilde and A.-J. van der Veen. *Time-Varying Systems and Computations*. Springer Verlag, 1998.
- P. Dewilde and A.-J. van der Veen. Inner-outer factorization and the inversion of locally finite systems of equations. *Linear Algebra and its Applications*, 313:53–100, 2001.
- B. G. Dijkstra. *Iterative Learning Control: with applications to a wafer stage*. PhD thesis, Delft University of Technology, 2004.
- P. S. R. Diniz. *Adaptive Filtering: Algorithms and Practical Implementations*. Springer, 2008.
- N. Doelman, R. Fraanje, I. Houtzager, and M. Verhaegen. Adaptive and real-time optimal control for adaptive optics systems. In *European Control Conference*, Budapest, Hungary, 2009a.
- N. Doelman, R. Fraanje, I. Houtzager, and M. Verhaegen. Adaptive and real-time optimal control for adaptive optics systems. *European Journal of Control*, 15(3–4): 480–488, 2009b.
- J. Dong. *Data Driven Fault Tolerant Control: A Subspace Approach*. PhD thesis, Delft University of Technology, 2009.
- J. Dong and M. Verhaegen. Cautious  $H_\infty$  optimal control using uncertain Markov parameters identified in closed loop. *Systems & Control Letters* 58, 58(5):378–388, 2009.
- J. Dong, M. Verhaegen, and E. Holweg. Closed-loop subspace predictive control for fault tolerant MPC design. In *17th IFAC World Congress*, Seoul, Korea, 2008.
- J. Doyle. Guaranteed margins for LQG regulators. *IEEE Transactions on Automatic Control*, 23(4):756–757, 1978.
- dSPACE. CONTROL DESK. Website. URL <http://www.dspaceinc.com>.
- DWIO. Wind Power Wiki. Website, 2009. URL <http://wiki.windpower.org>.
- Y. Eidelman and I. Gohberg. A modification of the Dewilde-van der Veen method for inversion of finite structured matrices. *Linear Algebra and its Applications*, 343–344: 1145–1201, 2001.
- Y. Eidelman, I. Gohberg, and V. Olshevsky. The QR iteration method for Hermitian quasiseparable matrices of an arbitrary order. *Linear Algebra and its Applications*, 404:305–324, 2005.



- W. P. Engels, S. Kanev, and T. G. Engelen. Distributed blade control. In *3rd Conference on the Science of Making Torque from Wind*, Heraklion, Crete, Greece, 2010.
- FACE International. THUNDER. Website. URL <http://www.faceinternational.com>.
- P. Fairley. Technical Review: Less May Be More for Wind Turbines. Website, 2009. URL <http://www.technologyreview.com/energy/23109/>.
- S. Faulstich, B. Hahn, and P. J. Tavner. Wind turbine downtime and its importance for offshore deployment. *Wind Energy*, 14:327–337, 2011.
- H. Fernando, K. V. nad Nicholson. Square-root solution for the discrete-time Lyapunov equation. 131(3):107–108, 1984.
- I. Font, Balaguer, M. Rossetti, S. Kanev, and D. Tchemiak. System identification methods on Alstom ECO 100 wind turbine. In *3rd Conference on the Science of Making Torque from Wind*, Heraklion, Crete, Greece, 2010.
- Garrad Hassan. Bladed. Website. URL <http://www.gl-garradhassan.com>.
- S. Gay. Dynamically regularized fast recursive least squares. In *4th International Workshop on Acoustic Echo and Noise Control*, Roros, Norway, 1995.
- P. M. O. Gebraad, J. W. van Wingerden, G. van der Veen, and M. Verhaegen. LPV subspace identification using a novel nuclear norm regularization method. In *American Control Conference*, San Francisco, CA, USA, 2011.
- M. Gevers. Identification for control: From the early achievements to the revival of experiment design. *European Journal of Control*, 11:1–18, 2005.
- M. Gevers and G. Li. Parametrizations in control, estimation and filtering problems: Accuracy aspects. In *Communication and Control Engineering Series*. Springer Verlag, 1993.
- Bladed: Theory Manual*. GH Garrad Hassan and Partners Limited, 20 edition, 2010a. 282/BR/009.
- Bladed: User Manual*. GH Garrad Hassan and Partners Limited, 20 edition, 2010b. 282/BR/010.
- M. Gibson and G. Mercère. Subspace-based optimal IV method for closed-loop system identification. In *14th IFAC Symposium on System Identification*, Newcastle, Australia, 2006.
- I. Goethals, L. Mevel, A. Benveniste, and B. De Moor. Recursive output-only subspace identification for in-flight flutter monitoring. In *22nd International Modal Analysis Conference*, Dearborn, MI, USA, 2004.
- Green Rhino Energy. Wind Energy: Wind Characteristics. Website, 2011. URL [http://www.greenrhinoenergy.com/renewable/wind/wind\\_characteristics.php](http://www.greenrhinoenergy.com/renewable/wind/wind_characteristics.php).
- M. Gu, X. S. Li, and P. S. Vassilevski. Direction-preserving and Schur-monotonic semiseparable approximations of symmetric positive definite matrices. *SIAM Journal on Matrix Analysis and Applications*, 31(5):2650–2664, 2010.

- S. Gunnarsson. On covariance modification and regularization in recursive least squares identification. In *10th IFAC Symposium on System Identification*, Copenhagen, Denmark, 1994.
- S. Gunnarsson. Combining tracking and regularization in recursive least squares identification. In *35th IEEE Decision and Control Conference*, Kobe, Japan, 1996.
- M. Gupta and J. Lee. Period-robust repetitive model predictive control. *Journal of Process Control*, 16:545–555, 2006.
- T. Gustafsson. Instrumental variable subspace tracking using projection approximation. *IEEE Transactions on Signal Processing*, 46(3):669–681, 1998.
- R. Hallouzi. *Multiple-Model Based Diagnosis for Adaptive Fault-Tolerant Control*. PhD thesis, Delft University of Technology, 2008.
- K. Hammerum, P. Brath, and N. K. Poulsen. A fatigue approach to wind turbine control. *Journal of Physics: The Science of Making Torque from Wind*, 75(012081), 2007.
- P. C. Hansen. Regularization tools: A MATLAB package for analysis and solution of discrete ill-posed problems. Technical report, Technical University of Denmark, Lyngby, Denmark, 1992. Manual.
- K. J. G. Hinnen. *Data-Driven Optimal Control of Adaptive Optics*. PhD thesis, Delft University of Technology, 2006.
- H. Hjalmarsson. From experiment design to closed-loop control. *Automatica*, 41(3):393–438, 2005.
- W. E. Holley, S. M. Rock, and K. Chaney. Control of variable speed wind turbines below-rated wind speed. In *3rd ASME/JSME Joint Fluids Engineering Conference*, San Francisco, CA, 1999.
- I. Houtzager. Predictor-Based Subspace Identification Toolbox version 0.4. Website, 2010. URL <http://www.dcsc.tudelft.nl/~datadriven/pbsid/>.
- I. Houtzager. External controller design tool for GH BLADED and SIMULINK. Website, 2011a. URL <http://www.dcsc.tudelft.nl/~datadriven/discon/>.
- I. Houtzager. Lifted Repetitive Control Toolbox version 0.2. Website, 2011b. URL <http://www.dcsc.tudelft.nl/~datadriven/lrc/>.
- I. Houtzager. Sequentially Semi-Separable Matrix Toolbox version 0.8. Website, 2011c. URL <http://www.dcsc.tudelft.nl/~datadriven/sss/>.
- I. Houtzager, J. W. van Wingerden, and M. Verhaegen. Fast-array recursive closed-loop subspace model identification. In *15th IFAC Symposium on System Identification*, Saint-Malo, France, 2009a.
- I. Houtzager, J. W. van Wingerden, and M. Verhaegen. VARMAX-based closed-loop subspace model identification. In *48th IEEE Conference on Decision and Control*, Shanghai, China, 2009b.

- I. Houtzager, B. A. Kulscár, J. W. van Wingerden, and M. Verhaegen. System identification for the control of wind turbine systems. In *3rd Conference on the Science of Making Torque from Wind*, Heraklion, Crete, Greece, 2010.
- I. Houtzager, J. W. van Wingerden, and M. Verhaegen. Recursive predictor-based subspace identification with application to the real-time closed-loop tracking of flutter. *Accepted for IEEE Transactions on Control System Technology*, 2011a.
- I. Houtzager, J. W. van Wingerden, and M. Verhaegen. Rejection of periodic wind disturbances on a smart rotor test section using lifted repetitive control. *Provisionally accepted for IEEE Transactions on Control System Technology*, 2011b.
- I. Houtzager, J. W. van Wingerden, and M. Verhaegen. Wind turbine load reduction by rejecting the periodic load disturbances. *Provisionally accepted for Wind Energy*, 2011c.
- I. Houtzager, J. W. van Wingerden, and M. Verhaegen. Rejection of periodic wind disturbances on an experimental “smart” rotor section using lifted repetitive control. In *IEEE Multi-Conference on Systems and Control*, Denver, CO, USA, 2011d.
- Analysis techniques for system reliability: procedure for failure mode and effects analysis (FMEA)*. IEC International Electrotechnical Commission, 1 edition, 1985. IEC 60812-1.
- Wind turbines part 1: design requirements*. IEC International Electrotechnical Commission, 3 edition, 2004. IEC 61400-1.
- M. Iribas. Closed loop identification for wind turbines. Technical Report IN-08.00289, CENER, 2006.
- M. Jansson. Subspace identification and ARX modeling. In *13th IFAC Symposium on System Identification*, Rotterdam, The Netherlands, 2003.
- K. E. Johnson, L. Y. Pao, M. J. Balas, and L. J. Fingersh. Control of variable-speed wind turbines: standard and adaptive techniques for maximizing energy capture. *IEEE Control Systems Magazine*, 26(3):70–81, 2006.
- J. Jonkman, S. Butterfield, W. Musial, and S. George. Definition of a 5-MW reference wind turbine for offshore system development. Technical Report TP 500-38060, National Renewable Energy Laboratory, 2008.
- J.-N. Juang and R. S. Pappa. An eigensystem realization algorithm for model parameter identification and model reduction. *Journal of Guidance, Control, and Dynamics*, 8(5):620–627, 1985.
- S. Kanev. System identification for wind turbine applications: We@Sea project 2007-014, phase 1. Technical report, Energy research Centre of the Netherlands, 2008.
- S. Kanev. Exploring the limits in individual pitch control. In *European Wind Energy Conference*, Marseille, France, 2009.
- S. Kanev, F. Savenije, I. Houtzager, and B. A. Kulscár. We@Sea project 2007-014 system identification. Technical report, We@Sea, 2009.

- T. Katayama. *Subspace Methods for System Identification*. Springer, 2005.
- T. Kind, I. Houtzager, T. Faes, and M. Hofman. Evaluation of model-independent deconvolution techniques to estimate blood perfusion. In *IEEE Engineering in Medicine and Biology Society*, Buenos Aires, Argentina, 2010.
- M. J. Kühn. *Dynamics and Design Optimisation of Offshore Wind Energy Conversion Systems*. PhD thesis, Delft University of Technology, 2001.
- B. Kulcsár and M. Verhaegen. Robust cautious data driven control with guaranteed mean square stability. In *16th Mediterranean Conference on Control and Automation*, Atlanta, GA, USA, 2011.
- A. L. Lackner and G. A. M. van Kuik. A comparison of smart rotor control approaches using trailing edge flaps and individual pitch control. *Wind Energy*, 13:117–134, 2010a.
- A. L. Lackner and G. A. M. van Kuik. The performance of wind turbines smart rotor control approaches during extreme loads. *Journal of Solar Energy*, 132(1), 2010b.
- W. E. Larimore. Canonical variate analysis in identification, filtering and adaptive control. In *29th IEEE Conference on Decision and Control*, Honolulu, HI, USA, 1990.
- V. B. Larin and F. A. Aliev. Construction of square root factor for solution of the Lyapunov matrix equation. 20(2):109–112, 1993.
- J. L. Larsen, H. A. Madsen, K. Thomsen, and F. Rasmussen. Reduction of teeter angle excursions for a two-bladed downwind rotor using cyclic pitch control. In *Proceedings of the European Wind Energy Conference*, Milan, Italy, 2007.
- J. H. Lee, S. Natarajan, and K. S. Lee. A model-based predictive control approach to repetitive control of continuous processes with periodic operations. *Journal of Process Control*, 11:195–207, 2001.
- C. Lindenberg. PHATAS user's manual: Program for horizontal axis wind turbine analysis and simulation. Technical Report ECN-I-05-005, Energy research Centre of the Netherlands, 2005.
- L. Ljung. On positive real transfer functions and the convergence of some recursive schemes. *IEEE Transactions on Automatic Control*, 22(4):539–551, 1977.
- L. Ljung. *System Identification: Theory for the user, 2nd Edition*. Prentice Hall, 1999.
- L. Ljung and T. McKelvey. Subspace identification from closed loop data. *Signal Processing*, 52:209–215, 1996.
- L. Ljung and T. Söderström. *Theory and Practice of Recursive Identification*. The MIT Press, 1983.
- R. W. Longman. Iterative learning control and repetitive control for engineering practise. *International Journal of Control*, 73:930–950, 2000.

- M. Lovera. Recursive subspace identification based on projector tracking. In *13th IFAC Symposium on System Identification*, Rotterdam, The Netherlands, 2003.
- M. Lovera, T. Gustafsson, and M. Verhaegen. Recursive subspace identification of linear and non-linear Wiener state-space models. *Automatica*, 36(11):1639–1650, 2000.
- C. Lyzell, M. Enqvist, and L. Ljung. Handling certain structure information in subspace identification. In *Proceedings of the 15th IFAC Symposium on System Identification*, Saint-Malo, France, 2009.
- J. F. Manwell, J. G. McGowan, and A. L. Rogers. *Wind Energy Explained: Theory, Design and Application, 2nd Edition*. John Wiley & Sons, 2010.
- Mathworks. MATLAB. Website, a. URL <http://www.mathworks.com/products/matlab>.
- Mathworks. SIMULINK. Website, b. URL <http://www.mathworks.com/products/simulink>.
- M. A. McEver, E. V. Ardelean, D. G. Cole, and R. L. Clark. Active control and closed-loop identification of flutter instability in typical section airfoil. *Journal of Guidance, Control, and Dynamics*, 30(3):733–740, 2007.
- G. Mercère and L. Bako. Parametrization and identification of multivariable state-space systems: a canonical approach. *Accepted to Automatica*, 2011.
- G. Mercère and M. Lovera. Convergence analysis of instrumental variable recursive subspace identification algorithms. *Automatica*, 43:1377–1386, 2007.
- G. Mercère, S. Lecœuche, and M. Lovera. Recursive subspace identification based on instrumental variable unconstrained quadratic optimization. *International Journal of Adaptive Control and Signal Processing*, 18(9–10):771–797, 2004.
- G. Mercère, L. Bako, and S. Lecœuche. Propagator-based methods for recursive subspace model identification. *Signal Processing*, 88(3):468–491, 2008.
- L. Mevel, M. Basseville, and A. Benveniste. Fast in-flight detection of flutter onset: A statistical approach. *Journal of Guidance, Control, and Dynamics*, 28(3):431–438, 2003.
- Microsoft. VISUAL STUDIO C++. Website. URL <http://www.microsoft.com/express>.
- D-P. Molenaar. *Cost effective design and operation of variable speed wind turbines*. PhD thesis, Delft University of Technology, 2003.
- V. Mukhopadhyay. Historical perspective on analysis and control of aeroelastic responses. *Journal of Guidance, Control, and Dynamics*, 26(5):673–684, 2003.
- R. R. Nino and P. J. Eecen. Turbulence and wind shear: A literature study and measurements. Technical Report DOWEC-F1W1-PE-01-044/00 P, Energy research Centre of the Netherlands, 2001.
- K. Ogata. *Discrete-Time Control Systems*. Prentice Hall, 1994.

- H. Oku and H. Kimura. Recursive 4SID algorithms using gradient type subspace tracking. *Automatica*, 38(6):1035–1043, 2002.
- H. Oku, Y. Ogura, and T. Fujii. Closed-loop subspace model identification based on QR-factorization. In *17th International Symposium on Mathematical Theory of Networks and Systems*, Kyoto, Japan, 2006.
- J. M. Olm, G. A. Ramos, and R. Costa-Castelló. Stability analysis of digital repetitive control systems under time-varying sampling period. *IET Control Theory and Applications*, 5(1):29–37, 2011.
- K. Z. Østergaard. *Robust, Gain-Scheduled Control of Wind Turbines*. PhD thesis, Aalborg University, 2008.
- L. Y. Pao and K. E. Johnson. Control of wind turbines: Approaches, challenges, and recent developments. *IEEE Control Systems Magazine*, 31(2):44–62, 2011.
- M. Phan, J.-N. Juang, and R. W. Longman. Identification of linear multivariable systems by identification of observers with assigned real eigenvalues. *Journal of the Astronautical Sciences*, 40(2):261–279, 1992.
- M. Phan, L. G. Horta, J.-N. Juang, and R. W. Longman. Improvement of observer/Kalman filter identification (OKID) by residual whitening. *Journal of Vibrations and Acoustics*, 117:232–238, 1995.
- G. Pipeleers, B. Demeulenaere, J. De Schutter, and J. Swevers. Robust high-order repetitive control: optimal performance trade-offs. *Automatica*, 44(10):2628–2634, 2008.
- G. Pipeleers, B. Demeulenaere, F. Al-Bender, J. De Schutter, and J. Swevers. Optimal performance trade-offs in repetitive control: experimental validation on an active air bearing setup. *IEEE Transactions on Control Systems Technology*, 17(4):970–979, 2009.
- S. J. Qin and L. Ljung. Closed-loop subspace identification with innovation estimation. In *13th IFAC Symposium on System Identification*, Rotterdam, The Netherlands, 2003.
- F. Rasmussen and A. Kretz. Dynamics and potentials for the two-bladed teetering rotor concept. In *Proceedings of the American Wind Energy Association Conference*, Seattle, WA, USA, 1992.
- E. Reynders and G. De Roeck. Subspace identification of ARARMAX, Box-Jenkins, and generalized model structures. In *Proceedings of the 15th IFAC Symposium on System Identification*, Saint-Malo, France, 2009.
- J. Ribrant and L. M. Bertling. Survey of failures in wind power systems with focus on swedish wind power plants during 1997-2005. *IEEE Transactions on Energy Conversion*, 22(1):167–173, 2007.
- J. K. Rice. *Efficient Algorithms for Distributed Control: A Structured Matrix Approach*. PhD thesis, Delft University of Technology, 2010.

- J. K. Rice and M. Verhaegen. Distributed control: A sequentially semi-separable approach for spatially heterogeneous linear systems. *IEEE Transaction on Automatic Control*, 54(6):1270–1283, 2009.
- J. K. Rice and M. Verhaegen. Robust and distributed control of a smart blade. *Wind Energy*, 13:103–116, 2010a.
- J. K. Rice and M. Verhaegen. A structured matrix approach to efficient calculation of LQG repetitive learning controllers in the lifted setting. *International Journal of Control*, 83(6):1265–1276, 2010b.
- J. K. Rice and M. Verhaegen. LPV analysis and control using fast iterative solutions to rationally parametric Lyapunov and Riccati equations. In *American Control Conference*, Baltimore, MD, USA, 2010c.
- J. K. Rice and M. Verhaegen. Efficient system identification of heterogeneous distributed systems via a structure exploiting extended Kalman filter. *Appears in IEEE Transactions on Automatic Control*, 99(1):1–1, 2011.
- J. A. Rivera, Jr., B. E. Dansberry, R. M. Bennett, M. H. Durham, and W. A. Silva. NACA 0012 benchmark model experimental flutter results with unsteady pressure distributions. In *33rd AIAA/ASME/ASCE/AHS/ASC Structures, Structural Dynamics and Materials Conference*, Dallas, TX, USA, 1992.
- C. R. Rojas and H. Hjarmarsson. Sparse estimation based on a validation criteria. In *50th IEEE Conference on Decision and Control and European Control Conference*, Orlando, FL, USA, 2011.
- M. Rossetti, M. Guadaya, O. Goula, and J. Prats. Wind turbine dynamics validation and controller optimisation. In *Proceedings of the European Wind Energy Conference*, Brussels, Belgium, 2008.
- A. H. Sayed. *Adaptive Filters*. John Wiley & Sons, 2008.
- K. Selvam, S. Kanev, J. W. van Wingerden, T. G. van Engelen, and M. Verhaegen. Feedback-feedforward individual pitch control for wind turbine load reduction. *International Journal of Robust and Nonlinear Control*, 19(1):72–91, 2009.
- Smart Material. Macro fiber composite. Website. URL <http://www.smart-material.com>.
- H. Song. *Model-based Control in Adaptive Optical Systems*. PhD thesis, Delft University of Technology, 2011.
- C. J. Spruce. Power control of active stall wind turbines. In *Proceedings of the European Wind Energy Conference*, London, UK, 2004.
- C. J. Spruce and J. K. Turner. Control algorithms for eliminating tower vibration events on active stall wind turbines. In *Proceedings of the European Wind Energy Conference*, Warsaw, Poland, 2010.
- M. Steinbuch. *Dynamic Modelling and Robust Control of a Wind Energy Conversion System*. PhD thesis, Delft University of Technology, 1989.

- M. Steinbuch. Repetitive control for systems with uncertain period time. *Automatica*, 38:2103–2109, 2002.
- M. Steinbuch, S. Weiland, and T. Singh. Design of noise and period-time robust high-order repetitive control, with application to optical storage. *Automatica*, 43:2086–2095, 2007.
- K. A. Stol. *Dynamics modelling and periodic control of horizontal-axis wind turbines*. PhD thesis, University of Colorado, 2001.
- F. K. Straub, V. R. Anand, T. S. Birchette, and B. H. Lau. Smart rotor development and wind tunnel test. In *35th European Rotorcraft Forum*, Hamburg, Germany, 2009.
- X. Sun. Superfast stable algorithm for solving the linear least squares problems with structured sequentially semi-separable matrices. Master's thesis, University of California, 1997.
- K. Tammi, J. Hatonen, and S. Daley. Novel adaptive repetitive algorithm for active vibration control of a variable-speed rotor. *Journal of Mechanical Science and Technology*, 21(6):855–859, 2007.
- P. Tavner, Y. Qiu, A. Korogiannos, and Y. Feng. The correlation between wind turbine turbulence and pitch failure. In *Proceedings of the European Wind Energy Conference*, Brussels, Belgium, 2011.
- T. Theodorsen. General theory of aerodynamic instability and the mechanism of flutter. Technical Report 496, National Advisory Committee for Aeronautics, 1935.
- S. C. Thomsen, H. Niemann, and N. K. Poulsen. Stochastic wind turbine control in multiblade coordinates. In *American Control Conference*, Baltimore, MD, USA, 2010.
- A. N. Tikhonov and V. Y. Arsenin. *Solutions of Ill-Posed Problems*. V. H. Winston & Sons, 1977.
- R. Top. Industrial Revelations: The European Story, Season 3, Episode 1: Reaping The Whirlwind. Discovery Channel, 2005.
- R. Tóth. *Modeling and Identification of Linear Parameter-Varying Systems: an Orthonormal Basis Function approach*. PhD thesis, Delft University of Technology, 2008.
- E. Tousain, R. van der Meché and O. Bosgra. Design strategy for iterative learning control based on optimal control. In *40th IEEE Conference on Decision and Control*, Orlando, FL, USA, 2001.
- M. Tsakiris, C. Lopes, and V. Nascimento. An array recursive least-squares algorithm with generic nonfading regularization matrix. *IEEE Signal Processing Letters*, 17(12): 1001–1004, 2010.
- T.-C. Tsao, Y.-X. Qian, and M. Nemani. Repetitive control for asymptotic tracking of periodic signals with an unknown period. *ASME Journal of Dynamic Systems, Measurement, and Control*, 122:364–369, 2000.



- G. E. van Baars and P. M. M. Bongers. Closed loop system identification of an industrial wind turbine system: Experiment design and first validation results. In *33rd IEEE Conference on Decision and Control*, Lake Buena Vista, Florida, USA, 1994.
- G. E. van Baars, H. Mosterd, and P. M. M. Bongers. Extension to standard system identification of detailed dynamics of a flexible wind turbine system. In *32nd Conference on Decision and Control*, San Antonio, TX, USA, 1993.
- Van Dale, editor. *Middelgroot woordenboek Nederlands, 1ste editie*. 2009.
- P. M. J. Van den Hof. System identification. Technical report, Delft University of Technology, Delft, The Netherlands, 2007. Lecture Notes.
- P. M. J. Van den Hof and R. J. P. Schrama. Identification and control – closed loop issues. *Automatica*, 31(12):1751–1770, 1995.
- E. L. van der Hooft, P. Schaak, and T. G. van Engelen. Wind turbine control algorithms. Technical Report DOWEC-F1W1-EH-03-094/0, Energy research Centre of the Netherlands, 2003.
- I. Van der Hoven. Power spectrum of horizontal wind speed in the frequency range from 0.0007 to 900 cycles per hour. *Journal of Meteorology*, 15:160–164, 1957.
- A.-J. van der Veen and P. Dewilde. On low-complexity approximation of matrices. *Linear Algebra and its Applications*, 205–206:1145–1201, 1994.
- G. J. van der Veen, J. W. van Wingerden, and M. Verhaegen. Data-driven modelling of wind turbines. In *American Control Conference*, San Francisco, CA, USA, 2011.
- S. van Eesbeek, E. de Vlugt, I. Houtzager, and M. Verhaegen. Time variant subspace identification of joint impedance. In *3rd Dutch Bio-Medical Engineering Conference*, Egmond aan Zee, The Netherlands, 2011.
- T. G. van Engelen. Design model and load reduction assessment for multi-rotational mode individual pitch control (higher harmonics control). In *European Wind Energy Conference*, Athens, Greece, 2006.
- T. G. van Engelen. Control design based on aero-hydro-servo-elastic linear models from TURBU (ECN). In *Proceedings of the European Wind Energy Conference*, Milan, Italy, 2007.
- T. G. van Engelen, H. Markou, T. Buhl, and B. Marrant. Morphological study of aeroelastic control concepts for wind turbines. Technical Report ENK5-CT-2002-00627, Energy research Centre of the Netherlands, 2007.
- G. A. M. van Kuik. Are wind turbines growing too fast? In *Proceedings of the European Wind Energy Conference*, Copenhagen, Denmark, 2001.
- P. Van Overschee and B. De Moor. *Subspace Identification for Linear Systems; Theory, Implementation, Applications*. Kluwer Academic Publishers, 1996.
- J. W. van Wingerden. *Control of Wind Turbines with “Smart” Rotors: Proof of Concept & LPV Subspace Identification*. PhD thesis, Delft University of Technology, 2008.

- J. W. van Wingerden and M. Verhaegen. Subspace identification of Bilinear and LPV systems for open and closed loop data. *Automatica*, 45(2):372–281, 2009.
- J. W. van Wingerden, T. Hulskamp, T. Barlas, B. Marrant, G. A. M. van Kuik, D-P. Molenaar, and M. Verhaegen. On the proof of concept of a “smart” wind turbine rotor blade for load alleviation. *Wind Energy*, 11(3):265–280, 2008.
- J. W. van Wingerden, I. Houtzager, F. Felici, and M. Verhaegen. Closed-loop identification of the time-varying dynamics of variable-speed wind turbines. *International Journal of Robust and Nonlinear Control*, 19(1):4–21, 2009.
- J. W. van Wingerden, T. Hulskamp, T. Barlas, I. Houtzager, H. Bersee, G. A. M. van Kuik, and M. Verhaegen. Smart dynamic rotor control: Part 3, advanced controller design. In *3rd Conference on the Science of Making Torque from Wind*, Heraklion, Crete, Greece, 2010.
- J. W. van Wingerden, T. Hulskamp, T. Barlas, I. Houtzager, H. Bersee, G. A. M. van Kuik, and M. Verhaegen. Two-degree-of-freedom active vibration control of a prototyped “smart” rotor. *IEEE Transactions on Control System Technology*, 19(2):284–296, 2011.
- H. F. Veldkamp. *Chances in Wind Energy: A Probabilistic Approach to Wind Turbine Fatigue Design*. PhD thesis, Delft University of Technology, 2006.
- M. Verhaegen. Round-off error propagation in four generally-applicable, recursive, least-squares estimation schemes. *Automatica*, 25(3):437–444, 1989.
- M. Verhaegen. Application of a subspace model identification technique to identify LTI systems operating in closed-loop. *Automatica*, 29(4):1027–1040, 1993.
- M. Verhaegen and V. Verdult. *Filtering and System Identification*. Cambridge University Press, 2007.
- Y. Wang, F. Gao, and F. J. Doyle III. Survey on iterative learning control, repetitive control, and run-to-run control. *Journal of Process Control*, 19(10):1589–1600, 2009.
- X. Wei and M. Verhaegen. Fault detection of large scale wind turbine systems: A mixed  $H_\infty/H_-$  index observer approach. In *16th Mediterranean Conference on Control and Automation*, Ajaccio, Corsica, France, 2008.
- X. Wei and M. Verhaegen. Fault detection of large scale off-shore wind turbine systems. Technical Report TR09-044, Delft University of Technology, 2009.
- X. Wei, M. Verhaegen, and I. Houtzager. Fault diagnosis for LTI systems based on subspace identification technique. In *IFAC Symposium on Fault Detection, Supervision and Safety of Technical Processes*, Barcelona, Spain, 2009.
- X. Wei, M. Verhaegen, and T. van Engelen. Sensor fault detection and isolation for wind turbines based on subspace identification and Kalman filter techniques. *International Journal of Adaptive Control and Signal Processing*, 24(8):687–707, 2010.

- H.-P. Wen and R. Longman. Repetitive control methods when the disturbance period is not an integer multiple of the sample time. In *AAS/AIAA Spaceflight Mechanics Meeting (Advances in the Astronautical Sciences)*, volume 108, pages 1301–1321, 2001.
- J. A. S. Witteveen, S. Sarkar, and H. Bijl. Modeling physical uncertainties in dynamic stall induced fluid-structure interaction of turbine blades using arbitrary polynomial chaos. *Computer & Structures*, 85(11–14):866–878, 2007.
- B. R. Woodley. *Model Free Subspace Based  $H_\infty$  Control*. PhD thesis, Stanford University, 2001.
- P. Wu, C. Yang, and Z. Song. Recursive subspace model identification based on vector autoregressive modelling. In *17th IFAC World Congress*, Seoul, Korea, 2008.
- B. Yang. Projection approximation subspace tracking. *IEEE Transactions on Signal Processing*, 43(1):95–107, 1995.



# List of Abbreviations

|       |                                                      |
|-------|------------------------------------------------------|
| 1P    | Once-per-revolution                                  |
| 2D    | Two-dimensional                                      |
| 2P    | Twice-per-revolution                                 |
| 3D    | Three-dimensional                                    |
| 3P    | Thrice-per-revolution                                |
| ARMAX | Auto-Regressive Moving Average with eXogenous inputs |
| ARX   | Auto-Regressive with eXogenous inputs                |
| CCA   | Canonical Correlation Analysis                       |
| CVA   | Canonical Variate Analysis                           |
| DARE  | Discrete-time Algebraic Riccati Equation             |
| ELS   | Extended Least Squares                               |
| ERA   | Eigensystem Realization Algorithm                    |
| FB    | Feedback                                             |
| FF    | Feedforward                                          |
| FMEA  | Failure Modes and Effects Analysis                   |
| GBN   | Generalized Binary Noise                             |
| GCV   | Generalized Cross Validation                         |
| HAWT  | Horizontal-Axis Wind Turbine                         |
| ILC   | Iterative Learning Control                           |
| IPC   | Individual Pitch Control                             |
| LMS   | Least Mean Squares                                   |
| LPV   | Linear Parameter-Varying                             |
| LQG   | Linear Quadratic Gaussian                            |
| LTI   | Linear Time-Invariant                                |
| LTV   | Linear Time-Varying                                  |
| MCS   | Monte-Carlo Simulations                              |
| MEM   | Micro-Electro-Mechanical                             |
| MFC   | Macro Fiber Composite                                |
| MIMO  | Multiple-Input and Multiple-Output                   |
| MPC   | Model Predictive Control                             |
| MW    | Megawatt                                             |
| OKID  | Observer/Kalman-filter Identification                |
| ORT   | Orthogonal Decomposition                             |
| PAST  | Projection Approximation Subspace Tracking           |
| PBSID | Predictor-Based Subspace Identification              |

---

|                      |                                                             |
|----------------------|-------------------------------------------------------------|
| PBSID <sub>opt</sub> | “optimized” Predictor-Based Subspace Identification         |
| PEM                  | Prediction Error Method                                     |
| PM                   | Propagator Method                                           |
| PRBS                 | Pseudo Random Binary Signal                                 |
| PSD                  | Power Spectral Density                                      |
| RLS                  | Recursive Least Squares                                     |
| RPM                  | Revolutions-per-minute                                      |
| RSID                 | Recursive Subspace Identification                           |
| SID                  | Subspace Identification                                     |
| SISO                 | Single-Input and Single-Output                              |
| SNR                  | Signal-to-Noise Ratio                                       |
| SPA                  | Spectral Analysis                                           |
| SRC                  | “Smart” Rotor Control                                       |
| SSS                  | Sequentially Semi-Separable                                 |
| STD                  | Standard Deviation                                          |
| SVD                  | Singular Value Decomposition                                |
| RC                   | Repetitive Control                                          |
| RPBSID               | Recursive Predictor-Based Subspace Identification           |
| VAF                  | Variance-Accounted-For                                      |
| VAR                  | Variance                                                    |
| VARMAX               | Vector Auto-Regressive Moving Average with eXogenous inputs |
| VARX                 | Vector Auto-Regressive with eXogenous inputs                |

# Summary

## **Towards Data-Driven Control for Modern Wind Turbines**

Ivo Houtzager

Further developments in data-driven control techniques for the load reduction of modern wind turbines can achieve an increased lifetime of components and make the scaling to larger rotor diameters possible, and therefore improve the cost effectiveness of modern wind turbines. Also the success of future rotor designs will heavily depend for their operation on new developments in active control technologies. This thesis proposes a novel control algorithm for the rejection of periodic disturbances. The load disturbances seen by an individual rotor blade are to a large extent periodic and repetitive; for instance, tower shadow, wind shear, yawed error, and gravity are depending on the azimuth angle and rotational speed, and will change slowly over time. The contribution of these periodic disturbances will become even larger in the future, due to the increasing length and mass of the rotor blades and the possible downwind location of a two-bladed rotor. By compensating only the smooth periodic disturbances, the control action would not create too much fatigue damage on the actuators.

In pursuit of this goal, the novel control synthesis requires a mathematical model description of the system to be controlled. The proposed way to obtain these models is from measured data using identification techniques. For this purpose, the “optimized” predictor-based subspace identification methods are proposed and their benefits for the estimation of modern wind turbine models are shown. These identification techniques do not require any controller related information, consequently the identified model becomes consistent no matter the wind turbine operates with or without controller in the loop. A modern wind turbine is not asymptotically stable in open-loop, and only closed-loop experiments are supported in reality. Further, the identification algorithms can efficiently deal with multi-input and multi-output measurements. This makes these methods very attractive for the new rotor concepts that require an increasing number of actuators and sensors to operate. Despite the short length of the measurement data, due to the limited amount of time in an operation region, and the low signal-to-noise ratio, due to the high turbulence intensity, the results show that an accurate batchwise identification of a realistic modern wind turbine model is possible, especially around the natural frequencies of the wind turbine.

The operational conditions of modern wind turbines will change slowly over time, due to changes in the mean wind speed. A novel recursive predictor-based subspace

identification algorithm is developed to track the slowly time-varying dynamics. The algorithm is suited for real-time implementation and is also able to operate in open loop or closed loop. The real-time implementation and the ability to work with systems operating in closed loop make this approach suitable for online estimation of unstable dynamics. Flutter detection is one of the applications of this approach and is demonstrated this on an experimental “smart” rotor test section. The “smart” rotor is a new rotor concept where the blades are equipped with a number of control devices that locally change the lift profile on the blade, combined with appropriate sensors. The algorithm is successfully able to track after some delay the natural frequencies and modal damping values.

The experimental “smart” rotor test section is also used to demonstrate that the novel repetitive controller can reject the periodic disturbances in a turbulent wind field. The repetitive control method is implemented in real-time for periodic disturbance rejection on linear systems with multi-inputs and multi-outputs and with both repetitive and non-repetitive disturbance components. For real-time implementation, the computational complexity can be reduced by exploiting the structure in the lifted state-space matrices. For relatively slow changing periodic and turbulent wind disturbances created by a wind generator it is shown that this repetitive control method could reduce the variance of the load signals of the “smart” rotor test section up to 42%. The cost of additional wear and tear of the “smart” actuators are kept small, because a smooth control action is generated as the controller mainly focuses on the reduction of periodic disturbances.

The load reduction capabilities of the repetitive controller using individually pitch controlled blades is verified on the UPWIND 5MW wind turbine in the aerolastic modelling software GH BLaDED. The repetitive control method is very generic, and can be extended to rotors with more actuators and sensors, or to other rotor designs; like, two-bladed rotors. An important drawback of repetitive control is also addressed, which is that a small mismatch between the controller period and the actual period of the disturbance signal can decrease the performance substantially. The formulation of the repetitive control problem is made more robust for small changes in period time. With these additions, the repetitive controller is still considered competitive for modern fixed-speed and modern variable-speed wind turbines operating above-rated with a pitch controller that will keep the rotational speed of the wind turbine close to the given rated rotor speed. The relative performance in terms of load reduction and control action is compared with typical individual pitch control and with different turbulence intensities and consequently different variations in the rotational rotor speed. The simulation results indicate that for relatively slow changing disturbances this method can significantly contribute to the reduction of vibrations in the wind turbine structure.



# Samenvatting

## Naar op data gebaseerde besturing voor moderne windturbines

Ivo Houtzager

Nieuwe op data gebaseerde regeltechnische ontwikkelingen voor de belastingvermindering van moderne windturbines kunnen de levensduur van componenten verlengen en maken het schalen naar grotere rotordiameters mogelijk, en verbeteren dus de kosteneffectiviteit van moderne windturbines. Ook het succes van toekomstige rotorontwerpen zullen sterk voor hun operatie afhankelijk zijn van deze nieuwe ontwikkelingen in de actieve regeltechnologie. Dit proefschrift stelt een nieuwe regelalgoritme voor om de periodieke verstoringen te compenseren. De belastingverstoringen op een individueel rotorblad zijn voor een groot deel periodiek en repetitief, bijvoorbeeld de torenschaduw, de windschering, de fout in gieren, en de zwaartekracht zijn allen afhankelijk van de azimut en de rotatie snelheid, en zal langzaam veranderen in de tijd. De bijdrage van deze periodieke verstoringen zullen nog groter worden in de toekomst, door het gevolg van de toenemende lengte en massa van de rotorbladen en door de mogelijke plaatsing van een tweebladige rotor met de wind mee. Door de periodieke verstoringen te compenseren, zullen de regelacties niet leiden tot veel vermoeidheidsschade aan de actuatoren.

Ter verwezenlijking van dit doel heeft de nieuwe regelaar ontwerpmethodes een wiskundig model benodigd van het geregelde systeem. De voorgestelde manier om deze modellen te verkrijgen is via de gemeten data met behulp van identificatietechnieken. Hiervoor zijn geoptimaliseerde op voorspeller gebaseerde subruimte identificatie methoden voorgesteld en zijn hun voordelen voor het schatten van moderne windturbine modellen aangetoond. Deze identificatietechnieken vereisen geen regelaar gerelateerde informatie, dus ook het geïdentificeerde model wordt consistent geschat ongeacht of de windturbine werkt met of zonder regelaar in de lus. Een moderne windturbine is niet asymptotisch stabiel in de open lus, dus alleen experimenten in de gesloten lus kunnen in de werkelijkheid worden ondersteund. Verder kunnen de identificatie algoritmen efficiënt omgaan met meerdere ingangsen uitgangsmetingen. Dit maakt deze methoden zeer aantrekkelijk voor de nieuwe rotor concepten die een toenemend aantal actuatoren en sensoren vereisen voor hun bediening. Ondanks de korte lengte van de meetgegevens, als gevolg van de beperkte hoeveelheid tijd in een operatiegebied, en de lage signaal-ruis verhouding, als gevolg van de hoge turbulentie-intensiteit, laten de resultaten zien dat een nauwkeurige batchgewijs identificatie van een realistische moderne windturbinemodel mogelijk is, vooral rond de natuurlijke frequenties van de windturbine.

De operationele omstandigheden van moderne windturbines veranderen langzaam in de tijd, als gevolg van veranderingen in de gemiddelde windsnelheid. Een nieuwe recursieve op voorspeller gebaseerde subruimte identificatie algoritme is ontwikkeld om het langzame tijdvariërend dynamische gedrag te volgen. Het algoritme is geschikt voor real-time implementatie en is ook in staat om te werken in zowel open lus of gesloten lus. De real-time implementatie en het vermogen om te werken met systemen die actief in gesloten lus zijn, maakt deze benadering geschikt voor het online schatten van instabiel dynamisch gedrag. Fladderdetectie is één van de toepassingen van deze benadering en dit wordt gedemonstreerd op een experimentele “slimme” rotor testsectie. De “slimme” rotor is een nieuw rotorconcept waarbij de bladen zijn voorzien van een aantal apparaten die lokaal op het blad het liftprofiel kan veranderen, gecombineerd met passende sensoren. Het algoritme heeft met succes de natuurlijke frequenties en modale dampingwaarden na enige vertraging gevolgd.

De experimentele “slimme” rotor testsectie is ook gebruikt om aan te tonen dat de nieuwe repetitieve regelaar ook de periodieke verstoringen kan compenseren in een turbulent windveld. De repetitieve regelaar ontwerpmethodologie is toegepast in real-time voor de compensatie van periodieke verstoring in lineaire systemen met meerdere ingangen en uitgangen, en met zowel repetitieve en niet-repetitieve verstoringcomponenten. Voor de real-time implementatie kan de berekeningscomplexiteit worden verminderd door gebruik te maken van de structuur in de matrices van de vergrote toestandsruimte. Voor relatief langzaam variërende periodieke en turbulente windverstoringen, gemaakt door een windgenerator, is aangetoond dat deze repetitieve regelaar de variatie van de belastingsignalen van de “slimme” rotor testsectie tot 42% kan verminderen. De kosten van extra slijtage van de “slimme” actuatoren zijn beperkt, omdat de regelactie gegenereerd door de regelaar zich vooral richt op de vermindering van de periodieke verstoringen.

De belastingverminderingcapaciteiten van de repetitieve regelaar met gebruik van individuele pitch geregelde bladen is geverifieerd op de UPWIND 5 MW windturbine in de aerolastische modelleringsoftware GH BLADED. De repetitieve regelmethodologie is zeer generiek en kan worden uitgebreid naar rotoren met meer actuatoren en sensoren of naar andere rotorontwerpen, zoals de tweebladige rotoren. Een belangrijk nadeel van repetitieve regelaars is ook behandeld, namelijk dat een kleine verschil tussen de periode van de regelaar en de werkelijke duur van het verstoringssignaal de prestaties aanzienlijk kunnen verminderen. De formulering van het repetitieve regelprobleem is meer robuust gemaakt voor kleine veranderingen in de periodetijd. Met deze toevoegingen kan de repetitieve regelaar nog steeds als concurrerend worden beschouwd voor de moderne vaste-snelheid en de moderne variabele-snelheid windturbines opererend in het bovenwerkgebied met een pitchregelaar, die de rotatiesnelheid van de windturbine dichtbij de opgegeven nominale rotorsnelheid zal houden. De relatieve prestaties in termen van belastingvermindering en regelactie is vergeleken met een typische pitchregelaar en met verschillende turbulentie intensiteiten, en daaruitvolgend ook verschillende variaties in de rotatiesnelheid. De simulatie resultaten geven aan dat voor de relatief langzame veranderlijke storingen deze methode aanzienlijk kan bijdragen tot de vermindering van trillingen in de windturbineconstructie.

# Curriculum Vitae

**I**VO HOUTZAGER was born on the 6th of January, 1982 in Krimpen aan den IJssel, The Netherlands.

In 1999 he graduated from high school (HAVO) at the Krimpernerwaard College in Krimpen aan den IJssel, The Netherlands.

In 2003 he received the B.Eng. degree in mechanical engineering with speciality in design and production from the Rotterdam University of Applied Sciences in Rotterdam, The Netherlands. His bachelor project entitled “Vierrollen rondbuigmachine: Een nieuw concept voor het coilen” was carried out at Fontijne Grotness B.V. in Vlaardingen, The Netherlands.

In 2007 he received the M.Sc. degree in mechanical engineering with speciality in systems and control from the Delft University of Technology in Delft, The Netherlands. His master project entitled “Adaptive learning control for wind turbines equipped with smart rotor: Applied to a NM80 wind turbine model” was carried out at Delft Center of Systems and Control (DCSC) in Delft, The Netherlands.

In 2007 he started pursuing a Ph.D. degree in systems and control engineering on the project “Data-Driven Control for Modern Wind Turbines” at Delft Center of Systems and Control (DCSC) and Delft University Wind Energy Research Institute (DUWIND) under the supervision of Michel Verhaegen and Jan-Willem van Wingerden. During his Ph.D. project, he participated in several international conferences, workshops, and graduate courses, for which he received a Dutch Institute of Systems and Control (DISC) certificate.

As of August 2011 he works as embedded software developer of the motion control facilities at ASML in Veldhoven, The Netherlands.

