

Control of Wind Turbines with
'Smart' Rotors: Proof of
Concept & LPV Subspace
Identification

Cover: Siemens wind turbines located in Middelgrunden, Denmark
Maarten Timmer, info@maartentimmer.com

CONTROL OF WIND TURBINES WITH 'SMART' ROTORS: PROOF OF CONCEPT & LPV SUBSPACE IDENTIFICATION

PROEFSCHRIFT

ter verkrijging van de graad van doctor
aan de Technische Universiteit Delft,
op gezag van de Rector Magnificus Prof. dr. ir. J.T. Fokkema,
voorzitter van het College voor Promoties,
in het openbaar te verdedigen op
maandag 17 november 2008 om 12:30 uur

door

Jan-Willem VAN WINGERDEN

werktuigkundig ingenieur
geboren te Ridderkerk

Dit proefschrift is goedgekeurd door de promotor:

Prof. dr. ir. M. Verhaegen

Samenstelling promotiecommissie:

Rector Magnificus,	voorzitter
Prof. dr. ir. M. Verhaegen,	Technische Universiteit Delft, promotor
Prof. dr. ir. G. A. M. van Kuik,	Technische Universiteit Delft
Prof. ir. O. H. Bosgra,	Technische Universiteit Delft
Prof. dr. drs. ir. H. Bijl,	Technische Universiteit Delft
Prof. dr. G. Balas,	University of Minnesota
Dr. ir. D-P. Molenaar,	Siemens Wind Power
Dr. M. Lovera,	Politecnico Di Milano
Prof. dr. R. Babuška,	Technische Universiteit Delft, reservelid



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



The work presented in this thesis has been supported by the Dutch Technology Foundation (STW) under project number: dwe6461.

ISBN: 978-90-9023583-7

Copyright © 2008 by J.W. van Wingerden.

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 in The Netherlands

To my family

Acknowledgments

Dear friends, I should thank a large number of people, but to avoid putting up a really long story, I will not. I only would like to thank Michel in particular for the freedom he gave me to do innovative research and the support I received throughout the last four years. The other people who read this part of this thesis to find their own name (which is how I read it): you know what you have done for me!

THANK YOU VERY MUCH.

Delft, September 2008
Jan-Willem van Wingerden

Contents

Acknowledgements	vii
1 Introduction	1
1.1 Introduction to the wind energy field	2
1.2 Control of wind turbines	4
1.2.1 State-of-the-art control of industrial wind turbines	4
1.2.2 Introduction to the ‘smart’ rotor concept	7
1.2.3 Modern model based control in the wind energy community	9
1.3 Introduction to LPV system identification	10
1.3.1 Modeling	10
1.3.2 LPV model structure	13
1.3.3 LPV system identification	14
1.4 Goals of this thesis	16
1.5 Contributions of this thesis	18
1.6 Organization of this thesis	19
I Experimental part	21
2 On the proof of concept of a ‘smart’ wind turbine rotor blade for load alleviation	23
2.1 Introduction	23
2.2 Experimental Setup	24
2.2.1 Wind tunnel	24
2.2.2 Blade Design	26
2.2.3 Actuators & sensors	27
2.2.4 Real-Time environment	28
2.3 Modeling of a ‘smart’ rotor blade and model validation	29

2.3.1	Aerodynamic model	29
2.3.2	Mechanical model	30
2.3.3	Controller Design	32
2.3.4	Experimental modeling	34
2.4	Wind tunnel measurements	36
2.4.1	Case 1: Feedforward control with a periodic disturbance	36
2.4.2	Case 2: Feedback control with a periodic disturbance	38
2.4.3	Case 3: Feedback control with a step disturbance	39
2.4.4	Case 4: Feedback control with a representative noise signal	39
2.5	Conclusions	41
 II Fundamental part		43
3	Subspace identification of MIMO LPV systems using a periodic scheduling sequence	45
3.1	Introduction	45
3.2	Problem formulation and assumptions	47
3.2.1	Problem formulation	47
3.2.2	Assumptions and notation	48
3.3	Periodic predictor-based subspace identification	50
3.3.1	Predictors	50
3.3.2	Extended observability times controllability matrix	52
3.4	A common state basis	54
3.4.1	Factorization of the extended observability matrix	54
3.4.2	Determination of the transformation matrices	56
3.5	Kernel method	58
3.6	Determination of the system matrices	59
3.7	Dedicated scheduling sequences	61
3.7.1	Almost Periodic scheduling	61
3.7.2	Piecewise constant scheduling	61
3.7.3	Parameter-varying K matrix	62
3.8	Simulation Examples	63
3.8.1	Example 1: Flapping dynamics of a wind turbine	63
3.8.2	Example 2: Third order MIMO system	64
3.9	Case study: Rotational dynamics of a wind turbine	67
3.9.1	First principles model of a Horizontal Axis Wind Turbine (HAWT)	67

3.9.2	Simulation of the closed-loop wind turbine model	71
3.9.3	Closed-loop LPV subspace identification results	72
3.10	Conclusion	72
4	Subspace identification of MIMO LPV systems using an arbitrary scheduling sequence	77
4.1	Introduction	77
4.2	Problem formulation and assumptions	79
4.2.1	Problem formulation	79
4.2.2	Assumptions and notation	80
4.3	Factorization of the LPV controllability matrix	81
4.4	LPV predictor-based subspace identification	84
4.4.1	Predictors	84
4.4.2	Extended observability times controllability matrix	85
4.4.3	‘Curse of dimensionality’	88
4.4.4	Summary of the LPV algorithms	88
4.5	Kernel method	89
4.5.1	Kernel Method	89
4.5.2	Computation of the kernel matrices	92
4.5.3	Regularization	93
4.5.4	Summary of the algorithm	93
4.6	Different model structures	94
4.6.1	Parameter-varying output equation and constant K	95
4.6.2	Parameter-varying output equation and parameter-varying K	95
4.6.3	Bilinear systems	96
4.7	Dedicated scheduling sequences	97
4.7.1	Periodic scheduling	97
4.7.2	Kernel selection	98
4.8	Simulation Examples	99
4.8.1	Example 1: Open-loop LPV identification	99
4.8.2	Example 2: Closed-loop LPV identification	101
4.8.3	Example 3: Bilinear identification	103
4.9	Case study: a ‘smart’ airfoil	105
4.9.1	Analytical LPV modeling	105
4.9.2	Discretization	108
4.9.3	Simulation results	110
4.10	Conclusion	113

5	Conclusions & Recommendations	115
5.1	'Smart' rotor concept	115
5.1.1	Conclusions	115
5.1.2	Recommendations	116
5.2	LPV system identification	117
5.2.1	Conclusions	117
5.2.2	Recommendations	118
A	Controller details of the experimental 'smart' rotor	119
B	LTI predictor-based subspace identification	121
C	Kernels for different model structures	125
	Bibliography	127
	List of Abbreviations	141
	Summary	143
	Samenvatting	145
	Curriculum Vitae	147

Introduction

Active control is becoming more and more important for the wind energy community. If we compare the ‘old’ stall regulated turbines with today’s individual pitch controlled turbines we see that the loads can be considerably reduced, leading to lighter or larger turbines. Still, there is a challenge to come up with novel designs and control concepts for the new generation of large scale offshore wind turbines.

The control engineering group in Delft, nowadays called the Delft Center for Systems and Control (DCSC), has a long history in the modeling, identification, and control of wind turbines. The main focus in the past was on first principles modeling of, and robust controller synthesis for flexible variable-speed wind turbines. However, the citations below emphasize the role of experimental validation and system identification in the complete design process.

“To demonstrate the cost-effectiveness of controlled wind turbines in practice, it is essential that the opportunity is given to implement the presented ideas and resulting control strategies in a flexible, variable-speed wind turbine.”

[Molenaar 03]

“..direct validation of models describing wind energy conversion systems by a direct comparison with measured data is of very limited use. One of the few possible solutions to this problem is the application of system identification.”

[Bongers 94]

These challenges are embedded in this thesis. We show the *proof of concept of the ‘smart’ rotor* and we develop novel *subspace linear parameter-varying system identification* algorithms. The goal of this chapter is to show how these contributions relate to the ‘state-of-the-art’ control and identification, and coincide with the long term perspectives in the wind energy community.

1.1 Introduction to the wind energy field

The current wind turbine technology is still rather young compared to that of fossil fuel. However, the wind energy community is maturing quickly. Figure 1.1 shows that in 1995 there was a capacity of only 4.8 GW of wind power installed worldwide, while in 2007 there was already 94 GW (GWEC 2008). With a predicted capacity of 160 GW in 2010 (WWEA 2008), the increasing human awareness of the need for sustainable energy, and the predicted lack of availability of fossil fuels, wind energy has a bright future.

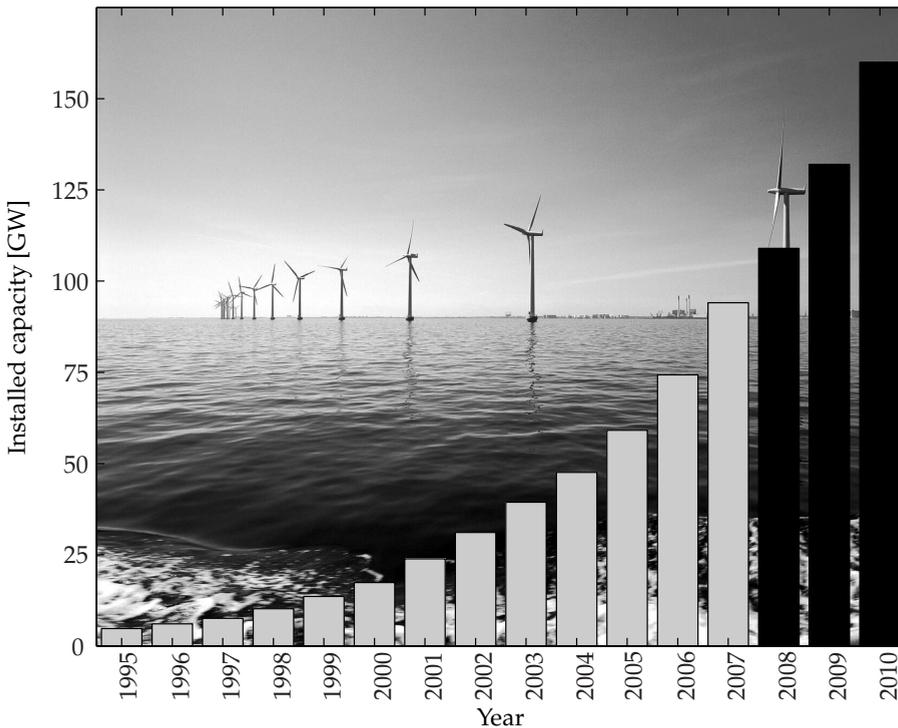


Figure 1.1: Total world wide installed and predicted installed capacity (WWEA 2008; GWEC 2008). Where the gray and black bars represent realized and predicted capacity, respectively.

Wind energy played an important role in the history of the Netherlands¹. Due to this history, the Netherlands has developed a strong position in the research and development in the wind energy field. The research institutes: 1). *Delft University Wind Energy Research Institute* (DUWIND 2008), 2). *Energy research Center of the Netherlands* (ECN 2008) and 3). *knowledge center Wind turbine Materials and Constructions* (WMC 2008) perform applied and fundamental research in the multi-disciplinary field of wind energy. These centers work on different international and national projects. One of the projects acquired by DUWIND is the project:

¹For a detailed overview of the history of wind energy we refer to Molenaar (2003)

“‘Smart’ dynamic rotor control of large offshore wind turbines” (van Kuik et al. 2003) and is sponsored by the Dutch national funding agency STW (STW 2008). As part of this project, this thesis mainly concentrates on the control and identification of wind turbines.

The motivation, highlighted in the project title, for focusing on the design of wind turbines of increasing size, is the trend in recent years to place the turbines offshore. This due to low turbulence levels offshore and the fact that in Europe the best onshore locations are already taken (van der Tempel 2006). However, the offshore foundations account for a large portion of the total wind turbine cost leading to the desire of the wind industry to increase the energy yield per wind turbine, and hence the rotor diameter, as much as possible. Hence, modern wind turbines designed for offshore application have become the largest rotating machines on earth with the length of one blade almost equal to the entire wing span of a Boeing 747. As illustrated in Figure 1.2 we see the progression from a 1.6 MW turbine realized in 1996 with a rotor diameter of 60 m to a sophisticated version of the E-126 6+ MW ENERCON turbine in 2008 (ENERCON 2008), which has a rotor diameter of 127 m. This trend in increasing rotor size and the trend to go

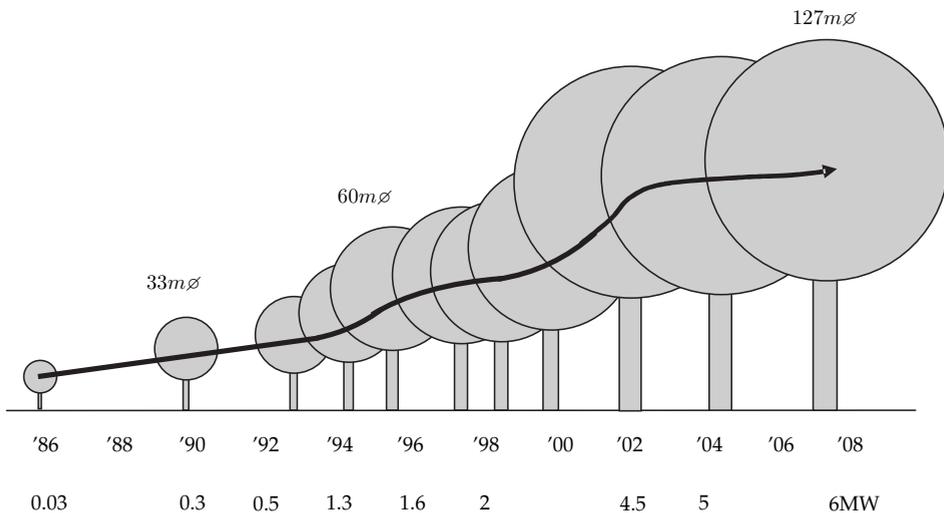


Figure 1.2: The trend in the development of wind turbines.

offshore explains the last part of the project title: “..large offshore wind turbines”. The first part of the project title relates to the tradeoff between the increasing size of the blades and the potentially damaging loads thereon. With the current control concept, the boundary of what is possible is reached, since increased turbine blade dimensions lead to drastically increased loads thereon. It is believed that more advanced control concepts and methodologies can surpass current limitations, resulting in even larger and more reliable wind turbines. One such novel

control concept is the so-called ‘smart’ rotor. This explains the first part of the project title: “‘*smart*’ *dynamic rotor control*..”. In the next section we focus on the state-of-the-art control of wind turbines and indicate a number of technologies, possibilities, and challenges to design the new generation of wind turbines from a control engineering perspective.

1.2 Control of wind turbines

The technology and science of control of wind turbines can be divided into two time frames: current state-of-the-art (including the short term developments) and future prospects. There is also a second distinction that we can make and that is the distinction between the academic wind community and the industrial counterpart. In this section we start with a brief introduction to the current state-of-the-art control of industrial wind turbines, then focus on some new technologies, possibilities, and challenges that may be applied to the next generation of wind turbines. Finally, we indicate that the academic wind energy community has a strong focus towards modern model based control.

1.2.1 State-of-the-art control of industrial wind turbines

There have been two main operation concepts to keep the loads on wind turbines (*e.g.* fatigue loads, power variations) within acceptable limits and to optimize the energy yield. The concept widely used from the seventies until the nineties of the previous century was the ‘Danish concept’ (Manwell et al. 2002). Such turbines combine constant rotor speed with stall of the flow around the rotor blades and are stable by design; increasing wind speeds automatically induce increasing drag forces that limit the produced power (this concept is also referred to as: stall turbine). In that period, all other control options were considered too complex and also the technology for variable speed control was not mature enough. Due to the development of dedicated converters, regulation concerning maximum allowable sound emissions and grid requirements, the most recent large wind turbines run at variable rotational speed, combined with the adjustment of the collective pitch angle of the blades (Bossanyi 2000; van der Hooft et al. 2003). This state-of-the-art control concept basically splits the operation range of a wind turbine into two parts: below-rated power and above-rated power; also referred to as ‘below-rated’ and ‘above-rated’, respectively. Below-rated is the operation region in which the wind does not contain enough kinetic energy to fully exploit the capacity of the generator. In this region the pitch position (angular position of the blade with respect to its longitudinal axis), θ_c , is kept constant (also referred to as: fine pitch) while the generator torque, T_{ge} , is controlled in such a way that the turbine tracks the optimal aerodynamic efficiency. In the above-rated power region, the objective is to produce rated power, where power is the product between generator torque and the generator rotational velocity. In the above-rated region these two quantities are kept constant and the pitch actuator takes over the control task. By

pitching the blades the lift forces can be reduced and consequently the aerodynamic moment can be controlled in such a way that the power is kept constant. The equilibrium solutions are illustrated in Figure 1.3, where we see the characteristics of the turbine for certain wind speeds; also referred to as operating points. However, a wind turbine is a dynamic system, so if a wind gust occurs, the wind turbine will go through a transient before converging to its new equilibrium. The transient behavior of a wind turbine is a complex combination of the aerodynamics and structural dynamics (together aeroelastics), the controllers, and the wind. A low-gain controller will cause the turbine to react slowly and sub-optimally to the changing wind conditions (see [Leith and Leithead \(1996\)](#)), while an aggressive, high-gain controller, will quickly approach the required equilibrium but impose a considerable load on the turbine. With a required lifetime of approximately 20 years, a trade-off is necessary between energy production and load control, better known as a multi-objective control problem ([Zhou et al. 1996](#)) to the control engineers. This problem is challenging because the emphasis of the controller should switch bumplessly from power tracking to load reduction depending on the operating point ([Leith and Leithead 1996](#); [Bianchi et al. 2007](#); [Østergaard et al. 2008a](#)). For example, low wind speeds generally produce relatively small structural loads and the energy yield can be increased if the aerodynamic efficiency is tracked accurately. At high wind speeds the loads are also high and the extracted power is already at its maximum, so the focus must switch to load control and the minimization of the power fluctuations.

If the wind speed is known the power set point can be generated and either the pitch or torque controller is active. There are two remarks to be made. The first is that the switching from below-rated to above-rated is presented in an adhoc manner. In practice, an effort is made to make the transition between below-rated and above-rated more smooth, resulting in a more complicated control scheme ([Bossanyi 2000](#); [van der Hooft et al. 2003](#)). The second remark is that the effective wind speed is hard to measure and generally is estimated using a large variety of techniques ([Bossanyi 2000](#); [van der Hooft et al. 2003](#); [van der Hooft and van Engelen 2003, 2004](#); [Østergaard et al. 2007a](#)). Finally, a number of extensions are presented in the literature, mainly to damp the tower and drive-train vibrational modes ([van der Hooft et al. 2003](#)).

For controller design it is also important to realize that the dynamics of wind turbines are dependent on the operational position, as clearly shown in [Hansen \(2007\)](#). However, for a certain operational position the dynamics can be considered to be Linear Time-Invariant (LTI), a requirement for linear controller design methods ([Franklin et al. 1994](#); [Zhou et al. 1996](#); [Ogata 1997](#)). Common practice in the wind industry is to make the gains of the designed controller dependent on the operating point, called "gain scheduling" ([Leith and Leithead 1996](#); [Bossanyi 2000](#); [van der Hooft et al. 2003](#)). However, as already indicated in [Leith and Leithead \(1996\)](#), interpolation between the different local controllers can result in unsatisfactory designs outside the points for which the controller was designed. Recently, more systematic scheduling methods have been proposed in the academic wind community based on the Linear Parameter-Varying (LPV) model structure to circumvent this problem ([Ohtsubo and Kajiwara 2004](#); [Bianchi et al. 2004, 2005, 2007](#);

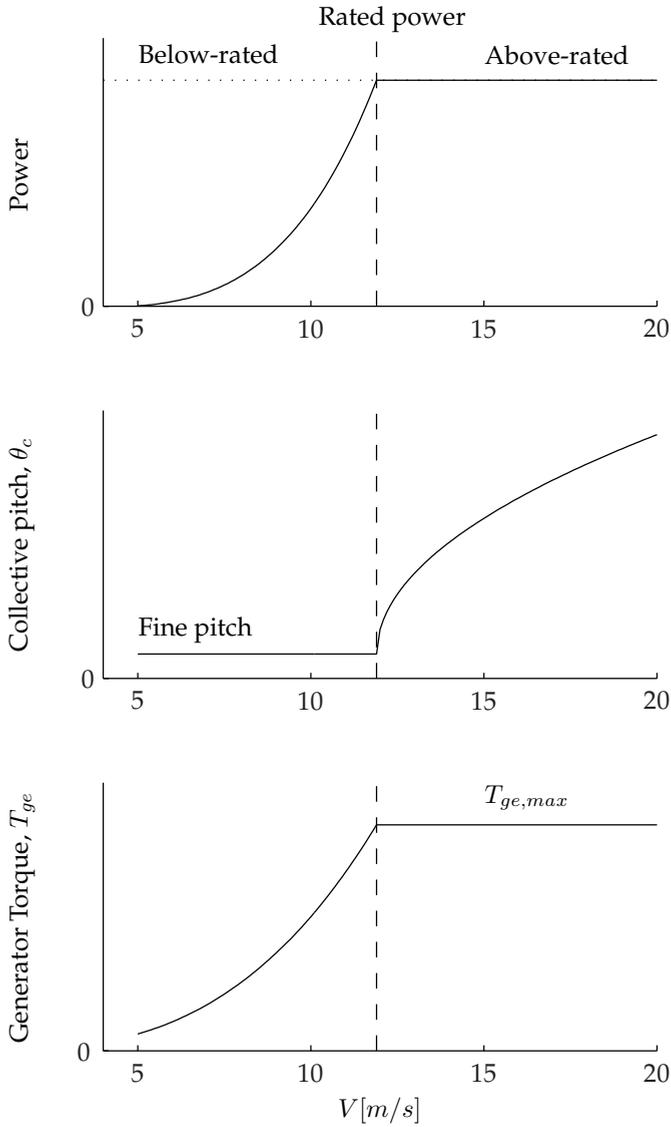


Figure 1.3: An example that indicates the steady state control actions for a variable-speed wind turbine. In the top, middle, and bottom figure we see the electrical power, pitch angle, and generator torque as function of the wind speed, respectively.

Lescher et al. 2006; Østergaard et al. 2008a,b; Østergaard 2008). The LPV model structure will be introduced in more detail in the next section.

Full-span collective pitch control, as previously discussed, is widely accepted in the wind energy community, but can only handle slow wind changes that affect the entire rotor. Because of the increasing rotor size it is necessary to react to the distributed nature of turbulence in a more detailed way: each blade separately and at several separate radial distances. This first item is dealt with by Individual Pitch Control (IPC) (Bossanyi 2003, 2005; van Engelen and van der Hooft 2004; Hansen et al. 2005; van Engelen 2006; Selvam et al. 2008), motivated by the helicopter industry (Ham 1980; Friedmann and Millott 1995; Lovera et al. 2006, 2007), which is the latest development in the wind turbine industry to further minimize the loads and is ready to be commercialized. With this concept each blade is pitched individually to suppress the periodic loads caused by tower shadow, wind shear, rotational sampling, yaw misalignment, etc. However, the performance of the IPC method is restricted by the limited bandwidth of and wear in the pitch actuators and because they only affect the load on the whole blade. A more advanced operation concept is required to further reduce the loads in order to optimize the rotor diameter with respect to weight and size.

To summarize one can say that compared to the stall turbines, full-span collective pitch control is a step forward; the control of the blade pitch angle has not only led to power regulation, but also to a significantly lighter blade construction due to the lower load spectrum and a lighter gear box due to shaved torque peaks. With the introduction of IPC, which will be implemented in the near future, the loads can be reduced considerably leading to even lighter or larger turbines. However, due to the increasing size of wind turbines it is necessary to look ahead to control concepts which can impose a force profile matching the distributed nature of turbulence in order to reduce the loads and to guarantee an economic lifetime of 20 years for the new generation of large scale offshore wind turbines (diameter over 150 meters). So, for the next generation of wind turbines we must look for novel control concepts that may be considered too complex at this point in time in the wind energy field, but so was IPC 20 years ago.

1.2.2 Introduction to the ‘smart’ rotor concept

There are a large number of concepts for the next generation of wind turbines. For all of these ideas, the most important design drivers are the loads (both extreme loads and fatigue loads). In the current designs this is solved by the mechanical design and collective pitch control (Fuglsang 2008). In the previous paragraph we discussed that pitch control is not a feasible solution for larger and more flexible machines, consequently more advanced concepts are required.

One advanced operation concept is to use a number of actuators that locally change the force profile on the wind turbine blade to cope with the spatial distributed nature of turbulence. This, in combination with sensors that measure the loads and a controller that manipulates the measured signals and generates an appropriate actuation signal, is defined as the ‘smart’ rotor concept. In Figure 1.4 an example of a ‘smart’ wind turbine is given.

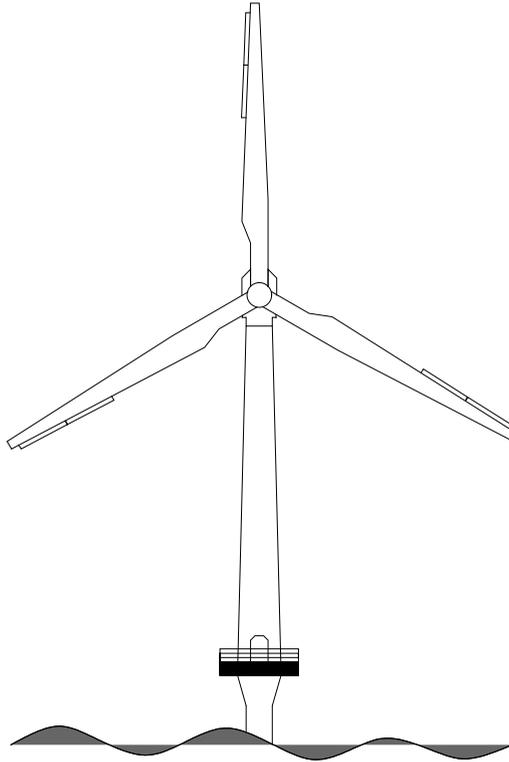


Figure 1.4: An ‘illustrative’ example of the new generation of wind turbines: the ‘smart’ rotor concept (Houtzager 2007). At the tip of the blade a number of additional control devices are drawn.

The ‘smart’ rotor concept is borrowed from the helicopter industry, where active devices like *e.g.* trailing edge flaps (Barret 1990; Chopra 2000), or Micro-Electro-Mechanical translational tabs (MEM-tabs) (Standish and Van Dam 2005) are proposed to reduce the loads. MEM-tabs and trailing edge flaps are illustrated in Figure 1.5 and Figure 1.6, respectively. Both of these concepts manipulate the boundary layer of the flow to change the aerodynamic forces and consequently control the loads. Trailing edge flaps are considered as the most promising idea for the ‘smart’ rotor concept (Marrant and Van Holten 2006). In Barlas et al. (2007) and Barlas and van Kuik (2007) a more detailed overview is given about these developments as well as some more exotic concepts.

The main goal of the ‘smart’ rotor is to reduce the fatigue loads to increase the lifetime of the wind turbine. However, when the lifetime constraint is reached the wind turbine rotor size may be increased or the rotor design may be optimized with respect to weight. Recently some research on this topic has been performed in the wind industry where trailing edge flaps (Joncas et al. 2005; Troldborg 2005; Buhl et al. 2005; Basualdo 2005; Andersen 2005; Andersen et al. 2006; Gaunaa 2006; Bak et al. 2007), and MEM-tabs (Zayas et al. 2006; Chow and van Dam 2007; van

Dam et al. 2007) have been used for load alleviation. The first step in the design of

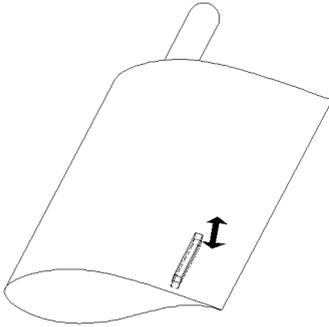


Figure 1.5: MEM-tab.

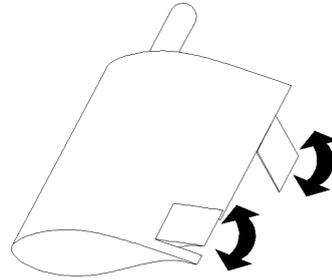


Figure 1.6: Trailing edge flaps.

this new control concept was a feasibility study. This study was performed by Basualdo (2005), where he showed the feasibility of a trailing edge flap applied on an airfoil (2-D study). In Andersen (2005) and Andersen et al. (2006) the feasibility of the ‘smart’ rotor concept on a rotating blade was demonstrated. The first proof of concept study was reported by Bak et al. (2007); a rigid cross-section with a trailing edge flap was used to validate the 2-D aerodynamic model of Gaunaa (2006).

In van Wingerden et al. (2008) (see also Chapter 2 of this thesis) a 3-D experiment is presented that also takes into account the blade aeroelastic effects and a feedback controller, thus proving the concept of a ‘smart’ rotor blade. This experimental work is significantly different from the work done by Bak et al. (2007); which used 2-D models without aeroelastics or a feedback controller.

1.2.3 Modern model based control in the wind energy community

So, it should be clear from the discussion so far that control is becoming more and more important for the wind energy field; progressing from stall induced turbines to IPC turbines and now the ‘smart’ rotor concept.

Currently, the control methodologies used in the wind industry are mainly based on Single-Input Single-Output (SISO) gain-scheduled PID regulators (Leith and Leithead 1996; van der Hooft et al. 2003). However, in the academic environments several articles are available that discuss optimal control (Stol 2001; Wright 2004; Hand and Balas 2007; Østergaard et al. 2007b), fuzzy logic control (Jauch et al. 2007), model predictive control (Henriksen 2007, 2008), robust control (Bongers 1994; Bianchi et al. 2007), and recently Linear Parameter-Varying (LPV) control (Ohtsubo and Kajiwara 2004; Bianchi et al. 2004, 2005, 2007; Lescher et al. 2006; Østergaard et al. 2008a,b; Østergaard 2008) of wind turbines. In general the industry is reluctant to use these novel methodologies due to their relative complexity and the expected small performance improvement. However, with the

more advanced operation concepts, such as the ‘smart’ rotor concept, the number of control variables is increasing and it becomes necessary to use Multiple-Input Multiple-Output (MIMO) model based control algorithms.

Still, the ‘smart’ rotor technology is a technology for the future and faces a number of challenges before the concept can be adopted by the industry, such as: development of suitable actuators and sensors, reliability, maintenance, integration of the actuators and sensors in a wind turbine, controller design, etc. Furthermore, the wind community should be convinced: a large number of proof of concept studies, studies in which the added value of these novel concepts is demonstrated and quantified, must be performed.

1.3 Introduction to LPV system identification

The increasing number of control variables in modern wind turbines will necessitate model based controller design for the wind energy community. In this section we motivate that Linear Parameter-Varying (LPV) system identification is a necessary building block for modern model based controller design for the wind community.

1.3.1 Modeling

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, the output, and the disturbances and should be accurate around the bandwidth of the controller. These requirements are based on implementation and conservatism issues. The two ways for obtaining a model and their application to the wind industry are listed below:

First Principles (FP) modeling: In FP modeling, the laws of physics are used to develop mathematical models. The main advantage of FP modeling is that you can have a model before the actual system is built and consequently the model can be used for system design and optimization. However, these FP models are not tailored for control requirements because they typically contain irrelevant dynamics and are nonlinear. A similar philosophy can be found in the wind energy community, where a large number of design codes are available (see [Molenaar \(2003\)](#) for a complete overview), but while controller design is a part of the design process, the models are not directly tailored to do so. However, there are a number of dedicated design tools that have the opportunity to derive models for controller synthesis (for example see [van Engelen \(2007\)](#); [Garrad Hassan \(2008\)](#)). Still, the amount of detail in the model is normally the choice of the user, who normally tends to ‘overmodel’ the system to make sure to capture all the dynamics. Another point that should be mentioned is that it is hard to have exact knowledge

of the material and aerodynamic properties, although small differences may produce significant different dynamics (Witteveen et al. 2007). Especially for detailed and complex models these uncertainties may have a big influence on the design and eventual implementation. In Witteveen et al. (2007) they include these uncertainties in their Computational Fluid Dynamics (CFD) models. Taking into account these uncertainties in the controller design process can lead to conservative but robust controller designs. Still, these models need to be calibrated based on measured data. With the high number of tunable parameters and the nonconvex nature of the problem, this is rather time-consuming.

Experimental modeling (system identification): In experimental modeling, also referred to as system identification, actual input and output data of the system is used to obtain a mathematical description of the system. Because this approach uses actual input and output data it will only model the dynamics present in the data. For a fixed operation point of a wind turbine, where the dynamics are assumed to be linear, LTI system identification (Ljung 1987; Verhaegen and Verdult 2007) is a well-established methodology to obtain a model for control. There are a number of scientific publications with applications in wind energy on this topic (see e.g. Bongers and van Baars (1991); van Baars and Bongers (1992, 1994); van Baars et al. (1993); James III et al. (1993); Knudsen et al. (1997); Marrant and Van Holten (2004); Hansen et al. (2006); Pires (2008)). The resulting models are only valid around one operating point. A typical problem for the application of system identification to wind turbines is that the data generating system, the wind turbine, has to operate in closed-loop, which requires some additional properties of the identification algorithms (Van den Hof and Schrama 1995; Van den Hof 2004). Another reason to use closed-loop system identification is given by Hjalmarsson et al. (1994), where they show that the iteration between closed-loop system identification and controller (re)design is beneficial for finding the ‘optimal’ controller.

We motivated that closed-loop system identification has a number of advantages compared to FP modeling, although an FP model is required to design wind turbines. Common practice outside the wind community is to use a mix of the two modeling approaches. This approach can be summarized as follows: develop an FP model for general design of the system and to synthesis a base-line controller. When implemented on the real system, the model used for controller synthesis should be refined by using system identification to obtain more accurate models around the bandwidth to obtain less conservative controllers. In the next step the controller synthesis should be performed on this identified model and if necessary the identification step and controller synthesis step can be repeated until a satisfactory result is obtained.

The application of system identification techniques to wind energy systems is currently not used by the wind energy industry². This is because up to now the

²However, Siemens recently performed a feasibility study to include system identification in their controller design process (Pires 2008).

wind industry has been satisfied with the performance of controllers based on FP models, but that does not mean that there is nothing to gain. This is acknowledged by the wind industry, and as a result the current state-of-the-art is to do actual measurements on a turbine at a number of different operation positions and compute the Power Spectral Density (PSD) for each. The PSD's are combined in a 3-D plot (the 2-D equivalent, is also referred to as a Campbell plot) to indicate the time-varying disturbances and dynamics (Rossetti et al. 2008). This plot is used to calibrate the FP models and based on this calibrated model an optimization step of the controller is performed. This calibration and updating process is repeated until a satisfactory performance is reached. A large number of tunable parameters and the nonconvexity of the problem make this a time-consuming process. Furthermore, as stated by Bongers (1994): '*...direct validation of the models describing the dynamics of the turbine by direct comparison with measured data is of very limited use*'. 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. By direct comparison of the measured PSD data with PSD data obtained from simulations, one can model the complete system, containing the dynamics of the wind turbine, the feedback controllers, and the disturbances. However, from this model it is still far from trivial how to isolate properties belonging to the dynamics of the wind turbine that are needed for controller synthesis. As indicated by Bongers (1994) the most promising solution is the application of system identification.

We motivated that system identification is a valuable tool for the wind community and is a logical next-step to be applied on a 'real' turbine for controller redesign. We also motivated that with the increasing dimensions of turbines and the application of more advanced control concepts MIMO control is required, and consequently we need MIMO system identification. There is one more point we would like to emphasize with respect to the identification of wind turbines. We mentioned that for a certain operation point the wind turbine can be considered to be LTI. However, a wind turbine switches from one operation point to the other quite rapidly as a result of the variations in wind speed. Common practice is to perform model identification and control at a number of different operating points, and then interpolate. As already indicated by Leith and Leithead (1996); the actual performance of such a controller can deviate from the expected performance in the transitions between different operating points. As mentioned earlier, more advanced scheduling methods have been proposed based on the Linear Parameter-Varying (LPV) model structure to circumvent this problem and to guarantee performance and stability in an operation region. For the identification procedure this implies that we must identify LPV models instead of LTI models. We introduced the identification of LTI models solely based on input and output data, also referred to as black-box modeling, which is a bit counter intuitive since known parameters are disregarded. In the LPV identification framework we include, in addition to the input and output data, the knowledge of the operation point of the wind turbine, and consequently the identification framework is referred to as gray box identification. In the next paragraph we elaborate more on the LPV model structure and the challenges for system identification.

1.3.2 LPV model structure

Linear Parameter-Varying (LPV) systems are a particular class of nonlinear systems which have attracted considerable attention in recent years. They can be thought of as a particular type of time-varying system, where the variation depends explicitly on a time-varying parameter referred to as the scheduling or weight sequence. For state-space systems, this results in the system matrices being a known function of this scheduling sequence. This is sometimes interpreted as an interpolation between different local linear models (Murray-Smith and Johansen 1997). The LPV model structure is represented by (continuous time)³:⁴

$$\begin{aligned}\dot{x} &= A(\mu)x + B(\mu)u + K(\mu)e, \\ y &= C(\mu)x + D(\mu)u + e,\end{aligned}$$

where x and \dot{x} are the state vector and its time derivative, respectively. A , B , C , D , and K are the system matrices. u , y , and e are the input, output, and noise signals, respectively. Finally, the parameter μ represents the parameter dependency of the model (for a wind turbine this might be the rotor speed, rotor position, or pitch angle).

The LPV model structure was introduced by Shamma and Athans (1991) and due to their close relation to Linear Fractional Transformation (LFT) descriptions (Lee and Poolla 1999), it has been possible to apply advanced control synthesis techniques to these systems, either continuous time or discrete time, which have guaranteed stability and performance (Becker and Packard 1994; Zhou et al. 1996; Apkarian and Adams 1998; Scherer 2001; Wu and Dong 2006). For wind turbines this translates to one global controller where the gains are dependent on the operation point with guaranteed stability in the transitions between different operation points, tracking of maximum power in the below-rated power region, and load reduction capacity above-rated power. For this controller framework, continuous or discrete time mathematical models are required in the synthesis step. However, other control methodologies are also making the step towards LPV controller synthesis such as data-driven control (Yoo and Rhee 2002) and model predictive control (Salcedo et al. 2007) which require discrete time models. This, together with the fact that sampled data is obtained from real systems, is the reason that we focus on discrete time model identification in this thesis. However, we also note the work presented in Tóth et al. (2008) where it is shown that the discretization of continuous time LPV systems is far from trivial. For example a nicely affine parameter-dependent model in continuous time is likely to translate into a more complex fractional parameter dependent model in discrete time. We will illustrate this difficulty in Section 4.9 with an example.

³For a discrete time formulation see Section 3.2 and 4.2.

⁴For a more elaborate discussion on the LPV model structure we refer to Casella and Lovera (2008).

1.3.3 LPV system identification

From the wind energy perspective we labeled the LPV identification problem as a kind of ‘gray-box’ identification problem because data (operational position) is used for the identification problem. However, the LPV identification problem can also be formulated in a black-box way assuming that the scheduling is not known. Intuitively this is a hard problem, which basically coincides with nonlinear system identification. In this thesis we assume that we know the scheduling sequence as in the LPV framework, where it is assumed that the scheduling sequence is measurable. To be more precise we will focus on state-space LPV identification with known scheduling. Similarly to the LTI case, a distinction can be made between state-space and input-output model representations of an LPV system, for which the most common identification procedures are Subspace Model Identification (SMI) and Prediction Error (PE) identification, respectively. In the input-output setting, results from LTI theory can be extended quite straightforwardly to the LPV setting (Bamieh and Giarre 2002; Previdi and Lovera 2004; Wei 2006; Tóth et al. 2007). The focus of this subsection is on state-space LPV identification methods. Although these methods face a number of challenges from a computational point of view, they have a number of advantages over the input-output setting:

- State-space methods have a straightforward extension to MIMO systems. In the input-output setting, the first step is to do model structure selection: to select the structure (*e.g.* OE, ARX) and the corresponding orders. In the state-space setting, the only degree of freedom is the state order. However, if we consider the subspace identification scheme, an estimation of the order is a part of the algorithm. The model structures for the input-output setting and their corresponding algorithms are hard to translate to the MIMO setting, while in the state-space setting this occurs naturally. This argument holds for the LPV case as well as for the LTI case.
- The state-space model structure is the desired structure for controller synthesis. The conversion from input-output model description to the state-space model description is one-to-one in the LTI case. In the LPV case the dependency on the scheduling parameter can change significantly (*e.g.* from static dependency to dynamic dependency), which makes this point important for LPV systems.

These issues motivate why the focus of this thesis is on state-space LPV identification. In general we can distinguish between four approaches to LPV system identification: 1.) *Interpolation* 2.) *Nonlinear optimization* 3.) *Subspace identification* 4.) *Identification using dedicated scheduling sequences*. Below a short overview of the different methods is given.

Interpolation: Common practice in industry is to identify a set of models belonging to different operating points, also referred to as stationary or local models. This is followed by an interpolation of the models into a particular parametrization (*e.g.* balanced state-space realization (Lovera and Mercere 2007)). For wind energy this translates into interpolation between the LTI

models corresponding to different operational points. A similar approach is followed in [Steinbuch et al. \(2003\)](#); [Groot Wassink et al. \(2005\)](#); [Wijnheijmer et al. \(2006\)](#) for mechanical motion systems. The followed approach is intuitive and simple and it leans on well-established realization and LTI system identification theory. However, the methods are only applicable if the application allows the scheduling to be constant for a certain period. Furthermore, in [Tóth et al. \(2007\)](#) it is shown that the interpolation between these stationary models can lead to unstable models of the LPV structure, even if the original system is stable. This phenomena was already acknowledged by [Leith and Leithead \(1996\)](#) where they stated that simply requiring local linear equivalence at the equilibrium points provides an inadequate basis for the choice of realization for the interpolated models.

Nonlinear optimization: The second approach is based on nonlinear optimization. In this approach the scheduling sequence is allowed to vary arbitrarily. It is well known that if the scheduling and full state information is available this results in a linear estimation problem ([Nemani et al. 1995](#); [Lovera 1997](#)). However, when only input and output data is used the problem turns into a nonconvex optimization problem. In [Lee and Poolla \(1999\)](#); [Verdult et al. \(2002, 2003\)](#) a parametric approach is followed where the LPV parameters are estimated using nonlinear optimization. The main disadvantage is that there is no guarantee of finding the global optimum. Furthermore, the state basis is fixed in the optimization algorithm, which can lead to ill-conditioned problems. The latest development in this particular category is from [Borges et al. \(2004\)](#) which splits the problem into a linear and a nonlinear part using Separable Least Squares (SLS).

Subspace identification: To overcome the drawbacks related to nonlinear optimization and the interpolation framework, the field of subspace LPV identification was founded. A milestone in this field is the Ph.D. thesis of [Verdult \(2002\)](#). This Ph.D. thesis extends the Bilinear identification scheme of [Favoreel \(1999\)](#) and [Favoreel et al. \(1997, 1999\)](#) and later improved by [Verdult et al. \(1998\)](#) and [Chen and Maciejowski \(2000\)](#) to LPV systems. Compared to the subspace LTI counterpart MOESP ([Verhaegen and Dewilde 1992](#)) this algorithm has the inherent drawback that it estimates the state sequences using a certain past window, possibly leading to biased results. Similar approximations are made in the subspace LTI algorithm: N4SID ([Van Overschee and De Moor 1996](#)), however, by making the past window larger and larger this bias will tend to zero. It turns out that identification of LPV systems with arbitrarily varying scheduling sequences is challenging from a numerical point of view ([Verdult and Verhaegen 2001, 2002](#)): the data matrices grow exponentially with the size of the past window. With the introduction of the kernel method ([Verdult and Verhaegen 2005](#)), the ‘curse of dimensionality’ was partially solved, however, a different bias was introduced ([Verdult and Verhaegen 2005](#)). In [van Wingerden and Verhaegen \(2008b, 2009\)](#) (see also Chapter 4 of this thesis) a significant dimension reduction is obtained, resulting in better estimates.

Identification using dedicated scheduling sequences: The fundamental problem

in the interpolation setting, and the numerical issues in the nonlinear optimization and subspace approach, forced researchers also to look at alternatives. Recently, a number of papers appeared where the structure of the scheduling sequence is exploited; it turns out that if the scheduling is periodic (Felici et al. 2007b; van Wingerden et al. 2008a) (see also Chapter 3 of this thesis), piecewise constant (Verdult and Verhaegen 2004; van Wingerden et al. 2007; van Wingerden and Verhaegen 2007), or white noise (Santos et al. 2005, 2006), well-established LTI subspace techniques can be extended to identify LPV or bilinear systems. The main advantage of this approach is that it leans on LTI techniques and does not require any approximation⁵. In Felici et al. (2007b); Verdult and Verhaegen (2004); van Wingerden et al. (2007); van Wingerden and Verhaegen (2007) a strong similarity appears with the interpolation methods. First a number of stationary models are estimated using well-established linear techniques but then, instead of interpolation, an intersection problem is formulated resulting in an LPV representation that is theoretically correct. Similarly as in the interpolation setting, the scheduling should have a certain structure, which can be restrictive for certain applications.

Open-loop versus closed-loop identification

Some of the applications on which LPV controller synthesis is of interest are unstable by nature and must operate in closed-loop to be identified, *e.g.* aerospace applications (Barker and Balas 2000) and wind turbines (Bianchi et al. 2007). It is well known that for LTI subspace identification the projector type of subspace algorithms (*e.g.* MOESP (Verhaegen and Dewilde 1992) and N4SID (Van Overschee and De Moor 1996)) give biased estimates if the identification data is generated under closed-loop conditions. The main reason for the bias is the constraint that the noise and the input should be uncorrelated. This assumption is clearly violated if there is a feedback loop present (as clearly explained by Ljung and McKelvey (1996)). Predictor-based subspace identification methods (*e.g.* PBSID (Chiuso and Picci 2005) and SSARX (Jansson 2005)) do not suffer from this drawback. The literature on LPV system identification indicated so far does not deal with closed-loop identification schemes, although the interpolation scheme can easily be extended to this particular setting. Recently, in van Wingerden et al. (2008a,b); van Wingerden and Verhaegen (2008a,b, 2009) (see also Chapter 3 and 4 of this thesis) novel subspace driven algorithms are presented that are based on their LTI counterparts PBSID (Chiuso and Picci 2005) and PBSID_{opt} (Chiuso 2007) for systems with arbitrary and dedicated scheduling sequences.

1.4 Goals of this thesis

In the previous sections we mainly highlighted two research areas: the *'smart' rotor concept* and *LPV system identification*. This thesis contributes to the development

⁵If we use the MOESP type of algorithms.

of these two research areas. In this section we specify what the goals of this thesis are.

The first step in the design of a ‘smart’ rotor is a feasibility study in which the basic concept of the ‘smart’ rotor is explored without too much detail on the boundary conditions (*e.g.* lightning strikes, robustness, etc). Within the project: *ADAPtive WING geometry for reduction of wind turbine loads* ([ADAPWING 2008](#)), done at the Danish research institute Risø, the focus was on a theoretical feasibility study. With respect to this work, the goal in this thesis is to show the feasibility of the ‘smart’ rotor under realistic wind turbine conditions (*e.g.* realistic disturbances, feedback control, and load measurements). Furthermore, we adopt the model based controller design cycle (modeling, identification, validation, and control) to control the ‘smart’ rotor. This cycle is used because it is common practice in high-tech mechatronic industries (*e.g.* the automotive, lithographic, and helicopter industry) and is therefore also suitable for the wind industry, since with the current evolution wind turbines also deserve the additive high-tech. The success of the realization might also help aerodynamicists and structural experts to embrace control engineering much earlier in their design cycle and use control engineering as a lever to create additional design freedom. To be more precise, we state the following goal:

Goal 1: *Experimental:* Show the experimental feasibility of the ‘smart’ rotor concept with emphasis on the controller design cycle.

In pursuit of this, we can partially achieve the desire highlighted in the abstract of this chapter; to show the contribution of control on an experimental wind turbine. With the first challenge we also include the desire highlighted in the second statement given by *Bongers*; to validate models using system identification. However, as indicated in this chapter, wind turbines are nonlinear systems and in order to apply LPV control on wind turbines there is also a need for LPV modeling. As introduced in the previous section, the identification of LPV systems is rather young; only a few algorithms have been developed and they only considered the open-loop situation⁶. The second goal is therefore a more fundamental one and is:

Goal 2: *Fundamental:* Develop efficient LPV identification techniques to obtain accurate LPV models of nonlinear systems given input, output, and scheduling data generated under open and closed-loop conditions.

These two goals allow us to contribute to the development of the new generation of wind turbines. First by showing the feasibility and second to come up with a building block for modern model based control.

⁶The closed-loop algorithms mentioned so far are a part of this thesis.

1.5 Contributions of this thesis

The contributions appear in two major subclasses: 1. *Proto-typing of a 'smart' rotor* and 2. *Discrete time LPV state space identification*:

- 1a. We showed the utility of active control on the 'smart' rotor under realistic wind turbine conditions, *i.e.* unknown disturbances, feedback control, and load measurements. In the wind tunnel we showed, with our experimental setup, that when the disturbance is known, perfect cancellation is possible. However, under realistic circumstances the disturbance is not known and feedback control is required. For this situation we showed the broadband load reduction capabilities of the 'smart' rotor for different load cases, *e.g.* with a sinusoidal disturbance we can reduce the amplitudes by 90% at the first eigenfrequency of the blade.
—see Chapter 2 and (van Wingerden et al. 2008)—
- 1b. We showed in our design that when you embrace control engineering in an early phase in the design cycle you create a lever to obtain additional design freedom. This result brings the cooperation between control engineers, aerodynamicists, and structural experts at the Delft University of Technology to a higher level.
—see Chapter 2 and (van Wingerden et al. 2008)—
- 2a. We showed that by using a periodic scheduling sequence the identification of LPV systems can be recasted into a number of linear time-invariant identification problems. Existing LTI subspace identification methods can then be used to determine the column space of the observability matrix for each LTI system. We have solved the crucial step in determining the original LPV system by ensuring that the obtained observability matrices and related state sequences are defined with respect to the same global state basis.
—see Chapter 3, (van Wingerden et al. 2008a), and (Felici et al. 2007a)—
- 2b. For LPV systems without periodic scheduling, we presented a novel subspace identification method. We derived a factorization that divides the parameter-varying extended controllability matrix into an unknown and a known part. Based on this factorization, a closed-loop identification method is developed to estimate the state sequence from which the LPV system matrices can be constructed.
—see Chapter 4 and (van Wingerden and Verhaegen 2008a,b, 2009)—
- 2c. The structure in the scheduling sequence in 2a and in the data matrices in 2b is explored in order to derive computationally efficient formulations for the identification methods.
—see chapter 3-4 and (van Wingerden and Verhaegen 2008a,b)—
- 2d. These algorithms are the first such LPV identification algorithms suited for data generated in closed-loop, which is a requirement for most aerospace and wind energy applications.
—see Chapter 3-4, (van Wingerden et al. 2008a), and (van Wingerden and Verhaegen 2009)—

1.6 Organization of this thesis

The two goals formulated in Section 1.4 have a totally different character and therefore this thesis is split up into two parts:

Part I: Experimental part The main challenge in the first part of this thesis is to demonstrate the effectiveness of the combination of modeling, identification, control, and validation on a practical demonstrator and to show the feasibility of the ‘smart’ rotor concept. Some results in this part are published elsewhere and are listed below:

- J.W. van Wingerden, A.W. 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”, in *Wind Energy*, 2008, **11**(3), 265-280.

Part II: Fundamental part In this fundamental part, a set of algorithms is developed to identify MIMO Linear Parameter-Varying systems for data generated in an open and closed-loop setting. The algorithms developed are not solely developed for wind energy but they may be applied to all kinds of dynamical systems with LPV characteristics. This part contains two chapters which can be read independently. In the first chapter we describe a novel LPV identification framework where we use dedicated scheduling sequences for the identification experiment. In the second chapter we deal with LPV systems where the scheduling is allowed to vary arbitrarily. At the end of both chapters the link with wind energy is highlighted using academic case studies. Some results in this part are published elsewhere and are listed below:

For periodic scheduling:

- J.W. van Wingerden, I. Houtzager, F. Felici, and M. Verhaegen, “Closed-loop identification of the time-varying dynamics of variable-speed wind turbines”, to appear in *International Journal of robust and nonlinear control*, special issue on Wind turbines: New challenges and advanced control solutions.
- F. Felici, J.W. van Wingerden and M. Verhaegen, “Subspace identification of MIMO LPV systems using a periodic weight sequence”, in *Automatica*, 2007, **43**(10), 1684-1697.

For arbitrary scheduling:

- J.W. van Wingerden and M. Verhaegen, “Subspace identification of Bilinear and LPV systems for open and closed loop data”, to appear in *Automatica*.
- J.W. van Wingerden and M. Verhaegen, “Subspace identification of multivariable LPV systems: a novel approach”, invited paper *IEEE Multi-conference on Systems and Control*, San Antonio, USA, September 2008.
- J.W. van Wingerden and M. Verhaegen, “Subspace identification of multivariable LPV systems: a PBSID approach”, invited paper *The 47th IEEE Conference on Decision and Control*, Cancun, Mexico, December 2008.

The outline of this thesis is visualized in Figure 1.7.

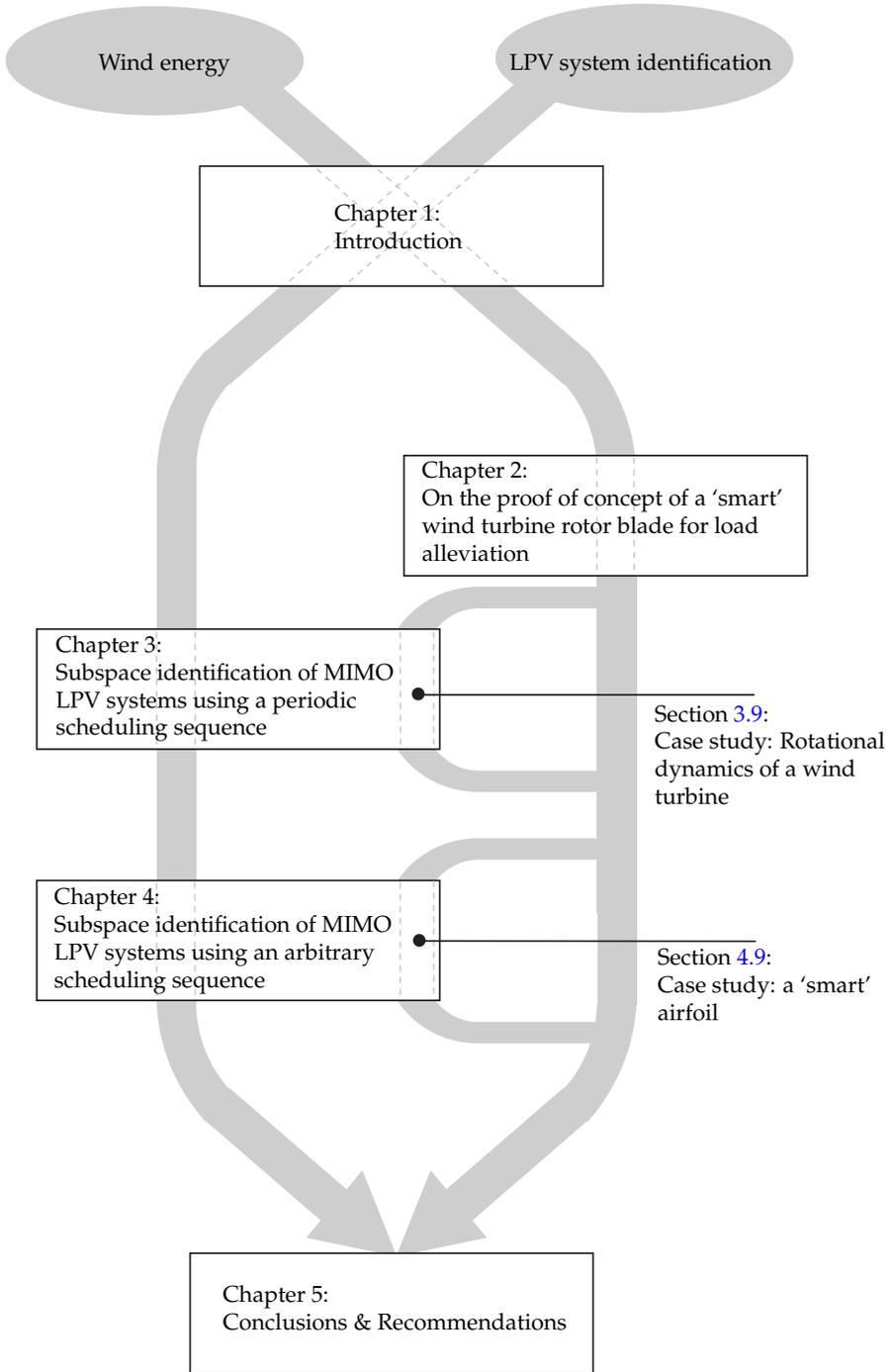


Figure 1.7: Visualization of the outline of this thesis.

Part I

Experimental part

On the proof of concept of a ‘smart’ wind turbine rotor blade for load alleviation

In this chapter a proof of concept study is performed to show the feasibility of the load alleviation abilities of a ‘smart’ blade; that is, a blade equipped with a number of control devices that locally change the lift profile on the blade, combined with appropriate sensors and feedback controllers. Theoretical and experimental models are developed of a scaled non-rotating rotor blade that is equipped with two trailing edge flaps and strain sensors to facilitate feedback control. A pitch actuator is used to induce disturbances with a similar character as a gust or turbulence. A feedback controller based on classical loop shaping is designed that minimizes the root bending moment in the flapping direction. We show that with appropriate control techniques the loads for periodic disturbances and for turbulence generated disturbances can be reduced up to 90% and 55%, respectively.

2.1 Introduction

In the previous chapter we discussed the state-of-the-art control in wind energy and motivated the ‘smart’ rotor concept. We also gave an overview of the work already done in this field. In this chapter we present a 3-D experiment that takes into account the blade aeroelastic effects and a feedback controller such that we have a proof of concept study of a ‘smart’ rotor blade. This experimental work is significantly different from the work done by [Bak et al. \(2007\)](#); they worked with 2-D models without aeroelastics, and they did not implement a feedback controller. We use a scaled non-rotating pitchable flexible rotor blade that is equipped with two trailing edge flaps. In the root two strain sensors are applied to facilitate

feedback control. The pitch degree of freedom is used to induce vibrations while the trailing edge flaps are used to compensate for the disturbances. Furthermore, we focus on the suppression of the 1P and 3P¹ loads in the flapping direction of a wind turbine blade and the ability to suppress the loads induced by gusts. For the scaled wind turbine blade the 1P and 3P frequencies are close to or below the first eigenfrequency of the blade (the first flapping mode). This means, from a mechanical point of view, that one vibration mode is dominantly present and the interaction between the mechanical and aerodynamic behavior is of critical interest.

The outline of this chapter is as follows. In Section 2.2 the experimental setup is presented, in Section 2.3 an analytical two-port model that facilitates controller design and model validation is presented, while in Section 2.4 the experimental results are presented. Finally, we end this chapter with the main conclusions.

2.2 Experimental Setup

In this section we present the experimental setup used to show the feasibility of the 'smart' rotor concept.

The 'smart' rotor that we use for our experimental validation is a non-rotating, pitchable blade with constant cross-section (see Figure 2.1). The blade is equipped with two trailing edge flaps to enable its use for future research. However, the two actuators are used together as one actuator, by applying the same control signal, because the main focus of this work is to suppress the first bending mode. For the same reason two strain sensors are applied in the root located on the central axis and at the leading edge of the blade, respectively. The experimental setup mainly consists of the following components: Wind tunnel, Blade, Actuators, Sensors, and Real-Time environment. Every element of the experimental setup will be described in this section.

2.2.1 Wind tunnel

A Low-Speed Low-Turbulence Wind Tunnel of the Delft University of Technology is used. It is an atmospheric tunnel of the closed-throat single-return type, with a maximum speed of 120 m/s. The test section is 1.8 m wide, 1.25 m high, and 2.6 m long. The total circuit length is 72.7 m and has a contraction ratio of 17.1 to 1. The free-stream turbulence level in the test section varies from 0.015% at 20 m/s to 0.07% at 75 m/s. These wind speeds correspond to Reynolds numbers in the range from $1.5 \cdot 10^5$ to $5 \cdot 10^5$, using 0.12 m chord models. In this wind tunnel it is not possible to tailor the incoming wind in such a way as to simulate disturbances that excite the model 1P and 3P disturbances. In the experimental setup the pitch actuator is used for this purpose.

¹once-per-revolution and three-times-per-revolution, respectively.

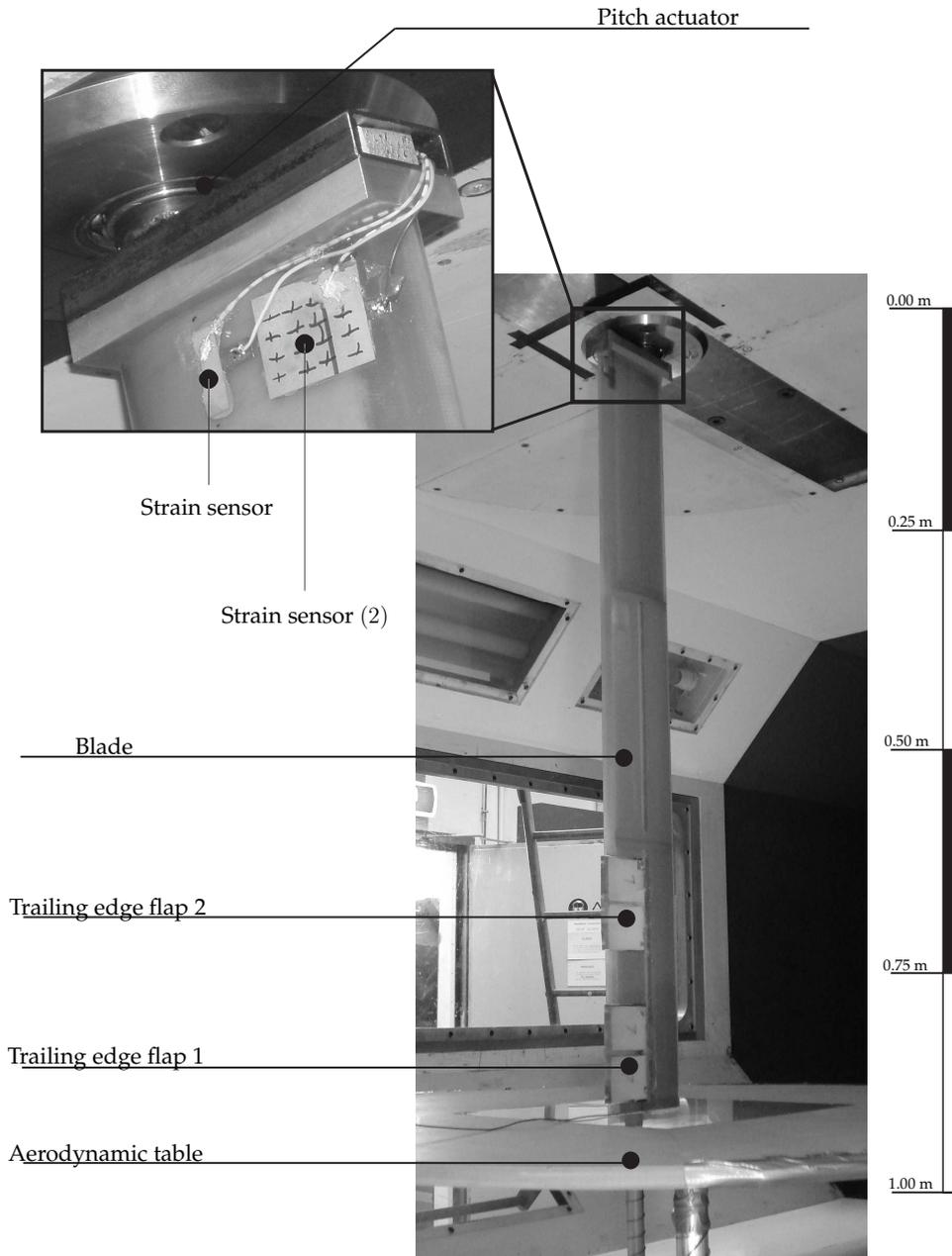


Figure 2.1: Photo of the 'smart' blade with the trailing edge flaps, the strain sensors, and the pitch actuator.

2.2.2 Blade Design

A blade of a reference turbine was used to scale our experimental model. The blade under consideration has a flap eigenfrequency of 1 Hz, a chord of 1.8 m at 75% blade length, and a span of 40 m. The maximum rotational speed of the blade is around 0.28 Hz with a nominal wind speed of 14 m/s. This implies that the 3P load, 0.84 Hz, is close to the first flapping mode of the blade at 1 Hz. The experimental blade is designed to have the same dynamic properties and the reduced frequency is used to scale the dynamics. The reduced frequency is given by:

$$k = \frac{\omega_k c}{2V_k}, \quad (2.1)$$

where k is the reduced frequency, c the chord length in meters, ω_k is the angular frequency of the unsteady disturbances in rad/s, and V_k is the velocity in m/s. We use the reduced frequency to scale the first flap eigenfrequency (also referred to as the flapping mode) and the 1P and 3P frequencies. This means that, with a wind speed of 45 m/s, we have a flapping frequency of 12.5 Hz. In Table 2.1 an overview of the parameters is presented.

Table 2.1: Scaling of the dynamic properties based on the 75% blade length values.

	Reference turbine	Experimental model
Chord [m]	1.8	0.12
Characteristic velocity [m/s]	54	45
1P load [Hz]	0.28	3.5
3P load [Hz]	0.84	10.5
1 st flap eigenfrequency [Hz]	1	12.5
k (1P)[-]	0.03	0.03
k (3P)[-]	0.09	0.09
k (eigenfrequency) [-]	0.1	0.1

The outer shape of the blade is given by the shape of the DU-W96-180 airfoil with 0.12 m chord and a length of 0.9 m, 0.35 m shorter than the height of the wind tunnel. To circumvent aerodynamic tip effects, an aerodynamic table is placed in the wind tunnel. The aerodynamic table prevents, by its size, flow around the tip of the blade. By this the flow remains 2-D, which has the advantage that the aerodynamic modeling is simpler than for a 3-D tip flow. In this phase of research this is considered to be appropriate. The table has a diameter of 1 m, and a distance below the blade tip of 3 mm. The blade is constructed in three different sections for structural reasons (Hulskamp et al. 2007). This is because the tip sections, in which the actuators are mounted, are also to be used in a future rotating experiment for which the root and middle section will be modified. The blade sections consist of foam cores, to which anodized aluminium inserts and

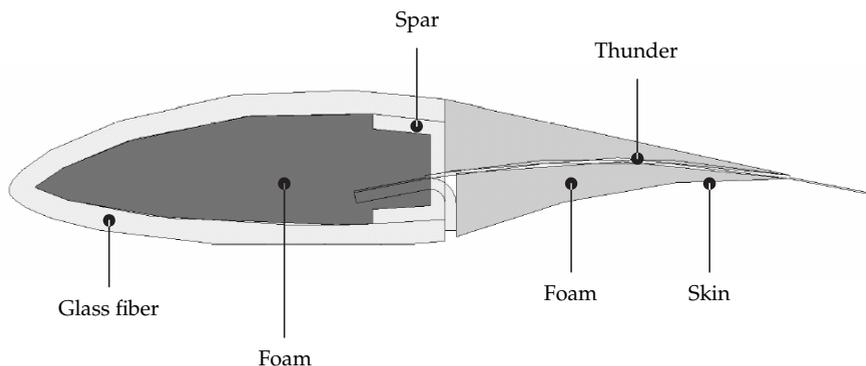


Figure 2.2: Schematic representation of the trailing edge flap.

prefabricated spars can be attached, wrapped with a number of glass fibre 8H-satin plies. This preform is placed in the aerodynamically contoured cavity of a rigid mold and vacuum infused with epoxy resin. In Figure 2.1 a photo of the blade is presented with the trailing edge flaps.

2.2.3 Actuators & sensors

The most challenging part in the design of the ‘smart’ rotor is the design of the actuator. In this chapter we design a specific actuator for the scaled blade model. The scalability is questionable with the actuator we propose; however, this is outside the scope of this research. In [Marrant and Van Holten \(2006\)](#) it is shown that the most effective control device is a trailing edge flap. Consequently, the blade was equipped with 2 trailing edge flaps in the outboard part of the blade, which is where the largest aerodynamic leverage can be obtained. The flaps have a width of 10.5 cm in the span direction and cover half the chord length (6 cm). The size is chosen in such a way that sufficient load reduction can be obtained for pitch variations of 2 degrees ([Marrant and Van Holten 2006](#); [Andersen et al. 2006](#)). The active part of the trailing edge flaps consist of Thundertm TH-6R actuators ([Face international cooperation 2008](#)). These are piezo-electric based benders that can deflect several millimeters under the application of a AC voltage from -450 V to 900 V. The actual deflection also depends on the structure around the bender and the aerodynamic loading. The actuators are shaped with soft foam to give them an aerodynamic shape. The foam is covered with a latex skin to provide a smooth surface. The actuators are attached to the blade through a bracket that is mounted on the spar. In Figure 2.2 the design of the flap is presented.

For control purposes, the blade is equipped with sensors that measure the dynamic behavior of the blade. Because the final goal for this non-rotating ‘smart’ blade is to reduce the fatigue loads, two piezoelectric patches (PZT) are adhered to the root to measure the high strains associated with the first bending mode. One PZT is placed on the neutral axis to measure the flapwise loads while the second PZT is placed near the leading edge of the blade to also measure the lead-lag

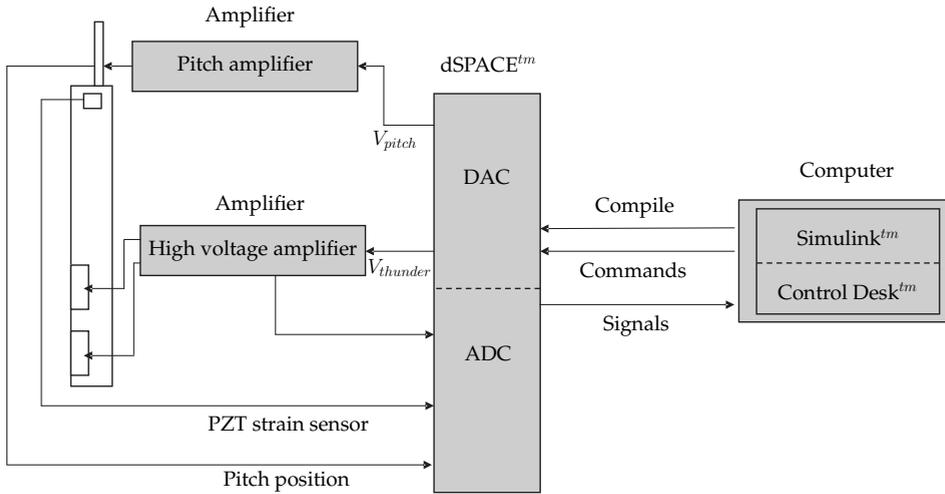


Figure 2.3: Schematic representation of all the available signals, hardware, and their interaction with the 'real-time' environment.

loads, the objective for future experiments. The first PZT signal is used as input for the feedback controller. The main advantage of a PZT is that no amplification is required to have a good signal-to-noise ratio. However, with the PZT it is not possible to do static measurements due to the capacitance behavior of the PZT. This high pass behavior is desirable for this experiment, as we want to control the dynamic behavior of the system, rather than the static deformations of the system. However, static deformations may be taken into account if different sensors (*e.g.* strain gauges or optic fibers) are used to measure the static strain. In Figure 2.1 the strain sensors can be found in the root of the blade.

For the pitching of the system a high force linear actuator with internal encoder is used. When a voltage is applied to the linear actuator it will elongate. However, a considerable amount of drift is present and a high gain feedback controller, where we compare the measured position with the reference position, is used to overcome this drawback. The reference signal can be chosen arbitrarily such that we can mimic the 1P, 3P, and gust loads with the described pitch system.

2.2.4 Real-Time environment

The 'smart' rotor described so far is not 'smart' when there is no control added. This controller intelligence and data acquisition capability are added with the inclusion of a dSPACE™ (dSPACE GmbH 2008) chip. The controller and data acquisition scheme are fully developed in the Matlab™ (The Mathworks 2008) and Simulink™ (The Mathworks 2008) environment and then compiled to the dSPACE™ chip. On a separate computer all the signals are monitored using Control Desk™ (dSPACE GmbH 2008) and the control parameters can be adjusted in real-time in the same environment.

In Figure 2.3 a schematic representation is given of the data processing system. The dSPACEtm chip has 8 Digital to Analog Converter (DAC) ports; 3 are used to generate signals to the pitch amplifier, V_{pitch} , and the high voltage amplifier, $V_{thunder}$. The outputs of the amplifiers are directly connected to the blade. Furthermore, 16 Analog to Digital Converter (ADC) ports are available; these are used to monitor the PZT’s and the voltages and currents of the high voltage amplifier. The encoder of the pitch actuator is attached to the encoder input of the dSPACEtm card. The dSPACEtm system is used with a sample frequency of 10kHz for control and data acquisition. With a controller incorporated in the real-time environment our blade deserves the additive ‘smart’.

2.3 Modeling of a ‘smart’ rotor blade and model validation

In this section we develop an analytic model of the experimental setup for controller design and for model validation. This means that we develop a compact model that includes the most dominant dynamics of every research field. Because of the multidisciplinary nature of a ‘smart’ rotor blade, a two-port approach is used to model the dynamics. The main advantage of such an approach is the exchangeability of the sub-models. For example: more complexity can be added to the mechanical block without altering the aerodynamic block. In Figure 2.4 the two-port representation of the model is presented. In the following subsections the different sub-models are presented and a feedback controller is introduced. At the end of the section we model the complete system in one step using an experimental approach. This experimental model is used to validate and update the first principles model.

2.3.1 Aerodynamic model

The dynamic behavior of the system is generated by the aerodynamic forces acting on the blade; these are caused by the wind and velocity perturbations imposed by the dynamic movement of the blade. This two-port coupling between the mechanics and the aerodynamics is also referred to as aeroelastics. Different models can be used to model this interaction. Theodorsen’s model (Theodorsen 1935) is a well known model that also takes into account rigid trailing edge flaps. However, in Gaunaa (2006) a more advanced model is presented that takes into account the deformation shape of the trailing edge flaps. For both models it holds that, when we assume that the static lift curve is linear, the aerodynamics can be modeled as a linear model for a given nominal wind speed (Theodorsen 1935; Lau and Krener 1999; Gaunaa 2006). The higher-order partial differential equations describing the aerodynamics can be rewritten as a number of first-order differential equations and they can be put in the state-space format (Kailath et al. 2000) given by:

$$\dot{x}_a = A_a(W)x_a + B_{a1}(W)y_m + B_{a2}(W)V_f, \quad (2.2a)$$

$$F = C_a(W)x_a + D_{a1}(W)y_m + D_{a2}(W)V_f, \quad (2.2b)$$

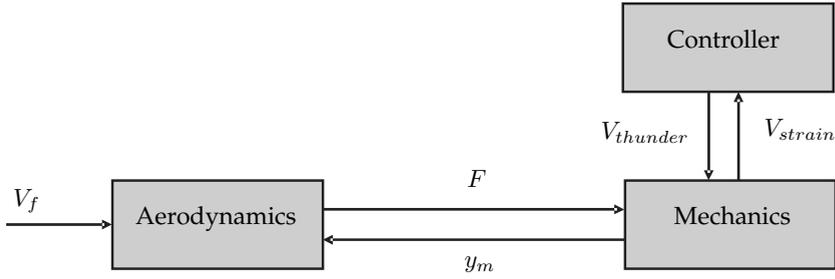


Figure 2.4: Two-port model of a ‘smart’ rotor blade.

where x_a and y_m represent the aerodynamic lag states and the mechanical outputs to the aerodynamic system, respectively. The forces acting on the blades are represented by F , while the nominal wind speed is represented by W , and the small perturbations in the wind speed are represented by V_f . The system matrices are $A_a(W)$, $B_{a1}(W)$, $B_{a2}(W)$, $C_a(W)$, $D_{a1}(W)$, and $D_{a2}(W)$; these can be directly derived from the aeroelastic equations of motion as demonstrated in [Theodorsen \(1935\)](#); [Lau and Krener \(1999\)](#); [Gaunaa \(2006\)](#). We use the representation presented in [Theodorsen \(1935\)](#), and, consequently, 4 lag states are added for every grid point. In the next section the blade is discretized in 4 segments. We apply the same aerodynamic grid and this implies that we have 16 lag states.

2.3.2 Mechanical model

The mechanical model of the blade has a two port coupling between the aerodynamics and the controller. The mechanical information fed to the aerodynamic model is the acceleration, velocity, and position profiles of the rotor blade while it experiences the aerodynamic force profile. The coupling with the controller takes place via the actuators and sensors.

The dynamics that form the coupling between the different input and output signals can be modeled with arbitrary complexity and accuracy. Different methods have been developed to describe the mathematical relations to get an accurate representation of the blade dynamics. These methods can roughly be classified into three types of systems; finite element, multibody, and modal representation.

For the experimental study, the multibody approach is chosen, because this approach is very appropriate for controller design, since the number of equations of motion remains small. The multibody approach approximates a flexible body as a set of rigid bodies coupled with springs and dampers. The spring stiffness and damping coefficients are functions of the material properties and the geometry of the original flexible body. This ‘superelement’ method has already been extensively tested and it is used for wind turbine simulation in DAWIDUM ([Molenaar 2003](#)). In [Figure 2.5](#) this approach is demonstrated. The mechanical model obtained is nonlinear but can easily be linearized. For this model we use a linear model with 8 states; this accurately represents the first 2 flapping modes. The

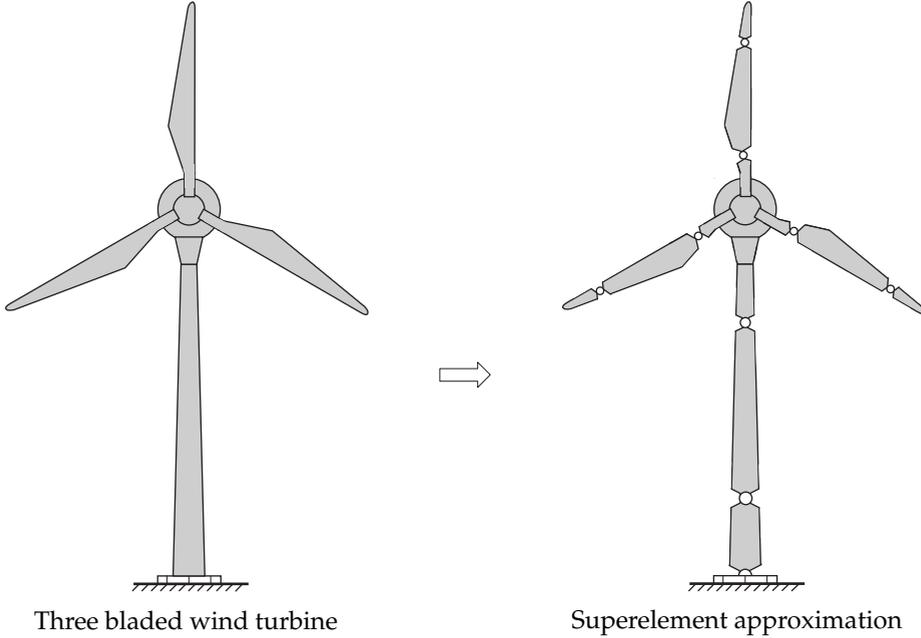


Figure 2.5: The multi body approximation of a wind turbine (Molenaar 2003).

torsion and lead-lag degrees of freedom are neglected because the main focus of the experiments in this chapter is to reduce the flapping motion. However, the lead-lag modes should be modeled for a 'real' wind turbine because they couple with the flapping modes.

In the mechanical modeling part we also include the sensors and actuators. The pitch actuator is modeled as a rotational torque generating device while the pitch angle is modeled as a measured input. The pitch actuator is only used to induce a vibration and is not used for load alleviation; thus it is modeled as a tracking controller that controls the position of the blade. The trailing edge flaps are modeled as a second-order filter that outputs the flap angles, angular velocities, and accelerations. The second-order systems can be tuned on the mechanical behavior of the trailing edge flap. The resulting linear mechanical model is represented in the following state-space representation:

$$\dot{x}_m = A_m x_m + B_{m1} F + B_{m2} V_{thunder}, \quad (2.3a)$$

$$y_m = C_{m1} x_m + D_{m1} F + D_{m2} V_{thunder}, \quad (2.3b)$$

$$V_{strain} = C_{m2} x_m, \quad (2.3c)$$

where x_m represents the mechanical states. The forces acting on the blades are represented by F , the control input is represented by $V_{thunder}$, and the output is given by y_m and V_{strain} that represent the mechanical outputs to the aerodynamic system and the measured strain, respectively. The system matrices are A_m , B_{m1} , B_{m2} , C_{m1} , C_{m2} , D_{m1} , and D_{m2} .

2.3.3 Controller Design

The main focus of the 'smart' rotor concept is to reduce the fatigue loads. For controller design it is important to know the dynamic relation between the actuator used for control, the control device, and the sensor used for control. The state-space formulations of the previous subsections can be coupled to obtain a 28th order first principles model that describes the desired relations. In Figure 2.6 the dynamic relation between the actuators and sensors is given by means of a Bode plot (Franklin et al. 1994; Ogata 1997). In Figure 2.6 we see the frequency dependent gain and phase (delay) of the dynamics. We will explain the Bode plot with the help of an example: Figure 2.6 presents the Bode plots between the pitch position and the measured strain and between the 'smart' actuators (that both have the same signal) and the measured strain. The measured strain is expressed in Volts. In the top left figure the gain of the pitch system with respect to the strain is given. If we have a sinusoid with a frequency of 4 Hz (enter in the x-axis) we observe a gain of 10. This means that for this specific sinusoid the strain is $10 \cdot 1 = 10$ Volt. If we apply a sinusoid signal with the same frequency but with an amplitude of -400 V on the trailing edge flaps we also find, Figure 2.6 top right, that the strain sensor will sense a sinusoid signal with an amplitude of $-400 \cdot 0.025 = -10$ Volt. In the subplots in the bottom of Figure 2.6 we can conclude that the output sinusoid has a phase shift of 20 degrees with respect to the input. Since this shift is the same for both signals at this specific input frequency the signals are in phase. If we apply both signals we use the superposition principle of linear systems to observe that the two effects will cancel out in the output and consequently the system will be motionless. In this case we assume that we know the disturbance beforehand and perfect cancellation can take place. Bak et al. (2007) also performed this exercise on a similar experimental setup for a 2-D airfoil where the lift was minimized. From now on we refer to this situation as feedforward control. In reality the disturbance is not known beforehand and real-time measured information is required to compensate for the unknown disturbances. In this case feedback control is applied.

In industry it is common practice to use the Bode plot between the input and output to design a feedback controller. The goal of feedback control is to suppress the disturbance as much as possible; however, the ability to do so is limited by the requirement that the system remains stable; a bounded input will result in a bounded output. In Figure 2.7 the structure of the feedback loop is presented. The feedback controller has as input the strain in the root expressed in volts (V_{strain}). Based on this signal the controller generates the deflection of the trailing edge flaps by applying a voltage to the Thunder actuators ($V_{thunder}$). For a stable non-minimum phase system, loop-shaping is a well known method to design stable feedback controllers based on the Bode plot of the system of interest (Franklin et al. 1994; Zhou et al. 1996; Ogata 1997). For the system considered, a smoothed Proportional-Derivative (PD) controller with additional notch filter will make a stable feedback controller with a considerable amount of disturbance rejection (Franklin et al. 1994; Zhou et al. 1996; Ogata 1997). This controller, used for the first principles and experimental model, is represented by the following

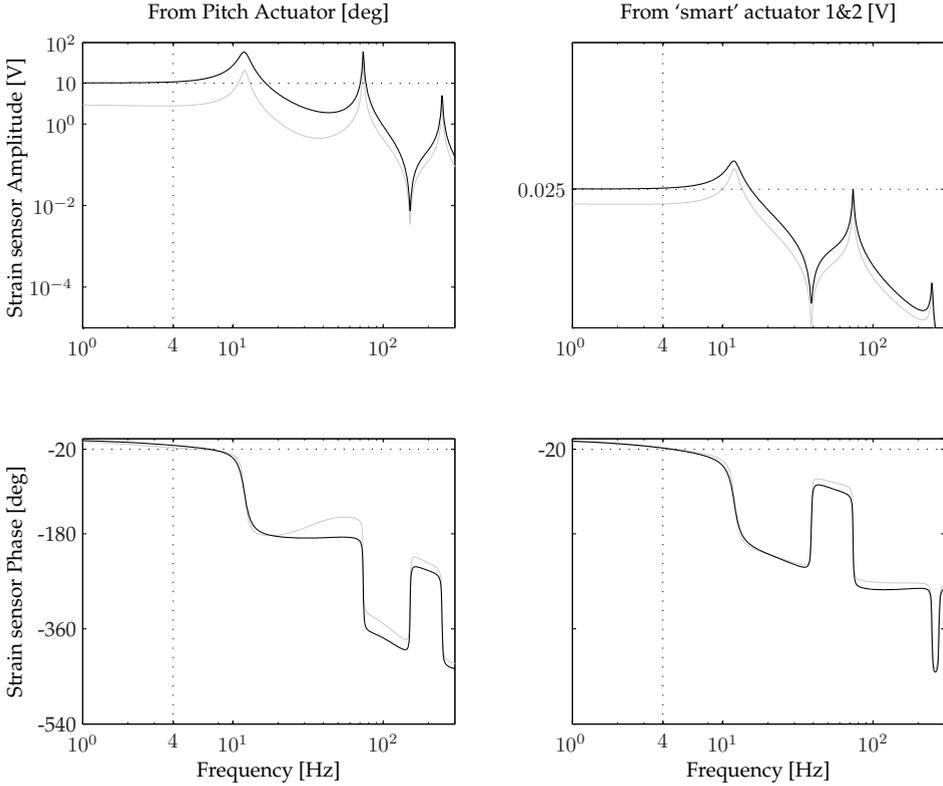


Figure 2.6: The linear models represented by their Bode Plot; gray is for 30 m/s and black 45 m/s. The three peaks correspond to the first 3 flapping modes.

transfer function:

$$C(s) = P \left(\frac{\frac{s}{\tau_1} + 1}{\frac{s}{\tau_2} + 1} \right) \left(\frac{s^2 + 2\omega\xi_1s + \omega^2}{s^2 + 2\omega\xi_2s + \omega^2} \right), \quad (2.4)$$

where $C(s)$ and s represent the controller and the Laplace operator, respectively. The variables P , τ_1 , τ_2 , ω , ξ_1 , and ξ_2 are parameters to tune the feedback controller for the theoretical and experimental study. These parameters are tuned based on Figure 2.7 using loop-shaping, to have a sufficient amount of gain margin, phase margin, and bandwidth (see Zhou et al. (1996) for more information). However, applying this analytically tuned feedback controller to the experimental setup is not advisable. Due to some unforeseen dynamics that have not been taken into account in the model and/or uncertainties in the parameters, the system can become unstable. Thus, we present experimental modeling in the next subsection to obtain a Bode plot from measured signals on which we can tune the controller given in (2.4).

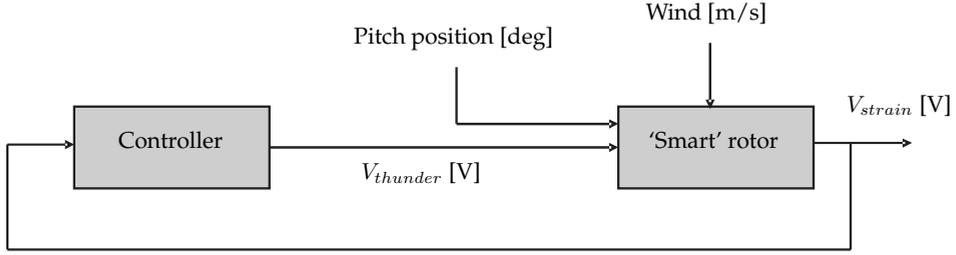


Figure 2.7: Schematic representation of the feedback problem.

2.3.4 Experimental modeling

The analytical modeling done so far strongly depends on a large number of parameters that completely determine the dynamic behavior of the system (*e.g.* stiffness, damping, shape of the flap, etc). Most of these parameters can be roughly estimated or calculated. Still, a large amount of uncertainty is present (*e.g.* the dynamics around the trailing edge flaps, actuator and sensor dynamics, etc); this makes it hard to design a stable feedback controller. In this subsection we use experimental modeling to build a linear dynamic model. A subspace identification method is used because of the numerical simplicity and the potential to work with Multiple-Input Multiple-Output (MIMO) systems (Verhaegen 1994; Van Overschee and De Moor 1996; Verhaegen and Verdult 2007). Subspace identification uses linear algebra (*e.g.* QR, SVD) to estimate a state-space model from measured data. From this state-space model, representing the dynamics of the system, a Bode plot with the measured gains and phase behavior can be made. The controller in (2.4) can then be tuned with this plot using loop shaping.

The control loops in the current system are the transfer functions between the trailing edge flaps, the pitch actuator, and the PZT sensor. The performance of the subspace 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 value is defined as:

$$\text{VAF}(y_k, \hat{y}_k) = \max \left\{ 1 - \frac{\text{var}(y - \hat{y})}{\text{var}(y)}, 0 \right\} * 100\%, \quad (2.5)$$

where \hat{y} denotes the output signal obtained by simulating the identified system, y is the measured signal, and $\text{var}()$ denotes the variance of a quasistationary signal.

The subspace identification is done with a step signal on the pitch actuator while a white noise signal with a bandwidth of 100 Hz is applied on the trailing edge flaps. These identifications can be done for different wind speeds and different angles of attack. However, it is expected that the angle of attack does not change the dynamic behavior of the blade. We performed subspace identification for two models: one for the 30 m/s situation and one for the 45 m/s situation, both with an angle of attack of 6 degrees. However, we will validate the model not only for this angle of attack, but also for an angle of attack of 3 degrees. In Table 2.2

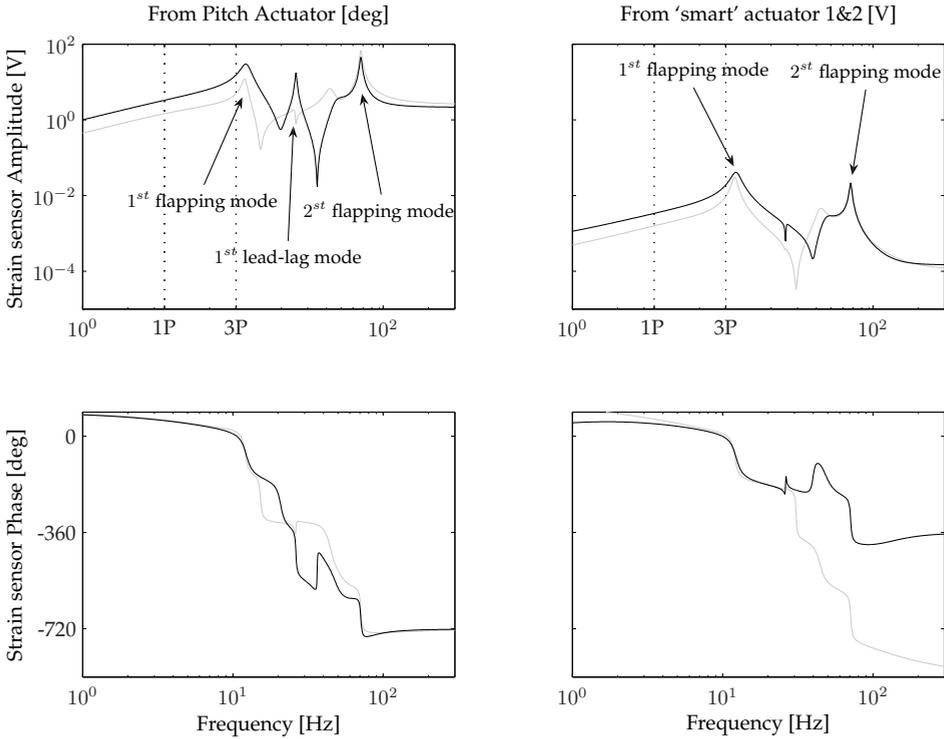


Figure 2.8: The linear subspace identification models Bode Plots; gray is for 30 m/s and black 45 m/s.

the settings and the VAF values are presented for the two subspace identification models and in Figure 2.8 the Bode plots are presented.

Table 2.2: The Variance Accounted For (VAF) for the different models.

V [m/s]	θ [deg]	VAF PZT [%]
30	3	85.2
30	6	86.2
45	3	89.5
45	6	91.6

From Figure 2.8 three resonance peaks can be observed; these represent the first flapping mode, the first lead-lag mode, and the second flapping mode. The third flapping mode is not seen because the excitation signal has a bandwidth of only 100 Hz, below the mode frequency of about 120 Hz. When we look at the effect of the wind speed in Figure 2.8, the gain is higher for a wind speed of 45 m/s, so we need gain scheduling. However, the eigenfrequencies hardly vary. Furthermore, we see in Table 2.2 from the VAF values of the PZT that the pitch

angle hardly changes the dynamic response of the system. This is because the system is still below stall and consequently the lift curve is still linear. The VAF values are higher for the higher wind speed. This is caused by the fact that with similar excitation the measured signals are larger and consequently the signal-to-noise ratio is higher, which has a direct effect on the VAF.

If we compare the subspace identification model with the first principles model (the model that incorporates the aerodynamic model and the mechanical model), we see globally the same behavior (compare Figure 2.6 with Figure 2.8). For the low frequencies we see a severe difference due to the unmodeled dynamics of the piezo sensors in the analytical model². Piezo electric sensors can not measure static loads and that explains the difference with the identified model. Furthermore, from subspace identification 10 states appeared to be dominantly present in the data resulting in a 10th order model, which is significantly smaller than the 28th order first principles model. For advanced control methodologies (e.g. *LQR*, *H₂*, *H_∞*), where the controller is of the same order as the model, implementation problems will arise (e.g. computational power) and model reduction should be performed.

The controller presented in the previous subsection is tuned on the subspace identification model. However, due to the unknown dynamics of the model at higher frequencies, a second-order roll-off filter with a cut-off frequency of 200 Hz is added to suppress the higher frequencies in the feedback signal. Furthermore, due to the low gain at the 1P frequency and inverted notch is added to have some disturbance rejection in the low frequency range. In Appendix A more controller details are given.

2.4 Wind tunnel measurements

In this section the main results of the experiments in the wind tunnel are presented. The subspace identification experiment has already been presented in the previous sections. In this section we present 4 cases that will show the possibilities of the ‘smart’ rotor concept. The first experiment is the so-called feedforward experiment where we use prior information to compensate for the disturbances. For the second experiment we do a similar experiment without prior information but now using the feedback controller. In the third experiment we show the response of the system to step disturbances with and without control. In the fourth experiment we show the response of the system to a filtered white noise with peaks at the 1P and 3P frequencies to mimic a typical turbulence spectrum.

2.4.1 Case 1: Feedforward control with a periodic disturbance

The first result that we present is based on the feedforward approach. A sinusoidal disturbance is applied to the pitch reference and a sinusoidal ‘smart’ rotor

²For analytical modeling of piezoelectric elements we refer to [van Wingerden \(2004a,b\)](#) and references therein.

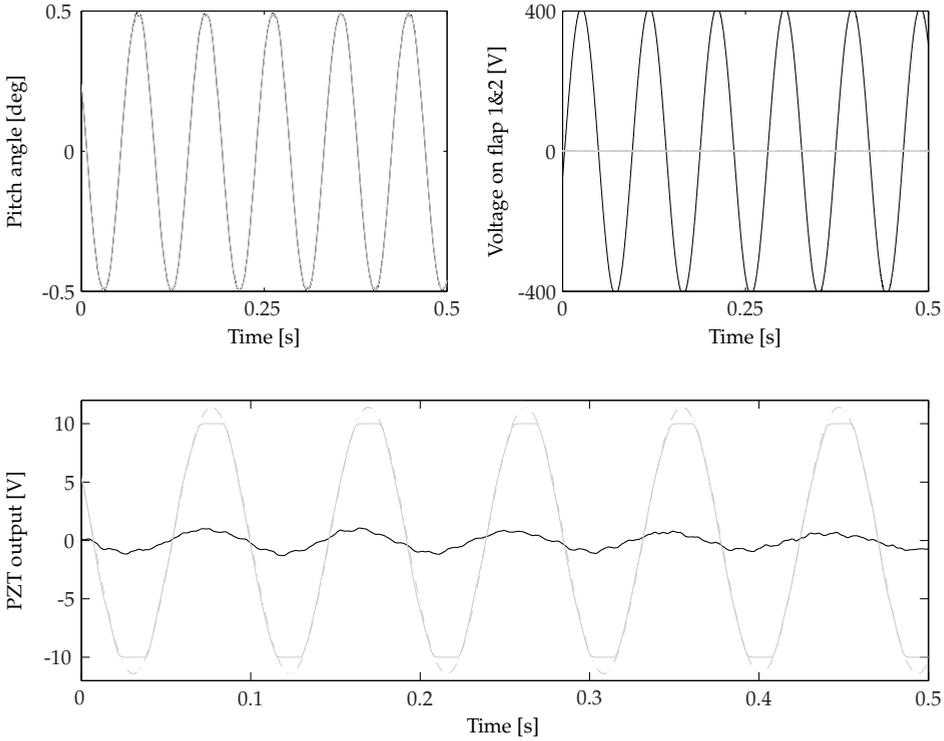


Figure 2.9: With (black) and without (gray) feedforward control for the 3P disturbance with an amplitude of 0.5 degrees on the pitch for a wind speed of 45 m/s. The dotted line represents the estimated strain [V].

actuation signal is applied to the flaps with different amplitude and phase. In Figure 2.9 the results are depicted for a sinusoidal disturbance with a frequency that corresponds to the 3P frequency with an amplitude of 0.5 degrees for a wind speed of 45 m/s. This signal is presented in the top left of Figure 2.9. From inspection of Figure 2.8 we decide to provide a sinusoidal input to the trailing edge flaps with a phase shift (delay) of 8 degrees and an amplitude of -400 volts. This control signal is depicted in the top right of Figure 2.9. The gray line represents the situation without control and the black line with control. The performance signal, the PZT signal, is presented in the lower part of Figure 2.9. It saturates at 10 V due to the sensor specifications. However, because we apply a pure sinusoid the estimated output is represented by the dashed line.

We observe that for the controlled case we see a reduction of close to 90% from the uncontrolled case at this particular frequency. However, we have to stress that the disturbance is known and the most ideal cancellation is reached. Theoretically the reduction should be 100%; however, limitations on the input amplitude play an important role. The same result is obtained for different frequencies and working conditions. In Bak et al. (2007) similar results are presented for the 2-D situation with a rigid blade.

2.4.2 Case 2: Feedback control with a periodic disturbance

With feed forward control it is assumed that the disturbance is known and perfect cancellation can take place when an accurate model of the system is available. When there are small uncertainties, the feed forward can be tuned online without any consequences for stability. For feedback control this unfortunately does not hold; an accurate model is required to design a feedback controller (see Section 2.3). The feedback controller described in (2.4) was tuned off-line and implemented. In Figure 2.10 we present a similar situation as in the feed forward case. However, the control signal is now generated by the feedback controller and we evaluate the controller at the first eigenfrequency of the blade. The load reduction

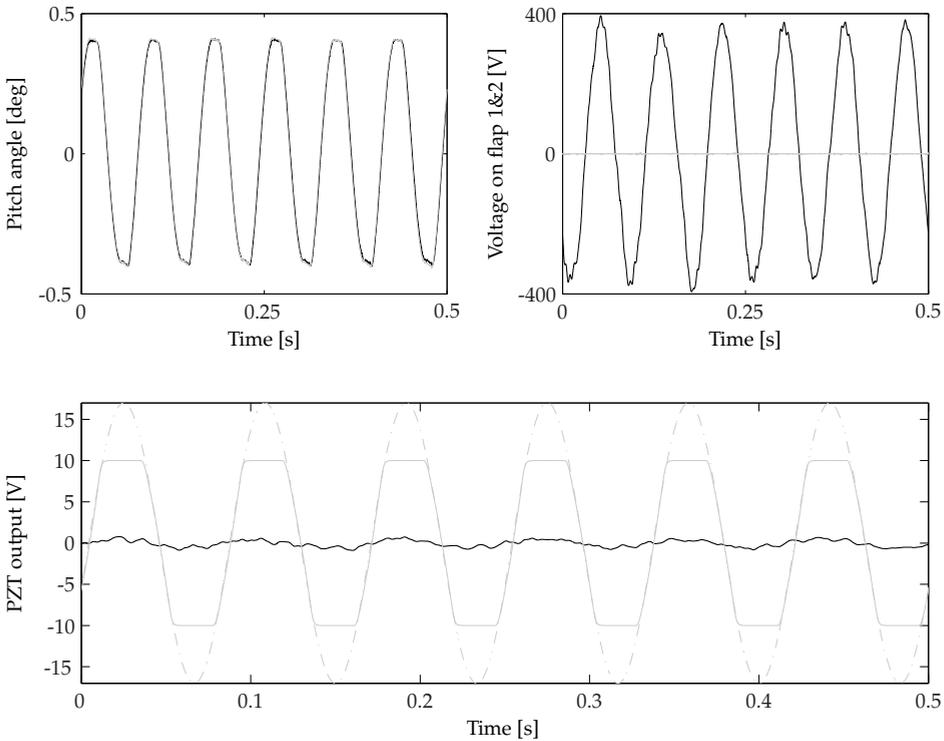


Figure 2.10: With (black) and without (gray) feedback control for the eigenfrequency with an amplitude of 0.5 degrees on the pitch for a wind speed of 45 m/s. The dotted line represents the estimated strain [V].

capability is around 90%. Due to the phase behavior of the system, this performance can not be reached for every frequency, but it can be expected for the 3P excitation, which is close to this eigenfrequency (see Appendix A for frequency domain details).

2.4.3 Case 3: Feedback control with a step disturbance

The main advantage of the feedback controller is that we can apply an arbitrary disturbance and the feedback controller will try to compensate for that. In the third case we mimic a wind gust by applying a step disturbance to the pitch system. The results are shown in Figure 2.11. We observe that the feedback controller works fine and damps the first flapping mode of the blade. It also attenuates the peak and the peak-to-peak amplitudes, thus reducing the fatigue loading.

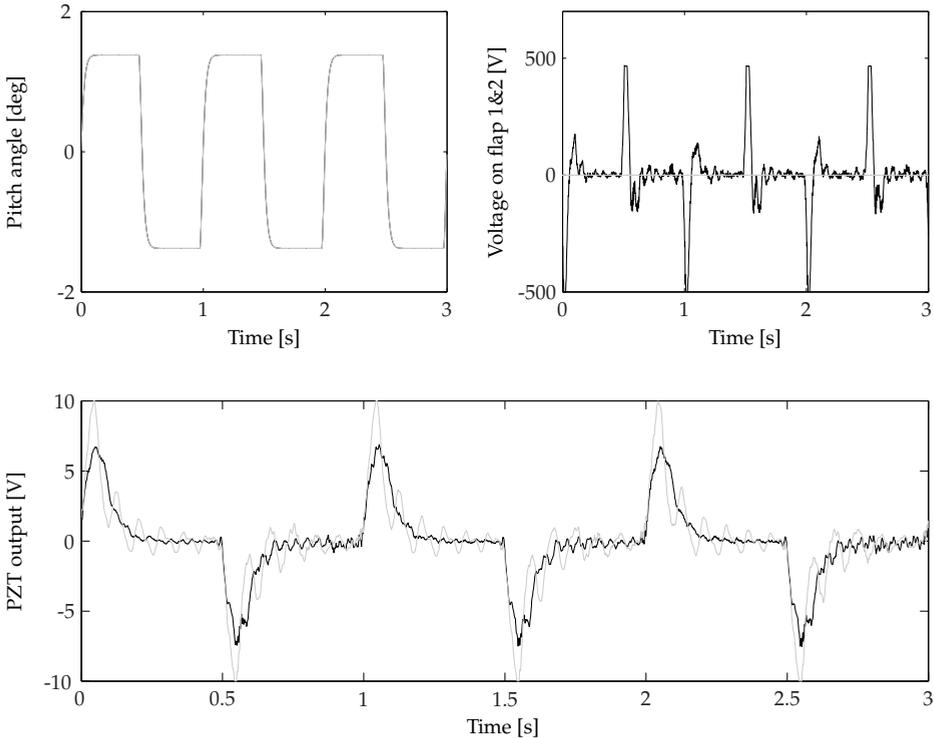


Figure 2.11: With (black) and without (gray) feedback control for a step disturbance on the pitch reference for a wind speed of 45 m/s.

2.4.4 Case 4: Feedback control with a representative noise signal

For the fourth experiment we use a filtered white noise to mimic the 1P and 3P loads on a real wind turbine blade. The input is completely unknown so every feed forward algorithm would fail. Because fatigue reduction is the main objective of this experiment, we express the performance using the Power Spectral Density (PSD), which is given by:

$$\Phi_u = \frac{1}{N} |U(\omega)|^2, \quad (2.6)$$

where Φ_u represents the PSD of the time signal u of length N and $U(\omega)$ represents the discrete time Fourier transform of the signal u . In Figure 2.12 the disturbance

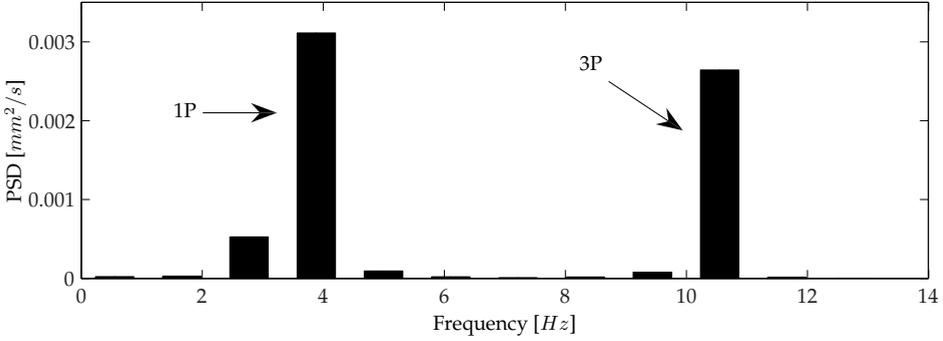


Figure 2.12: The PSD of the pitch input signal.

spectrum is presented for a load spectrum with peaks at 1P and 3P. Observe that the 1P peak is a little bit higher than the 3P peak. In Figure 2.13 the PSDs of the

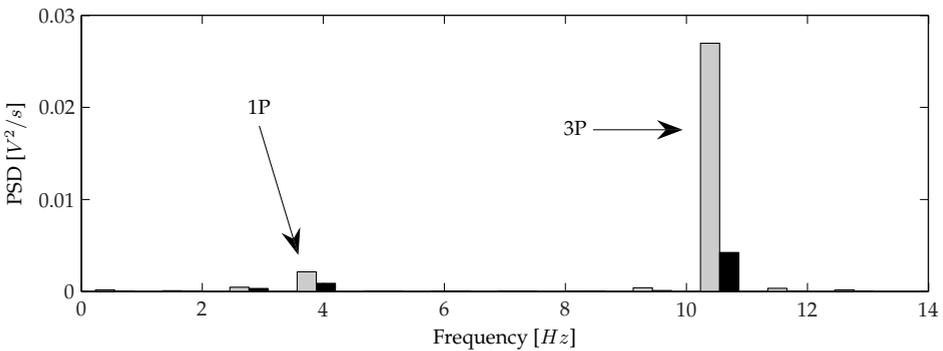


Figure 2.13: The PSD of the measured strains with (black) and without (gray) feedback control.

PZT sensor with and without control are given. In this figure we see that without control the 3P peak has a much larger amplification than the 1P peak. This is caused by the fact that the 3P peak is close to the first eigenfrequency. This can also be seen in Figure 2.8 where we see a larger gain at the 3P frequency than at the 1P frequency. If we compare the feedback results with the results without feedback we can conclude that a significant reduction in the standard deviation is obtained at the 1P (37%) and 3P (55%) frequencies.

2.5 Conclusions

In this chapter we developed an experimental model and a theoretical model of a 'smart' rotor blade. We scaled down the dynamics of a representative wind turbine blade using the reduced frequency and applied two trailing edge flaps as additional control devices. In the root, strain sensors are added to facilitate feedback control. The pitch system is used to impose realistic disturbances. It is shown that, when the disturbance is known, perfect cancellation can be realized; however, due to actuator saturation the reduction in the amplitude that was reached was limited to 90%. In real life the disturbance is not known and feedback control is required. For this situation a model was required and we used experimental modeling to validate the theoretical model and designed the feedback controller using the loop shaping technique. Furthermore, we showed that, with a relatively simple model, the dynamics can be represented with a sufficient amount of detail to design feedback controllers. The feedback controllers have been tested on the experimental setup for different cases. With a sinusoidal disturbance we showed that we can reduce the amplitudes by 90% at the first eigenfrequency of the blade. In the second feedback case a step was applied on the pitch, to mimic a gust, and it was shown that the oscillation corresponding to the first mode was almost completely removed. In the last feedback case a noise signal with a representative turbulence spectrum was applied and it was shown, in the PSD, that the 1P disturbance was reduced by 37% while the 3P frequency was reduced by 55%. The feasibility of the 'smart' rotor concept was demonstrated on this non-rotating experiment.

Part II

Fundamental part

Subspace identification of MIMO LPV systems using a periodic scheduling sequence

In this chapter we present a novel algorithm to identify LPV systems with affine parameter dependence. This algorithm is applicable for data generated in either open or closed-loop. The key assumption is that there is structure in the scheduling sequence; it is periodic or almost periodic. This allows us to use methods from LTI subspace identification to determine the column space of the periodic observability matrices and the corresponding periodic state sequences. It is shown that the crucial step in determining the original LPV system is to ensure that the obtained observability matrices and state sequences are defined with respect to the same global state basis. Once the LPV model has been identified, it is valid for arbitrary scheduling sequences as well.

3.1 Introduction

Linear Parameter-Varying (LPV) systems, already introduced in Chapter 1, are a particular class of nonlinear systems which have attracted considerable attention in recent years. Although we motivated the use of LPV system identification from a wind energy perspective, LPV system theory can be applied to a much wider field of applications. Recently, a number of industrial applications of LPV systems were published:

- compressors ([Giarre et al. 2006](#)),
- wind turbines ([Bianchi et al. 2007](#); [Østergaard 2008](#)),
- aerospace applications ([Barker and Balas 2000](#)),

- biomedical applications (Takahashi and Massaquoi 2007),
- motion platforms (Groot Wassink et al. 2005; Wijnheijmer et al. 2006),
- web server (Qin and Wang 2007; Tanelli et al. 2008),
- etc.

Earlier we motivated the need for efficient state-space LPV system identification algorithms in wind energy. Similar arguments can be used to motivate LPV system identification for the applications listed above. In this thesis we contribute to the development of system identification by introducing two novel subspace based identification frameworks to identify LPV systems under open and closed-loop conditions. In this chapter we develop a novel identification framework for LPV systems with a particular scheduling sequence; periodic. In the next chapter we address LPV systems with arbitrary scheduling as well.

Recently, a number of papers appeared where the structure of the scheduling sequence is exploited; it turns out that if the scheduling is periodic (Felici et al. 2007b; van Wingerden et al. 2008a), piecewise constant (Verdult and Verhaegen 2004; van Wingerden et al. 2007; van Wingerden and Verhaegen 2007), or white noise (Favoreel et al. 1999; Santos et al. 2005, 2006), well-established LTI subspace techniques can be used to identify LPV or bilinear systems.

For Linear Time-Varying (LTV) systems it is well known that ensemble identification can be used to obtain accurate models (MacNeil et al. 1992). In these algorithms the system undergoes the same time-variation multiple times and LTI identification techniques can be employed to estimate a set of models. If subspace algorithms are considered, multiple models are obtained corresponding to a certain window of the time-variation (stationary sequence models). It is well known that all these stationary models are identified in a different state basis and consequently, if the order of appearance of the time-varying dynamics changes, the LTV model is no longer valid (Verhaegen and Yu 1995). For LPV systems the time-variation is given by the scheduling sequence and every stationary sequence model is related to a stationary scheduling sequence. The crucial step in the identification framework presented in this chapter is that we transform all the stationary sequence models to the same global state basis, which makes it possible to reconstruct the LPV system matrices.

This chapter is set up as follows. We start in Section 3.2 with the problem formulation for this framework where we assume periodic scheduling. In Section 3.3 it will be shown how LTI subspace algorithms can be used to construct the observability matrices and the state sequences of the so-called stationary sequence models. These observability matrices and state sequences are all identified in a different state basis. In Section 3.4 we show how these observability matrices and state sequences can be transformed to the same state basis. In Section 3.5 we show how to circumvent the ‘curse of dimensionality’. In Section 3.6 the result of the intersection problem is used to transform all the stationary models to the same global state basis and we estimate the LPV system matrices. In the following section, Section 3.7, we discuss extensions of the proposed algorithm with respect to

the model structure and the structure in the scheduling sequence. In Section 3.8 two simple simulation studies are performed while in Section 3.9 the algorithm is applied on the rotational dynamics of a wind turbine. We end this chapter with our conclusions concerning periodic LPV identification.

Some results published in this chapter are published elsewhere, see Felici et al. (2007b); van Wingerden et al. (2007); van Wingerden and Verhaegen (2007); van Wingerden et al. (2008a).

3.2 Problem formulation and assumptions

In this section we present the model structure that we consider in this chapter. Furthermore, some assumptions are listed and some notation is introduced.

3.2.1 Problem formulation

For the derivation of the algorithm we consider the following LPV system:

$$x_{k+1} = \sum_{i=1}^m \mu_k^{(i)} \left(A^{(i)} x_k + B^{(i)} u_k \right) + K e_k, \quad (3.1a)$$

$$y_k = \sum_{i=1}^m \mu_k^{(i)} \left(C^{(i)} x_k + D^{(i)} u_k \right) + e_k, \quad (3.1b)$$

where $x_k \in \mathbb{R}^n$, $u_k \in \mathbb{R}^r$, $y_k \in \mathbb{R}^\ell$, are the state, input, and output vectors. $e_k \in \mathbb{R}^\ell$ denotes the zero-mean white innovation process. The matrices $A^{(i)} \in \mathbb{R}^{n \times n}$, $B^{(i)} \in \mathbb{R}^{n \times r}$, $C^{(i)} \in \mathbb{R}^{\ell \times n}$, $D^{(i)} \in \mathbb{R}^{\ell \times r}$, $K \in \mathbb{R}^{n \times \ell}$ are the local system, input, output, direct feedthrough, and the observer matrices; and $\mu_k^{(i)} \in \mathbb{R}$ the local weights. The index m is referred to as the number of local models or scheduling parameters. Note that the system, input, output, and direct feedthrough matrices depend linearly on the time-varying scheduling vector. The time-varying system matrix is now given by:

$$A_k = \sum_{i=1}^m \mu_k^{(i)} A^{(i)},$$

and a similar thing can be done for the other system matrices. We can rewrite (3.1a)-(3.1b) in the predictor form as:¹

$$x_{k+1} = \sum_{i=1}^m \mu_k^{(i)} \left(\tilde{A}^{(i)} x_k + \tilde{B}^{(i)} u_k \right) + K y_k, \quad (3.2a)$$

$$y_k = \sum_{i=1}^m \mu_k^{(i)} \left(C^{(i)} x_k + D^{(i)} u_k \right) + e_k, \quad (3.2b)$$

¹Observe that if K is allowed to be parameter-dependent we have for example the product between $K^{(i)} C^{(j)}$, which significantly makes the notation for LPV system identification more cumbersome as we will discuss later.

with:

$$\tilde{A}^{(i)} = A^{(i)} - KC^{(i)}, \quad \tilde{B}^{(i)} = B^{(i)} - KD^{(i)}.$$

It is well-known that an invertible linear transformation of the state does not change the input-output behavior of a state-space system. Therefore, we can only determine the system matrices up to a similarity transformation $T \in \mathbb{R}^{n \times n}$: $T^{-1}A^{(i)}T, T^{-1}B^{(i)}, T^{-1}K, C^{(i)}T$, and $D^{(i)}$.

The identification problem can now be formulated as:

Problem Description 3.1 (LPV system identification)

Given the input sequence u_k , the output sequence y_k , and the scheduling sequence μ_k over a time interval $k = \{0, \dots, N - 1\}$; find, if they exist, the LPV system matrices $A^{(i)}, B^{(i)}, C^{(i)}, D^{(i)}$, and K up to a global similarity transformation.

3.2.2 Assumptions and notation

The scheduling sequence:

$$\mu_k = \left[\mu_k^{(1)}, \dots, \mu_k^{(m)} \right]^T,$$

is assumed to be known and periodic with period π , defined over Π periods:

$$\mu_k = \mu_{k+\tau\pi} \quad \forall \tau \in \{1, 2, \dots, \Pi - 1\}, \quad \forall k \in \{0, 1, \dots, \pi - 1\}.$$

For persistency of excitation it is also required that the scheduling sequence satisfies the following relation:

$$\text{rank} \left(\left[\mu_0, \mu_1, \dots, \mu_{\pi\Pi-p-f} \right] \right) = m,$$

and $\pi\Pi - p - f + 1 > m$ where p and f are referred to as the past and future window length, respectively. Furthermore, we assume that the feedback problem is well-posed. So, there is either a delay in the feedback controller or in the system.

We also define the transition matrix for discrete time time-varying systems and this is given by (Rugh 1996):

$$\phi_{j,k} = \tilde{A}_{k+j-1} \cdots \tilde{A}_{k+1} \tilde{A}_k, \quad (3.3)$$

and for simplicity we define:

$$\overline{B}^{(i)} = \left[\tilde{B}^{(i)}, K \right].$$

Similar as in Jansson (2005); Chiuso (2007) the past and future windows are used

to define the following stacked vector:

$$\bar{z}_k^p = \begin{bmatrix} z_k \\ z_{k+1} \\ \vdots \\ z_{k+p-1} \end{bmatrix},$$

with $z_k = [u_k^T, y_k^T]^T$ and we define the time-varying observability matrix:

$$\tilde{\Gamma}_k^f = \begin{bmatrix} C_{k+p} \\ C_{k+p+1}\phi_{1,k+p} \\ \vdots \\ C_{k+p+f-1}\phi_{f-1,k+p} \end{bmatrix} \in \mathbb{R}^{f\ell \times n}, \quad (3.4)$$

for which we assume that it has full column rank for all k , which is equivalent to requiring that the system is observable on all the intervals of length f according to the condition for observability of LPV systems (Rugh 1996). Note that the fulfillment of this requirement depends both on the system matrices and on the scheduling sequence. To illustrate this time-varying observability matrix see the following example:

Example 3.1 ($\tilde{\Gamma}_k^f$)

For $m = 2$, $f = 3$, and $k = p = 0$ one obtains:

$$\tilde{\Gamma}_0^3 = \begin{bmatrix} C^{(1)} \otimes \mu_0^{(1)} + C^{(2)} \otimes \mu_0^{(2)} \\ \left(C^{(1)} \otimes \mu_1^{(1)} + C^{(2)} \otimes \mu_1^{(2)} \right) \left(\tilde{A}^{(1)} \otimes \mu_0^{(1)} + \tilde{A}^{(2)} \otimes \mu_0^{(2)} \right) \\ \left(C^{(1)} \otimes \mu_2^{(1)} + C^{(2)} \otimes \mu_2^{(2)} \right) \left(\tilde{A}^{(1)} \otimes \mu_1^{(1)} + \tilde{A}^{(2)} \otimes \mu_1^{(2)} \right) \cdots \\ \cdots \left(\tilde{A}^{(1)} \otimes \mu_0^{(1)} + \tilde{A}^{(2)} \otimes \mu_0^{(2)} \right) \end{bmatrix}.$$

The LPV controllability matrix $\bar{\mathcal{K}}_k^p \in \mathbb{R}^{n \times (r+\ell)p}$ is defined as:

$$\bar{\mathcal{K}}_k^p = [\phi_{p-1,k+1}\check{B}_k, \quad \cdots, \quad \phi_{1,k+p-1}\check{B}_{k+p-2}, \quad \check{B}_{k+p-1}], \quad (3.5)$$

with $\check{B}_k = [\tilde{B}_k, K]$. It is assumed that $\bar{\mathcal{K}}_k^p$ has full row rank, n , for all k , which can be seen as a condition for reachability of LPV systems. For a factorization of this particular matrix we refer to Section 4.3.

In the next section we will use these assumptions to extrapolate a number of LTI predictor-based subspace algorithms to periodic systems.

3.3 Periodic predictor-based subspace identification

It is well known that the projector type of subspace algorithms (*e.g.* MOESP (Verhaegen and Dewilde 1992) and N4SID (Van Overschee and De Moor 1996)) give biased estimates if the data is generated under closed-loop conditions. The main reason for the bias is the constraint that the noise, e_k , and the input, u_k , should be uncorrelated. This assumption is clearly violated if there is a feedback loop present (as clearly explained in Ljung and McKelvey (1996)). Predictor-based subspace identification methods (*e.g.* PBSID (Chiuso and Picci 2005) and SSARX (Jansson 2005)) do not suffer from this drawback if either the direct feedthrough of the system or the feedback controller is zero. These methods use high-order ARX models to remove the correlation between the input and noise sequence. In this chapter, we extend these predictor-based algorithms to the periodic situation. The reason to use only the predictor framework is threefold:

1. The projector based algorithms can be made periodic in a similar way, see Felici et al. (2007b).
2. The predictor-based methods can cope with data generated in an open and closed-loop setting.
3. Predictor-based methods can be applied to LPV system identification where arbitrary scheduling is used (see next chapter).

In predictor-based subspace identification methods we use a number of predictors to estimate the Markov parameters. From the estimates the product between the extended observability matrix and state sequence can be constructed. From this point, an SVD can be used to estimate the observability matrix. However, because we work in the innovation framework an estimate of the observability matrix is not enough to determine the system matrices and that is why we also estimate the state sequence. In this section we show how two of these methods can be extended to the periodic setting².

3.3.1 Predictors

The state at time instance $k+p$ is a function of the known past inputs and outputs, \bar{z}_k^p , and the initial state, x_k . Using (3.2a) the state x_{k+p} is given by:

$$x_{k+p} = \phi_{p,k} x_k + \bar{\mathcal{K}}_k^p \bar{z}_k^p, \quad (3.6)$$

where $\bar{\mathcal{K}}_k^p$ are matrices depending on \tilde{A}_k and \tilde{B}_k . The key approximation in this algorithm is that we assume that $\phi_{j,k} \approx 0$ for all $j \geq p$. It can be shown that if the system in (3.2a)-(3.2b) is uniformly exponentially stable the approximation error can be made arbitrarily small (Knudsen 2001; Verdult and Verhaegen 2002).

²When the reader is not familiar with PBSID we would like to refer to Appendix B, where a short summary of the algorithm is given.

The same approximation is made in the N4SID algorithm (Van Overschee and De Moor 1996; Verhaegen and Verdult 2007). With this approximation, (3.6) can be substituted in (3.2b) to obtain the following set of equations:

$$\left\{ \begin{array}{l} y_{k+p} \approx C_{k+p} \overline{\mathcal{K}}_k^p \overline{z}_k^p + D_{k+p} u_{k+p} + e_{k+p} \\ y_{k+p+1} \approx C_{k+p+1} \overline{\mathcal{K}}_k^{p+1} \overline{z}_k^{p+1} + D_{k+p+1} u_{k+p+1} + e_{k+p+1} \\ \vdots \\ y_{k+p+f-1} \approx C_{k+p+f-1} \overline{\mathcal{K}}_k^{p+f-1} \overline{z}_k^{p+f-1} + D_{k+p+f-1} u_{k+p+f-1} + e_{k+p+f-1} \end{array} \right. ,$$

for all $k \in \{0, \dots, \Pi\pi - p - f\}$. This set of equations looks really cumbersome. However, at this point we are going to exploit the periodic nature of the scheduling sequence. In this case we can show that this set of equations can be viewed at as a set of linear multi-variable ARX regression problems. To show this, we define the following data matrices:

$$\overline{\mathcal{Z}}_k^p = \left[\overline{z}_k^p, \overline{z}_{k+\pi}^p, \overline{z}_{k+2\pi}^p, \dots, \overline{z}_{k+(\Pi-1)\pi}^p \right], \quad (3.7)$$

and we define:

$$\overline{Y}_k = \left[y_{k+p}, y_{k+p+\pi}, y_{k+p+2\pi}, \dots, y_{k+p+(\Pi-1)\pi} \right], \quad (3.8)$$

and a similar thing can be done for \overline{U}_k and \overline{X}_k . At this point it is important to stress that if the periodic state sequences are known, we can also construct the whole state sequence, X . Now we use the fact that we have periodic scheduling so this implies that:

$$\begin{aligned} C_k &= C_{k+\pi}, \\ \mathcal{K}_k^{p+i} &= \mathcal{K}_{k+\pi}^{p+i}, \\ D_k &= D_{k+\pi}, \end{aligned}$$

for all $k \in \{0, \dots, \pi(\Pi - 1)\}$. With other words the system matrices are periodic and if we batch the data accordingly we obtain π multi-variable ARX regression problems. With the data matrices defined periodically in (3.7)-(3.8) and this observation we can formulate the following linear problem:

$$\min_{C_{k+p+i} \overline{\mathcal{K}}_k^{p+i}, D_{k+p+i}} \|\overline{Y}_{k+i} - C_{k+p+i} \overline{\mathcal{K}}_k^{p+i} \overline{\mathcal{Z}}_k^{p+i} - D_{k+p+i} \overline{U}_{k+i}\|_F^2, \quad (3.9)$$

for all $i \in \{0, \dots, f - 1\}$ and for all $k \in \{0, \dots, \pi - 1\}$. The notation: $\|\cdot\|_F$ represents the Frobenius norm (Golub and Loan 1996), while also the condition should hold that the following matrix has full row rank:

$$\begin{bmatrix} \overline{\mathcal{Z}}_k^{p+i} \\ \overline{U}_{k+i} \end{bmatrix}.$$

For finite p the solution of this linear problem will be biased due the approximation made that $\phi_{j,k} \approx 0$ for all $j \geq p$. In the LTI literature a number of papers appeared that studied the effect of the window size and although they proved the asymptotic properties of the algorithms (if $p \rightarrow \infty$ the bias disappears) it is hard to quantify the effect for finite p (Knudsen 2001; Chiuso and Picci 2005; Chiuso 2007). Furthermore, the linear problem stated in (3.9) looks rather cumbersome with all the indices, but for given i and k we end up with a really simple linear problem. In the next subsection we introduce two algorithms that make use of the obtained estimates to construct the extended observability matrix times the state sequence.

3.3.2 Extended observability times controllability matrix

The product $\overline{\mathcal{K}}_k^p \overline{\mathcal{Z}}_k^p$, which by definition represents the periodic state sequence \overline{X}_k , can not directly be estimated. In the LTI literature it is common practice to use the estimates of similar matrices as $C_{k+p+i} \overline{\mathcal{K}}_k^{p+i}$ to construct an estimate of the extended observability matrix times the extended controllability matrix, $\tilde{\Gamma}_k^f \overline{\mathcal{K}}_k^p$. The structure of this matrix is demonstrated by the following example:

Example 3.2 (Observability matrix times controllability matrix)

For $k = 0$, $\pi = 6$, and $p = f = 3$ one obtains:

$$\tilde{\Gamma}_0^3 \overline{\mathcal{K}}_0^3 = \begin{bmatrix} C_3 \tilde{A}_2 \tilde{A}_1 \overline{B}_0 & C_3 \tilde{A}_2 \overline{B}_1 & C_3 \overline{B}_2 \\ C_4 \tilde{A}_3 \tilde{A}_2 \tilde{A}_1 \overline{B}_0 & C_4 \tilde{A}_3 \tilde{A}_2 \overline{B}_1 & C_4 \tilde{A}_3 \overline{B}_2 \\ C_5 \tilde{A}_4 \tilde{A}_3 \tilde{A}_2 \tilde{A}_1 \overline{B}_0 & C_5 \tilde{A}_4 \tilde{A}_3 \tilde{A}_2 \overline{B}_1 & C_5 \tilde{A}_4 \tilde{A}_3 \overline{B}_2 \end{bmatrix}.$$

The algorithms PBSID and PBSID_{opt} differ from each other by the construction of the matrix $\tilde{\Gamma}_k^f \overline{\mathcal{K}}_k^p$. For PBSID the estimate of the matrix is given by:

$$\widehat{\tilde{\Gamma}_k^f \overline{\mathcal{K}}_k^p} = \begin{bmatrix} \widehat{C_{k+p} \overline{\mathcal{K}}_k^p}(:, 1 : (r + \ell)p) \\ \widehat{C_{k+p+1} \overline{\mathcal{K}}_k^{p+1}}(:, 1 : (r + \ell)p) \\ \vdots \\ \widehat{C_{k+p+f-1} \overline{\mathcal{K}}_k^{p+f-1}}(:, 1 : (r + \ell)p) \end{bmatrix}, \quad (3.10)$$

for all $k \in \{0, \dots, \pi - 1\}$, where $(:, 1 : (r + \ell)p)$ is Matlab notation for taking the first $(r + \ell)p$ columns of the matrix. Note that for constructing these matrices we have to solve $f \times \pi$ linear problems. In the case of the periodic-PBSID_{opt} algorithm we only have to solve π linear problems because for this particular solution we again exploit the approximation that $\phi_{j,k} \approx 0$ for all $j \geq p$ leading to an upper block

triangular realization of $\tilde{\Gamma}_k^f \overline{\mathcal{K}}_k^p$ that is given by (we assume $f = p$):

$$\widehat{\tilde{\Gamma}_k^f \overline{\mathcal{K}}_k^p} \approx \begin{bmatrix} \widehat{C_{k+p} \overline{\mathcal{K}}_k^p} \\ \left[\begin{array}{cc} 0_{\ell \times \ell+r} & \widehat{C_{k+p+1} \overline{\mathcal{K}}_{k+1}^p}(:, 1 : (p-1)(r+\ell)) \end{array} \right] \\ \vdots \\ \left[\begin{array}{cc} 0_{\ell \times (p-1)(\ell+r)} & \widehat{C_{k+p+f-1} \overline{\mathcal{K}}_{k+f-1}^p}(:, 1 : (r+\ell)) \end{array} \right] \end{bmatrix}. \quad (3.11)$$

To construct this matrix we only have to solve the set of linear equations given in (3.9) for $i = 0$ resulting in π linear problems. This is illustrated in the following example:

Example 3.3 (Periodic-PBSID_{opt} solution)

For $k = 0$ and $\pi = p = f = 3$ one obtains:

$$\begin{aligned} \tilde{\Gamma}_0^3 \overline{\mathcal{K}}_0^3 &= \begin{bmatrix} C_0 \tilde{A}_2 \tilde{A}_1 \overline{B}_0 & C_0 \tilde{A}_2 \overline{B}_1 & C_0 \overline{B}_2 \\ 0 & C_1 \tilde{A}_0 \tilde{A}_2 \overline{B}_1 & C_1 \tilde{A}_0 \overline{B}_2 \\ 0 & 0 & C_2 \tilde{A}_1 \tilde{A}_0 \overline{B}_2 \end{bmatrix}, \\ \tilde{\Gamma}_1^3 \overline{\mathcal{K}}_1^3 &= \begin{bmatrix} C_1 \tilde{A}_0 \tilde{A}_2 \overline{B}_1 & C_1 \tilde{A}_0 \overline{B}_2 & C_1 \overline{B}_0 \\ 0 & C_2 \tilde{A}_1 \tilde{A}_0 \overline{B}_2 & C_2 \tilde{A}_1 \overline{B}_0 \\ 0 & 0 & C_0 \tilde{A}_2 \tilde{A}_1 \overline{B}_0 \end{bmatrix}, \\ \tilde{\Gamma}_2^3 \overline{\mathcal{K}}_2^3 &= \begin{bmatrix} C_2 \tilde{A}_1 \tilde{A}_0 \overline{B}_2 & C_2 \tilde{A}_1 \overline{B}_0 & C_2 \overline{B}_1 \\ 0 & C_0 \tilde{A}_2 \tilde{A}_1 \overline{B}_0 & C_0 \tilde{A}_2 \overline{B}_1 \\ 0 & 0 & C_1 \tilde{A}_0 \tilde{A}_2 \overline{B}_1 \end{bmatrix}. \end{aligned}$$

Observe that from all the first rows, the 3 solutions of (3.9) for $i=0$, we can construct the three matrices.

By computing a Singular Value Decomposition (SVD) of $\widehat{\tilde{\Gamma}_k^f \overline{\mathcal{K}}_k^p \overline{\mathcal{Z}}_k^p}$, constructed by the periodic-PBSID_{opt} or the periodic-PBSID algorithm, we can estimate the state sequence and the order of the π stationary models. We will use the following SVD:

$$\widehat{\tilde{\Gamma}_k^f \overline{\mathcal{K}}_k^p \overline{\mathcal{Z}}_k^p} = [\mathcal{U}_k \quad \mathcal{U}_{k,\perp}] \begin{bmatrix} \Sigma_{k,n} & 0 \\ 0 & \Sigma_k \end{bmatrix} \begin{bmatrix} \mathcal{V}_k \\ \mathcal{V}_{k,\perp} \end{bmatrix},$$

where $\Sigma_{k,n}$ is the diagonal matrix containing the n largest singular values and \mathcal{V}_k is the corresponding row space. Note that we can find the largest singular values by detecting a gap between the singular values (Verhaegen and Verdult 2007). The periodic state and periodic extended observability matrices are now estimated by:

$$\widehat{\mathcal{X}}_k = T_k^{-1} \Sigma_{k,n} \mathcal{V}_n, \quad (3.12a)$$

$$\tilde{\Gamma}_k^f = \mathcal{U}_k T_k. \quad (3.12b)$$

This can be done for all $k = \{0, \dots, \pi - 1\}$, obtaining different column spaces and periodic state sequences. The similarity transformations $T_{k,}$ similar as in periodic PO-MOESP, will also be different at each time, which can be interpreted as each of the state sequences \widehat{X}_k being in a different basis. We would like to stress that if the stationary models are in the same state basis the problem is solved. It was already shown in [Lovera \(1997\)](#) that if the state and the scheduling is known the problem reduces to a linear estimation problem.

3.4 A common state basis

In the previous section we showed how two predictor-based subspace identification methods can be extended to periodic systems. We found a set of observability matrices and state sequences, but all these observability matrices and state sequences are identified in a different state basis. In this section we show that with a special factorization we can combine all these observability matrices in a global state basis and this is the key to identify the LPV system matrices. The work presented in this section shows strong similarities with the work presented in [Felici et al. \(2007b\)](#).

3.4.1 Factorization of the extended observability matrix

In this section we define a fundamental factorization in which we separate the unknown system matrices from the known weighting sequence. The time-varying extended observability matrix can be factorized in a matrix containing only the scheduling terms and a constant matrix that depends only on the system matrices $\tilde{A}^{(i)}$, and $C^{(i)}$. Before we formulate this factorization in a lemma we have to introduce a number of definitions. We start with the following definition:

Definition 3.1 *We define the matrix:*

$$\mathcal{P}_j = \begin{bmatrix} \mathcal{P}_{j-1} \tilde{A}^{(1)} \\ \vdots \\ \mathcal{P}_{j-1} \tilde{A}^{(m)} \end{bmatrix},$$

with

$$\mathcal{P}_1 = \begin{bmatrix} C^{(1)} \\ \vdots \\ C^{(m)} \end{bmatrix}.$$

To illustrate this definition see the following example

Example 3.4 (\mathcal{P}_j)

For $m=2$ one obtains:

$$\mathcal{P}_1 = \begin{bmatrix} C^{(1)} \\ C^{(2)} \end{bmatrix}, \quad \mathcal{P}_2 = \begin{bmatrix} C^{(1)}\tilde{A}^{(1)} \\ C^{(2)}\tilde{A}^{(1)} \\ C^{(1)}\tilde{A}^{(2)} \\ C^{(2)}\tilde{A}^{(2)} \end{bmatrix}, \quad \mathcal{P}_2 = \begin{bmatrix} C^{(1)}\tilde{A}^{(1)}\tilde{A}^{(1)} \\ C^{(2)}\tilde{A}^{(1)}\tilde{A}^{(1)} \\ C^{(1)}\tilde{A}^{(2)}\tilde{A}^{(1)} \\ C^{(2)}\tilde{A}^{(2)}\tilde{A}^{(1)} \\ C^{(1)}\tilde{A}^{(1)}\tilde{A}^{(2)} \\ C^{(2)}\tilde{A}^{(1)}\tilde{A}^{(2)} \\ C^{(1)}\tilde{A}^{(2)}\tilde{A}^{(2)} \\ C^{(2)}\tilde{A}^{(2)}\tilde{A}^{(2)} \end{bmatrix}.$$

The amount of block-rows grows exponentially as m^j .

This operator is used in the expression for the extended observability matrix in the following lemma.

Lemma 3.1 (Factorization of the LPV observability matrix) Let $\tilde{\Gamma}_k^f$ be defined as in (3.4), and let

$$S = \begin{bmatrix} \mathcal{P}_1 \\ \mathcal{P}_2 \\ \vdots \\ \mathcal{P}_f \end{bmatrix} \in \mathbb{R}^{q \times n}, \quad (3.13)$$

and let $M_k \in \mathbb{R}^{f\ell \times q}$ be given as

$$M_k = \begin{bmatrix} \mu_{k+p}^T & 0 & \cdots & 0 \\ 0 & \mu_{k+p}^T \otimes \mu_{k+p+1}^T & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & \mu_{k+p}^T \otimes \cdots \otimes \mu_{k+p+f-1}^T \end{bmatrix} \otimes I_\ell, \quad (3.14)$$

where \otimes denotes the Kronecker product (Brewer 1978) and now it holds, for all k , that

$$\tilde{\Gamma}_k^f = M_k S. \quad (3.15)$$

Proof: The proof is by straightforward computation. □

Note that the number of rows of S (columns of M_k), denoted by q , increases exponentially with f according to the relation $q = \sum_{j=1}^f \ell m^j$. In the next example we illustrate the proposed factorization:

Example 3.5 (M_k and S)

For $m = 2$, $\ell = 1$, and $p = k = 0$ one obtains:

$$M_0 = \begin{bmatrix} \mu_0^{(1)} & 0 & 0 \\ \mu_0^{(2)} & 0 & 0 \\ 0 & \mu_0^{(1)} \mu_1^{(1)} & 0 \\ 0 & \mu_0^{(1)} \mu_1^{(2)} & 0 \\ 0 & \mu_0^{(2)} \mu_1^{(1)} & 0 \\ 0 & \mu_0^{(2)} \mu_1^{(2)} & 0 \\ 0 & 0 & \mu_0^{(1)} \mu_1^{(1)} \mu_2^{(1)} \\ 0 & 0 & \mu_0^{(1)} \mu_1^{(1)} \mu_2^{(2)} \\ 0 & 0 & \mu_0^{(1)} \mu_1^{(2)} \mu_2^{(1)} \\ 0 & 0 & \mu_0^{(1)} \mu_1^{(2)} \mu_2^{(2)} \\ 0 & 0 & \mu_0^{(2)} \mu_1^{(1)} \mu_2^{(1)} \\ 0 & 0 & \mu_0^{(2)} \mu_1^{(1)} \mu_2^{(2)} \\ 0 & 0 & \mu_0^{(2)} \mu_1^{(2)} \mu_2^{(1)} \\ 0 & 0 & \mu_0^{(2)} \mu_1^{(2)} \mu_2^{(2)} \end{bmatrix}^T, \quad S = \begin{bmatrix} C^{(1)} \\ C^{(2)} \\ C^{(1)} \tilde{A}^{(1)} \\ C^{(2)} \tilde{A}^{(1)} \\ C^{(1)} \tilde{A}^{(2)} \\ C^{(2)} \tilde{A}^{(2)} \\ C^{(1)} \tilde{A}^{(1)} \tilde{A}^{(1)} \\ C^{(2)} \tilde{A}^{(1)} \tilde{A}^{(1)} \\ C^{(1)} \tilde{A}^{(2)} \tilde{A}^{(1)} \\ C^{(2)} \tilde{A}^{(2)} \tilde{A}^{(1)} \\ C^{(1)} \tilde{A}^{(1)} \tilde{A}^{(2)} \\ C^{(2)} \tilde{A}^{(1)} \tilde{A}^{(2)} \\ C^{(1)} \tilde{A}^{(2)} \tilde{A}^{(2)} \\ C^{(2)} \tilde{A}^{(2)} \tilde{A}^{(2)} \end{bmatrix},$$

multiplication of M_0 with S results in the matrix given in Example 3.1.

3.4.2 Determination of the transformation matrices

We now formulate a theorem to determine the transformation matrices T_k in (3.12a)-(3.12b). Since the algorithms presented in the previous section are unbiased estimators for the observability matrices under specific conditions, $p \rightarrow \infty$ and $\Pi \rightarrow \infty$, we present the theorem for this case.

Theorem 3.1 (Transformation into common basis) Let \mathcal{U}_k be equal to $\tilde{\Gamma}_k^f$ up to a similarity transformation, obtained from (3.12b) in the noiseless and unbiased case, such that:

$$\mathcal{U}_k T_k = \tilde{\Gamma}_k^f, \quad (3.16)$$

Define

$$\tilde{\mathcal{U}} = \begin{bmatrix} \mathcal{U}_0 & 0 & \cdots & 0 \\ 0 & \mathcal{U}_1 & \cdots & 0 \\ \vdots & \vdots & \ddots & 0 \\ 0 & 0 & \cdots & \mathcal{U}_{\pi-1} \end{bmatrix} \in \mathbb{R}^{\ell f \pi \times n \pi}, \quad (3.17a)$$

$$\tilde{\Gamma} = \begin{bmatrix} \tilde{\Gamma}_0^f & 0 & \cdots & 0 \\ 0 & \tilde{\Gamma}_1^f & \cdots & 0 \\ \vdots & \vdots & \ddots & 0 \\ 0 & 0 & \cdots & \tilde{\Gamma}_{\pi-1}^f \end{bmatrix} \in \mathbb{R}^{\ell f \pi \times n \pi}, \quad (3.17b)$$

$$\hat{T} = [T_0^T, T_1^T, \dots, T_{\pi-1}^T]^T \in \mathbb{R}^{n\pi \times n}, \quad (3.17c)$$

$$\tilde{M} = [M_0^T, M_1^T, \dots, M_{\pi-1}^T]^T \in \mathbb{R}^{\ell f \pi \times q}, \quad (3.17d)$$

where M_k is defined in (3.14). Also, define \tilde{T} equal to \hat{T} up to an unknown square invertible matrix $T \in \mathbb{R}^{n \times n}$.

$$\tilde{T} = \hat{T}T. \quad (3.18)$$

Finally, let it hold that

$$\text{rank} \left(\begin{bmatrix} \tilde{\Gamma} & \tilde{M} \end{bmatrix} \right) = \text{rank}(\tilde{\Gamma}) + \text{rank}(\tilde{M}) - n. \quad (3.19)$$

Then the system of equations:

$$\tilde{U}\phi = \tilde{M}\psi, \quad (3.20a)$$

$$\phi^T \tilde{U}^T \tilde{U} \phi = I_n, \quad (3.20b)$$

$$\psi^T \tilde{M}^T \tilde{M} \psi = I_n, \quad (3.20c)$$

with unknowns ϕ and ψ has a solution $\phi \in \mathbb{R}^{n\pi \times n}$, which is unique up to a square orthogonal matrix $Q \in \mathbb{R}^{n \times n}$. Moreover, there exists an invertible $T \in \mathbb{R}^{n \times n}$ such that $\phi = \tilde{T}T = \hat{T}$.

Proof: It is easy to see that $\tilde{U}\hat{T} = \tilde{\Gamma} = \tilde{M}S$. That implies that the combination of \hat{T} and S is a solution of the intersection problem given in (3.20a). However, with a post-multiplication with an invertible square matrix T we still satisfy (3.20a). This degree of freedom represents in which basis the system is identified. Now we have to show that we find a solution that is unique up to this matrix T . To have a unique solution the intersection should be n dimensional. This directly results in condition (3.19). So, with the intersection problem defined in (3.20a) and the condition stated in (3.19) we find the similarity matrices up to a global similarity transformation T . \square

Condition (3.19) is essential for this theorem. It is a requirement both on the system and on the scheduling sequence, through $\tilde{\Gamma}$ and \tilde{M} , respectively. It might be true that certain systems have an $\tilde{\Gamma}$ such that the condition is never satisfied. This might pose a restriction on which LPV systems can be identified in this manner. On the other side, it can be seen that in some cases an inappropriate choice of the scheduling sequence might give an \tilde{M} such that (3.19) is violated.

The system of equations (3.20a) – (3.20c) can not be solved exactly in the presence of noise. For this reason, we replace (3.20a) by the minimization problem $\min_{\phi, \psi} \|\tilde{U}\phi - \tilde{M}\psi\|_F^2$. Under these modifications, these problems are known as Canonical Correlation Analysis (CCA) problems (Van Overschee and De Moor 1996). Such problems can be interpreted as attempting to determine the intersection between the column spaces of two known matrices, or as a generalization of angles between subspaces. The constraints (3.20b), (3.20c) ensure orthogonality of the projections. Different methods for solution exist on which we will not elaborate further (Seber 1984; Krzanowski 1988). It is however important to note that

Table 3.1: Total number of rows in the matrix S for $r = l = 1$

	m=1	m=2	m=3	m=4	m=5
f=2	2	6	12	20	30
f=4	4	30	120	340	155
f=6	6	126	1092	5460	19530
f=8	8	510	9840	87380	488280

when the dimension of the target space (the number of rows in (3.20a)) is small compared to the dimension of the signal space (the number of columns of \tilde{U} and \tilde{M}), CCA methods may obtain solution directions which are determined mainly by noise, which in our case is present because $U_k T_k \approx \tilde{\Gamma}_k^f$. This is also known as overfitting. A method to decrease the variance of the solutions, at the cost of introducing a bias, is to employ regularization as presented in [De Bie and De Moor \(2003\)](#). It can be shown that a solution of (3.20a) - (3.20c) can be found by solving the regularized generalized eigenvalue problem:

$$\begin{bmatrix} 0 & \tilde{U}^T \tilde{M} \\ \tilde{M}^T \tilde{U} & 0 \end{bmatrix} \begin{bmatrix} \phi \\ \psi \end{bmatrix} = \lambda \begin{bmatrix} \tilde{U}^T \tilde{U} + \nu_1 I_{n\pi} & 0 \\ 0 & \tilde{M}^T \tilde{M} + \nu_2 I_q \end{bmatrix} \begin{bmatrix} \phi \\ \psi \end{bmatrix},$$

where λ is referred to as the canonical correlation coefficient and will be between 0 and 1, indicating the closeness of the subspaces. The generalized eigenvectors corresponding to the highest canonical coefficients are chosen as solutions. The regularization parameters ν_1, ν_2 should be chosen appropriately as will be discussed in Section 3.8.2.

3.5 Kernel method

A drawback of (3.20a) is that the matrix ψ has a large number of rows, q , which is demonstrated in Table 3.1. Since, we are not particularly interested in ψ , it is possible to determine ϕ while avoiding computations in this high-dimensional space. A similar method is used in [Verdult and Verhaegen \(2005\)](#) to alleviate problems due to large matrix dimensions. The basic intuition is that the column space of \tilde{M} is equal to that of $\tilde{M}\tilde{M}^T$. Not only does the latter matrix have fewer columns, it is also rather straightforward to compute. The result is formulated in the following theorem.

Theorem 3.2 (More efficient determination of T_k) *Let the same definitions and conditions as Theorem 3.1 hold. Additionally, let $\Theta \in \mathbb{R}^{\ell f\pi \times \ell f\pi}$ be defined as*

$$\Theta = \tilde{M}\tilde{M}^T = \begin{bmatrix} M_0 M_0^T & M_0 M_1^T & \cdots & M_0 M_{\pi-1}^T \\ M_1 M_0^T & M_1 M_1^T & \cdots & M_1 M_{\pi-1}^T \\ \vdots & \vdots & \ddots & \vdots \\ M_{\pi-1} M_0^T & M_{\pi-1} M_1^T & \cdots & M_{\pi-1} M_{\pi-1}^T \end{bmatrix}, \quad (3.21)$$

with M_k as defined in (3.14). Finally, let

$$\text{rank} \left(\begin{bmatrix} \tilde{\Gamma} & \Theta \end{bmatrix} \right) = \text{rank}(\tilde{\Gamma}) + \text{rank}(\Theta) - n. \quad (3.22)$$

Then the system of equations:

$$\tilde{U}\phi = \Theta\Upsilon, \quad (3.23a)$$

$$\phi^T \tilde{U}^T \tilde{U} \phi = I_n, \quad (3.23b)$$

$$\Upsilon^T \Theta^T \Theta \Upsilon = I_n, \quad (3.23c)$$

with unknowns ϕ and Υ has the same solution $\phi \in \mathbb{R}^{n\pi \times n}$ unique up to a square orthogonal matrix $Q \in \mathbb{R}^{n \times n}$, as Theorem 3.1.

Proof: The matrix ψ which solves (3.20a) can be written as $\psi = \tilde{M}^T \Upsilon + \psi_N$ where ψ_N is an arbitrary matrix in the null space of \tilde{M} : $\tilde{M}\psi_N = 0$, and $\tilde{M}^T \Upsilon$ lies in the row space of \tilde{M} . Since ψ_N does not influence the solution ϕ it may be disregarded, thus we can parameterize the solutions in the mentioned row space as $\psi = \tilde{M}^T \Upsilon$ with $\Upsilon \in \mathbb{R}^{f\pi \times n}$. Since $\tilde{M}\psi = \tilde{M}\tilde{M}^T \Upsilon = \Theta\Upsilon$, \tilde{T} may be equivalently obtained by (3.23a) – (3.23c). \square

Note that the resulting problem is much smaller since $\psi \in \mathbb{R}^{q \times n}$ and $\Upsilon \in \mathbb{R}^{\ell f \pi \times n}$ and for large $f, q \gg \ell f \pi$. Also, the matrix Θ can be constructed from the known model weights directly without having to compute the (large) matrices M_k , since:

$$M_{i-p} M_{j-p}^T = \begin{bmatrix} \mu_i^T \mu_j & 0 & \cdots & 0 \\ 0 & (\mu_i^T \mu_j)(\mu_{i+1}^T \mu_{j+1}) & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & \left((\mu_i^T \mu_j) \cdots (\mu_{i+f-2}^T \mu_{j+f-1}) \right) \end{bmatrix} \otimes I_\ell.$$

This kernel method makes it possible to solve the equations in a computationally efficient way.

3.6 Determination of the system matrices

In the previous sections we applied LTI subspace techniques to obtain the periodic state sequences up to an unknown similarity transformation (see (3.12a)). Followed by two sections how we can transform all these local periodic state sequences to the same global state basis. With the similarity transformations given in Theorem 3.1 and applying them on (3.12a) the whole state sequence is given in the same global state base, $T^{-1}X$. Given this state sequence the computation of the system matrices, up to a global similarity transformation, becomes trivial (see also Lovera (1997)) and can be done in two steps:

Compute $C^{(i)}$, $D^{(i)}$ and the noise sequence e_k With the estimated state, the input, the output, and the scheduling sequence given, (3.1b) reduces to a linear problem in the unknowns $C^{(i)}$, $D^{(i)}$, which can be formulated as a regression problem. The residual of the regression problem is e_k ($e_k = y_k - \sum_{i=1}^m \mu_k^{(i)} (C^{(i)} x_k + D^{(i)} u_k)$).

Compute $A^{(i)}$, $B^{(i)}$, and K With the estimated state, the input, the scheduling, and the noise sequence given, (3.1a) reduces to a linear problem in the unknowns $A^{(i)}$, $B^{(i)}$, and K , which can be formulated as a regression problem.

This ends the description of the algorithm. A brief algorithmic summary is given below:

Algorithm 3.1 (Periodic LPV-PBSID_{opt})

The algorithm can be summarized as follows:

1. Create the matrices \bar{Z}_k^p , \bar{U}_k , and \bar{Y}_k for all $k \in \{0, \dots, \pi - 1\}$ using (3.7) and (3.8),
2. Solve the linear problem (3.9) for $i = 0$ and for all $k \in \{0, \dots, \pi - 1\}$,
3. Construct $\tilde{\Gamma}_k^f \bar{\mathcal{K}}_k^p \bar{Z}_k^p$ for all $k \in \{0, \dots, \pi - 1\}$ using (3.11),
4. Compute the state sequences and observability matrices using (3.12a) and (3.12b) for all $k \in \{0, \dots, \pi - 1\}$,
5. Solve intersection problem using Theorem 3.1 or Theorem 3.2 and transform all the state sequences to the same basis,
6. With the estimated state, use the linear relations (3.1a)-(3.1b) to obtain the system matrices.

Algorithm 3.2 (Periodic LPV-PBSID)

The algorithm can be summarized as follows:

1. Create the matrices \bar{Z}_k^{p+i} , \bar{U}_{k+i} , and \bar{Y}_{k+i} for all $i \in \{0, \dots, f - 1\}$ and for all $k \in \{0, \dots, \pi - 1\}$ using (3.7) and (3.8),
2. Solve the linear problem (3.9) for all $i \in \{0, \dots, f - 1\}$ and for all $k \in \{0, \dots, \pi - 1\}$,
3. Construct $\tilde{\Gamma}_k^f \bar{\mathcal{K}}_k^p \bar{Z}_k^p$ for all $k \in \{0, \dots, \pi - 1\}$ using (3.10),
4. Compute the state sequences and observability matrices using (3.12a) and (3.12b) for all $k \in \{0, \dots, \pi - 1\}$,
5. Solve intersection problem using Theorem 3.1 or Theorem 3.2 and transform all the state sequences to the same basis,
6. With the estimated state, use the linear relations (3.1a)-(3.1b) to obtain the system matrices.

3.7 Dedicated scheduling sequences

In the previous sections we discussed the identification of LPV systems using a periodic scheduling sequence. In this section we briefly discuss the issues that you face when you have almost periodic scheduling or a scheduling referred to as piecewise constant scheduling. Furthermore, we discuss the model structure with parameter-varying noise structure (parameter-varying $K; K^{(i)}$).

3.7.1 Almost Periodic scheduling

The rotational dynamics of variable-speed wind turbines is not perfectly periodic. When there is a wind gust the rotor speed will increase for a moment until the feedback control is able to compensate for the gust. Consequently, the periodic loads that are related to the rotor speed will vary and the scheduling vector is not purely periodic anymore. Still, there are parts in the data that experience the same parameter-varying dynamic behavior. In a similar way a number of stationary sequence models can be estimated by selecting parts of the scheduling sequence with the same time-variation. In [van Wingerden et al. \(2008a\)](#) this approach is followed and applied on the rotational dynamics of a variable-speed wind turbine. The example is also included at the end of this chapter.

The big challenge in this approach is to find a number of stationary sequence models from the data. There are a number of options, going from doing it by hand to using advanced clustering algorithms. We will not discuss the different methods in more detail, because in the next chapter we introduce the framework for arbitrary scheduling and in this framework we can also exploit structure in the scheduling without looking for stationary sequences. However, for more information we refer to [van Wingerden et al. \(2008a\)](#).

3.7.2 Piecewise constant scheduling

We define piecewise constant scheduling as a scheduling sequence that is constant for a number of samples. In [Figure 3.1](#) and [Figure 3.2](#) the shapes of two possible scheduling sequences are sketched that satisfy this requirement. For this kind of

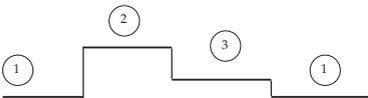


Figure 3.1: Illustrative example of a $\mu_k^{(i)}$ sequence with hard switching between the different stationary sequence models.

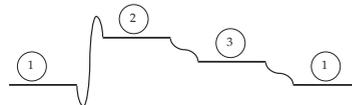


Figure 3.2: Illustrative example of a $\mu_k^{(i)}$ sequence with soft switching between the different stationary sequence models.

scheduling sequence it is straightforward to localize data sequences that experience the same dynamic behavior; the sequences where the scheduling is constant. The data points where the scheduling is constant can be combined and used in an LTI identification algorithm. Similar as for the periodic case, we can do this for a number of constant values; stationary models. Again all these models are identified in a different state basis. For these stationary models we can formulate the intersection problem formulated in Theorem 3.1. However, if we only use data obtained in intervals where the scheduling is constant it turns out that the rank conditions specified in the same theorem do not hold. In [van Wingerden et al. \(2007\)](#); [van Wingerden and Verhaegen \(2007\)](#) we showed that if we use the transitions from 1 local stationary model, for instance model 1 in Figure 3.1 or 3.2, to the other models a global LPV model can be identified. Again we will not discuss this algorithm in detail in this thesis because in the next chapter we introduce the framework for arbitrary scheduling and in this framework we can also exploit structure in the scheduling without searching for stationary sequences.

3.7.3 Parameter-varying K matrix

In this chapter we worked with a model structure with a parameter-invariant K matrix. The main reason for this assumption is to keep the description of the algorithm as transparent as possible. For the case that we have a parameter-dependent K matrix we obtain the following predictor form:

$$x_{k+1} = \sum_{i=1}^m \mu_k^{(i)} \left(A^{(i)} x_k + B^{(i)} u_k + K^{(i)} y_k - K^{(i)} \sum_{j=1}^m \mu_k^{(j)} \left(C^{(j)} x_k + D^{(j)} u_k \right) \right),$$

$$y_k = \sum_{i=1}^m \mu_k^{(i)} \left(C^{(i)} x_k + D^{(i)} u_k \right) + e_k,$$

For the first steps in the algorithm, the estimation of the stationary models, this is not an issue. This model structure becomes an issue if we are going to solve the intersection problem. To be more precise the factorization of the parameter-varying observability matrix becomes more involved. However, the factorization is really cumbersome and we do not require it for the computation. So, we do not give an explicit formula for this matrix. However, the matrix M_k is required to solve the intersection problem. We change the definition given in Definition 3.14 in the following way:

$$M_k = \begin{bmatrix} \tilde{\mu}_{k+p}^T & 0 & \cdots & 0 \\ 0 & \tilde{\mu}_{k+p}^T \otimes \tilde{\mu}_{k+p+1}^T & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & \tilde{\mu}_{k+p}^T \otimes \cdots \otimes \tilde{\mu}_{k+p+f-1}^T \end{bmatrix} \otimes I_\ell,$$

with $\tilde{\mu}_k^T = \mu_k^T \otimes \mu_k^T$. With this substitution the problem can be solved with a parameter-dependent K matrix. The disadvantage of a parameter-dependent K matrix is the ‘curse of dimensionality’. Due to the substitutions listed above the

number of rows of S (columns of M_k) is now given by $q = \sum_{j=1}^f \ell m^{j+1}$ instead of $q = \sum_{j=1}^f \ell m^j$.

3.8 Simulation Examples

We have tested the proposed identification approach on two simple simulated models. The first model is based on the dynamics of a wind turbine and has two states and a parameter-dependent A matrix. The second example is a MIMO system with three states and parameter-dependent A , B , C , and D matrices, which better shows the capabilities of the algorithm. In the next section we discuss a more detailed wind turbine example.

3.8.1 Example 1: Flapping dynamics of a wind turbine

The first example might represent a simple model of the out-of-plane dynamics of a flexible rotor blade of a fixed speed wind turbine (see [Eggleston and Stoddard \(1987\)](#)). Among other phenomena, gravity will lead to a nonlinear description of the flapping dynamics. This gravity induced nonlinearity can be introduced in an LPV model by choosing the scheduling sequence as the cosine of the blade rotation angle, resulting in periodic scheduling. After choosing some wind turbine parameters, the following LPV system description can be derived from the differential equations governing the flapping dynamics:

$$\begin{aligned} [A^{(1)} \mid A^{(2)}] &= \begin{bmatrix} 0 & 0.0734 & \mid & -0.0021 & 0 \\ -6.5229 & -0.4997 & \mid & -0.0138 & 0.5196 \end{bmatrix}, \\ [B^{(1)} \mid B^{(2)}] &= \begin{bmatrix} -0.7221 & \mid & 0 \\ -9.6277 & \mid & 0 \end{bmatrix}, \\ [C^{(1)} \mid C^{(2)}] &= [1 \ 0 \ \mid \ 0 \ 0], \\ [D^{(1)} \mid D^{(2)}] &= [0 \ \mid \ 0], \\ [K^{(1)} \mid K^{(2)}] &= \begin{bmatrix} 0.1 & \mid & 0 \\ 0.02 & \mid & 0 \end{bmatrix}. \end{aligned}$$

The flapping dynamics is excited using a constant wind speed with added turbulence modeled by a white noise input signal u_k with $E[u_k] = 10$ and $\text{var}(u_k) = 1$. The periodic scheduling is given by $\mu_k = [1, \cos(\frac{2\pi}{p}k)]^T$. For the identification experiment we used $\Pi = 500$, $\pi = 10$, and $p = f = 8$. The collected data u_k , y_k , and μ_k is then used in the identification algorithm without using regularization in the CCA step. The performance of the identified system is evaluated by looking at the eigenvalues of the $A^{(i)}$ matrices and the value of the Variance-Accounted-For (VAF) on a data set different from the one used for identification. The VAF is defined as:

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

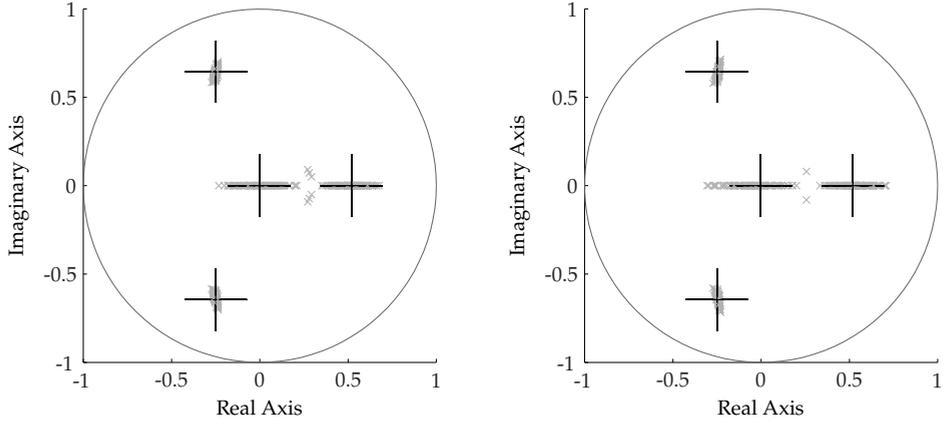


Figure 3.3: Eigenvalues of the estimated $A^{(1)}$ and $A^{(2)}$ matrices in one plot (\times), for 100 experiments with SNR = 10dB and using the periodic LPV-PBSID and periodic LPV-PBSID_{opt} algorithm, the left and right figure, respectively. The big crosses correspond to the real values of the eigenvalues of the matrices.

where \hat{y}_k denotes the output signal obtained by simulating the identified LPV system, y_k is the output signal of the true LPV system.

To investigate the sensitivity of the identification algorithm with respect to noise, a Monte-Carlo simulation with 100 runs was carried out. For each of the 100 simulations a different realization of the input u_k is used. During the simulation noise is added with a specific Signal-to-Noise Ratio (SNR):

$$\text{SNR(dB)} = 10 \log_{10} \frac{\text{var}(y_k)}{\text{var}(e_k)}.$$

An LPV model was identified for a SNR ratio of 10dB, with the Periodic LPV-PBSID_{opt} and the Periodic LPV-PBSID algorithm. In Figure 3.3 the eigenvalues of the estimated models are compared to the true values for the Periodic LPV-PBSID_{opt} and Periodic LPV-PBSID algorithm. Figure 3.4 shows the corresponding histograms of the VAF values on a fresh validation data set but with the same scheduling vector. As expected the quality of the identified models is affected by the noise. There, is no significant difference between the two identification algorithms. However, we have to stress that the Periodic LPV-PBSID_{opt} algorithm has a lower computational complexity.

3.8.2 Example 2: Third order MIMO system

The first example is based on a system that has a periodic scheduling by nature. This assumption only holds for a number of practical applications. However, the proposed algorithm also works for other systems where the scheduling can be

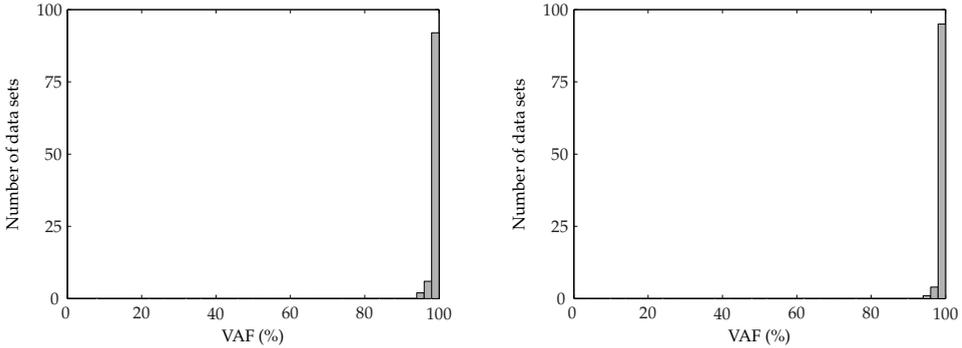


Figure 3.4: Histogram of VAF values (%) obtained for validation data with the models estimated using a data set with SNR = 10dB and using the periodic LPV-PBSID and periodic LPV-PBSID_{opt} algorithm, the left and right figure, respectively. The range of VAF values from 0 to 100% is divided into bins of 2%. For each bin, it is shown how many data sets out of the total 100 resulted in VAF values that fall into that bin.

made periodic during the identification experiment. In this second example a more general case is considered: a MIMO system with parameter-dependent A , B , C , and D matrices where the validation is done with a scheduling that is not periodic. In this example we also illustrate the effect of the regularization parameters. The following third order model is used:

$$\begin{aligned} [A^{(1)} \mid A^{(2)}] &= \left[\begin{array}{ccc|ccc} 0 & 0.9 & 0.2 & 0.6 & 0.5 & 0.5 \\ -0.9 & 0.5 & 0 & 0.5 & 0.6 & 0 \\ -0.2 & 0 & 0.2 & -0.5 & 0 & 0.6 \end{array} \right], \\ [B^{(1)} \mid B^{(2)}] &= \left[\begin{array}{c|c} 1 & 0.4 \\ 1 & 0.2 \\ 1 & 0.12 \end{array} \right], \\ [C^{(1)} \mid C^{(2)}] &= \left[\begin{array}{ccc|ccc} 0.2 & 1 & 0.5 & 0.2 & 0.1 & 1 \\ 0.2 & 0.1 & 1 & 0.3 & 0.4 & 0.8 \end{array} \right], \\ [D^{(1)} \mid D^{(2)}] &= \left[\begin{array}{c|c} 0.1 & 0.2 \\ 0.2 & 0.1 \end{array} \right], \\ [K^{(1)} \mid K^{(2)}] &= \left[\begin{array}{cc|cc} 0.0130 & 0.0225 & 0 & 0 \\ 0.0089 & 0.0060 & 0 & 0 \\ 0.0002 & -0.0010 & 0 & 0 \end{array} \right]. \end{aligned}$$

As input we take a zero-mean white noise signal with $\text{var}(u_k) = \frac{1}{2}\sqrt{2}$. For the identification experiment we used $\Pi = 1000$, $\pi = 18$, and $p = f = 7$. The collected data $(u_k, y_k, \text{ and } \mu_k)$ is used to identify an LPV system. The obtained model is validated by simulating it with a non-periodic scheduling sequence where $\mu_k^{(1)} + \mu_k^{(2)} = 1 \forall k$ and $\mu_k^{(1)}$ is a white noise signal with $E[\mu_k^{(1)}] = 0.5$ and $\text{var}(\mu_k^{(1)}) = 0.01$. The input used in the validation has the same properties as that used in the

experiment.

An LPV model was identified for a signal-to-noise ratio of 30dB. For this noise level it was found that unless regularization is employed the results are completely unsatisfactory. Experiments showed that the first parameter ν_1 has little effect on the performance of the algorithm, probably due to the fact that \tilde{U} is well conditioned. Therefore ν_1 is chosen to be 0. The choice of ν_2 was then made by plotting the average validation VAF for 50 simulations against a logarithmic grid of different values of the parameter and choosing the value corresponding to a peak. It was found, however, that different choices of the periodic scheduling sequence μ_k have an important effect on the degree of regularization required. To illustrate this, we have repeated the identification for two different scheduling sequences:

$$\mu_{k,1} = [0.2 \cos(\frac{2\pi k}{p}) + 0.8, \quad 0.2 \sin(\frac{2\pi k}{p}) + 0.8],$$

and

$$\mu_{k,2} = [0.8 \cos(\frac{2\pi k}{p}) + 0.2, \quad 0.8 \sin(\frac{2\pi k}{p}) + 0.2].$$

The results are summarized in Figures 3.5 to 3.6. In the first case, it can be seen that a relatively high value of regularization is required to obtain acceptable values of the VAF, while in the second case a high VAF is obtained using smaller values. The consequence is that the results obtained using $\mu_{k,1}$ are less accurate than those obtained using $\mu_{k,2}$

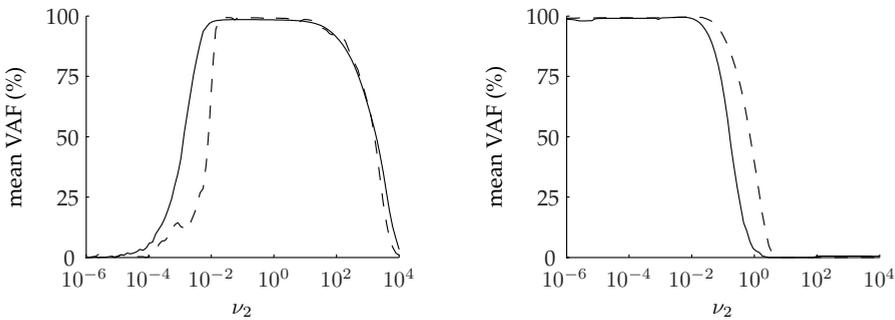


Figure 3.5: The VAF for the validation data as function of the regularization parameter ν_2 for identification data generated with $\mu_{k,1}$ (left) and $\mu_{k,2}$ (right), SNR = 30dB. For the periodic LPV-PBSID (solid) and the periodic LPV-PBSID_{opt}(dashed) algorithm, respectively.

We draw the conclusion that the choice of the magnitude and average value of the periodic scheduling is crucial for the performance of the algorithm. Different choices yield different requirements on the regularization parameter.

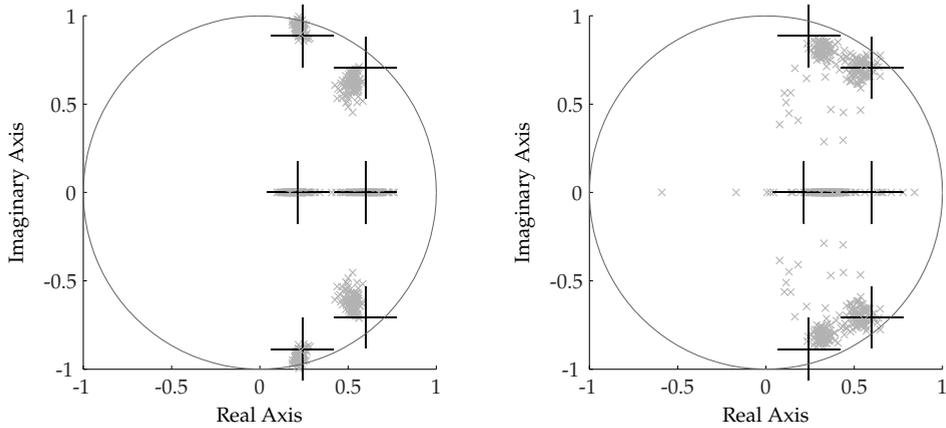


Figure 3.6: True and estimated eigenvalues of $A^{(1)}$ and $A^{(2)}$ in one plot for 50 experiments with SNR = 30dB using the periodic LPV-PBSID algorithm. The small crosses (\times) are the estimated eigenvalues for the scheduling sequence $\mu_{k,1}$ (left with $\nu_2 = 1 \times 10^{-1}$) and $\mu_{k,2}$ (right with $\nu_2 = 1 \times 10^{-4}$).

3.9 Case study: Rotational dynamics of a wind turbine

Linear Time-Invariant (LTI) system identification is well-established and a few applications can be reported in the wind energy (Bongers and van Baars 1991; van Baars and Bongers 1992, 1994; van Baars et al. 1993; James III et al. 1993; Knudsen et al. 1997; Marrant and Van Holten 2004; Hansen et al. 2006). However, the techniques used are all based on the open-loop setting and will give biased results in the closed-loop setting (Van den Hof and Schrama 1995). In this chapter we presented an identification approach to identify LPV systems assuming that the scheduling sequence repeats itself a number of times. In this section we use a nonlinear model of the rotational dynamics to illustrate the effectiveness of the proposed algorithm. We start this section with the description of the wind turbine model used. Followed by a simulation to obtain input-output data. In the last part of this section the simulation results are presented.

3.9.1 First principles model of a Horizontal Axis Wind Turbine (HAWT)

In this example, we consider a seven degrees of freedom model as described in van Engelen (2006) and van Engelen et al. (2007). The model describes the rotational dynamics of a wind turbine around a particular operating point. The model contains degrees of freedom for the main rotation, first torsion mode of the drive train, the first fore-aft, and sideward bending mode of the tower. In this model

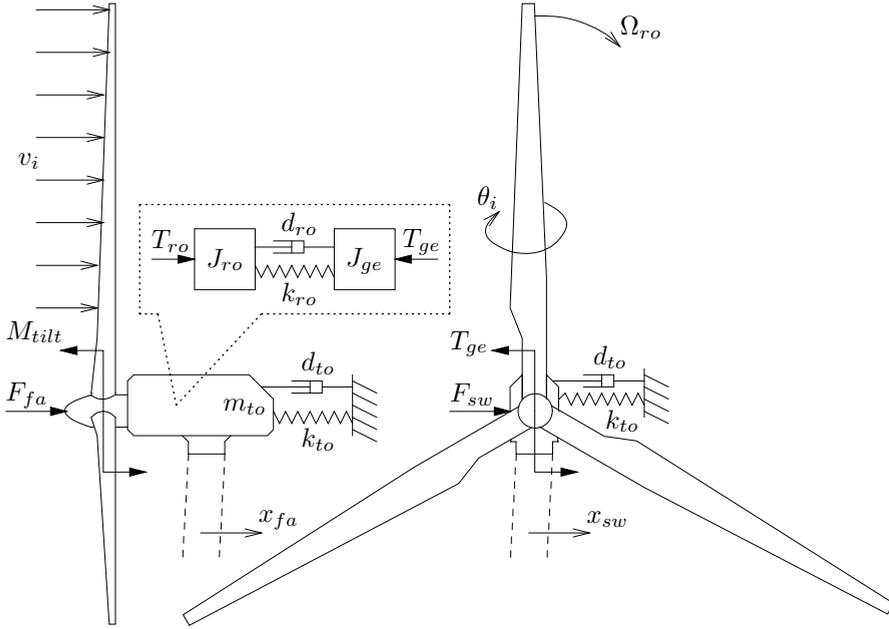


Figure 3.7: Schematic representation of the wind turbine model.

the blades are considered to be rigid. In Figure 3.7, a schematic representation of the model is given

Using a linearized conversion of the aerodynamic behavior, the model equations can be written in the following continuous time LPV system:

$$\dot{x} = Ax + \left(B^{(1)} + \sum_{i=1}^3 B^{(i+1)}\varphi^{(i)} \right) u + \left(F^{(1)} + \sum_{i=1}^3 F^{(i+1)}\varphi^{(i)} \right) v, \quad (3.24a)$$

$$y = \left(C^{(1)} + \sum_{i=1}^3 C^{(i+1)}\varphi^{(i)} \right) x + Du + Gv, \quad (3.24b)$$

where the matrices $B^{(i)}$, $C^{(i)}$, and $F^{(i)}$ are multiplied with the scheduling sequence, which is the azimuth angle $\varphi^{(i)}$ of the accompanying rotor blade. The wind turbine model under consideration has three rotor blades ($i = 1, 2, 3$) and is normally used to design IPC controllers. The system state, input, disturbance, and output vector are given by:

$$\begin{aligned} x &= [\delta\Omega_{ro}, x_{fa}, \dot{x}_{fa}, x_{sw}, \dot{x}_{sw}, \varepsilon, \dot{\varepsilon}]^T, \\ u &= [\delta\theta_1, \delta\theta_2, \delta\theta_3, \delta T_{ge}]^T, \\ v &= [\delta v_1, \delta v_2, \delta v_3]^T, \\ y &= [\delta\Omega_{ge}, \dot{x}_{fa}, \dot{x}_{sw}, \delta M_1, \delta M_2, \delta M_3]^T, \end{aligned}$$

respectively. This model contains thus the control inputs for the variation in generator torque δT_{ge} and the pitch angle $\delta\theta_i$ of each rotor blade. Furthermore, the model contains the inputs for the wind speed disturbance δv_i on each of the three rotor blades. The outputs are the variations in generator speed $\delta\Omega_{ge}$, the fore-aft velocity \dot{x}_{fa} and sideward velocity \dot{x}_{sw} of the tower, and the blade root bending moment δM_i of each rotor blade. The state contains the variations in rotor speed $\delta\Omega_{ro}$, the fore-aft displacement x_{fa} and velocity \dot{x}_{fa} , the sideward displacement x_{sw} and velocity \dot{x}_{sw} , and the drive-train displacement ε and speed $\dot{\varepsilon}$.

The model under consideration has a constant A matrix while the input and output matrices depend on the azimuth angle, φ . In [van Engelen \(2006\)](#) the Coleman transformation is used to transform this model to an LTI model. The Coleman transformation is a nonlinear transformation that is used to transform the outputs defined in the rotating frame to the fixed non-rotating frame and on a similar way this can be done for the inputs. However, this transformation can not cope with a failing sensor/actuator, gravity, and yaw misalignment. If the Coleman transformation is applied to these models still periodic components will be present in the dynamics. However, all the mentioned phenomena will still lead to an LPV model where the system undergoes the same time-variation a number of times. Still, in this chapter we selected the model given in (3.24a)-(3.24b) based on its simplicity, available documentation ([van Engelen 2006](#); [van Engelen et al. 2007](#)), and the mentioned phenomena will not change the proposed LPV system identification algorithm.

The constant state-space matrix A , is given by:

$$A = \begin{bmatrix} 0 & 0 & -\frac{3hM_x}{J_{ro}} & 0 & 0 & -\frac{k_{ro}}{J_{ro}} & -\frac{d_{ro}}{J_{ro}} \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & -\frac{k_{to}}{m_{to}} & q_1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & -\frac{27R}{16H^2} \frac{hF_z}{m_{to}} & -\frac{k_{to}}{m_{to}} & -\frac{d_{to}}{m_{to}} & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & -\frac{3hM_x}{J_{ro}} & 0 & 0 & -\frac{J_{ro}+J_{ge}}{J_{ro}J_{ge}} k_{ro} & -\frac{J_{ro}+J_{ge}}{J_{ro}J_{ge}} d_{ro} \end{bmatrix},$$

with $q_1 = \frac{81R}{32H^2} \frac{hM_z}{m_{to}} - \frac{d_{to}}{m_{to}} + \frac{3hF_x}{m_{to}}$. The constant state-space matrices D and G are given by:

$$D = \begin{bmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ k_{M_z} & 0 & 0 & 0 \\ 0 & k_{M_z} & 0 & 0 \\ 0 & 0 & k_{M_z} & 0 \end{bmatrix},$$

$$G = \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \\ h_{Mz} & 0 & 0 \\ 0 & h_{Mz} & 0 \\ 0 & 0 & h_{Mz} \end{bmatrix}.$$

The state-space matrices B , C , and F do have an LPV structure and are given by:

$$\left[\begin{array}{c|c} B^{(0)} & B^{(i+1)} \end{array} \right] = \left[\begin{array}{cccc|ccc} \frac{k_{Mx}}{J_{ro}} & \frac{k_{Mx}}{J_{ro}} & \frac{k_{Mx}}{J_{ro}} & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & \frac{3}{2H} \frac{k_{Mz}}{m_{to}} & 0 \\ \frac{k_{Fx}}{m_{to}} & \frac{k_{Fx}}{m_{to}} & \frac{k_{Fx}}{m_{to}} & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & -\frac{k_{Fz}}{m_{to}} & 0 \\ 0 & 0 & 0 & \frac{3}{2H} \frac{1}{m_{to}} & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ \frac{k_{Mx}}{J_{ro}} & \frac{k_{Mx}}{J_{ro}} & \frac{k_{Mx}}{J_{ro}} & \frac{1}{J_{ge}} & 0 & 0 & 0 \end{array} \right] \begin{array}{l} 0_{7 \times (i-1)} \\ 0_{7 \times (4-i)} \end{array},$$

ith column

$$\left[\begin{array}{c|c} F^{(0)} & F^{(i+1)} \end{array} \right] = \left[\begin{array}{ccc|ccc} \frac{h_{Mx}}{J_{ro}} & \frac{h_{Mx}}{J_{ro}} & \frac{h_{Mx}}{J_{ro}} & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ \frac{h_{Fx}}{m_{to}} & \frac{h_{Fx}}{m_{to}} & \frac{h_{Fx}}{m_{to}} & \frac{3}{2H} \frac{h_{Mz}}{m_{to}} & 0 & 0 \\ 0 & 0 & 0 & 0 & -\frac{h_{Fz}}{m_{to}} & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ \frac{h_{Mx}}{J_{ro}} & \frac{h_{Mx}}{J_{ro}} & \frac{h_{Mx}}{J_{ro}} & 0 & 0 & 0 \end{array} \right] \begin{array}{l} 0_{7 \times (i-1)} \\ 0_{7 \times (3-i)} \end{array},$$

ith column

$$\left[\begin{array}{c} C^{(0)} \\ C^{(i+1)} \end{array} \right] = \left[\begin{array}{ccccccc|c} 1 & 0 & 0 & 0 & 0 & 0 & -1 & \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & \\ 0 & 0 & -h_{Mz} & 0 & 0 & 0 & 0 & \\ 0 & 0 & -h_{Mz} & 0 & 0 & 0 & 0 & \\ 0 & 0 & -h_{Mz} & 0 & 0 & 0 & 0 & \\ \hline 0 & 0 & \frac{9Rh_{Mz}}{8H} & 0 & 0 & 0 & 0 & \end{array} \right] \begin{array}{l} 0_{(i+2) \times 7} \\ \text{} \\ 0_{(3-i) \times 7} \end{array} \left. \vphantom{\begin{array}{c} C^{(0)} \\ C^{(i+1)} \end{array}} \right\} (i+3)\text{th row}$$

In these matrices the parameters k_{Mx} , k_{Mz} , k_{Fx} , and k_{Fz} describe the aerodynamic gains from the pitch angle to the root moment, flap moment, root force, and flap force, respectively. The parameters h_{Mx} , h_{Mz} , h_{Fx} , and h_{Fz} describe the gain from the wind speed to the root moment, flap moment, root force, and flap force, respectively. The constants R and H are the rotor radius and the height of the hub, respectively; the mass moment of inertia J , the mass m , the stiffness k and the damping d . Furthermore, the subscripts ro , to , and ge refer to the rotor, tower, and generator, respectively. The aerodynamic constants are listed in Table 3.2 and

are derived for a wind speed of 16 m/s, a pitch angle of 10 degrees, and a rotor speed of 1.795 rad/s.

Table 3.2: Numerical values of the model parameters (van Engelen et al. 2007).

parameter	value	parameter	value
H	55.953 m	h_{M_x}	8.3806×10^4 Ns
R	40 m	h_{F_x}	7.2019×10^3 Ns/m
J_{ge}	1.067×10^6 kgm ²	h_{M_z}	-1.8948×10^5 Ns
J_{ro}	7.187×10^6 kgm ²	h_{F_z}	4.0683×10^3 Ns/m
k_{ro}	1.262×10^8 N/m	k_{M_x}	-3.7711×10^4 Nm
d_{ro}	1.262×10^5 Ns/m	k_{F_x}	-6.1478×10^3 N
m_{to}	1.5657×10^5 kg	k_{M_z}	1.6174×10^5 Nm
k_{to}	1.235×10^6 N/m	k_{F_z}	-1.8306×10^3 N
d_{to}	2.7995×10^3 Ns/m		

3.9.2 Simulation of the closed-loop wind turbine model

The LPV system given in (3.24a) and (3.24b) is used to obtain the input, output, and the scheduling sequence for the identification algorithm. For this purpose, the equations are converted to discrete time using a naive zero-order hold discretization method with a sample time of 0.1 s. The naive approach omits the switching behaviors of the sampled scheduling signals. For our case, where the scheduling sequence is a function of the azimuth angles the scheduling sequences are given by the following smooth signals:

$$\varphi_k = \left[\sin\left(\frac{2\pi k}{v}\right), \sin\left(\frac{2\pi k}{v} + \frac{2\pi}{3}\right), \sin\left(\frac{2\pi k}{v} + \frac{4\pi}{3}\right) \right]^T.$$

When an appropriate sample time is chosen this method gives a good approximation of the continuous time LPV system.

The wind turbine system is not asymptotically stable, it has an integrator. A collective pitch controller in a feedback loop is added to stabilize the system. The controller used, can be found in van Engelen et al. (2007) where the collective pitch controller is parameterized. For the pitch-angle inputs we take an additional zero-mean white noise with $\text{var}(\theta_{k,i}) = 1$ deg, which is added to the control signal of the collective pitch controller. As input for the generator torque we take also a zero-mean white noise signal with $\text{var}(T_{ge,k}) = 1 \cdot 10^6$ Nm. The wind disturbance signals are also zero-mean white noise with $\text{var}(v_{k,i}) = 1$ m/s but these signals are assumed to be unknown.

3.9.3 Closed-loop LPV subspace identification results

The collected data of u_k , y_k , and μ_k from the simulations are used in the identification experiments. The scheduling sequence can be rewritten as:

$$\mu_k = [1, \quad \varphi_{k,1}, \quad \varphi_{k,2}]^T,$$

to fulfill the assumption that this scheduling matrix must be of full rank. The third azimuth angle can be written as a linear combination of the other two angles. For the identification experiments we used $\Pi = 1000$, $\pi = 35$, $f = 16$, and $p = 10$. For this particular example we used the periodic LPV-PBSID algorithm.

The performance of the identified system is evaluated by looking at the eigenvalues of the A matrix and the value of the VAF on a data set different from the one used for identification. For meaningful VAF values the system under consideration must be asymptotically stable, otherwise a small error will give low VAF values due to the increasing or decreasing characteristic of the outputs. This problem occurs for the output of the generator speed, therefore Bode diagrams at a fixed scheduling vector are used to evaluate the performance at those specific channels.

To investigate the sensitivity of the identification algorithm with respect to the wind disturbances, a Monte-Carlo simulation with 100 runs was carried out. For each of the 100 simulations a different realization of the input u_k and wind disturbance v_k is used. In Figure 3.8 the eigenvalues of the estimated models are compared with the true values. It shows that the identified eigenvalues are very close to the true eigenvalues and that the variance and bias is very small. Figure 3.9 shows the corresponding histograms of the VAF values on a fresh validation set with the same scheduling vector and without the wind disturbances. The outputs of the blade root moments M_1 , M_2 , and M_3 score very high VAF values, all within 98% and 100%. The outputs \dot{x}_{fa} and \dot{x}_{sw} are more affected by the wind disturbance. However, the values are still satisfactory high. The bode diagrams with the generator speed Ω_{ge} as output are given in Figure 3.10. Also in this figure satisfactory fits are shown, especially for the transfer function with as input the pitch angles. However, for the transfer function between generator torque and generator speed, the low frequent behavior shows a large variance due to the high disturbance that has a significant effect on the estimation of the pole belonging to the integrator. However, this is a well-known phenomena in LTI system identification. The resonance frequency is well estimated and for controller design this resonance frequency will significantly limit the bandwidth.

3.10 Conclusion

Wind turbines are nonlinear systems, although their nonlinearity is linearly dependent on measurable scheduling signals and therefore they can be modeled in the LPV framework. With LPV controller synthesis, which is strongly related to

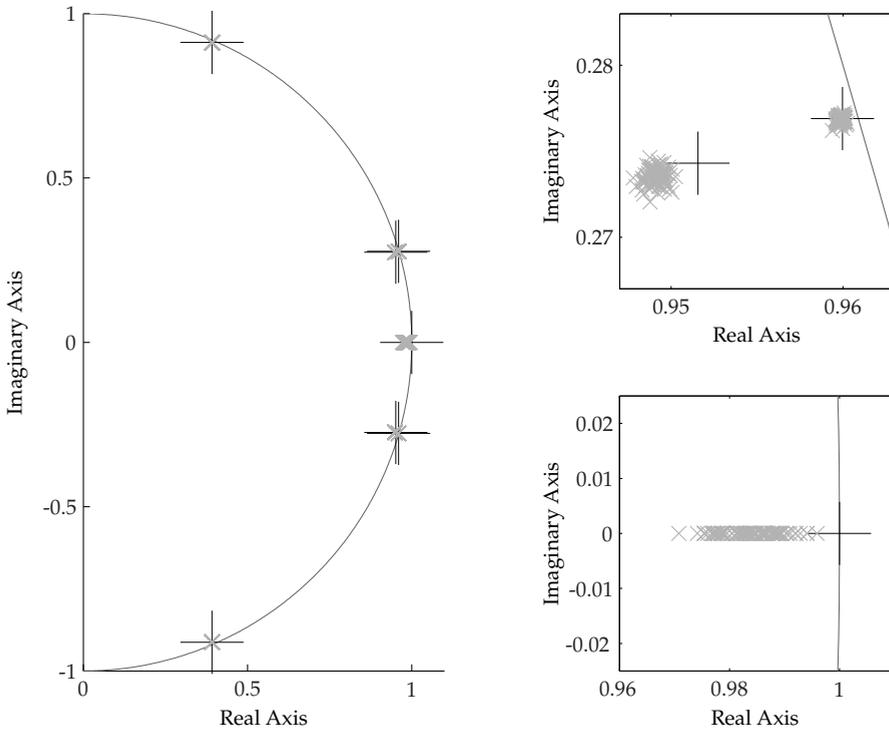


Figure 3.8: Eigenvalues of the estimated A matrix in the complex plane, for 100 experiments with a wind disturbance of $\text{var}(v) = 1 \text{ m/s}$. The big crosses correspond to the real values of the eigenvalues of the matrix. The boxes to the right show a magnification of three pole locations.

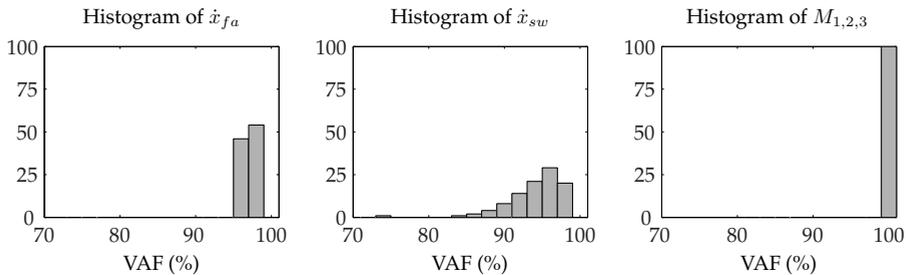


Figure 3.9: Histogram of VAF values (%) of the outputs \dot{x}_{fa} , \dot{x}_{sw} , and $M_{1,2,3}$. The range of VAF values from 0 to 100% is divided into bins of 2%. For each bin, it is shown how many data sets out of the total 100 resulted in VAF values that fall into that bin.

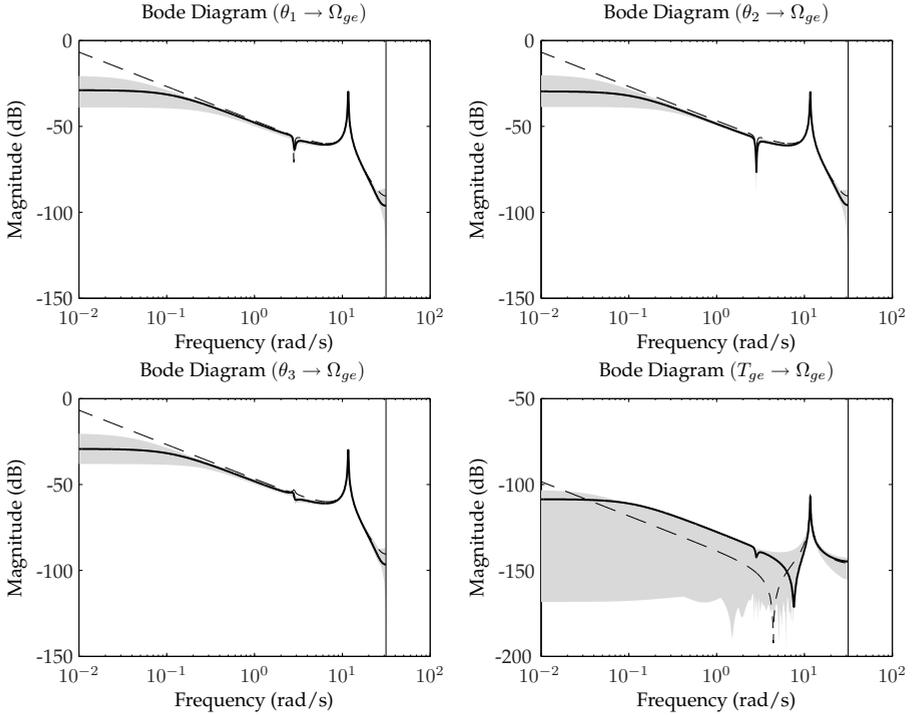


Figure 3.10: Bode diagrams of the original transfer functions (dashed) and the identified transfer functions of the experiment with the highest mean VAF value (bold). The transfer functions of the other 99 experiments are within the gray confidence region. To determine the Bode diagram, the azimuth angles are fixed at the values $\varphi = [0, \sqrt{3}/2, -\sqrt{3}/2]$.

robust controller design, gain-scheduled controllers can be calculated with guaranteed stability and performance margins. In this chapter we discussed LPV system identification using periodic scheduling sequences. The obtained LPV model is then valid for different scheduling sequences as well. We exploited the fact that the system experienced the same time-variation a number of times. We used LTI system identification techniques, PBSID and $\text{PBSID}_{\text{opt}}$, to identify a number of observability matrices and state sequences which are, inherent to subspace identification, identified in a different state basis. We showed that by formulating an intersection problem all the states can be reconstructed in a global state basis from which the system matrices could be estimated. The algorithm is first implemented on two simple test examples. The first shows the effectiveness of the periodic LPV- $\text{PBSID}_{\text{opt}}$ and the periodic LPV-PBSID algorithm. The second illustrates that the performance of the algorithm is affected by the choice of the scheduling sequence. Further research could shed more light on the issue of the choice of this scheduling sequence in relation to the use of regularization. Finally, we applied the approach successfully on the rotational dynamics of a horizontal axis wind turbine.

Subspace identification of MIMO LPV systems using an arbitrary scheduling sequence

In this chapter we present a novel algorithm to identify LPV systems with affine parameter dependence. This algorithm is applicable for data generated in either open or closed-loop. With respect to the previous chapter the scheduling is now allowed to vary arbitrarily. A factorization is introduced that makes it possible to form a predictor, which is based on past inputs, outputs, and scheduling data. The predictor contains the LPV equivalent of the Markov parameters. Using this predictor, ideas from closed-loop LTI identification are developed to estimate the state sequence from which the LPV system matrices can be constructed. A numerically efficient implementation is presented using the kernel method. It turns out that if structure is present in the scheduling sequence the computational complexity reduces even more.

4.1 Introduction

From a system theoretic point of view the identification and control of Linear Parameter-Varying (LPV) systems has attracted considerable attention in recent years as already highlighted in the introduction of this thesis and the introduction of the previous chapter. In the previous chapter we developed a set of algorithms that exploit the structure in the scheduling sequence to enable the use of well-known LTI techniques to identify LPV systems. This can be seen as the latest development in the field of LPV system identification. When it is possible to control the scheduling sequence during the data acquisition, the approach presented in the previous chapter is appropriate. However, for many applications you can not control the scheduling sequence. For instance for a wind turbine the operational position is strongly correlated with the wind and we all know that the wind

is not controllable. For systems where there is no structure in the scheduling sequence we talk about LPV systems with arbitrary scheduling sequences, which is the topic of this chapter.

For the case of arbitrary scheduling we presented in the introduction two approaches with the potential to tackle the identification problem: subspace identification and nonlinear optimization based approaches. Previous work showed that these two methods should be developed simultaneously because for the nonlinear optimization technique good initial guesses are required while for the subspace approaches it is typical to have a biased estimate. In the field of subspace LPV identification a milestone contribution is presented in [Verdult \(2002\)](#). [Verdult \(2002\)](#) extends the Bilinear identification scheme of [Favoreel et al. \(1997, 1999\)](#) and [Favoreel \(1999\)](#), and later improved by [Verdult et al. \(1998\)](#) and [Chen and Maciejowski \(2000\)](#), to LPV systems. Compared to the subspace LTI counterpart MOESP ([Verhaegen and Dewilde 1992](#)) this algorithm has the inherent drawback that it estimates the state sequence using a certain past window, possibly leading to biased results. Similar approximations are made in the subspace LTI algorithm: N4SID ([Van Overschee and De Moor 1996](#)), however, by making the past window larger and larger this bias will tend to zero. It turns out that identification of LPV systems with arbitrarily varying scheduling sequences is challenging from a numerical point of view ([Verdult and Verhaegen 2001, 2002](#)); the data matrices involved in this algorithm grow exponentially with the size of the past window. With the introduction of the kernel method, the ‘curse of dimensionality’ was partially solved, however, a different bias was introduced ([Verdult and Verhaegen 2005](#)). Still, they showed that the obtained estimates are good initial guesses for nonlinear optimization approaches ([Lee and Poolla 1999](#); [Verdult et al. 2003](#)). Another disadvantage of the approach presented in [Verdult \(2002\)](#) is that the algorithm is only suited for data generated in open loop.

In this chapter we discuss a novel LPV identification algorithm that can cope with data generated in open and closed-loop, which is the first main contribution of this chapter. We present a factorization that makes it possible to formulate a predictor that contains the LPV equivalent of the Markov parameters. The second main contribution is that we present an approach that stays close to the formulations given in [Chiuso \(2007\)](#) and that can be seen as an extension of the algorithms presented therein. The computational complexity of the algorithm is significantly smaller than the algorithms in [Verdult \(2002\)](#), but still the dimensions grow exponentially. Similar to what is done in [Verdult and Verhaegen \(2005\)](#) we present the kernel method to reduce the computational complexity. However, we derive computationally efficient formulations of the kernels, which is the third main contribution of this chapter. We also show that if the scheduling sequence is periodic, piecewise constant, or structured in some sense, the identification procedure significantly simplifies even more from a computational point of view.

The outline of this chapter is as follows; we start in [Section 4.2](#) with the problem formulation and assumptions. In [Section 4.3](#) we present a factorization that separates the unknown system matrices from the known input, output, and scheduling data. In [Section 4.4](#) the basic idea behind the identification scheme is presented and the ‘curse of dimensionality’ will appear. In [Section 4.5](#) the kernel

method is presented, where compact formulations of the kernels are presented. The algorithm is presented for a specific model structure, in Section 4.6 we will discuss extensions to other model structures as well. In Section 4.7 we show that dedicated scheduling sequences significantly reduce the computational complexity. In Section 4.8 three simulation examples are presented that show the potential of the proposed algorithm. At the end of this chapter we briefly go back to the ‘smart’ rotor concept and present a case study to identify the dynamics of a ‘smart’ airfoil, Section 4.9. We end this chapter with our conclusions concerning this novel algorithm.

Some results published in this chapter are published elsewhere, see [van Wingerden and Verhaegen \(2008a,b, 2009\)](#).

4.2 Problem formulation and assumptions

In this section we present the model structure we consider in this chapter. Furthermore, some assumptions are listed and some notation is introduced.

4.2.1 Problem formulation

For the derivation of the algorithm we consider the LPV model structure defined in (4.1a)-(4.1b). However, similar results can be derived for LPV systems with parameter-varying output equation, which we will briefly discuss in Section 4.6. For now we consider:

$$x_{k+1} = \sum_{i=1}^m \mu_k^{(i)} \left(A^{(i)} x_k + B^{(i)} u_k + K^{(i)} e_k \right), \quad (4.1a)$$

$$y_k = C x_k + D u_k + e_k, \quad (4.1b)$$

where $x_k \in \mathbb{R}^n$, $u_k \in \mathbb{R}^r$, $y_k \in \mathbb{R}^\ell$, are the state, input, and output vectors. $e_k \in \mathbb{R}^\ell$ denotes the zero-mean white innovation process. The matrices $A^{(i)} \in \mathbb{R}^{n \times n}$, $B^{(i)} \in \mathbb{R}^{n \times r}$, $C \in \mathbb{R}^{\ell \times n}$, $D \in \mathbb{R}^{\ell \times r}$, $K^{(i)} \in \mathbb{R}^{n \times \ell}$ are the local system, input, output, direct feedthrough, and the observer matrices; and $\mu_k^{(i)} \in \mathbb{R}$ the local weights. The index m is referred to as the number of local models or scheduling parameters. Note that the system, input, and the observer matrices depend linearly on the time-varying scheduling vector. The time-varying system matrix is now given by:

$$A_k = \sum_{i=1}^m \mu_k^{(i)} A^{(i)},$$

and a similar thing can be done for the other system matrices. We assume that we have an affine dependence and the scheduling is given by:

$$\mu_k = \left[1, \mu_k^{(2)}, \dots, \mu_k^{(m)} \right]^T.$$

We can rewrite (4.1a)-(4.1b) in the predictor form as:

$$x_{k+1} = \sum_{i=1}^m \mu_k^{(i)} \left(\tilde{A}^{(i)} x_k + \tilde{B}^{(i)} u_k + K^{(i)} y_k \right), \quad (4.2a)$$

$$y_k = C x_k + D u_k + e_k, \quad (4.2b)$$

with

$$\tilde{A}^{(i)} = A^{(i)} - K^{(i)} C, \quad \tilde{B}^{(i)} = B^{(i)} - K^{(i)} D.$$

It is well-known that an invertible linear transformation of the state does not change the input-output behavior of a state-space system. Therefore, we can only determine the system matrices up to a similarity transformation $T \in \mathbb{R}^{n \times n}$: $T^{-1}A^{(i)}T$, $T^{-1}B^{(i)}$, $T^{-1}K^{(i)}$, CT , and D .

The identification problem can now, similar as in the previous chapter, be formulated as:

Problem Description 4.1 (LPV system identification)

Given the input sequence u_k , the output sequence y_k , and the scheduling sequence μ_k over a time interval $k = \{0, \dots, N-1\}$; find, if they exist, the LPV system matrices $A^{(i)}$, $B^{(i)}$, $K^{(i)}$, C , and D up to a global similarity transformation.

4.2.2 Assumptions and notation

First we define the transition matrix for discrete time time-varying systems (Rugh 1996) and this is given by:

$$\phi_{j,k} = \tilde{A}_{k+j-1} \cdots \tilde{A}_{k+1} \tilde{A}_k. \quad (4.3)$$

To make the notation more transparent we define:

$$\begin{aligned} z_k &= \begin{bmatrix} u_k \\ y_k \end{bmatrix}, \\ \check{B}_k &= \begin{bmatrix} \tilde{B}_k & K_k \end{bmatrix}, \\ \overline{B}^{(i)} &= \begin{bmatrix} \tilde{B}^{(i)} & K^{(i)} \end{bmatrix}. \end{aligned}$$

Similar as in Jansson (2005) and Chiuso (2007) we define a past window denoted by p . This window is used to define the following stacked vector:

$$\overline{z}_k^p = \begin{bmatrix} z_k \\ z_{k+1} \\ \vdots \\ z_{k+p-1} \end{bmatrix},$$

which is a vector containing past and future data. We assume that the state sequence:

$$X = [x_{p+1}, \dots, x_N],$$

has full row rank and the matrix:

$$\Gamma^f = \begin{bmatrix} C \\ C\tilde{A}^{(1)} \\ \vdots \\ C(\tilde{A}^{(1)})^{f-1} \end{bmatrix}, \quad (4.4)$$

has full column rank and where f is referred to as the future window. This last matrix can be interpreted as the extended observability matrix of the first local model. For persistency of excitation it is also required that the scheduling sequence satisfies the following relation:

$$\text{rank} \left([\mu_0, \mu_1, \dots, \mu_{N-p-f}] \right) = m,$$

and $N-p-f+1 > m$. Furthermore, we assume that the feedback problem is well-posed. So, there is either a delay in the feedback controller or in the system. The problem formulation so far does not require any assumptions on the correlation between the input and noise sequence, which opens the possibility to apply the algorithm in closed-loop.

These definitions and assumptions are used in Section 4.4 but first we define a factorization to extend the predictor-based subspace identification approaches (see [Chiuso \(2007\)](#) for an overview) to LPV systems.

4.3 Factorization of the LPV controllability matrix

In this section we define a fundamental factorization in which we separate the unknown system matrices from the known weighting sequence. To be more precise we will factorize the time-varying extended controllability matrix, which is defined in the following definition.

Definition 4.1 *Given the transition matrix in (4.3) the time-varying extended controllability matrix is given by:*

$$\overline{\mathcal{K}}_k^p = [\phi_{p-1,k+1}\check{B}_k, \dots, \phi_{1,k+p-1}\check{B}_{k+p-2}, \check{B}_{k+p-1}].$$

To illustrate this definition see the following example:

Example 4.1 ($\bar{\mathcal{K}}_k^p$)

For $m = 2$, $k = 0$, and $p = 3$ one obtains:

$$\bar{\mathcal{K}}_0^3 = \begin{bmatrix} \left(\tilde{A}^{(1)} + \tilde{A}^{(2)} \otimes \mu_2^{(2)} \right) \left(\tilde{A}^{(1)} + \tilde{A}^{(2)} \otimes \mu_1^{(2)} \right) \left(\bar{B}^{(1)} + \bar{B}^{(2)} \otimes \mu_0^{(2)} \right), \dots \\ \left(\tilde{A}^{(1)} + \tilde{A}^{(2)} \otimes \mu_2^{(2)} \right) \left(\bar{B}^{(1)} + \bar{B}^{(2)} \otimes \mu_1^{(2)} \right), \left(\bar{B}^{(1)} + \bar{B}^{(2)} \otimes \mu_2^{(2)} \right) \end{bmatrix}.$$

From this example we clearly see that this time-varying controllability matrix is depending on the known scheduling sequence and the LPV system matrices. The next step is to factorize the time-varying extended controllability matrix in a matrix containing only the scheduling terms and a constant matrix, which depends only on the system matrices $\tilde{A}^{(i)}$, and $\bar{B}^{(i)}$. Before we formulate this factorization in a lemma we have to introduce a number of definitions. We start with the following definition:

Definition 4.2 We define the matrix:

$$\mathcal{L}_j = \left[\tilde{A}^{(1)} \mathcal{L}_{j-1}, \dots, \tilde{A}^{(m)} \mathcal{L}_{j-1} \right],$$

with

$$\mathcal{L}_1 = \left[\bar{B}^{(1)}, \dots, \bar{B}^{(m)} \right].$$

To illustrate this definition see the following example:

Example 4.2 (\mathcal{L}_j)

For $m = 2$ one obtains:

$$\begin{aligned} \mathcal{L}_1 &= \left[\bar{B}^{(1)}, \bar{B}^{(2)} \right], \\ \mathcal{L}_2 &= \left[\tilde{A}^{(1)} \bar{B}^{(1)}, \tilde{A}^{(1)} \bar{B}^{(2)}, \tilde{A}^{(2)} \bar{B}^{(1)}, \tilde{A}^{(2)} \bar{B}^{(2)} \right], \\ \mathcal{L}_3 &= \left[\tilde{A}^{(1)} \tilde{A}^{(1)} \bar{B}^{(1)}, \tilde{A}^{(1)} \tilde{A}^{(1)} \bar{B}^{(2)}, \tilde{A}^{(1)} \tilde{A}^{(2)} \bar{B}^{(1)}, \tilde{A}^{(1)} \tilde{A}^{(2)} \bar{B}^{(2)}, \dots \right. \\ &\quad \left. \tilde{A}^{(2)} \tilde{A}^{(1)} \bar{B}^{(1)}, \tilde{A}^{(2)} \tilde{A}^{(1)} \bar{B}^{(2)}, \tilde{A}^{(2)} \tilde{A}^{(2)} \bar{B}^{(1)}, \tilde{A}^{(2)} \tilde{A}^{(2)} \bar{B}^{(2)} \right]. \end{aligned}$$

The number of block-columns grows exponentially as m^j . Using this definition we define the matrix \mathcal{K}^p , which we refer to as LPV extended controllability matrix.

Definition 4.3 The operator \mathcal{L}_j is used to define the LPV extended controllability matrix:

$$\mathcal{K}^p = \left[\mathcal{L}_p, \mathcal{L}_{p-1}, \dots, \mathcal{L}_1 \right] \in \mathbb{R}^{n \times \tilde{q}},$$

with $\tilde{q} = (r + \ell) \sum_{j=1}^p m^j$.

To present the factorized expression of the time-varying extended controllability matrix in Lemma 4.1, we still need the following two definitions:

Definition 4.4 We define the matrix:

$$P_{p|k} = \mu_{k+p-1} \otimes \cdots \otimes \mu_k \otimes I_{r+\ell},$$

with $P_{p|k} \in \mathbb{R}^{m^p(r+\ell) \times (r+\ell)}$ and \otimes represents the Kronecker product (Brewer 1978).

Now we define:

Definition 4.5 With Definition 4.4 we can define:

$$N_k^p = \begin{bmatrix} P_{p|k} & & & 0 \\ & P_{p-1|k+1} & & \\ & & \ddots & \\ 0 & & & P_{1|k+p-1} \end{bmatrix},$$

with $N_k^p \in \mathbb{R}^{\bar{q} \times p(r+\ell)}$.

To illustrate this, see the following example:

Example 4.3 (N_k^p)

For $m = 2$, $k = 0$, and $p = 3$ one obtains:

$$N_0^3 = \begin{bmatrix} 1 & 0 & 0 \\ \mu_0^{(2)} & 0 & 0 \\ \mu_1^{(2)} & 0 & 0 \\ \mu_1^{(2)} \mu_0^{(2)} & 0 & 0 \\ \mu_2^{(2)} & 0 & 0 \\ \mu_2^{(2)} \mu_0^{(2)} & 0 & 0 \\ \mu_2^{(2)} \mu_1^{(2)} & 0 & 0 \\ \mu_2^{(2)} \mu_1^{(2)} \mu_0^{(2)} & 0 & 0 \\ 0 & 1 & 0 \\ 0 & \mu_1^{(2)} & 0 \\ 0 & \mu_2^{(2)} & 0 \\ 0 & \mu_2^{(2)} \mu_1^{(2)} & 0 \\ 0 & 0 & 1 \\ 0 & 0 & \mu_2^{(2)} \end{bmatrix} \otimes I_{\ell+r}.$$

Now we can state the following lemma:

Lemma 4.1 *Given the model structure in (4.2a)-(4.2b) we use Definition 4.3, and 4.5 to obtain:*

$$\overline{\mathcal{K}}_k^p = \mathcal{K}^p N_k^p,$$

where $\overline{\mathcal{K}}_k^p$ is the time-varying extended controllability matrix, which equals Definition 4.1, N_k^p depends on the known scheduling sequence, and \mathcal{K}^p is an unknown matrix defined in Definition 4.3. Note that the number of columns of \mathcal{K}^p (rows of N_k^p), denoted by \tilde{q} , increases exponentially with p according to the relation $\tilde{q} = (r + \ell) \sum_{j=1}^p m^j$.

Proof: Proof follows through straightforward computations. \square

Using the examples in this section this definition can be illustrated.

4.4 LPV predictor-based subspace identification

With the factorization defined in Section 4.3 we now can present the core of this chapter. We will use similar ideas as discussed in [Chiuso \(2007\)](#) but now we introduce the LPV counterpart.

4.4.1 Predictors

The first objective of the algorithm is to reconstruct the state sequence up to a similarity transformation. The state x_{k+p} is given by:

$$x_{k+p} = \phi_{p,k} x_k + \mathcal{K}^p N_k^p \overline{z}_k^p,$$

where $\phi_{p,k}$ is the transition matrix given in (4.3), \mathcal{K}^p is the time-invariant extended LPV controllability matrix, and the matrix N_k^p is a matrix solely depending on the scheduling sequence. The key approximation in this algorithm is that we assume that $\phi_{j,k} \approx 0$ for all $j \geq p$. Similar as in the LTI case it can be shown that if the system in (4.2a)-(4.2b) is uniformly exponentially stable the approximation error can be made arbitrarily small by making p large ([Verdult and Verhaegen 2002](#)). With this assumption the state x_{k+p} is approximately given by:

$$x_{k+p} \approx \mathcal{K}^p N_k^p \overline{z}_k^p. \quad (4.5)$$

In a number of LTI subspace methods it is well known to make this step ([Van Overschee and De Moor 1996](#); [Jansson 2005](#); [Chiuso 2007](#)). The input-output behavior is now approximately given by:

$$\begin{cases} y_{k+p} & \approx C \mathcal{K}^p N_k^p \overline{z}_k^p + D u_{k+p} + e_{k+p} := y_{k+p}^{(p)} \\ y_{k+p+1} & \approx C \mathcal{K}^{p+1} N_k^{p+1} \overline{z}_k^{p+1} + D u_{k+p+1} + e_{k+p+1} \\ \vdots & \\ y_{k+p+f-1} & \approx C \mathcal{K}^{p+f-1} N_k^{p+f-1} \overline{z}_k^{p+f-1} + D u_{k+p+f-1} + e_{k+p+f-1} \end{cases} \quad (4.6)$$

Now we define the stacked matrices U_i , Y_i , and Z_i :

$$U_i = [u_{p+i}, \dots, u_{N-f+i+1}], \quad (4.7a)$$

$$Y_i = [y_{p+i}, \dots, y_{N-f+i+1}], \quad (4.7b)$$

$$Z_i = [N_0^{p+i} \bar{z}_0^{p+i}, \dots, N_{N-p-f}^{p+i} \bar{z}_{N-p-f}^{p+i}], \quad (4.7c)$$

for all $i \in \{0, \dots, f-1\}$. If the matrix $[Z_i^T, U_i^T]^T$ has full row rank for all $i \in \{0, \dots, f-1\}$, the matrices CK^{p+i} and D can be estimated by solving the following linear regression problem:

$$\min_{CK^{p+i}, D} \|Y_i - CK^{p+i}Z_i - DU_i\|_F^2, \quad (4.8)$$

for all $i \in \{0, \dots, f-1\}$ and where $\|\dots\|_F$ represents the Frobenius norm (Golub and Loan 1996). For finite p the solution of this linear problem will be biased due the approximation made in (4.5). In the LTI literature a number of papers appeared that studied the effect of the window size and although they proved the asymptotic properties of the algorithms (if $p \rightarrow \infty$ the bias disappears) it is hard to quantify the effect for finite p (Knudsen 2001; Chiuso and Picci 2005; Chiuso 2007). In Section 4.8 we demonstrate that a rather large approximation does not directly imply a large bias in the estimate of the system matrices.

4.4.2 Extended observability times controllability matrix

The product $K^p Z_0$, which represents by definition the state sequence, X , can not directly be estimated. In the LTI literature it is common practice to use the estimate of the linear-time invariant equivalent of CK^{p+i} to construct the extended observability matrix times the extended controllability matrix. For the LPV situation a similar approach can be followed. However, in this case we look at the product $\Gamma^f K^p$, where Γ^f is defined in (4.4). This matrix can be constructed with Definition 4.2 to equal the following matrix (we assume $f = p$):

$$\Gamma^f K^p = \begin{bmatrix} C\mathcal{L}_p & C\mathcal{L}_{p-1} & \dots & C\mathcal{L}_1 \\ C\tilde{A}^{(1)}\mathcal{L}_p & C\tilde{A}^{(1)}\mathcal{L}_{p-1} & \dots & C\tilde{A}^{(1)}\mathcal{L}_1 \\ \vdots & \vdots & \ddots & \vdots \\ C\left(\tilde{A}^{(1)}\right)^{f-1}\mathcal{L}_p & C\left(\tilde{A}^{(1)}\right)^{f-1}\mathcal{L}_{p-1} & \dots & C\left(\tilde{A}^{(1)}\right)^{f-1}\mathcal{L}_1 \end{bmatrix}. \quad (4.9)$$

This particular matrix is constructed in the LPV-PBSID algorithm. The following matrix is used in the LPV-PBSID_{opt} algorithm (we assume $f = p$):

$$\Gamma^f K^p \approx \begin{bmatrix} C\mathcal{L}_p & C\mathcal{L}_{p-1} & \dots & C\mathcal{L}_1 \\ 0 & C\tilde{A}^{(1)}\mathcal{L}_{p-1} & \dots & C\tilde{A}^{(1)}\mathcal{L}_1 \\ & & \ddots & \vdots \\ 0 & & & C\left(\tilde{A}^{(1)}\right)^{f-1}\mathcal{L}_1 \end{bmatrix}. \quad (4.10)$$

The zeros appear in this equation based on the approximation that $\phi_{j,k} \approx 0$ for all $j \geq p$. This matrix represents the extended observability matrix of the first local model times the LPV controllability matrix. Observe that from the linear regression problems formulated in (4.8) we only need the solution for $i = 0$ to construct the matrix given in (4.10). With the factorizations presented in Definition 4.2 we can construct this product between the extended observability matrix of the first local model and the extended LPV controllability matrix solely from the elements of CK^p . To illustrate this we refer to Table 4.1 where an example is given. For the LPV-PBSID algorithm we have to construct the matrix given in (4.9) or alternatively given by:

$$\Gamma^f \mathcal{K}^p = \begin{bmatrix} CK^p \\ C \left(\tilde{A}^{(1)} \right) \mathcal{K}^p \\ \vdots \\ C \left(\tilde{A}^{(1)} \right)^{f-1} \mathcal{K}^p \end{bmatrix}.$$

Observe that from the linear problems formulated in (4.8) and the factorizations presented in Definition 4.2 we can construct the $(i+1)^{th}$ block row of $\Gamma^f \mathcal{K}^p$ from the estimate of CK^{p+i} . From which we can construct $\Gamma^f \mathcal{K}^p Z_0$, which equals by definition the extended observability matrix times the state sequence, $\Gamma^f X$. The construction of $\Gamma^f \mathcal{K}^p$ is a rather cumbersome task. However, when we introduce the kernel method the construction of this matrix simplifies significantly.

From the constructed matrix $\Gamma^f \mathcal{K}^p$, either the LPV-PBSID or LPV-PBSID_{opt} way, we can compute $\Gamma^f \mathcal{K}^p Z_0$, which equals by definition the extended observability matrix times the state sequence, $\Gamma^f X$. By computing a Singular Value Decomposition (SVD) of this estimate we can estimate the state sequence and the order of the system. We will use the following SVD:

$$\widehat{\Gamma^f \mathcal{K}^p Z_0} = \begin{bmatrix} \mathcal{U} & \mathcal{U}_\perp \end{bmatrix} \begin{bmatrix} \Sigma_n & 0 \\ 0 & \Sigma \end{bmatrix} \begin{bmatrix} \mathcal{V} \\ \mathcal{V}_\perp \end{bmatrix}, \quad (4.11)$$

where Σ_n is the diagonal matrix containing the n largest singular values and \mathcal{V} is the corresponding row space. Note that we can find the largest singular values by detecting a gap between the singular values (Verhaegen and Verdult 2007). The state is now estimated by:

$$\hat{X} = \Sigma_n \mathcal{V}. \quad (4.12)$$

It is well known that when the state, input, output, and scheduling sequence are known the system matrices can be estimated (Nemani et al. 1995; Lovera 1997; Verdult and Verhaegen 2002). First we use (4.1b), which is now a linear relation in C and D and where e_k represents white noise. From this equation an estimate can be found of the C and D matrix while also the noise sequence can be estimated. The estimated noise sequence is used to transform (4.1a) into a linear expression depending on $A^{(i)}$, $B^{(i)}$, and $K^{(i)}$ and consequently all the system matrices can be estimated.

Table 4.1: Example of the matrix $\Gamma^p \mathcal{K}^p$ for $p = 3$, $m = 2$, and constant input matrix; $B^{(i)} = 0$ for $i > 1$.

$$\Gamma^3 \mathcal{K}^3 \approx C \left[\begin{array}{cccc|cc|c} \tilde{A}^{(1)} \tilde{A}^{(1)} \overline{B}^{(1)} & \tilde{A}^{(1)} \tilde{A}^{(2)} \overline{B}^{(1)} & \tilde{A}^{(2)} \tilde{A}^{(1)} \overline{B}^{(1)} & \tilde{A}^{(2)} \tilde{A}^{(2)} \overline{B}^{(1)} & \tilde{A}^{(1)} \overline{B}^{(1)} & \tilde{A}^{(2)} \overline{B}^{(1)} & \overline{B}^{(1)} \\ 0 & 0 & 0 & 0 & \tilde{A}^{(1)} \tilde{A}^{(1)} \overline{B}^{(1)} & \tilde{A}^{(1)} \tilde{A}^{(2)} \overline{B}^{(1)} & \tilde{A}^{(1)} \overline{B}^{(1)} \\ 0 & 0 & 0 & 0 & 0 & 0 & \tilde{A}^{(1)} \tilde{A}^{(1)} \overline{B}^{(1)} \end{array} \right].$$

Table 4.2: Total number of rows in the matrix Z_i for $m = 4$ and $r = \ell = 1$

	$i = 0$	$i = 1$	$i = 2$	$i = 3$	$i = 4$
$p = 2$	40	168	680	2728	10920
$p = 3$	168	680	2728	10920	43688
$p = 4$	680	2728	10920	43688	174760
$p = 5$	2728	10920	43688	174760	699048

4.4.3 ‘Curse of dimensionality’

Similar as in [Verdult and Verhaegen \(2002\)](#) the presented method suffers from the ‘curse of dimensionality’. The number of rows of Z_i grow exponentially with the past and future window. The number of rows of Z_i is given by:

$$\rho_{Z_i} = (r + \ell) \sum_{j=1}^{p+i} m^j.$$

In [Table 4.2](#) the ‘curse of dimensionality’ is demonstrated. Observe that the growth of the dimensions is almost similar to the work of [Verdult and Verhaegen \(2002\)](#) if we consider the LPV-PBSID algorithm (see [Table 1](#) in [Verdult and Verhaegen \(2002\)](#)). However, we would like to stress that for the LPV-PBSID_{opt} algorithm we only need Z_0 and consequently resulting is a smaller set of regression parameters that we have to estimate. So, we can conclude that the computational complexity of the LPV-PBSID is similar to the algorithm presented in [Verdult and Verhaegen \(2002\)](#) but is much higher compared to the LPV-PBSID_{opt} case. Still, the ‘curse of dimensionality’ plays an important role. In the next section the kernel method is derived for the LPV-PBSID_{opt} and the LPV-PBSID algorithm.

4.4.4 Summary of the LPV algorithms

The predictor-based subspace algorithms are presented in a conceptual way. No explicit algorithm recipe is given. This is due to the fact that the construction of the matrices [\(4.9\)](#) and [\(4.10\)](#) is quite cumbersome. In the next section we will derive explicit formulas for the algorithms. A sketch of the algorithms is now given below:

Algorithm 4.1 (LPV-PBSID_{opt})

The LPV-PBSID_{opt} algorithm can be summarized as follows:

1. Create the matrices U_i , Y_i , and Z_i using (4.7a), (4.7b) and (4.7c) for $i = 0$,
2. Solve the linear problem given in (4.8) for $i = 0$,
3. Construct $\Gamma^f \mathcal{K}^p Z_0$ using (4.7c) and (4.10),
4. Compute the state sequence using (4.11) and (4.12),
5. With the estimated state, use the linear relations (4.1a)-(4.1b) to obtain the system matrices.

Algorithm 4.2 (LPV-PBSID)

The LPV-PBSID algorithm can be summarized as follows:

1. Create the matrices U_i , Y_i , and Z_i using (4.7a), (4.7b) and (4.7c) for all $i \in \{0, \dots, f-1\}$,
2. Solve the linear problem given in (4.8) for all $i \in \{0, \dots, f-1\}$,
3. Construct $\Gamma^f \mathcal{K}^p Z_0$ using (4.7c) and (4.9),
4. Compute the state sequence using (4.11) and (4.12),
5. With the estimated state, use the linear relations (4.1a)-(4.1b) to obtain the system matrices.

4.5 Kernel method

The LPV identification methods presented in the previous paragraph suffer from the ‘curse of dimensionality’. However, similar as in [Verdult and Verhaegen \(2005\)](#) we can use the kernel method to overcome this drawback. In Section 4.5.1 we present the kernel method for the proposed LPV identification schemes. In Section 4.5.2, a computationally efficient formula is presented for the proposed model structure. The kernel method is normally ill-conditioned, but in Section 4.5.3 regularization is proposed to overcome this drawback. In the last subsection a summary of the algorithms with kernels is given.

4.5.1 Kernel Method

The LPV identification approach presented in the previous section resulted in a linear problem formulated in (4.8), from now on also referred to as primal problem. This equation can be solved by using traditional Least Squares (LS). How-

ever, the data matrices grow exponentially with the past window for LPV-PBSID_{opt} and with the past and future window for LPV-PBSID. In [Verdult and Verhaegen \(2005\)](#) it was shown that the solution of this linear problem is equal to the solution of the dual problem if the solution with the minimum-norm is considered ([Golub and Loan 1996](#)). In this subsection we show how the kernel method can be exploited for the presented LPV identification schemes.

The linear problem in (4.8) has a unique solution if the matrix:

$$\begin{bmatrix} Z_i \\ U_i \end{bmatrix},$$

has full row rank and this solution is given by:

$$\begin{bmatrix} \widehat{CK^{p+i}} & \widehat{D} \end{bmatrix} = Y_i \begin{bmatrix} Z_i^T & U_i^T \end{bmatrix} \left(\begin{bmatrix} Z_i \\ U_i \end{bmatrix} \begin{bmatrix} Z_i^T & U_i^T \end{bmatrix} \right)^{-1}.$$

When the matrix $\begin{bmatrix} Z_i^T & U_i^T \end{bmatrix}^T$ has missing row rank the solution is not unique. This will occur when the past window is large. However, the solution with the smallest norm, $\min \| \begin{bmatrix} CK^{p+i} & D \end{bmatrix} \|_{F^r}^2$, can still be computed by using the SVD of the matrix:

$$\begin{bmatrix} Z_i \\ U_i \end{bmatrix} = \begin{bmatrix} \mathcal{U}_i & \mathcal{U}_{\perp,i} \end{bmatrix} \begin{bmatrix} \Sigma_{n,i} & 0 \\ 0 & 0 \end{bmatrix} \begin{bmatrix} \mathcal{V}_i^T \\ \mathcal{V}_{\perp,i}^T \end{bmatrix},$$

where $\Sigma_{n,i}$ is the diagonal matrix containing the non-zero singular values and \mathcal{V}_i^T and \mathcal{U}_i are the corresponding row and column space, respectively. The solution with the minimum-norm is now given by:

$$\begin{bmatrix} \widehat{CK^{p+i}} & \widehat{D} \end{bmatrix} = Y \mathcal{V}_i \Sigma_{n,i}^{-1} \mathcal{U}_i^T.$$

The computations take place in a large dimensional space spanned by the columns of Z_i . If we consider the minimum-norm solution of (4.8) the dual problem avoids computations in this large dimensional space ([Suykens et al. 2002](#)). The dual problem results in:

$$\min_{\alpha_i} \|\alpha_i\|_{F^r}^2 \quad \text{with} \quad Y_i - \alpha_i \begin{bmatrix} Z_i^T Z_i + U_i^T U_i \end{bmatrix} = 0, \quad (4.13)$$

where α_i are the Lagrange Multipliers and $\begin{bmatrix} Z_i^T Z_i + U_i^T U_i \end{bmatrix}$ is referred to as the kernel matrix. If the matrix $\begin{bmatrix} Z_i^T & U_i^T \end{bmatrix}^T$ has full column rank the solution to this dual problem is given by:

$$\begin{aligned} \hat{\alpha}_i &= Y_i \left(\begin{bmatrix} Z_i^T Z_i + U_i^T U_i \end{bmatrix} \right)^{-1}, \\ &= Y_i \mathcal{V}_i \Sigma_{n,i}^{-2} \mathcal{V}_i^T. \end{aligned}$$

The estimate of $[CK^{p+i}, D]$ is now given by:

$$\begin{aligned} [\widehat{CK^{p+i}}, \widehat{D}] &= \widehat{\alpha}_i [Z_i^T, U_i^T], \\ &= Y_i \mathcal{V}_i \Sigma_{n,i}^{-1} \mathcal{U}_i^T. \end{aligned}$$

The construction of the matrix $\widehat{CK^{p+i}}$ from the dual problem requires the matrix Z_i , explicitly. However, due to the ‘curse of dimensionality’ this can lead to dimension problems. For the construction of $CK^{p+i}Z_i$ we do not need the matrix Z_i explicitly, we only have to construct $Z_i^T Z_i$ and $U_i^T U_i$ for the computation of α_i and an estimate of $CK^{p+i}Z_i$ is given by:

$$\widehat{CK^{p+i}}Z_i = \widehat{\alpha} Z_i^T Z_i. \quad (4.14)$$

With this above we can not reconstruct the extended controllability matrix times the state sequence directly. However, in the previous section we used these estimates to build the matrix given in (4.9) and (4.10). We mentioned that this was a rather cumbersome task. However, in the next two lemma’s things clear up:

Lemma 4.2 (Extended observability matrix times state sequence (LPV-PBSID_{opt}))

Given the model structure in (4.1a)-(4.1b) and Definition 4.4, we can define the following matrices:

$$Z^{i,j} = [P_{p-j|j-i} z_{j-i}, \quad \dots, \quad P_{p-j|\overline{N}+j-i-1} z_{\overline{N}+j-i-1}],$$

with $Z^{i,j} \in \mathbb{R}^{m^{p-j+1}(r+\ell) \times \overline{N}}$ and $\overline{N} = N - p - f + 1$. Now we have:

$$C \left(\tilde{A}^{(1)} \right)^i \mathcal{L}_{p-j} = \alpha_0 (Z^{i,j})^T,$$

and then we can construct the matrix $\Gamma^f \mathcal{K}^p Z_0$ as follows:

$$\Gamma^f \mathcal{K}^p Z_0 = \begin{bmatrix} \alpha_0 \sum_{j=0}^{p-1} (Z^{0,j})^T Z^{0,j} \\ \alpha_0 \sum_{j=1}^{p-1} (Z^{1,j})^T Z^{0,j} \\ \vdots \\ \alpha_0 \sum_{j=p-1}^{p-1} (Z^{f-1,j})^T Z^{0,j} \end{bmatrix}. \quad (4.15)$$

Proof: The proof follows from the derivation of the dual problem. \square

Lemma 4.3 (Extended observability matrix times state sequence (LPV-PBSID))

Given the model structure in (4.2a)-(4.2b) and the Definition 4.4. Now we have:

$$C \left(\tilde{A}^{(1)} \right)^i \mathcal{K}^p = \alpha_i Z_0^T.$$

So with $\alpha = [\alpha_0^T, \alpha_1^T, \dots, \alpha_{f-1}^T]^T$. We can construct the matrix $\Gamma^f \mathcal{K}^p Z_0$ as

follows

$$\Gamma^f \mathcal{K}^p Z_0 = \alpha Z_0^T Z_0. \quad (4.16)$$

Proof: The proof follows from the derivation of the dual problem. \square

With one of the previous lemma's we can go back to the original problem and compute an SVD of this estimate to find the state sequence. Again it is important to stress that we do not require Z_i but we only need $Z_i^T Z_i$. This observation makes it possible to derive a computationally more efficient implementation.

4.5.2 Computation of the kernel matrices

In the previous subsection it was already stressed that we do not have to compute Z_i but we only need $Z_i^T Z_i$. In this section an analytical expression is given that does not require the calculation of Z_i . First we define the matrix

$$\tilde{N} = [0, \quad 1, \quad \dots, \quad N - f - p],$$

and the following lemma:

Lemma 4.4 *Given the vectors $\lambda_1, \lambda_2, \dots, \lambda_\nu \in \mathbb{R}^{\kappa \times 1}$ and $\theta_1, \theta_2, \dots, \theta_\nu \in \mathbb{R}^{\kappa \times 1}$ the product*

$$(\lambda_1 \otimes \lambda_2 \otimes \dots \otimes \lambda_\nu)^T (\theta_1 \otimes \theta_2 \otimes \dots \otimes \theta_\nu),$$

is given by:

$$\prod_{j=1}^{\nu} \lambda_j^T \theta_j. \quad (4.17)$$

Proof: With the properties of the Kronecker product (\otimes) defined in [Brewer \(1978\)](#), $(A \otimes B)(C \otimes D) = AC \otimes BD$, we can rewrite (4.4) as: $\lambda_1^T \theta_1 \otimes \dots \otimes \lambda_\nu^T \theta_\nu$ and observing that all the elements between the Kronecker products are scalars results in (4.17). \square

With Lemma 4.4 we can define the kernels for the model structure given in (4.1a)-(4.1b) for both LPV-PBSID_{opt} and LPV-PBSID.¹

Theorem 4.1 (Kernels for LPV-PBSID_{opt}) *Given Lemma 4.4 and the model structure given in (4.1a)-(4.1b) we have for $j \geq i$*

$$(Z^{i,j})^T Z^{0,j} = \left(\prod_{v=0}^{p-j-1} \mu_{\tilde{N}+v+j-i}^T \mu_{\tilde{N}+v+j} \right) \left(z_{\tilde{N}+j-i}^T z_{\tilde{N}+j} \right)$$

¹We define $\mu_{\tilde{N}+v} = [\mu_v, \quad \dots, \quad \mu_{N-f-p+v}]$.

and

$$Z_0^T Z_0 = \sum_{j=0}^{p-1} (Z^{0,j})^T Z^{0,j}.$$

We can solve (4.13) for $i = 0$ and construct (4.15).

Proof: Using Lemma 4.4 the proof follows by straightforward computations. \square

Theorem 4.2 (Kernels for LPV-PBSID) *Given Lemma 4.4 and the model structure given in (4.1a)-(4.1b) we have:*

$$Z_i^T Z_i = \sum_{j=0}^{p+i-1} \left(\left(\prod_{v=j}^{p+i-1} \mu_{\tilde{N}+v}^T \right) \left(z_{\tilde{N}+j}^T z_{\tilde{N}+j} \right) \right)$$

We can solve (4.13) for all $i \in \{0, \dots, f-1\}$ and construct (4.16).

Proof: Using Lemma 4.4 the proof follows by straightforward computations. \square

For $N \gg p$ and $N \gg m$ the computational complexity of the indirect computation, so first constructing Z_i and then computing the product $(Z_i^T Z_i)$, is of order $\mathcal{O}(N^2 \hat{q})$ with $\hat{q} = \sum_{j=1}^p m^j$ and direct construction of $(Z_i^T Z_i)$ is of the order $\mathcal{O}(N^2)$, which illustrates the computational efficiency of the algorithm.

The kernels are valid for the model structure given in (4.1a)-(4.1b). The whole derivation of the kernel matrices can be repeated for different model structures as well and these kernels can be found in Appendix C.

4.5.3 Regularization

The kernel $Z_i^T Z_i$ described in the previous paragraph is square and has the size of the number of data points available. This normally leads to an ill-conditioned set of equations. This conditioning problem can be circumvented through regularization. There are a number of regularization techniques (for a detailed overview see Sima (2006)). In Verdult and Verhaegen (2005) a simulation study is performed to select the optimal regularization technique and corresponding regularization parameter selection method. In this study they conclude that Tikhonov regularization (Tikh) with Generalized Cross Validation (GCV) regularization parameter selection gives the best result. In this thesis we will adopt these settings.

4.5.4 Summary of the algorithm

We end this section with the summary of the closed-loop kernel LPV identification algorithms.

Algorithm 4.3 (LPV-PBSID_{opt} (kernel))

The algorithm can be summarized as follows:

1. Create the matrices $(Z^{i,j})^T Z^{i,j}$ and $Z_0^T Z_0$ using Theorem 4.1 and compute $U_0^T U_0$ using (4.7a) for $i = 0$,
2. Solve the linear problem given in (4.13) for $i = 0$. If desired regularized,
3. Construct $\Gamma^f \mathcal{K}^p Z_0$ using (4.15),
4. Compute the state sequence using (4.11) and (4.12),
5. With the estimated state, use the linear relations (4.1a)-(4.1b) to obtain the system matrices.

Algorithm 4.4 (LPV-PBSID (kernel))

The algorithm can be summarized as follows:

1. Create the matrices $Z_i^T Z_i$ using Theorem 4.2 and compute $U_0^T U_0$ using (4.7a) for all $i \in \{0, \dots, f-1\}$,
2. Solve the linear problem given in (4.13) for all $i \in \{0, \dots, f-1\}$. If desired regularized,
3. Construct $\Gamma^p \mathcal{K}^p Z_0$ using (4.16),
4. Compute the state sequence using (4.11) and (4.12),
5. With the estimated state, use the linear relations (4.1a)-(4.1b) to obtain the system matrices.

4.6 Different model structures

For the derivation of the LPV identification algorithm we used the model structure given in (4.1a)-(4.1b). This model structure has a parameter-invariant output equation. The main reason to make this assumption is that the derivation of the algorithm becomes more cumbersome with parameter-varying output equation. Another reason could be that most applications also have a time-invariant output equation. This argument will not hold because converting these first principles models to discrete time normally results in a parameter-varying output equation (this will be demonstrated in Section 4.9).

In this section we discuss other model structures. We start with the model structure with parameter-varying output equation but with a constant K and we discuss the consequences for the proposed algorithm. Followed by a discussion

about the model structure where all the matrices are parameter-varying. Finally, we discuss the bilinear model structure.

4.6.1 Parameter-varying output equation and constant K

In this subsection we consider the following model structure:

$$x_{k+1} = \sum_{i=1}^m \mu_k^{(i)} \left(A^{(i)} x_k + B^{(i)} u_k \right) + K e_k, \quad (4.18a)$$

$$y_k = \sum_{i=1}^m \mu_k^{(i)} \left(C^{(i)} x_k + D^{(i)} u_k \right) + e_k, \quad (4.18b)$$

which has a parameter-varying output equation but a parameter-invariant K matrix. The predictor representation of this model structure is given by:

$$x_{k+1} = \sum_{i=1}^m \mu_k^{(i)} \left(\tilde{A}^{(i)} x_k + \tilde{B}^{(i)} u_k \right) + K y_k, \quad (4.19a)$$

$$y_k = \sum_{i=1}^m \mu_k^{(i)} \left(C^{(i)} x_k + D^{(i)} u_k \right) + e_k, \quad (4.19b)$$

with

$$\tilde{A}^{(i)} = A^{(i)} - K C^{(i)}, \quad \tilde{B}^{(i)} = B^{(i)} - K D^{(i)}.$$

This model structure will affect the proposed identification method at two points. The factorization presented in Section 4.3 is affected by the fact that K is parameter-invariant and this has consequences for the data matrices. In Appendix C the kernels are given for different model structures and also for a parameter-invariant K . The second issue is caused by the parameter-varying output equation. The linear problem given by (4.8) is now also dependent on: $C^{(2)} \mathcal{K}^{p+i}, \dots, C^{(m)} \mathcal{K}^{p+i}$ and $D^{(2)}, \dots, D^{(m)}$ but still the terms $C^{(1)} \mathcal{K}^{p+i}$ are estimated and these terms are again used to construct the extended observability matrix times the extended LPV controllability matrix. From this point on the algorithm proceeds in a similar way as discussed in the previous sections. In Appendix C explicit formulas are given for LPV system identification for this particular model structure.

4.6.2 Parameter-varying output equation and parameter-varying K

In this subsection we consider the following model structure:

$$x_{k+1} = \sum_{i=1}^m \mu_k^{(i)} \left(A^{(i)} x_k + B^{(i)} u_k + K^{(i)} e_k \right), \quad (4.20a)$$

$$y_k = \sum_{i=1}^m \mu_k^{(i)} \left(C^{(i)} x_k + D^{(i)} u_k \right) + e_k, \quad (4.20b)$$

which has a parameter-varying output equation and a parameter-varying K matrix. The predictor representation of this model structure is given by:

$$x_{k+1} = \sum_{i=1}^m \mu_k^{(i)} \left(A^{(i)} x_k + B^{(i)} u_k + K^{(i)} y_k - K^{(i)} \sum_{j=1}^m \mu_k^{(j)} \left(C^{(j)} x_k + D^{(j)} u_k \right) \right),$$

$$y_k = \sum_{i=1}^m \mu_k^{(i)} \left(C^{(i)} x_k + D^{(i)} u_k \right) + e_k,$$

If we compare this model structure with the model structure with parameter-invariant K we see a significant increase of complexity. With respect to the previous subsection we have to change the factorization of the time-varying controllability matrix. This factorization can be obtained with a similar recipe as presented in this chapter. However, the consequence is that the ‘curse of dimensionality’ will play an even more important role. However, to circumvent this problem we assume the following predictor model:

$$x_{k+1} = \sum_{i=1}^m \mu_k^{(i)} \left(\tilde{A}^{(i)} x_k + \tilde{B}^{(i)} u_k + K^{(i)} y_k \right), \quad (4.21a)$$

$$y_k = \sum_{i=1}^m \mu_k^{(i)} \left(C^{(i)} x_k + D^{(i)} u_k \right) + e_k, \quad (4.21b)$$

with

$$\tilde{A}^{(i)} = A^{(i)} - K^{(i)} C^{(i)}, \quad \tilde{B}^{(i)} = B^{(i)} - K^{(i)} D^{(i)}.$$

This model structure is also considered in the appendix.

4.6.3 Bilinear systems

Bilinear systems are nonlinear systems due to the product of the states with the input. Many important processes in engineering, system biology, and economics may be modeled by bilinear systems (Bruno et al. 1974). Due to their nice properties bilinear system identification received considerable attention in recent years.

We consider a discrete time bilinear system of the form:

$$x_{k+1} = A^{(1)} x_k + \sum_{i=2}^{r+1} u_k^{(i-1)} \left(A^{(i)} x_k \right) + B u_k + K e_k, \quad (4.22a)$$

$$y_k = C x_k + D u_k + e_k. \quad (4.22b)$$

This model structure can be considered to be a subset of the LPV model structure given in (4.1a)-(4.1b). However, now we have a parameter-invariant input and noise matrix and the scheduling is given by the input. For this model structure we can directly use the proposed algorithm described in this chapter. For explicit formulas we refer to Appendix C.

4.7 Dedicated scheduling sequences

Recently, a number of papers appeared where the structure of the scheduling sequence is exploited to overcome the computational complexity of the general LPV identification approach (see also Chapter 3). It turns out that if periodic scheduling (Felici et al. 2007b; van Wingerden et al. 2008a), piecewise constant scheduling (Verdult and Verhaegen 2004; van Wingerden et al. 2007; van Wingerden and Verhaegen 2007), or white noise scheduling (Favoreel et al. 1999; Santos et al. 2005, 2006) is used well-established LTI subspace techniques can be used to identify LPV systems or bilinear systems. In Felici et al. (2007b); van Wingerden et al. (2007); van Wingerden and Verhaegen (2007); van Wingerden et al. (2008a) they have to solve a numerical sensitive intersection problem and in Verdult and Verhaegen (2004); van Wingerden et al. (2007); van Wingerden and Verhaegen (2007) the local models are interconnected by formulating a number of least squares problems, which require accurate local models. In Santos et al. (2005, 2006) an iterative approach is used to obtain the system matrices. In this section we show how structure in the scheduling sequence further reduces the computational complexity of the algorithm presented in this chapter.

4.7.1 Periodic scheduling

We will use the kernel algorithm to show that the computational complexity significantly reduces when the scheduling is periodic, piecewise constant or is structured in some sense.

The matrix $Z_0^T Z_0$ is given by:

$$Q^T \begin{bmatrix} (N_0^p)^T N_0^p & (N_0^p)^T N_1^p & \cdots & (N_0^p)^T N_{\overline{N}}^p \\ (N_1^p)^T N_0^p & (N_1^p)^T N_1^p & \cdots & (N_1^p)^T N_{\overline{N}}^p \\ \vdots & & \ddots & \\ \left(N_{\overline{N}}^p\right)^T N_0^p & \left(N_{\overline{N}}^p\right)^T N_1^p & \cdots & \left(N_{\overline{N}}^p\right)^T N_{\overline{N}}^p \end{bmatrix} Q,$$

with $Q = \text{diag}([\overline{z}_0^p, \overline{z}_2^p, \dots, \overline{z}_{\overline{N}}^p])$ and $\overline{N} = N - f - p$. If the system is periodic with period π we observe that $N_i = N_{i+\pi}$ and consequently the matrix $Z_0^T Z_0$ will lose rank. In the kernel method we are only interested in the row space of $CK^p Z_0$. In the previous section an estimate of $CK^p Z_0$ was given by $\alpha_0 Z_0^T Z_0$ if the matrix $Z_0^T Z_0$ has no full row rank we can select the rows that span the row space of this matrix without altering the row space of $CK^p Z_0$. This means that we only have to select the rows of $Z_0^T Z_0$ that span the row space of this matrix. For periodic scheduling it is sufficient to select the first $\pi \times (r + \ell)p$ rows if the matrix Q has full rank. For piecewise constant scheduling (van Wingerden et al. 2007) the same time-variation will be present a number of times and in this case the matrix $Z_i^T Z_i$ will lose rank. For this situation it is harder to select the rows that span the row space of $Z_i^T Z_i$. There are more scheduling sequences where the same or almost the same time-variation in the dynamics occurs. In the next

subsection we present a sketch of an algorithm how to select the correct rows for structured scheduling sequences.

4.7.2 Kernel selection

To exploit structure like periodic or piecewise constant scheduling it is important to select the linear independent rows of $Z_i^T Z_i$. However, we can also select the rows that are independent up to a certain threshold, ϵ . In this way only the most dominant rows are selected. There are different methods to select the most dominant rows but we summarize the most straightforward approach in Algorithm 4.5.

Algorithm 4.5 (Kernel selection algorithm using LPV-PBSID_{opt})

The algorithm can be summarized as follows:^a

Init Give a tolerance level ϵ

define the matrix:

$$\mathcal{Z} = ((Z_0)^T(Z_0))(1, :) + ((U_0)^T(U_0))(1, :)$$

$$\mathcal{D} = \{1\}$$

For $j=2:\bar{N}$

Compute:

$$\bar{\mathcal{Z}} = ((Z_0)^T(Z_0))(j, :) + ((U_0)^T(U_0))(j, :)$$

If $\min_{\Xi} \|\mathcal{Z}\Xi - \bar{\mathcal{Z}}\|_F^2 > \epsilon$

$$\mathcal{Z} = \begin{bmatrix} \mathcal{Z}^T & \bar{\mathcal{Z}}^T \end{bmatrix}^T$$

Add j to the set \mathcal{D}

End

End

Solve $\min_{\tilde{\alpha}} \|Y_0 - \tilde{\alpha}_0 \mathcal{Z}\|_F^2$. The estimate of the extended observability matrix is now given by (4.15) where only the rows indicated by the set \mathcal{D} of $\sum_{j=i}^{p-1} (Z^{i,j})^T Z^{0,j}$ are used. Similar as in the original algorithm an SVD can be computed to obtain an estimated of the state and consequently the system matrices can be estimated.

^aMatlab notation is used for convenience

With this algorithm the number of selected rows can still be large. More advanced kernel selection algorithms can be used to select the most dominant rows of $Z_0^T Z_0$. In [Suykens et al. \(2002\)](#) and [Espinoza et al. \(2006\)](#) they use fixed-sized kernels. In machine learning literature more methods can be found on how to select a fixed number of kernels (see for instance [Smola and Scholkopf \(2000\)](#)).

The advantage of the method presented above is that we can deal with large data sets when the row space of $Z_0^T Z_0$ is small. In the next section we present simulation examples that show the potential of the proposed LPV identification algorithm. We also show that we can deal with a large data set if we have periodic scheduling.

4.8 Simulation Examples

In this section we show some features of the novel algorithm on three different simulation examples.

4.8.1 Example 1: Open-loop LPV identification

In the first example we demonstrate the effectiveness of the LPV-PBSID_{opt} algorithm on an open-loop problem. Furthermore, we explore the effect of regularization and the window size. We use the benchmark model used in [Verduyt and Verhaegen \(2000, 2002, 2005\)](#). This is a fourth order MIMO open-loop LPV model with $m = 4$, $r = 2$, and $\ell = 3$. The collected data u_k , y_k , and μ_k are used for the identification algorithm. The system matrices are given by:

$$\begin{aligned}
 A^{(1)} &= \begin{bmatrix} -1.3 & -0.6325 & -0.1115 & 0.0596 \\ 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix}, \\
 A^{(2)} &= \begin{bmatrix} -0.51 & -0.1075 & -0.007275 & -0.0000625 \\ 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix}, \\
 A^{(3)} &= \begin{bmatrix} 0.2 & 0 & 0 & 0 \\ 0 & 0.4 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix}, & A^{(4)} &= \begin{bmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0.3 & 0 \\ 0 & 0 & 0 & 0.3 \end{bmatrix}, \\
 B^{(1)} &= \begin{bmatrix} 0 & 1 \\ 1 & 0 \\ 1 & 0 \\ 0 & 1 \end{bmatrix}, & B^{(2)} = B^{(3)} = B^{(4)} &= \begin{bmatrix} 0 & 0 \\ 0 & 0 \\ 0 & 0 \\ 0.3 & 0.3 \end{bmatrix}, \\
 K &= \begin{bmatrix} 0.16 & 0 & 0 \\ 0 & 0.16 & 0 \\ 0 & 0 & 0.16 \\ 0.16 & 0 & 0 \end{bmatrix}, & C &= \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix}, & D &= \begin{bmatrix} 0 & 0 \\ 0 & 0 \\ 0 & 0 \end{bmatrix}.
 \end{aligned}$$

The scheduling and input sequence are given by:

$$\mu_k = \begin{bmatrix} 1 \\ \rho_k \\ \frac{1}{2} \sin\left(\frac{2\pi k}{100}\right) \rho_k \\ \frac{1}{2} \cos\left(\frac{2\pi k}{100}\right) \rho_k \end{bmatrix}, \quad \text{and} \quad u_k = \begin{bmatrix} G(q)\eta_k + 4H(q)\nu_k \\ \xi_k \end{bmatrix},$$

where ρ_k is a uniformly distributed random sequence with unit variance, η_k , ν_k , and ξ_k are zero-mean white Gaussian noise sequences of unit variance, q denotes the shift operator, $G(q) = 0.75 + 1.05q^{-1} + 0.15q^{-2}$, and $H(q)$ is a second-order

Table 4.3: The VAF on a fresh data set for 100 monte carlo simulations. The experiments are performed for different settings.

$p = 3$	Kernel LPV-PBSID _{opt}			Kernel LPV-PBSID _{opt} +Tikh+GCV		
	Output 1	Output 2	Output 3	Output 1	Output 2	Output 3
N=500, SNR=40	93.2	93.5	93.8	96.8	96.7	96.8
N=500, SNR= ∞	100.0	100.0	100.0	99.9	100.0	100.0
N=1000, SNR=40	98.0	98.0	98.0	98.4	98.4	98.5
N=1000, SNR= ∞	100.0	100.0	100.0	99.9	99.9	99.9
	VAF($y_k, y_k^{(p)}$)					
	Output 1	Output 2	Output 3			
N=1000, SNR= ∞	54.0	34.5	15.7			
$p = 5$	Kernel LPV-PBSID _{opt}			Kernel LPV-PBSID _{opt} +Tikh+GCV		
	Output 1	Output 2	Output 3	Output 1	Output 2	Output 3
N=500, SNR=40	66.5	67.2	67.6	90.5	90.6	90.8
N=500, SNR= ∞	99.4	99.4	99.4	99.2	99.3	99.3
N=1000, SNR=40	17.2	19.1	21.0	94.8	95.0	95.1
N=1000, SNR= ∞	100.0	100.0	100.0	100.0	100.0	100.0
	VAF($y_k, y_k^{(p)}$)					
	Output 1	Output 2	Output 3			
N=1000, SNR= ∞	70.8	67.5	56.8			

low-pass Butterworth filter with a cut-off frequency of one-fifth of the sampling frequency.

The performance of the identified system is evaluated by looking at the value of the Variance-Accounted-For (VAF) on a data set different from the one used for identification. The VAF value is defined as:

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

where \hat{y}_k denotes the output signal obtained by simulating the identified LPV system, y_k is the output signal of the true LPV system, and $\text{var}(\cdot)$ denotes the variance of a quasi-stationary signal. To investigate the sensitivity of the identification algorithm with respect to output and process noise, a Monte-Carlo simulation with 100 runs was carried out. For each of the 100 simulations a different realization of the input u_k and scheduling sequence μ_k is used.

In Table 4.3 the results of the different identification methods are summarized, where the VAF values are based on a validation data set. If we look at the identification results for the system with $N = 1000$ and no noise (SNR= ∞) the results are significantly better than the results presented in [Verdult and Verhaegen \(2005\)](#). Remarkably, we observe that the VAF is 100% for a finite p . However, if we compute the VAF between y_k and $y_k^{(p)}$ defined in (4.6), which basically indicates the approximation we make, the VAF values are significantly smaller. This stresses the point that although the approximation error is rather large we can not predict how large the bias term is and how it transfers to the final estimate of the system matrices. That is why from a theoretical point of view we can state that it is better to pick a large p because then the consistency can be proven ([Verdult and Verhaegen 2002](#)). However, a large p , due to the ‘curse of dimensionality’, and a finite N implies a large number of variables to be estimated leading to minimum-norm solutions with a larger variance. From this perspective it is better to choose a small p . In this particular simulation example this trade-off already appears for a small p . Estimation with a finite p is an interesting research field for both LTI and LPV system identification. In Section 4.8.3 we clearly show the trade-off between bias and variance on a simple bilinear identification example.

4.8.2 Example 2: Closed-loop LPV identification

In this section we demonstrate the operation of the algorithm with data collected in closed-loop, we compare the LPV-PBSID_{opt} and LPV-PBSID, and we exploit the structure in the scheduling sequence. We use the model described in [Felici et al. \(2007b\)](#) (and in Section 3.8.1), which might represent the flapping dynamics of a wind turbine blade ([Eggleston and Stoddard 1987](#)). To apply state feedback we now assume that this model has an identity output matrix. We take a past and future window of 8 and limit ourselves to 2000 data points. To show the potential of the closed-loop setting a time-varying state feedback is used. The controller is synthesized using a discrete time periodic Riccati equation ([Hench and Laub 1994; Varga 2005](#)). Process noise is added with a variance of $\sqrt{0.1}$.

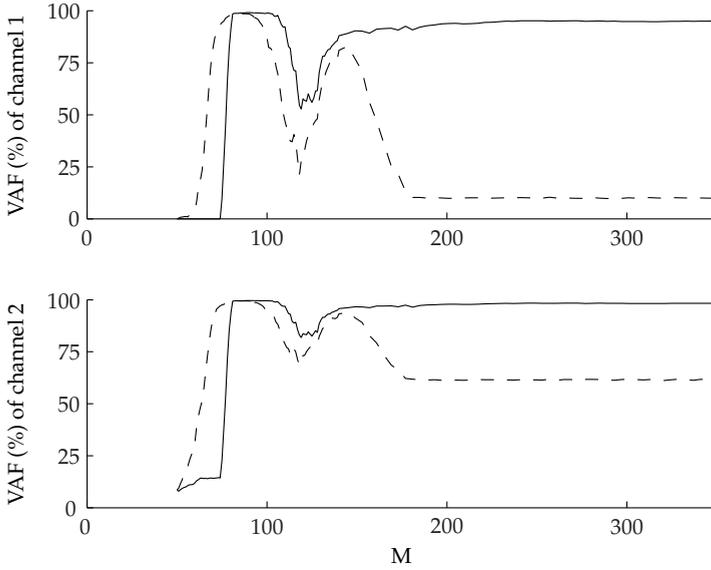


Figure 4.1: The VAF for 100 Monte Carlo simulations as a function of the numbers of rows of $Z_i^T Z_i$ that taken into account. The dashed line and the solid line represent the regularized and the un-regularized version of the LPV-PBSID_{opt} algorithm.

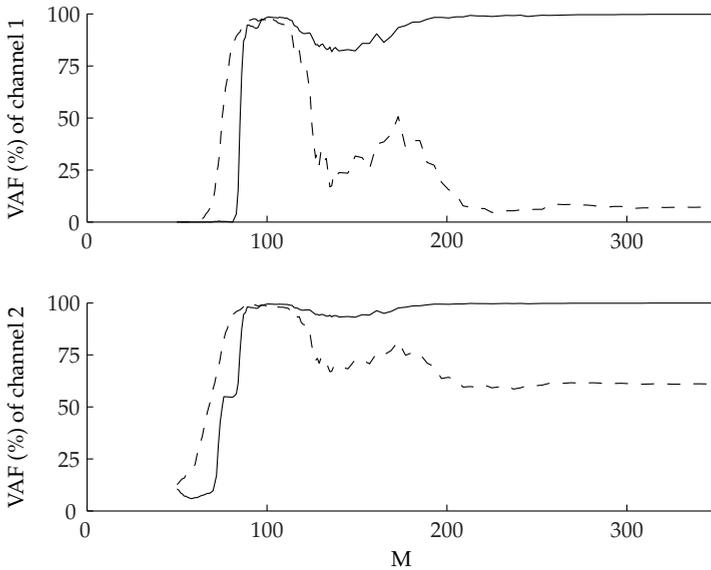


Figure 4.2: The VAF for 100 Monte Carlo simulations as a function of the numbers of rows of $Z_i^T Z_i$ that taken into account. The dashed line and the solid line represent the regularized and the un-regularized version of the LPV-PBSID algorithm.

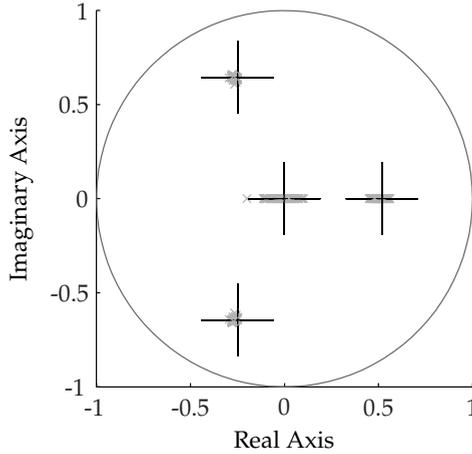


Figure 4.3: Eigenvalues of the estimated $A^{(1)}$ and $A^{(2)}$ in one plot for the LPV-PBSID_{opt} algorithm, for 100 experiments and $M = 100$. The big crosses correspond to the real values of the eigenvalues of the matrices.

An LPV model was identified using the kernel version of the LPV-PBSID_{opt} and the LPV-PBSID algorithm and we exploit the fact that the scheduling sequence contains structure, it is periodic. We take the first M rows of $Z_i^T Z_i$ into account. As mentioned earlier there are smarter ways to select these rows, however, this will illustrate the effectiveness of the approach. In Figure 4.1 and 4.2 the mean VAF value as a function of M is presented for the regularized and the unregularized LPV-PBSID_{opt} and LPV-PBSID algorithm, respectively. We observe that we can select an optimum of 100 rows and we see that regularization is not required anymore. Furthermore, we observe that for this particular simulation example the LPV-PBSID_{opt} and LPV-PBSID algorithm have a similar performance.

In Figure 4.3 the eigenvalues of the estimated models are compared with their true values. As expected the closed-loop algorithm gives consistent results but a small bias arises due to the approximation made in the algorithm.

4.8.3 Example 3: Bilinear identification

In the previous two examples we demonstrated a number of key features of the proposed algorithm. In this example we demonstrate the effect of the past window using a low order open loop bilinear model. As indicated in this chapter a small p will lead in general to biased estimates while for large p , even without noise, the variance will increase due to the ‘curse of dimensionality’. The bilinear system is given by:

$$\left[A^{(1)}, A^{(2)} \right] = \left[\begin{array}{cc|cc} 0.5 & 0.5 & 0.2 & 0.2 \\ -0.5 & 0.5 & -0.2 & 0.2 \end{array} \right],$$

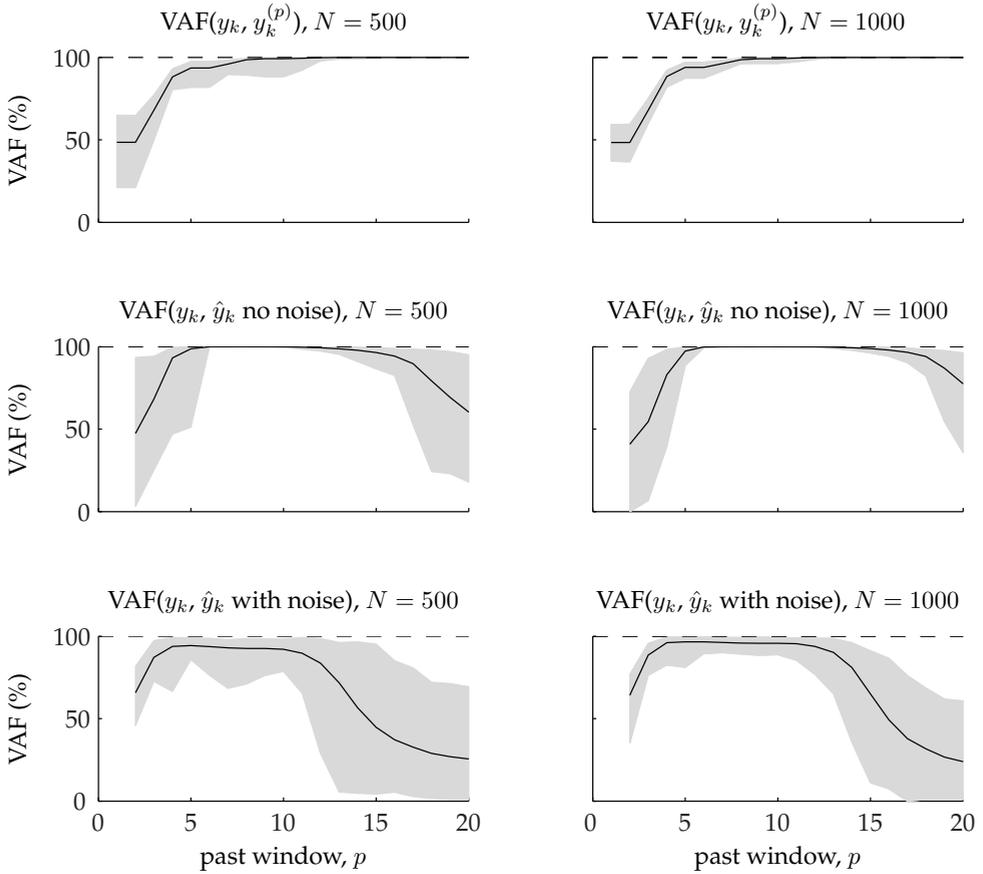


Figure 4.4: The mean value of the VAF over 100 experiments (solid line) is presented with respect to the past window size using 500 (left) and 1000 (right) data points. The VAF values of the 100 experiments are within the gray confidence region.

$$B = \begin{bmatrix} 1 \\ 0 \end{bmatrix}, \quad C = [1 \quad 1], \quad D = [0].$$

The input is generated by filtering a zero-mean white noise sequence with a fourth-order low pass Butterworth filter that has a cutoff frequency of 0.8 times the Nyquist frequency. Because we work with a bilinear system the scheduling is given by $\mu_k = [1, u_k^T]^T$. To investigate the sensitivity of the identification algorithm with respect to the realization of the input and noise, a Monte-Carlo simulation with 100 runs was carried out. Compared with the previous two examples the complexity of this example is small, which enables us to clearly demonstrate the trade-off between a large p and a small p . In Figure 4.4 we demonstrate the effect of p on the amount of approximation we introduce and the consequence for the identification algorithm. The top figures present the VAF between y_k and

$y_k^{(p)}$, defined in (4.6), which basically indicates the amount of approximation we incorporated in the algorithm. The system is uniformly exponentially stable and therefore the approximation error will converge to zero if p goes to infinity. In the plots in the middle the VAF is given between the signal generated by the estimated model, using noise free data, and the validation data. In this figure we see that for really small p we have a biased estimate, while for large p the variance increases due to the fact that we look for a minimum-norm solution because the number of unknowns exceeds the number of data points. In the bottom figure, noise with an SNR of 15dB is added to the identification data and we clearly see a larger variance for a larger p . This simulation example illustrates the trade-off problem for the identification for both bilinear and LPV systems. With increasing complexity it is expected that it is more worthwhile to pick a small p , as already seen in the first example. However, if the bias is too large the biased estimate may serve as an initial estimate for optimization based identification algorithms (Lee and Poolla 1999; Verdult et al. 2003), which was already suggested in Verhaegen (2002).

4.9 Case study: a ‘smart’ airfoil

In this section we apply the proposed LPV identification approach on a ‘smart’ 2-D airfoil. This is an airfoil with at the trailing edge a control surface, trailing edge flap. The lay-out of such an airfoil is illustrated in Figure 4.5. A similar model is used for flutter control and detection and a detailed description of this model can be found in a large number of scientific papers (see for instance Ko et al. (1997); Zeng and Singh (1998); Block and Strganac (1998); Lind and Baldelli (2005); Lee and Singh (2007)). The control of a ‘smart’ airfoil and flutter control is already a research topic for several years (see also Chapter 2). The main issue is that the dynamics is strongly dependent on the free stream wind speed. The system can become unstable when the wind speed reaches a certain wind speed; flutter limit. In Lau and Krener (1999) and Barker and Balas (2000) LPV controller synthesis is proposed to solve this problem. In this section we show that the algorithms presented in this chapter can be used to obtain a model for LPV control.

In this example we first derive an analytical LPV model of the system under consideration. In Section 4.9.2 we show that due to discretization we lose the ‘nice’ affine LPV structure. Then we derive a time-varying state-feedback control law that stabilizes the system over the whole trajectory. On the controlled flutter model we apply our closed-loop identification scheme.

4.9.1 Analytical LPV modeling

There are a large number of 2-D ‘smart’ airfoil models available in the literature. The main difference between these models is the number of mechanical degrees of freedom that are taken into account and the unsteadiness of the aerodynamic model. The general mechanical degrees of freedom are the plunge, h , the pitch,

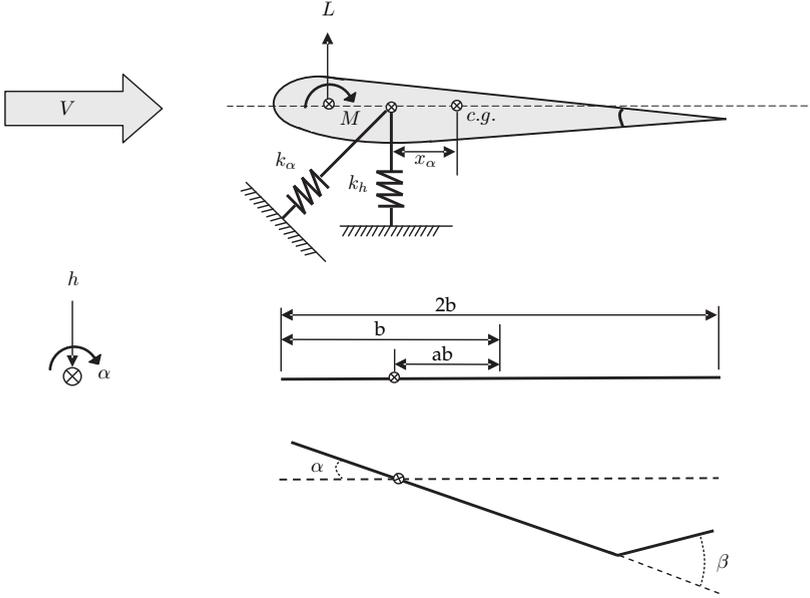


Figure 4.5: Schematic representation of the ‘smart’ airfoil model used in this simulation example.

α , and the trailing edge flap, β . The trailing edge flap is normally seen as the control variable and if this control surface is considered to be infinitely stiff the corresponding dynamics can be ignored. This is also the approach we will follow in this example. The equations of motion are now given by:

$$\begin{bmatrix} m_t & m_w x_\alpha b \\ m_w x_\alpha b & I_\alpha \end{bmatrix} \begin{bmatrix} \ddot{h} \\ \ddot{\alpha} \end{bmatrix} + \begin{bmatrix} c_h & 0 \\ 0 & c_\alpha \end{bmatrix} \begin{bmatrix} \dot{h} \\ \dot{\alpha} \end{bmatrix} + \begin{bmatrix} k_h & 0 \\ 0 & k_\alpha \end{bmatrix} \begin{bmatrix} h \\ \alpha \end{bmatrix} = \begin{bmatrix} -L \\ M \end{bmatrix},$$

where m_w is the mass of the wing, m_t is the total mass, b is the semi-chord of the wing, I_α is the moment of inertia, x_α is the non dimensionalized distance of the center of mass from the elastic axis, the thicknesses and damping coefficients of the plunge and pitch degree of freedom are k_h , k_α , c_h , and c_α , respectively. The inputs L and M are the aerodynamic force and moment, respectively. This set of second-order differential equations can also be written as a set of first-order differential equations:

$$\begin{bmatrix} \dot{h} \\ \dot{\alpha} \\ \ddot{h} \\ \ddot{\alpha} \end{bmatrix} = \begin{bmatrix} 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \\ \frac{I_\alpha k_h}{q_1} & \frac{-m_w x_\alpha b}{q_1} k_\alpha & \frac{I_\alpha c_h}{q_1} & \frac{-m_w x_\alpha b}{q_1} c_\alpha \\ \frac{-m_w x_\alpha b}{q_1} k_h & \frac{m_t}{q_1} k_\alpha & \frac{-m_w x_\alpha b}{q_1} c_h & \frac{m_t}{q_1} c_\alpha \end{bmatrix} \begin{bmatrix} h \\ \alpha \\ \dot{h} \\ \dot{\alpha} \end{bmatrix} +$$

$$\begin{bmatrix} 0 & 0 \\ 0 & 0 \\ \frac{-I_\alpha}{q_1} & \frac{m_{uw}x_\alpha b}{q_1} \\ \frac{m_{uw}x_\alpha b}{q_1} & \frac{-m_t}{q_1} \end{bmatrix} \begin{bmatrix} -L \\ M \end{bmatrix}, \quad (4.23)$$

with $q_1 = -m_t I_\alpha + m_w^2 x_\alpha^2 b^2$.

The aerodynamic forces can be modeled in several ways. With unsteady aerodynamics the dynamics of the airflow around the blade is modeled. A number of models are available to describe this dynamic behavior (see [Theodorsen \(1935\)](#); [Leishman \(2002\)](#); [Gaunaa \(2006\)](#)) leading to an additional set of differential equations. In this section we assume quasi-steady aerodynamics and that means that the unsteadiness is ignored and we only have static relations between the motion degrees of freedom and the aerodynamic forces. This is a valid assumption if the reduced frequency is considered to be small. For the quasi-steady aerodynamic model we have the following aerodynamic forces:

$$L = q_2 [0, V^2, V, q_6 V] \begin{bmatrix} h \\ \alpha \\ \dot{h} \\ \dot{\alpha} \end{bmatrix} + q_3 V^2 \beta, \quad (4.24a)$$

$$M = q_4 [0, V^2, V, q_6 V] \begin{bmatrix} h \\ \alpha \\ \dot{h} \\ \dot{\alpha} \end{bmatrix} + q_5 V^2 \beta, \quad (4.24b)$$

with:

$$\begin{aligned} q_2 &= \rho b s_p c_{l\alpha}, \\ q_3 &= \rho b c_{l\beta} s_p, \\ q_4 &= \rho b^2 s_p c_{m\alpha}, \\ q_5 &= \rho b^2 c_{m\beta} s_p, \\ q_6 &= \left[\frac{1}{2} - a \right] b, \end{aligned}$$

with V the free stream velocity, ρ the air density, a is the nondimensionalized distance from the midchord to the elastic axis, s_p is the span, $c_{l\alpha}$ and $c_{m\alpha}$ are the lift and moment coefficients per angle of attack, and $c_{l\beta}$ and $c_{m\beta}$ are lift and moment coefficients per control surface deflection β .

If we substitute (4.24a) and (4.24b) in (4.23) we obtain a model that describes the aeroelastics of a 'smart' rotor blade. The dynamics is depending on the free stream velocity. If we assume that we can measure the plunge and pitch motion

of the blade we obtain the following continuous time LPV model:

$$\begin{bmatrix} \dot{h} \\ \dot{\alpha} \\ \ddot{h} \\ \ddot{\alpha} \end{bmatrix} = (A_1 + A_2V + A_3V^2) \begin{bmatrix} h \\ \alpha \\ \dot{h} \\ \dot{\alpha} \end{bmatrix} + B_3V^2\beta \quad (4.25a)$$

$$\begin{bmatrix} h \\ \alpha \end{bmatrix} = C_1 \begin{bmatrix} h \\ \alpha \\ \dot{h} \\ \dot{\alpha} \end{bmatrix} \quad (4.25b)$$

with:

$$\begin{aligned} A_1 &= \begin{bmatrix} 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \\ \frac{I_\alpha}{q_1}k_h & \frac{-m_w x_\alpha b}{q_1}k_\alpha & \frac{I_\alpha}{q_1}c_h & \frac{-m_w x_\alpha b}{q_1}c_\alpha \\ \frac{-m_w x_\alpha b}{q_1}k_h & \frac{m_t}{q_1}k_\alpha & \frac{-m_w x_\alpha b}{q_1}c_h & \frac{m_t}{q_1}c_\alpha \end{bmatrix}, \\ A_2 &= \begin{bmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & \frac{I_\alpha}{q_1}q_2 + \frac{m_w x_\alpha b}{q_1}q_4 & \frac{I_\alpha}{q_1}q_2q_6 + \frac{m_w x_\alpha b}{q_1}q_4q_6 \\ 0 & 0 & \frac{-m_w x_\alpha b}{q_1}q_2 + \frac{-m_t}{q_1}q_4 & \frac{-m_w x_\alpha b}{q_1}q_2q_6 + \frac{-m_t}{q_1}q_4q_6 \end{bmatrix}, \\ A_3 &= \begin{bmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & \frac{I_\alpha}{q_1}q_2 + \frac{m_w x_\alpha b}{q_1}q_4 & 0 & 0 \\ 0 & \frac{-m_w x_\alpha b}{q_1}q_2 + \frac{-m_t}{q_1}q_4 & 0 & 0 \end{bmatrix}, \\ B_3 &= \begin{bmatrix} 0 \\ 0 \\ \frac{I_\alpha}{q_1}q_3 + \frac{m_w x_\alpha b}{q_1}q_5 \\ \frac{-m_w x_\alpha b}{q_1}q_3 + \frac{-m_t}{q_1}q_5 \end{bmatrix}, \\ C_1 &= \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \end{bmatrix}. \end{aligned}$$

The parameters used are given in Table 4.4.

4.9.2 Discretization

The LPV system given in (4.25a) and (4.25b) is used to obtain the input, output, and scheduling sequence for the identification algorithm. We assume that the scheduling variable V can be chosen arbitrarily. The LPV identification algorithms introduced in this chapter are valid for discrete time systems and the modeling so far is done in continuous time. The ‘nice’ affine LPV structure that we have in continuous time will be lost if we convert it to discrete time as highlighted in [Tóth et al. \(2008\)](#). In this example we use Tustin discretization, which for this model is

Table 4.4: Numerical values of the model parameters of the simple flutter model (Lee and Singh 2007).

parameter	value	parameter	value
b	0.135 m	a	-0.6847 -
k_h	2844.4 N/m	k_α	2.84 N/m
c_h	27.43 Ns/m	c_α	$0.036 \times \text{Ns/m}$
$c_{m\alpha}$	-1.1599 -	$c_{l\alpha}$	6.28 -
$c_{m\beta}$	-0.635 -	$c_{l\beta}$	3.358 -
m_w	2.049 kg	m_t	12.387 kg
I_α	0.0558 kgm^2	x_α	0.3314 -
ρ	1.225 kg/m^3	s_p	1 m

given by:

$$\begin{aligned}
 A_d \left(V, V^2, \frac{1}{V}, \dots \right) &= \left(I + \frac{T_s}{2} A(V, V^2) \right) \left(I - \frac{T_s}{2} A(V, V^2) \right)^{-1}, \\
 B_d \left(V, V^2, \frac{1}{V}, \dots \right) &= \sqrt{T_s} \left(I - \frac{T_s}{2} A(V, V^2) \right)^{-1} B(V^2), \\
 C_d \left(V, V^2, \frac{1}{V}, \dots \right) &= \sqrt{T_s} C \left(I - \frac{T_s}{2} A(V, V^2) \right)^{-1}, \\
 D_d \left(V, V^2, \frac{1}{V}, \dots \right) &= \frac{T_s}{2} C \left(I - \frac{T_s}{2} A(V, V^2) \right)^{-1} B(V^2),
 \end{aligned}$$

where T_s is the sample time. There are two important things we would like to stress at this point. The first is that in the continuous time representation of the system the output equation is parameter-invariant. After discretization, the output equation is depending on the scheduling because in the discretization formula the continuous time matrices A and C are used for computing the discrete time system matrices: C_d and D_d . The second issue is that we lose the affine LPV structure. This is caused by the fact that in the discretization formula we have an inverse of the parameter dependent A matrix. The main conclusion is that we completely lose the 'nice' affine LPV structure with parameter-invariant output equation. However, still we have the freedom to force an affine model on the data we obtain. In the sequel we try to identify the LPV model with an affine LPV model with parameter-varying output equation on data generated by the Tustin discretized model.

In this example we use a sampling time of 0.08 s. The pole trajectory in the continuous time and discrete time, for a wind speed going from 0 m/s to 14 m/s, is given in Figure 4.6 and Figure 4.7. In these figures we clearly see that the system becomes unstable. Consequently, to do system identification we have to apply feedback control. The focus of this chapter is on identification and not on control that is why we use a finite-horizon state-feedback controller (Ogata 1995) that

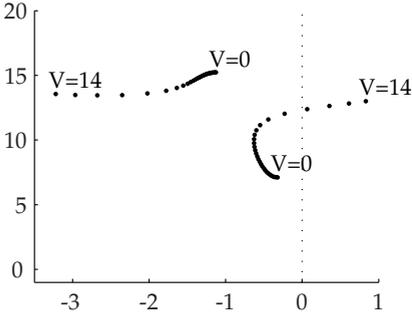


Figure 4.6: Poles of the continuous time LPV model for different wind speeds.

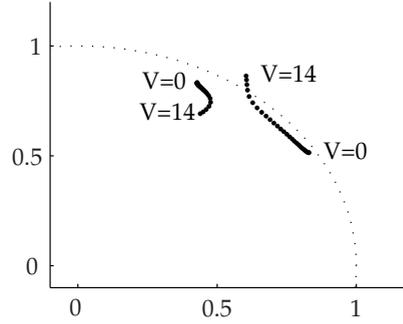


Figure 4.7: Poles of the discrete time LPV model for different wind speeds.

stabilizes the system.

4.9.3 Simulation results

For the identification algorithm we use the following wind signal:

$$V_k = 5 + 3 \sin\left(\frac{2\pi}{10}kT_s\right) + 2 \cos\left(\frac{12\pi}{10}kT_s\right) + w_k,$$

with w_k a white noise with $\text{var}(1)$. On the actuator, the flap β , we apply a white noise sequence with unit variance. With these two signals we can simulate the dynamic system given by (4.25a)-(4.25b) and discretized using the described Tustin discretization method. To investigate the sensitivity of the identification algorithm with respect to output and process noise, a Monte-Carlo simulation with 100 runs was carried out. For each of the 100 simulations a different realization of the input u_k and scheduling μ_k is used. We added process and measurement noise with a Signal-to-Noise Ratio (SNR) of 40dB. We collect 2500 data points for each trial.

For the identification procedure we assume to have the following affine scheduling sequence:

$$\mu_k = [1, V_k, V_k^2]^T.$$

All the signals are scaled such that they have a variance of 1. Concerning the identification, we observed that we obtain the best results if we assume parameter-varying A , B , D , and K matrices. We used the kernel version of the LPV-PBSID_{opt} algorithm with $p = 3$ and we select the first 140 rows of $Z_0^T Z_0$. The collected scaled data u_k , y_k , and μ_k is then used in the proposed identification algorithm.

For this example we do not use the VAF but we look at the Bode plots for a frozen parameter because for meaningful VAF values the system under consideration must be asymptotically stable, otherwise a small error will give low VAF values due to the increasing or decreasing characteristic of the outputs. In Fig-

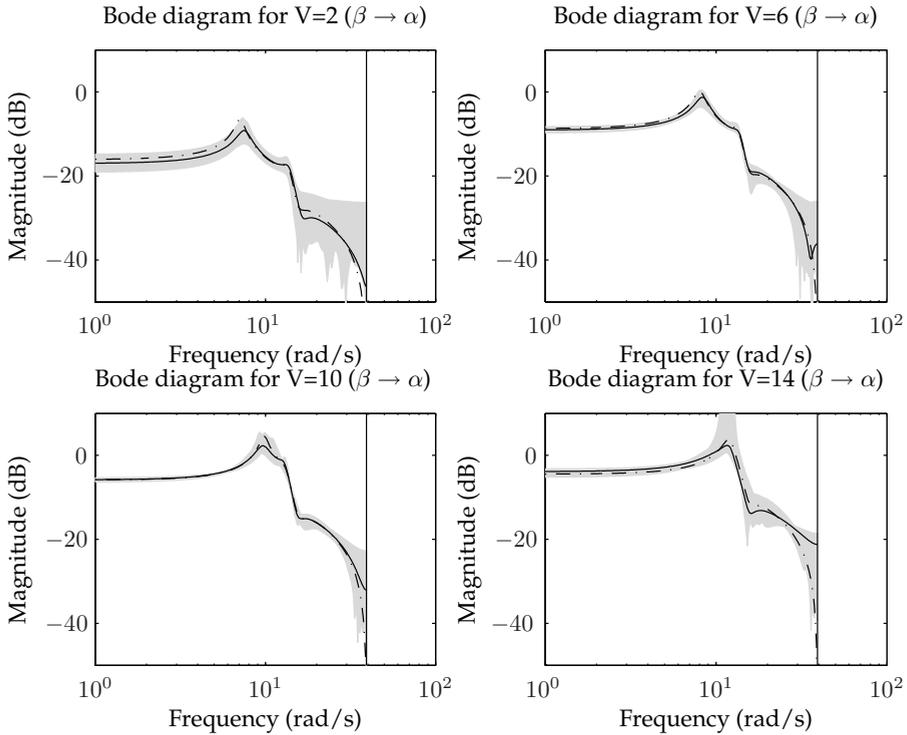


Figure 4.8: Bode diagrams of the original transfer functions (dashed) and the identified transfer functions of the experiment with the highest mean correlation coefficients (solid). The transfer functions of the other 99 experiments are within the gray confidence region. The determine the Bode diagram, the free stream velocity is fixed at the values $V = \{2, 6, 10, 14\}$.

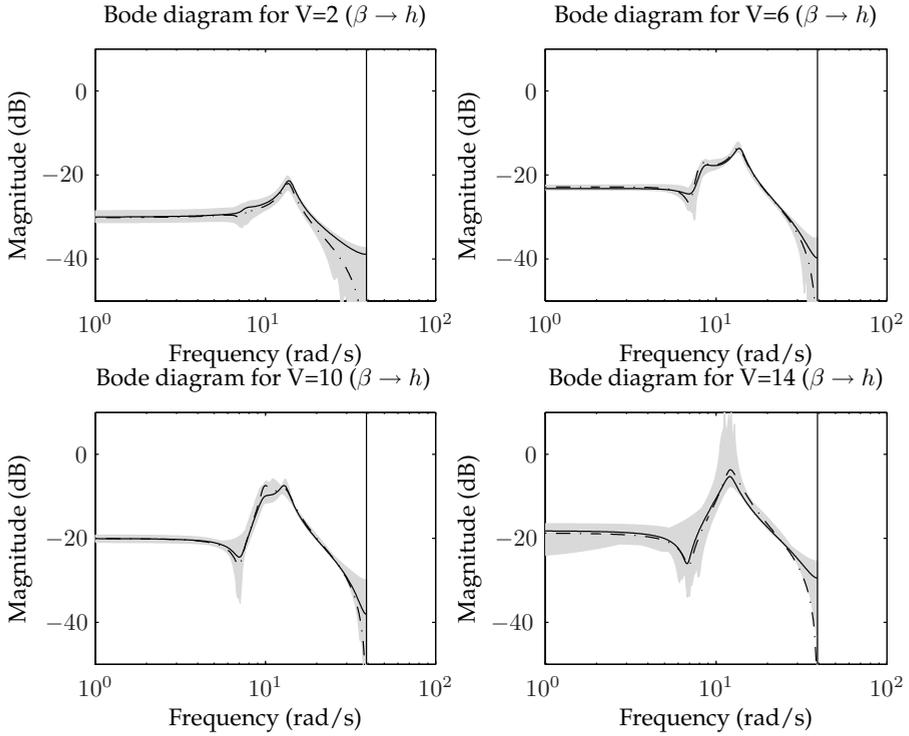


Figure 4.9: Bode diagrams of the original transfer functions (dashed) and the identified transfer functions of the experiment with the highest mean correlation coefficients (solid). The transfer functions of the other 99 experiments are within the gray confidence region. The determine the bode diagram, the free stream velocity is fixed at the values $V = \{2, 6, 10, 14\}$.

ure 4.8 and 4.9 the Bode plots for the pitch position and plunge position, respectively, are presented for 4 frozen values of the free stream velocity, $V = 2$, $V = 6$, $V = 10$, and $V = 14$. In these figures the dashed line represents the ‘real’ system and the solid line the fit with the highest correlation coefficients (Seber 1984; Krzanowski 1988) between the ‘real’ states and the identified states. The gray area represents the interval in which the other 99 estimates are located. We see that we have ‘good’ fits with a small variance (the gray area is small).

4.10 Conclusion

In this chapter we presented an open-loop and closed-loop LPV subspace identification method, which is an extension of Predictor-Based Subspace IDentification (PBSID) algorithms. This methodology from closed-loop LTI subspace identification is used to formulate the input-output behavior of an LPV system. From this input-output behavior the LPV equivalent of the Markov parameters can be estimated. We showed that with this estimate the product between the extended observability matrix and state sequence can be reconstructed and an SVD can be used to estimate the state sequence and consequently the system matrices. The ‘curse of dimensionality’ in subspace LPV identification appeared and the kernel method was proposed. A computationally efficient representation of the kernel is presented, which makes the approach numerical attractive. Furthermore, we showed that if there is structure in the scheduling, then the computational complexity reduces even more. The algorithm was illustrated with three simulation examples. Furthermore, we applied the algorithm on a simple ‘smart’ rotor model.

Conclusions & Recommendations

The drive to make wind energy more attractive and competitive to fossil fuel power plants will provide an enormous stimulus for technological breakthroughs. One such advance, potentially to be realized on the next generation of wind turbines, is the ‘smart’ rotor. In this thesis we have demonstrated the potential of using ‘smart’ surfaces along the span of a wind turbine blade for load reduction and developed a methodology for deriving mathematical models to tune model based controllers for optimizing turbine performance. Although these topics may be considered too complex to be implemented on existing wind turbines, so was individual pitch control 20 years ago. The conclusions and recommendations relating to the two different topics are now discussed separately.

5.1 ‘Smart’ rotor concept

5.1.1 Conclusions

In the first part of this thesis a novel control concept is introduced to change the force profile on wind turbine blades using local control devices to copy the spatially distributed nature of turbulence. The success of distributed load reduction greatly depends on the selection of appropriate sensors that measure the loads and a controller that manipulates the measured signals and generates an actuation signal. This overall combination of sensors, actuators, and control is defined as the ‘smart’ rotor concept. We contribute in this thesis to the development of this new concept by showing the feasibility of the ‘smart’ rotor under realistic wind turbine conditions (*i.e.* realistic disturbances, feedback control, and load measurements).

We developed an experimental model and a theoretical model of a ‘smart’ rotor blade by scaling down the dynamics of a representative wind turbine blade using the reduced frequency and applied a trailing edge flap as control device. In the root, strain sensors are added to facilitate feedback control. The pitch system

is used to impose realistic disturbances. In the wind tunnel it is shown that, when the disturbance is known, perfect cancellation can be realized; however, due to actuator saturation the reduction in the amplitude was limited to 90%. In real life the disturbance is not known and feedback control is required. For this situation we used experimental modeling to validate the theoretical model and designed a feedback controller using a loop-shaping technique. The feedback controllers have been tested on the experimental setup for different load cases. With a sinusoidal disturbance we showed that it is possible to reduce the amplitudes by 90% at the first eigenfrequency of the blade. In the second case a step was applied on the pitch, to simulate a gust, and it was shown that the oscillation corresponding to the first mode was almost completely removed. In the last case a noise signal with a spectrum representative of turbulence was applied, and it was shown that the 1P disturbance was reduced by 37% while the 3P frequency was reduced by 55%. The success of this proof of concept should help aerodynamicists and structural experts to embrace control engineering much earlier in their design cycle, and utilize it as a lever to create additional design freedom.

5.1.2 Recommendations

The ‘smart’ rotor is a technology for the future and faces a number of challenges before the concept will be adopted by the wind industry. We focused on the capability to do load reduction around the 1P and 3P frequencies and on damping enhancement of the first flapping mode. However, it is desirable to also look at distributed control with more actuators and sensors to further reduce the loads and take into account other vibration modes. Furthermore, the work presented is for a non-rotating blade, which does not guarantee that the load reduction capabilities of this concept can be extended to the loads in the drive train, for example. Within the [UPWIND \(2008\)](#) project, a rotating experiment will be performed to show the feasibility of a rotating ‘smart’ rotor.

After a successful rotating study the concept should be scaled towards a ‘real’ wind turbine. The up-scalability of the actuator concept used in this thesis is questionable for obvious reasons: voltage requirements, robustness, linearity, etc. The big challenge for the scaling of the ‘smart’ rotor concept is the robust design of actuators which have at least the same reliability as the current state-of-the-art.

Naturally, there are also challenges for control. The first is already mentioned: the use of distributed control¹ to react on turbulence in a more detailed way. The challenge for control is the development of efficient robust control design methodologies for this concept. Finally, we discussed LPV system identification for LPV controller synthesis. This controller methodology should certainly be applied on the ‘smart’ rotor because the gains are strongly dependent on the operational position of the wind turbine. This, in combination with the intended high number of actuators and sensors, requires significant advances in computationally efficient algorithms (see [Rice and Verhaegen \(2008, 2009\)](#) for recent developments).

¹The interested reader is referred to [Bamieh et al. \(2002\)](#); [D’Andrea and Dullerud \(2003\)](#); [Langbort et al. \(2004\)](#) and references therein.

5.2 LPV system identification

5.2.1 Conclusions

To enable model based controller synthesis for a variable speed ‘smart’ rotor, we developed tools to obtain experimental models by nonlinear system identification. We suggested the use of the Linear Parameter-Varying (LPV) framework because wind turbine dynamics are dependent on the operational condition (rotor speed, rotor position, pitch angle). We introduced two novel LPV identification frameworks for coping with periodically varying dynamics and arbitrarily varying dynamics, respectively. Although this work is introduced from a wind turbine perspective it can also be used for other applications.

The conclusions with respect to the LPV identification framework can be summarized as follows:

Periodic scheduling In the case of periodic dynamics we developed open and closed-loop LPV system identification approaches where we exploit the periodicity of the dynamics. The obtained LPV model is then valid for arbitrary scheduling sequences as well. We used LTI predictor-based subspace identification techniques to identify a number of observability matrices and state sequences which are identified in different state bases. We showed that by formulating an intersection problem, the states can be reconstructed in a global state basis from which the system matrices can be estimated. The algorithm is first demonstrated on two simple test examples. The first shows the effectiveness of the periodic LPV algorithm. The second illustrates that the performance of the algorithm is affected by the choice of the scheduling sequence. Finally, we applied the approach successfully on a virtual model of the rotational dynamics of a horizontal axis wind turbine.

Arbitrary scheduling In the case of arbitrarily parameter-dependent dynamics an open and closed-loop LPV subspace identification method was developed. The methodology from LTI predictor-based subspace identification is used to formulate the input-output behavior of an LPV system. From this input-output behavior the LPV equivalent of the Markov parameters can be estimated. We showed that with this estimate the product between the extended observability matrix and state sequence can be reconstructed. From this product the state and consequently the system matrices can be estimated. The ‘curse of dimensionality’ in subspace LPV identification reared its ugly head but we showed that by exploiting structure in the problem, this complexity problem can be circumvented. Some features of the algorithm were demonstrated on three simulation examples in which we showed the effect of a number of algorithm parameters (*e.g.* window size, regularization). Finally, we applied the algorithm on a virtual ‘smart’ rotor model to emphasize the application towards the new generation of wind turbines.

5.2.2 Recommendations

In the fundamental part of this thesis we focused mainly on the development of the actual identification algorithms, not the details of how to choose the past and future window, the scheduling sequence, or the regularization parameter (although we indicated with a number of examples the effects of the different parameters). It is of practical interest to have a detailed study on how these parameters should be chosen.

With respect to the ‘curse of dimensionality’ of the different algorithms, we explored the field of selecting a small number of kernels and showed the potential of this approach. Still, research is required on how to select the kernels and we believe that answers can be found in the field of Support Vector Machines (SVM).

In our development, we assumed that the scheduling vector was known exactly. However, for wind turbines (and most other LPV systems) the scheduling sequence is measured and consequently contains a stochastic component. To include this in the identification framework is a topic for future research.

The identification approach was motivated from a control perspective; we would like to use the obtained models for controller synthesis. Moreover, the first step of the presented algorithm, the identification of the LPV Markov parameters, can also serve as the first step for data-driven LPV control, a promising adaptive control scheme.

Finally, we suggested LPV system identification as a kind of gray-box system identification. Yet, in our current LPV framework we can only use a limited amount of prior knowledge. It would be of interest to make the next step and use all the prior knowledge in the identification procedure.

A

APPENDIX

Controller details of the experimental 'smart' rotor

In Chapter 2 a rather brief description of the controller is given, although the emphasis of that particular chapter is not on controller design, in this appendix we present some additional details with respect to the final controller design.

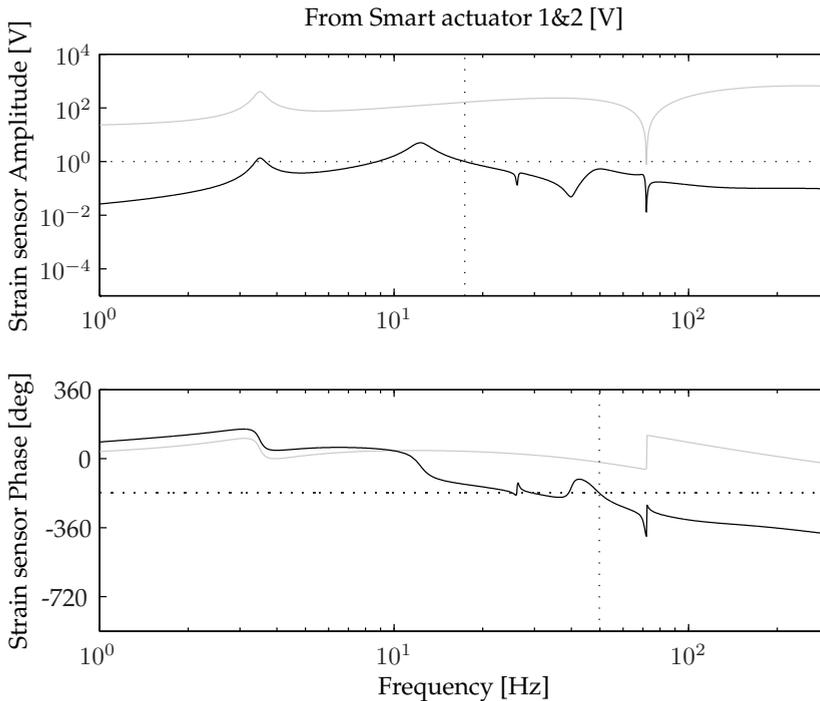


Figure A.1: The controller (gray) and the open loop (black) dynamics represented by their Bode Plot, for a wind speed of 45 m/s.

In Figure A.1 the open loop Bode plot and a Bode plot of the controller are presented. In this plot it is clearly shown that we added an inverted notch at the 1P frequency and a normal notch at the second flapping frequency. The inverted notch at the 1P frequency is added to have disturbance attenuation for the 1P load while the second notch is added for stability reasons (to have a sufficient amount of gain margin). The derivative-action is added to increase the phase margin around the cross-over frequency. The controller is manually tuned using traditional loop-shaping techniques. We ended up with a phase margin of 46.3 degrees (at 17.4 Hz) and a gain margin of 5.53 dB (at 49.8 Hz), which are considered to be appropriate numbers to have a sufficient amount of robustness. In Figure A.2 we present the open and closed loop bode plots and we clearly see that we have disturbance rejection capabilities in the frequency range of interest (1P, 3P, first flapping mode).

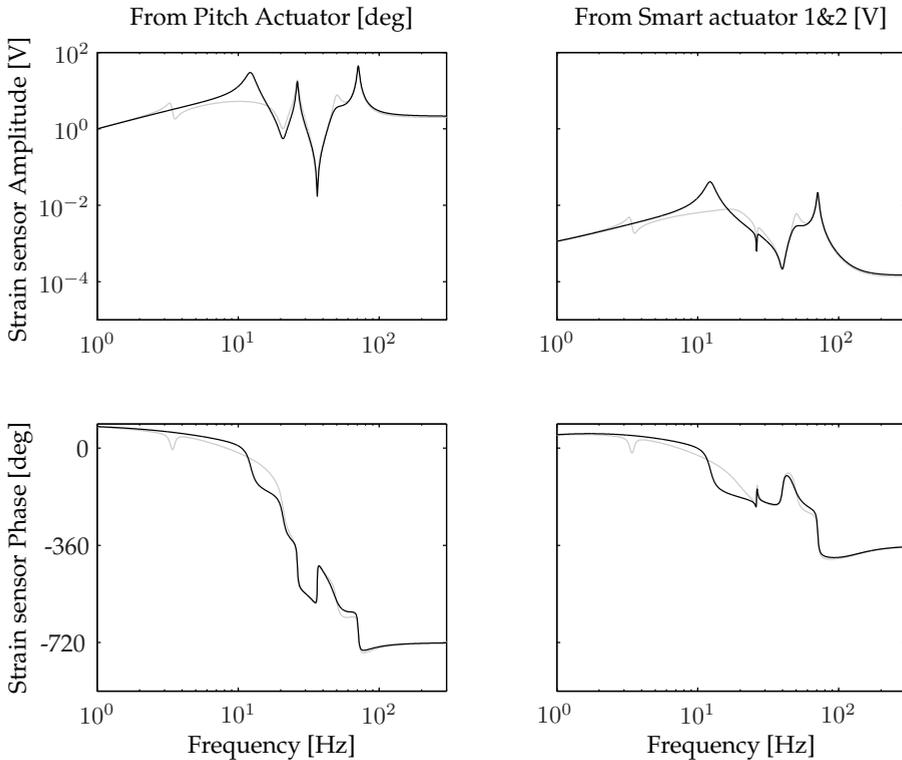


Figure A.2: The open loop and closed loop Bode figures given by the black and gray line, respectively, for a wind speed of 45 m/s.

LTI predictor-based subspace identification

This appendix deals with the identification of LTI systems using predictor-based subspace identification algorithms¹. The notation and the basic idea presented in this section is also used in Section 3.3 and Section 4.4 where we deal with predictor-based LPV identification.

It is well known that the projector type of subspace algorithms (*e.g.* MOESP (Verhaegen and Dewilde 1992) and N4SID (Van Overschee and De Moor 1996)) give biased estimates if the identification data is generated under closed-loop conditions. The main reason for the bias is the constraint that the noise, e_k , and the input, u_k , should be uncorrelated. This assumption is clearly violated if there is a feedback loop present (as clearly explained by Ljung and McKelvey (1996)). Predictor-based subspace identification (*e.g.* PBSID (Chiuso and Picci 2005) and SSARX (Jansson 2005)) methods do not suffer from this drawback. These methods use high order ARX models to remove the correlation between the input and noise sequence. We introduce in this appendix two closely related LTI predictor-based identification schemes, which are used extensively throughout this thesis: PBSID and PBSID_{opt}.

Predictors

The first objective of the predictor-based algorithms is to reconstruct the state sequence up to a similarity transformation. The state x_{k+p} is given by:

$$x_{k+p} = \tilde{A}^p x_k + \underbrace{\left[\tilde{A}^{p-1} \tilde{B}, \tilde{A}^{p-2} \tilde{B}, \dots, \tilde{B} \right]}_{\mathcal{K}^p} \bar{z}_k^p,$$

¹Observe that for $m = 1$ and $\mu_k^{(1)} = 1 \quad \forall k$ the model structure presented in (3.1a)-(3.1b) or (4.1a)-(4.1b) is actually an LTI system.

with $\bar{B} = [\tilde{B}, K]$. The key approximation in this algorithm is that we assume that $\tilde{A}^j \approx 0$ for all $j \geq p$. It can be shown that if the system in (4.2a)-(4.2b) is uniformly exponentially stable, the approximation error can be made arbitrarily small by making p large (Knudsen 2001; Chiuso and Picci 2005; Chiuso 2007). With this assumption the state x_{k+p} is approximately given by:

$$x_{k+p} \approx \underbrace{[\tilde{A}^{p-1}\bar{B}, \tilde{A}^{p-2}\bar{B}, \dots, \bar{B}]}_{\mathcal{K}^p} \bar{z}_k^p. \quad (\text{B.1})$$

In a number of other LTI subspace methods it is well known to make this step (e.g. N4SID, SSARX, PBSID, PBSID_{opt}). The input-output behavior is now approximately given by:

$$\left\{ \begin{array}{l} y_{k+p} \approx C \underbrace{[\tilde{A}^{p-1}\bar{B}, \tilde{A}^{p-2}\bar{B}, \dots, \bar{B}]}_{\mathcal{K}^p} \bar{z}_k^p + Du_{k+p} + e_{k+p} := y_{k+p}^{(p)} \\ y_{k+p+1} \approx C \underbrace{[\tilde{A}\mathcal{K}^p, \bar{B}]}_{\mathcal{K}^{p+1}} \bar{z}_k^{p+1} + Du_{k+p+1} + e_{k+p+1} \\ \vdots \\ y_{k+p+f-1} \approx C \underbrace{[\tilde{A}\mathcal{K}^{p+f-2}, \bar{B}]}_{\mathcal{K}^{p+f-1}} \bar{z}_k^{p+f-1} + Du_{k+p+f-1} + e_{k+p+f-1} \end{array} \right. .$$

Now we define the stacked matrices U_i , Y_i , and Z_i :

$$U_i = [u_{p+i}, \dots, u_{N-f+i+1}], \quad (\text{B.2a})$$

$$Y_i = [y_{p+i}, \dots, y_{N-f+i+1}], \quad (\text{B.2b})$$

$$Z_i = [\bar{z}_0^{p+i}, \dots, \bar{z}_{N-p-f}^{p+i}], \quad (\text{B.2c})$$

for all $i \in \{0, \dots, f-1\}$. If the matrix $[Z_i^T, U_i^T]^T$ for all $i \in \{0, \dots, f-1\}$ has full row rank the matrices $C\mathcal{K}^{p+i}$ and D can be estimated by solving the following linear regression problem:

$$\min_{C\mathcal{K}^{p+i}, D} \|Y_i - C\mathcal{K}^{p+i}Z_i - DU_i\|_F^2, \quad (\text{B.3})$$

for all $i \in \{0, \dots, f-1\}$ and where $\|\dots\|_F$ represents the Frobenius norm (Golub and Loan 1996). For finite p the solution of this linear problem will be biased due the approximation made in (B.1). In the LTI literature a number of papers appeared that studied the effect of the window size and although they proved the asymptotic properties of the algorithms (if $p \rightarrow \infty$ the bias disappears) it is hard to quantify the effect for finite p (Knudsen 2001; Chiuso and Picci 2005; Chiuso 2007).

Extended observability times controllability

The product $\mathcal{K}^p Z_0$ that represents by definition the state sequence, X , can not be estimated directly. In the predictor-based identification algorithms $C\mathcal{K}^{p+i}$ is used to construct the extended observability matrix times the extended controllability matrix. This matrix is given by (we assume $f = p$):

$$\Gamma^f \mathcal{K}^p = \begin{bmatrix} C\tilde{A}^{p-1}\bar{B} & C\tilde{A}^{p-2}\bar{B} & \dots & C\bar{B} \\ C\tilde{A}^p\bar{B} & C\tilde{A}^{p-1}\bar{B} & \dots & C\tilde{A}\bar{B} \\ \vdots & \vdots & \ddots & \vdots \\ C\tilde{A}^{p+f-2}\bar{B} & C\tilde{A}^{p+f-3}\bar{B} & \dots & C\tilde{A}^{p-1}\bar{B} \end{bmatrix}. \quad (\text{B.4})$$

This particular matrix is constructed in the PBSID algorithm. The following upper block triangular matrix is used in the PBSID_{opt} algorithm (we assume $f = p$):

$$\Gamma^f \mathcal{K}^p \approx \begin{bmatrix} C\tilde{A}^{p-1}\bar{B} & C\tilde{A}^{p-2}\bar{B} & \dots & C\bar{B} \\ 0 & C\tilde{A}^{p-1}\bar{B} & \dots & C\tilde{A}\bar{B} \\ & & \ddots & \vdots \\ 0 & & & C\tilde{A}^{p-1}\bar{B} \end{bmatrix}. \quad (\text{B.5})$$

The zeros appear in this equation based on the approximation that $\tilde{A}^j \approx 0$ for all $j \geq p$. Observe that from the linear regression problems formulated in (B.3) we only need the solution for $i = 0$, because from the first row in (B.5) we can construct the other rows.

From the constructed matrix $\Gamma^f \mathcal{K}^p$, either the PBSID or PBSID_{opt} way, we can compute $\Gamma^f \mathcal{K}^p Z_0$ which equals by definition the extended observability matrix times the state sequence, $\Gamma^f X$. By computing a Singular Value Decomposition (SVD) of this estimate we can estimate the state sequence and the order of the system. We will use the following SVD:

$$\widehat{\Gamma^f \mathcal{K}^p Z_0} = \begin{bmatrix} \mathcal{U} & \mathcal{U}_\perp \end{bmatrix} \begin{bmatrix} \Sigma_n & 0 \\ 0 & \Sigma \end{bmatrix} \begin{bmatrix} \mathcal{V} \\ \mathcal{V}_\perp \end{bmatrix}, \quad (\text{B.6})$$

where Σ_n is the diagonal matrix containing the n largest singular values and \mathcal{V} is the corresponding row space. Note that we can find the largest singular values by detecting a gap between the singular values (Verhaegen and Verdult 2007). The state is now estimated by:

$$\hat{X} = \Sigma_n \mathcal{V}. \quad (\text{B.7})$$

It is well known that when the state, input, and output are known the system matrices can be estimated (Verhaegen and Verdult 2007). First we use (4.1b), which is now a linear relation in C and D and where e_k represents the innovation process. From this equation an estimate can be found of the C and D matrix while also the noise sequence can be estimated. The estimated noise sequence is used to transform (4.1a) (for $m=1$) into a linear expression depending on A , B , and K and consequently all the system matrices can be estimated.

Summary of the LTI algorithms

The described algorithms can be summarized by the following two algorithms:

Algorithm B.1 (PBSID_{opt})

The PBSID_{opt} algorithm can be summarized as follows:

1. Create the matrices U_i , Y_i , and Z_i using (B.2a), (B.2b), and (B.2c) for $i = 0$,
2. Solve the linear problem given in (B.3) for $i = 0$,
3. Construct $\Gamma^f \mathcal{K}^p Z_0$ using (B.2c) and (B.5),
4. Compute the state sequence using (B.6) and (B.7),
5. With the estimated state, use the linear relations (4.1a)-(4.1b) to obtain the system matrices (for $m=1$).

Algorithm B.2 (PBSID)

The PBSID algorithm can be summarized as follows:

1. Create the matrices U_i , Y_i , and Z_i using (B.2a), (B.2b), and (B.2c) for all $i \in \{0, \dots, f-1\}$,
2. Solve the linear problem given in (B.3) for all $i \in \{0, \dots, f-1\}$,
3. Construct $\Gamma^f \mathcal{K}^p Z_0$ using (B.2c) and (B.4),
4. Compute the state sequence using (B.6) and (B.7),
5. With the estimated state, use the linear relations (4.1a)-(4.1b) to obtain the system matrices (for $m=1$).

Kernels for different model structures

In Chapter 4 we derived an LPV identification scheme for the model structure given in (4.1a)-(4.1b). In Section 4.6 we discussed different model structures. In this appendix the kernel algorithm is given for all of the mentioned algorithms.

The model structures presented in Chapter 4 can be summarized using the following table where: pv= parameter-varying, c=constant.

Model	B	K	C	D	μ_k	
(4.1a)-(4.1b)	pv	pv	c	c	μ_k	Parameter-invariant output eq.
(4.18a)-(4.18b)	pv	c	pv	pv	μ_k	Parameter-varying output eq.
(4.20a)-(4.20b)	pv	pv	pv	pv	μ_k	Parameter-varying output eq.
(4.22a)-(4.22b)	c	c/pv	c	c	u_k	Bilinear model

The consequences for the presented identification approach are already discussed in Section 4.6. In this section we give the kernel driven LPV identification algorithm for the different model structures in a generalized framework.

The kernels are now given by:

$$\begin{aligned}
 (Z^{i,j})^T Z^{0,j} = & \left(\prod_{v=Const_B}^{p-j-1} \mu_{\tilde{N}+v+j-i}^T \mu_{\tilde{N}+v+j} \right) \left(u_{\tilde{N}+j-i}^T u_{\tilde{N}+j} \right) + \\
 & \dots \left(\prod_{v=Const_K}^{p-j-1} \mu_{\tilde{N}+v+j-i}^T \mu_{\tilde{N}+v+j} \right) \left(y_{\tilde{N}+j-i}^T y_{\tilde{N}+j} \right), \quad (C.1)
 \end{aligned}$$

where $Const_B = 1$ if B is constant and a similar definition holds for $Const_K$ and

later for $Const_C$ and $Const_D$. In a similar way we can define ¹:

$$Z_i^T Z_i = \sum_{j=0}^{p+i-1} \left(\left(\prod_{v=j+Const_B}^{p+i-1} \mu_{\tilde{N}+v}^T \mu_{\tilde{N}+v} \right) \left(u_{\tilde{N}+j}^T u_{\tilde{N}+j} \right) \right) + \dots \\ \sum_{j=0}^{p+i-1} \left(\left(\prod_{v=j+Const_K}^{p+i-1} \mu_{\tilde{N}+v}^T \mu_{\tilde{N}+v} \right) \left(y_{\tilde{N}+j}^T y_{\tilde{N}+j} \right) \right), \quad (C.2)$$

and

$$\bar{Z}_i^T \bar{Z}_i = \sum_{j=0}^{p+i-1} \left(\left(\prod_{v=j+Const_B}^{p+i-Const_C} \mu_{\tilde{N}+v}^T \mu_{\tilde{N}+v} \right) \left(u_{\tilde{N}+j}^T u_{\tilde{N}+j} \right) \right) + \dots \\ \sum_{j=0}^{p+i-1} \left(\left(\prod_{v=j+Const_K}^{p+i-Const_C} \mu_{\tilde{N}+v}^T \mu_{\tilde{N}+v} \right) \left(y_{\tilde{N}+j}^T y_{\tilde{N}+j} \right) \right), \quad (C.3)$$

and

$$Q_i^T Q_i = \left(\prod_{v=1}^{1-Const_D} \mu_{\tilde{N}+p+i}^T \mu_{\tilde{N}+p+i} \right) \left(u_{\tilde{N}+p+i}^T u_{\tilde{N}+p+i} \right). \quad (C.4)$$

We replace the linear problem given in (4.13) by:

$$\min_{\alpha_i} \|\alpha_i\|_F \quad \text{with} \quad Y_i - \alpha_i \left[\bar{Z}_i^T \bar{Z}_i + Q_i^T Q_i \right] = 0, \quad (C.5)$$

Algorithm C.1 (LPV-identification (kernel))

The algorithm can be summarized as follows:

1. Select the model structure you want to identify (select μ_k , $Const_B$, $Const_C$, $Const_D$, and $Const_K$) and the identification algorithm,
2. Create the matrices $(Z^{i,j})^T Z^{i,j}$, $Z_i^T Z_i$, $\bar{Z}_i^T \bar{Z}_i$, and $Q_i^T Q_i$ using (C.1)-(C.4),
3. Solve the linear problem given in (C.5). If desired regularized,
4. Construct $\Gamma^f K^p Z_0$ using (4.15) or (4.16) for LPV-PBSID_{opt} or LPV-PBSID, respectively,
5. Compute the state sequence using (4.11) and (4.12),
6. With the estimated state, the selected model structure is linear in the unknowns and consequently the system matrices can be estimated.

¹ We define $\prod_{a=1}^0 = 1$

Bibliography

- ADAPWING (2008). ADAPtive WING geometry for reduction of wind turbine loads. Available from Internet: <http://www.risoe.dk/vea/adapwing/>. Last visited 21 July 2008.
- Andersen, P. B. (2005). Load alleviation on wind turbine blades using variable airfoil geometry (2D and 3D study). M.sc thesis, Technical University of Denmark.
- Andersen, P. B., M. Guanaa, C. Bak and T. Buhl (2006). Load alleviation on wind turbine blades using variable airfoil geometry. In *Proceedings of the European Wind Energy Conference (EWEC)*, Athens, Greece.
- Apkarian, P. and R. J. Adams (1998). Advanced gain scheduling techniques for uncertain systems. *IEEE Transactions on Control Systems Technology* **6**(1), 21–32.
- Bak, C., M. Gaunaa, P.B. Andersen, T. Buhl, P. Hansen, K. Clemmensen and R. Moeller (2007). Wind tunnel test on wind turbine airfoil with adaptive trailing edge geometry. In *Proceedings of the 45th AIAA Aerospace Sciences Meeting and Exhibit (ASME)*, Reno, USA.
- Bamieh, B. and L. Giarre (2002). Identification of linear parameter varying models. *International Journal of Robust and Nonlinear Control* **12**(9), 841–853.
- Bamieh, B., F. Paganini and M. A. Dahleh (2002). Distributed control of spatially invariant systems. *IEEE Transactions on Automatic Control* **47**(7), 1091–1107.
- Barker, J. M. and G. J. Balas (2000). Comparing linear parameter-varying gain-scheduled control techniques for active flutter suppression. *Journal of Guidance, Control, and Dynamics* **23**(5), 948–955.
- Barlas, T., A.W. Hulskamp, J. W. van Wingerden, A. Mroz, P. van Langen, M. Landa and S. Apinaniz (2007). Smart rotor blades and rotor control for wind turbines state of the art. Knowledge base report for upwind wp 1b3, UPWIND.
- Barlas, T. and G. A. M. van Kuik (2007). State of the art and perspectives of smart rotor control for wind turbines. *Journal of Physics, Conference series: The Science of Making Torque from Wind* **75**(012080).
- Barret, R. (1990). Intelligent rotor blade actuation through directionally attached piezoelectric crystals. In *Proceedings of the 46th Annual forum and technology display*, Washington DC, USA.

- Basualdo, S. (2005). Load alleviation on wind turbine blades using variable airfoil geometry. *Wind Engineering* **29**(2), 169–182.
- Becker, A. and A. Packard (1994). Robust performance of linear parametrically varying systems using parametrically-dependent linear feedback. *Systems & Control Letters* **23**(3), 205–215.
- Bianchi, F.D., H. De Battista and R.J. Mantz (2007). *Wind Turbine Control Systems; Principles, modelling and gain scheduling design*. Springer-Verlag London.
- Bianchi, F.D., R.J. Mantz and C.F. Christiansen (2004). Control of variable-speed wind turbines by LPV gain scheduling. *Wind Energy* **7**(1), 1–8.
- Bianchi, F.D., R.J. Mantz and C.F. Christiansen (2005). Gain scheduling control of variable-speed wind energy conversion systems using quasi-LPV models. *Control Engineering Practice* **13**(2), 247–255.
- Block, J. J. and T. W. Strganac (1998). Applied active control for a nonlinear aeroelastic structure. *Journal of Guidance, Control and Dynamics* **21**(6), 838–845.
- Bongers, P. M. M. (1994). *Modeling and identification of flexible wind turbines and a factorizational approach to robust control*. Ph. D. thesis, Delft University of Technology.
- Bongers, P. M. M. and G. E. van Baars (1991). Experimental validation of a flexible wind turbine model. In *Proceedings of the 30th IEEE Conference on Decision and Control (CDC)*, Brighton, England.
- Borges, J., V. Verdult, M. Verhaegen and M. A. Botto (2004). Separable least squares for projected gradient identification of composite local linear state-space models. In *Proceedings of the 16th international symposium on mathematical theory of networks and systems*, Leuven, Belgium.
- Bossanyi, E.A. (2000). The design of closed loop controllers for wind turbines. *Wind Energy* **3**(3), 149–163.
- Bossanyi, E.A. (2003). Individual blade pitch control for load reduction. *Wind Energy* **6**(2), 119–128.
- Bossanyi, E.A. (2005). Further load reductions with individual pitch control. *Wind Energy* **8**(4), 481–485.
- Brewer, J. W. (1978). Kronecker products and matrix calculus in system theory. *IEEE Transactions on circuits and systems* **25**(9), 772–781.
- Bruno, C., G. Di Pillo and G. Koch (1974). Bilinear systems: an appealing class of near linear systems in theory and applications. *IEEE Transactions on Automatic Control* **19**(4), 334–348.
- Buhl, T., M. Gaunaa and C. Bak (2005). Potential of load reduction using airfoils with variable trailing edge geometry. *Journal of Solar Energy Engineering* **127**(4), 503–516.

- Casella, F. and M. Lovera (2008). LPV/LFT modelling and identification: overview, synergies and a case study. In *Proceedings of the IEEE Multi-conference on Systems and Control (MSC)*, San Antonio, USA.
- Chen, H. and J. Maciejowski (2000). An improved subspace identification method for bilinear systems. In *Proceedings of the 39th IEEE Conference on Decision and Control (CDC)*, Sydney, Australia.
- Chiuso, A. (2007). The role of vector auto regressive modeling in predictor based subspace identification. *Automatica* **43**(6), 1034–1048.
- Chiuso, A. and G. Picci (2005). Consistency analysis of certain closed-loop subspace identification methods. *Automatica* **41**(3), 377–391.
- Chopra, I. (2000). Status of application of smart structures technology to rotorcraft systems. *American Helicopter Society* **45**(4), 228–252.
- Chow, R. and C.P. van Dam (2007). Computational investigations of deploying load control microtabs on a wind turbine airfoil. In *Proceedings of the 45th AIAA Aerospace Sciences Meeting and Exhibit (ASME)*, Reno, USA.
- D’Andrea, R. and G.E. Dullerud (2003). Distributed control design for spatially interconnected systems. *IEEE Transactions on Automatic Control* **48**(9), 1478–1495.
- De Bie, T. and B. De Moor (2003). On the regularization of canonical correlation analysis. In *Proceedings of the 4th International symposium on independent component analysis and blind signal separation (ICA2003)*, Nara, Japan.
- dSPACE GmbH (2008). website dSPACE. Available from Internet: www.dSPACE.com. Last visited 22 January 2008.
- DUWIND (2008). Delft university wind energy research institute (DUWIND). Available from Internet: <http://www.tudelft.nl/duwind>. Last visited 22 January 2008.
- ECN (2008). Energy research center the netherlands (ECN). Available from Internet: www.ecn.nl. Last visited 22 January 2008.
- Eggleston, D. M. and F. S. Stoddard (1987). *Wind turbine engineering design*. Van Nostrand Reinhold.
- ENERCON (2008). ENERCON GmbH. Available from Internet: <http://www.enercon.de/>. Last visited 14 February 2008.
- Espinoza, M., J. A. K. Suykens and B. De Moor (2006). Fixed-size least squares support vector machines: A large scale application in electrical load forecasting. *Computational Management Science* **3**(2), 113–129.
- Face international cooperation (2008). Thundertm TH-6R data sheet. Available from Internet: <http://www.faceinternational.com/>. Last visited 22 January 2008.

- Favoreel, W. (1999). *Subspace methods for identification and control of linear and bilinear systems*. Ph. D. thesis, Katholieke universiteit leuven.
- Favoreel, W., B. De Moor and P. Van Overschee (1997). A bilinear extension of subspace identification for systems subject to white noise. In *Proceedings of the American Control Conference (ACC)*, Albuquerque, New Mexico, USA.
- Favoreel, W., B. De Moor and P. Van Overschee (1999). Subspace identification of bilinear systems subject to white inputs. *IEEE Transactions on Automatic Control* **44**(6), 1157–1165.
- Felici, F., J. W. van Wingerden and M. Verhaegen (2007a). Dedicated periodic scheduling sequences for LPV system identification. In *Proceedings of the European Control Conference (ECC)*, Kos, Greece.
- Felici, F., J. W. van Wingerden and M. Verhaegen (2007b). Subspace identification of MIMO LPV systems using a periodic scheduling sequence. *Automatica* **43**(10), 1684–1697.
- Franklin, G.F., J.D. Powell and A. Emani-Naeini (1994). *Feedback control of dynamic systems*. Addison Wesley.
- Friedmann, P.P. and T. A. Millott (1995). Vibration reduction in rotorcraft using active control: A comparison of various approaches. *Journal of Guidance, Control, and Dynamics* **18**(4), 664–673.
- Fuglsang, L. (2008). Integrated design of turbine rotors. In *Proceedings of the European Wind Energy Conference (EWEC)*, Brussels, Belgium.
- Garrad Hassan (2008). Gh bladed: Wind turbine design software. Available from Internet: <http://www.garradhassan.com/>. Last visited 18 February 2008.
- Gaunaa, M. (2006). Unsteady 2D potential-flow forces on a thin variable geometry airfoil undergoing arbitrary motion. Technical Report Risø-R-1478(EN), Risø.
- Giarre, L., D. Bauso, P. Falugi and B. Bamieh (2006). LPV model identification for gain scheduling control: an application to rotating stall and surge control problem. *Control Engineering Practice* **14**(4), 351–361.
- Golub, G. H. and C. F. van Loan (1996). *Matrix computations*. The John Hopkins University Press.
- Groot Wassink, M., M. Van De Wal, C. W. Scherer and O. Bosgra (2005). LPV control for a wafer stage: beyond the theoretical solution. *Control Engineering Practice* **13**(2), 231–245.
- GWEC (2008). Global Wind Energy Council (GWEC). Available from Internet: <http://www.gwec.net>. Last visited 14 February 2008.
- Ham, N. D. (1980). A simple system for helicopter individual-blade-control using modal decomposition. *Vertica* **4**(1), 23–28.

- Hand, M.M. and M.J. Balas (2007). Blade load mitigation control design for a wind turbine operating in the path of vortices. *Wind Energy* **10**(4), 339–355.
- Hansen, M. H. (2007). Aeroelastic instability problems for wind turbines. *Wind Energy* **10**(6), 551–577.
- Hansen, M. H., A. Hansen, T.J. Larsen, O. Stig, P. Sorensen and P. Fuglsang (2005). Control design for pitch-regulated, variable-speed wind turbine. Technical Report Riso-R-1500(EN), Riso National Laboratory.
- Hansen, M. H., K. Thomsen, P. Fuglsang and T. Knudsen (2006). Two methods for estimating aeroelastic damping of operational wind turbine modes from experiments. *Wind Energy* **9**(1-2), 179–191.
- Hench, J. J. and A.J. Laub (1994). Numerical solution of the discrete-time periodic riccati equation. *IEEE Transactions on Automatic Control* **39**(6), 1197–1210.
- Henriksen, L. C. (2007). Model predictive control of a wind turbine. Master's thesis, Technical University of Denmark.
- Henriksen, L. C. (2008). Model predictive control of a wind turbine with constraints. In *Proceedings of the European Wind Energy Conference (EWEC)*, Brussels, Belgium.
- Hjalmarsson, H., M. Gevers, F. De Bruyne and J. Leblond (1994). Identification for control: closing the loop gives more accurate controllers. In *Proceedings of the 33rd IEEE Conference on Decision and Control (CDC)*, Lake Buena Vista, USA.
- Houtzager, I. (2007). Adaptive learning control for wind turbines equipped with smart rotor: Applied to a NM80 wind turbine model. Master's thesis, Delft University of Technology; Delft Center for Systems and Control.
- Hulskamp, A.W., A. Beukers, H.E.N. Bersee, J. W. van Wingerden and T. Barlas (2007). Design of a wind tunnel scale model of an adaptive wind turbine blade for active aerodynamic load control experiments. In *Proceedings of the 16th international conference on composite materials*, Tokyo, Japan.
- James III, G.H., T.G. Carne and J.P. Lauffer (1993). The natural excitation technique (NExt) for modal parameter extraction from operating wind turbines. Technical report, Sandia.
- Jansson, M. (2005). A new subspace identification method for open and closed loop data. In *Proceedings of the 16th IFAC World Congress*, Prague, Czech Republic.
- Jauch, C., T. Cronin, P. Sorensen and B. B. Jensen (2007). A fuzzy logic pitch angle controller for power system stabilization. *Wind Energy* **10**(1), 10–39.
- Joncas, S., O. Bergsma and A. Beukers (2005). Power regulation and optimization of offshore wind turbines through trailing edge flap control. In *Proceedings of the 43th AIAA Aerospace Sciences Meeting and Exhibit (ASME)*, Reno, USA.

- Kailath, T., A. H. Sayed and B. Hassibi (2000). *Linear estimation*. Prentice Hall Information and System Sciences Series. New Jersey: Prentice Hall.
- Knudsen, T. (2001). Consistency analysis of subspace identification methods based on a linear regression approach. *Automatica* **37**(1), 81–89.
- Knudsen, T., P. Andersen and S. Toffner-Clausen (1997). Comparing PI and robust pitch controller on a 400 kW wind turbine by full scale tests. In *Proceedings of the European Wind Energy Conference (EWEC)*, Dublin, Ireland.
- Ko, J., A. J. Kurdila and T. W. Strganac (1997). Nonlinear control of a prototypical wing section with torsion nonlinearity. *Journal of Guidance, Control and Dynamics* **20**(6), 1181–189.
- Krzanowski, W. J. (1988). *Principles of Multivariate analysis*. Oxford University Press.
- Langbort, C., R. S. Chandra and R. D'Andrea (2004). Distributed control design for systems interconnected over an arbitrary graph. *IEEE Transactions on Automatic Control* **49**(8), 1502–1519.
- Lau, E. and A. J. Krener (1999). LPV control of two dimensional wing flutter. In *Proceedings of the 38th IEEE Conference on Decision and Control (CDC)*, Phoenix, USA.
- Lee, K. W. and S. N. Singh (2007). Global robust control of an aeroelastic system using output feedback. *Journal of Guidance, Control, and Dynamics* **30**(1), 271–275.
- Lee, L. H. and K. Poolla (1999). Identification of linear parameter-varying systems using non-linear programming. *Journal of Dynamic Systems, Measurements and control* **121**(1), 71–78.
- Leishman, J. G. (2002). Challenges in modelling the unsteady aerodynamics of wind turbines. *Wind Energy* **5**(2-3), 85–132.
- Leith, D. J. and W. E. Leithead (1996). Appropriate realization of gain-scheduled controller with application to wind turbine regulation. *International Journal of Control* **65**(2), 223–248.
- Lescher, F., J. Y. Zhao and A. Martinez (2006). Multiobjective H_2/H_∞ control of a pitch regulated wind turbine for mechanical load reduction. In *Proceedings of the European Wind Energy Conference (EWEC)*, Athens, Greece.
- Lind, R and D. H. Baldelli (2005). Identifying parameter-dependent volterra kernels to predict aeroelastic instabilities. *AIAA journal* **43**(12), 2496–2502.
- Ljung, L. (1987). *System identification; Theory for the user*. Prentice Hall.
- Ljung, L. and T. McKelvey (1996). Subspace identification from closed loop data. *Signal Processing* **52**(2), 209–215.
- Lovera, M. (1997). *Subspace identification methods: theory and applications*. Ph. D. thesis, Politecnico Di Milano.

- Lovera, M., P. Colaneri, C. Malpica and R. Celi (2006). Closed-loop aeromechanical stability of hingeless rotor helicopters with higher harmonic control. *Journal of Guidance, Control, and Dynamics* **29**(1), 179–189.
- Lovera, M., P. Colaneri, C. Malpica and R. Celi (2007). Discrete-time, closed-loop aeromechanical stability analysis of helicopters with higher harmonic control. *Journal of Guidance, Control, and Dynamics* **30**(5), 1249–1260.
- Lovera, M. and G. Mercere (2007). Identification for gain-scheduling: balanced subspace approach. In *Proceedings of the American Control Conference (ACC)*, New York City, USA.
- MacNeil, J. B., R. E. Kearney and I. W. Hunter (1992). Identification of time-varying biological systems from ensemble data. *IEEE Transactions on Biomedical Engineering* **39**(12), 1213–1225.
- Manwell, J.F., J.G. McGowan and A.L. Rogers (2002). *Wind energy explained; Theory, design and application*. Chichester: John Wiley and Sons, Ltd.
- Marrant, B. and Th. Van Holten (2004). System identification for the analysis of aeroelastic stability of wind turbine blades. In *Proceedings of the European Wind Energy Conference (EWEC)*, Londen, England.
- Marrant, B.A.H. and Th. Van Holten (2006). Comparison of smart rotor blade concepts for large offshore wind turbines. In *Offshore Wind Energy and other Renewable Energies in Mediterranean and European Seas*, Civitavecchia, Italy.
- Molenaar, D-P. (2003). *Cost effective design and operation of variable-speed wind turbines*. Ph. D. thesis, Technical university of Delft.
- Murray-Smith, R. and T. A. Johansen (1997). *Multiple Model Approaches to Modelling and Control*. London: Taylor & Franchis.
- Nemani, M., R. Ravikanth and B. A. Bamieh (1995). Identification of linear parametrically varying systems. In *Proceedings of the 34th IEEE Conference on Decision and Control (CDC)*, New Orleans, USA.
- Ogata, K. (1995). *Discrete-Time Control Systems (2nd edition)*. Prentice Hall.
- Ogata, K. (1997). *Modern Control Engineering*. Prentice Hall.
- Ohtsubo, K. and H. Kajiwara (2004). LPV technique for rotational speed control of wind turbines using measured wind speed. *OCEANS '04. MTTs/IEEE TECHNO-OCEAN '04* **4**, 1847–1853.
- Østergaard, K. Z. (2008). *Robust, Gain-Scheduled Control of Wind Turbines*. Ph. D. thesis, Aalborg University.
- Østergaard, K. Z., P. Brath and J. Stoustrup (2007a). Estimation of effective wind speed. *Journal of Physics, Conference series: The Science of Making Torque from Wind* **75**(012082).

- Østergaard, K. Z., P. Brath and J. Stoustrup (2007b). Gain-scheduled linear quadratic control of wind turbines operating at high wind speed. In *Proceedings of the 16th IEEE International Conference on Control Applications (CCA), Part of IEEE Multi-conference on Systems and Control*, Singapore.
- Østergaard, K. Z., J. Stoustrup and P. Brath (2008a). Linear parameter varying control of wind turbines covering both partial and full load conditions. To appear in *International Journal of Robust and Nonlinear Control*, special issue on Wind turbines: New challenges and advanced control solutions -(-), -.
- Østergaard, K. Z., J. Stoustrup and P. Brath (2008b). Rate bounded linear parameter varying control of a wind turbine in full load operation. In *Proceedings of the 17th IFAC world congress*, Seoul, South-Korea.
- Pires, L. (2008). Closed-loop system identification for model-based control design; feasibility study on a SWT 2.3-93 wind turbine. Master's thesis, Delft University of Technology.
- Previdi, F. and M. Lovera (2004). Identification of nonlinear parametrically varying models using separable least squares. *International Journal of Control* 77(16), 1382–1392.
- Qin, W. and Q. Wang (2007). An LPV approximation for admission control of an internet web server: identification and control. *Control Engineering Practice* 15(12), 1457–1467.
- Rice, J.K. and M. Verhaegen (2008). Distributed control: A sequentially semi-separable approach. In *Proceedings of the 47th IEEE Conference on Decision and Control (CDC)*, Cancun, Mexico.
- Rice, J.K. and M. Verhaegen (2009). Distributed control: A sequentially semi-separable approach for spatially heterogeneous linear systems. To appear in *IEEE Transactions on Automatic Control* -(-), -.
- Rossetti, M., M. Guadayal, O. Goula and J. Prats (2008). Wind turbine dynamics validation and controller optimisation. In *Proceedings of the European Wind Energy Conference (EWEC)*, Brussels, Belgium.
- Rugh, W.J. (1996). *Linear System Theory*. Upper Sadle River.
- Salcedo, J.V., M. Martinez, C. Ramos and J.M. Herrero (2007). Predictive LPV control of a liquid-gas separation process. *Advances in Engineering Software* 38(7), 466–474.
- Santos, P.L. dos, J. A. Ramos and J. L. M. de Carvalho (2006). Subspace identification of linear parameter varying systems with innovation-type noise models. In *Controlo; 7th portuguese conference on automatic control*, Lisbon, Portugal.
- Santos, P. L. dos, J. A. Ramos and J. L. M. de Carvalho (2005). Identification of bilinear systems using an iterative deterministic-stochastic subspace approach. In *Proceedings of the 44th IEEE Conference on Decision and Control (CDC), and the European Control Conference (ECC)*, Seville, Spain.

- Scherer, C. W. (2001). LPV control and full block multipliers. *Automatica* **37**(3), 361–375.
- Seber, G. A. F. (1984). *Multivariate Observations*. Wiley.
- Selvam, K., S. Kanev, J. W. van Wingerden, T. van Engelen and M. Verhaegen (2008). Feedback-feedforward individual pitch control for wind turbine load reduction. To appear in *International Journal of Robust and Nonlinear Control*, special issue on Wind turbines: New challenges and advanced control solutions -(–), –.
- Shamma, J. and M. Athans (1991). Guaranteed properties of gain scheduled control for linear parameter varying plants. *Automatica* **27**(3), 559–564.
- Sima, D. M. (2006). *Regularization techniques in model fitting and parameter estimation*. Ph. D. thesis, Katholieke Universiteit Leuven.
- Smola, A. J. and B. Scholkopf (2000). Sparse greedy matrix approximation for machine learning. In *Proceedings of the 17th International Conference on Machine Learning*, San Francisco, USA.
- Standish, K.J. and C.P. Van Dam (2005). Computational analysis of a microtab-based aerodynamic load control system for rotor blades. *Journal of the american helicopter society* **50**(3), 249–258.
- Steinbuch, M., R. van den Molengraft and A. van der Voort (2003). Experimental modelling and LPV control of a motion system. In *Proceedings of the 2003 American Control Conference (ACC)*, Denver, USA.
- Stol, K. A. (2001). *Dynamics modelling and periodic control of horizontal-axis wind turbines*. Ph. D. thesis, Faculty of the Graduate School of the University of Colorado.
- STW (2008). Dutch national funding agency STW. Available from Internet: <http://www.stw.nl/>. Last visited 9 June 2008.
- Suykens, J. A. K., T. V. van Gestel, J. De Brabanter, B. De Moor and J. Vandewalle (2002). *Least squares support vector machines*. World Scientific.
- Takahashi, K. and S. G. Massaquoi (2007). Neuroengineering model of human limb control - gainscheduled feedback control approach. In *Proceedings of the 46th IEEE Conference on Decision and Control (CDC)*, New Orleans, USA.
- Tanelli, M., Ardagna D. and M. Lovera (2008). LPV model identification for power management of web service systems. In *Proceedings of the IEEE Multi-conference on Systems and Control (MSC)*, San Antonio, USA.
- The Mathworks (2008). Website simulink. Available from Internet: <http://www.mathworks.com/products/simulink/>. Last visited 22 January 2008.
- The Mathworks (2008). Website matlab. Available from Internet: <http://www.mathworks.com/products/matlab/>. Last visited 22 January 2008.

- Theodorsen, Th. (1935). General theory of aerodynamic instability and the mechanism of flutter. Report 496, N.A.C.A.
- Tóth, R., F. Felici, P. S. C. Heuberger and P. M. J. Van den Hof (2007). Discrete time LPV I/O and state-space representations, differences of behavior and pitfalls of interpolation. In *Proceedings of the European Control Conference (ECC)*, Kos, Greece.
- Tóth, R., F. Felici, P. S. C. Heuberger and P. M. J. Van den Hof (2008). Crucial aspects of zero-order-hold LPV state-space system discretization. In *Proceedings of the 17th IFAC world congress*, Seoul, South-Korea.
- Tóth, R., P. S. C. Heuberger and P. M. J. Van Den Hof (2007). LPV system identification with globally fixed orthonormal basis functions. In *Proceedings of the 46th IEEE Conference on Decision and Control (CDC)*, New Orleans, USA.
- Troldborg, N. (2005). Computational study of the Riso-B1-18 airfoil with a hinged flap providing variable trailing edge geometry. *Wind Engineering* **29**(2), 89–113.
- UPWIND (2008). UpWind project site. Available from Internet: <http://www.upwind.eu/>. Last visited 19 June 2008.
- van Baars, G. E. and P. M. M. Bongers (1992). Wind turbine control design and implementation based on experimental models. In *Proceedings of the 31st IEEE Conference on Decision and Control (CDC)*, Tucson, Arizona, USA.
- van Baars, G. E. and P. M. M. Bongers (1994). Closed loop system identification of an industrial wind turbine system: Experiment design and first validation results. In *Proceedings of the 33rd IEEE Conference on Decision and Control (CDC)*, Lake Buena Vista, Florida, USA.
- van Baars, G. E., H. Mosterd and P. M. M. Bongers (1993). Extension to standard system identification of detailed dynamics of a flexible wind turbine. In *Proceedings of the 32nd IEEE Conference on Decision and Control (CDC)*, San Antonio, Texas, USA, pp. 3514–3519.
- van Dam, C. P., R. Chow, J.R. Zayas and D. A. Berg (2007). Computational investigations of small deploying tabs and flaps for aerodynamic load control. *Journal of Physics, Conference series: The Science of Making Torque from Wind* **75**(012027).
- Van den Hof, P. M. J. (2004). *System identification, lecture notes*.
- Van den Hof, P. M. J. and R. J. P. Schrama (1995). Identification and control – closed-loop issues. *Automatica* **31**(12), 1751–1770.
- van der Hooft, E.L., P. Schaak and T.G. van Engelen (2003). Wind turbine control algorithms. Technical Report DOWEC-F1W1-EH-030094/0, ECN.
- van der Hooft, E. L. and T.G. van Engelen (2004). Estimated wind speed feed forward control. In *Proceedings of the European Wind Energy Conference (EWEC)*, London, England.

- van der Hooft, E. L. and T. G. van Engelen (2003). Feed forward control of estimated wind speed. Technical report ECN-C03-137, Energy research Center of the Netherlands.
- van der Tempel, J. (2006). *Design of support structure for offshore wind turbines*. Ph. D. thesis, Delft University of Technology.
- van Engelen, T.G. (2006). Design model and load reduction assessment for multi-rotational mode individual pitch control. In *Proceedings of the European Wind Energy Conference (EWEC)*, Athens, Greece.
- van Engelen, T. (2007). Control design based on aero-hydro-servo-elastic linear models from TURBU (ECN). In *Proceedings of the European Wind Energy Conference (EWEC)*, Milan, Italy.
- van Engelen, T., H. Markou, T. Buhl and B. Marrant (2007). Morphological study of aeroelastic control concepts for wind turbines. Technical report, Energy Research Center (ECN).
- van Engelen, T.G. and E.L. van der Hooft (2004). Dynamic inflow compensation for pitch controlled wind turbines. In *Proceedings of the European Wind Energy Conference (EWEC)*, London, England.
- van Kuik, G.A.M., Th. van Holten and M. Verhaegen (2003). STW project proposal: Smart dynamic rotor control of large offshore windturbines. Research proposal, University of technology.
- Van Overschee, P. and B. De Moor (1996). *Subspace Identification for linear systems*. Kluwer Academic Publishers.
- van Wingerden, J. W. (2004a). Analytical modeling of piezoelectric actuated mechanical structures; literature survey. Technical Report CTB 534-04-1266, Philips Applied Technologies.
- van Wingerden, J. W. (2004b). Control of flexible motion systems using piezoelectric over-actuation. Technical Report CTB 534-04-1610, Philips Applied Technologies.
- van Wingerden, J. W., F. Felici and M. Verhaegen (2007). Subspace identification of MIMO LPV systems using piecewise constant scheduling sequence with hard/soft switching. In *Proceedings of the European Control Conference (ECC)*, Kos, Greece.
- van Wingerden, J. W., I. Houtzager, F. Felici and M. Verhaegen (2008a). Closed-loop identification of the time-varying dynamics of variable-speed wind turbines. To appear in *International Journal of Robust and Nonlinear Control*, special issue on Wind turbines: New challenges and advanced control solutions -(-), -.
- van Wingerden, J. W., I. Houtzager, F. Felici and M. Verhaegen (2008b). Closed-loop identification of the time-varying dynamics of variable-speed wind turbines. In *Proceedings of the 17th IFAC world congress*, Seoul, South-Korea.

- van Wingerden, J. W., A.W. Hulskamp, T. Barlas, B. Marrant, G. A. M. Van Kuik, D-P. Molenaar and M. Verhaegen (2008). On the proof of concept of a 'smart' wind turbine rotor blade for load alleviation. *Wind Energy* **11**(3), 265–280.
- van Wingerden, J. W. and M. Verhaegen (2007). Subspace identification of bilinear systems using a dedicated input sequence. In *Proceedings of the 46th IEEE Conference on Decision and Control (CDC)*, New Orleans, USA.
- van Wingerden, J. W. and M. Verhaegen (2008a). Subspace identification of MIMO LPV systems: the PBSID approach. In *Proceedings of the 47th IEEE Conference on Decision and Control (CDC)*, Cancun, Mexico.
- van Wingerden, J. W. and M. Verhaegen (2008b). Subspace identification of multivariable LPV systems: a novel approach. In *Proceedings of the IEEE Multi-conference on Systems and Control (MSC)*, San Antonio, USA.
- van Wingerden, J. W. and M. Verhaegen (2009). Subspace identification of Bilinear and LPV systems for open and closed loop data. To appear in *Automatica* (-), -.
- Varga, A. (2005). On solving discrete-time periodic Riccati equations. In *Proceedings of the 16th IFAC World Congress*, Prague, Czech Republic.
- Verdult, V. (2002). *Nonlinear system identification; A state-space approach*. Ph.D. thesis, University of Twente.
- Verdult, V., N. Bergboer and M. Verhaegen (2003). Identification of fully parameterized linear and nonlinear state-space systems by projected gradient search. In *Proceedings of the IFAC symposium on system identification (SYSID)*, Rotterdam, The Netherlands.
- Verdult, V., L. Ljung and M. Verhaegen (2002). Identification of composite local linear state-space models using a projected gradient search. *International Journal of Control* **75**(16), 1385–1398.
- Verdult, V. and M. Verhaegen (2000). Identification of multivariable linear parameter-varying systems based on subspace techniques. In *Proceedings of the 39th IEEE Conference on Decision and Control (CDC)*, Sydney, Australia.
- Verdult, V. and M. Verhaegen (2001). Identification of multivariable bilinear state-space systems based on subspace techniques and separable least squares optimization. *International Journal of Control* **74**(18), 1824–1836.
- Verdult, V. and M. Verhaegen (2002). Subspace identification of multivariable linear parameter-varying systems. *Automatica* **38**(5), 805–814.
- Verdult, V. and M. Verhaegen (2004). Subspace identification of piecewise linear systems. In *Proceedings of the 43rd IEEE Conference on Decision and Control (CDC)*, Atlantis, Paradise Island, Bahamas, USA.
- Verdult, V. and M. Verhaegen (2005). Kernel methods for subspace identification of multivariable LPV and bilinear systems. *Automatica* **41**(9), 1557–1565.

- Verdult, V., M. Verhaegen, C.T. Chou and M. Lovera (1998). Efficient and systematic identification of MIMO bilinear state-space models. In *Proceedings of the 37th IEEE Conference on Decision and Control (CDC)*, Tampa, Florida, USA.
- Verhaegen, M. (1994). Identification of the deterministic part of MIMO state-space models given in innovations form from input-output data. *Automatica* **30**(1), 61–74.
- Verhaegen, M. and P. Dewilde (1992). Subspace model identification part 1: The output-error state-space model identification class of algorithms. *International Journal of Control* **56**(5), 1187–1210.
- Verhaegen, M. and V. Verdult (2007). *Filtering and System Identification: An introduction*. Cambridge University Press.
- Verhaegen, M. and X. Yu (1995). A class of subspace model identification algorithms to identify periodically and arbitrarily time varying systems. *Automatica* **31**(2), 201–216.
- Wei, X. (2006). *Adaptive LPV techniques for Diesel engines*. Ph. D. thesis, Johannes Kepler University.
- Wijnheijmer, F., G. Naus, W. Post, M. Steinbuch and P. Teerhuis (2006). Modelling and LPV control of an electro-hydraulic servo system. In *Proceedings of the 2006 IEEE International Conference on Control Applications (CCA)*, Munich, Germany.
- Witteveen, J. A. S., S. Sarkar and H. Bijl (2007). Modeling physical uncertainties in dynamic stall induced fluid-structure interaction of turbine blades using arbitrary polynomial chaos. *Computer and Structures* **85**(11-14), 866–878.
- WMC (2008). The Knowledge Centre WMC (Wind turbine, Materials and Constructions). Available from Internet: <http://www.wmc.eu/>. Last visited 22 January 2008.
- Wright, A. D. (2004). *Modern control design for flexible wind turbines*. Ph. D. thesis, National Renewable Energy Laboratory.
- Wu, F. and K. Dong (2006). Gain-scheduling control of LFT systems using parameter-dependent Lyapunov functions. *Automatica* **42**(1), 39–50.
- WWEA (2008). World Wide Energy Association (WWEA). Available from Internet: <http://www.wwindea.org/>. Last visited 14 February 2008.
- Yoo, K-Y. and H-K. Rhee (2002). LF-LPV input/output data-based predictive controller design for nonlinear systems. *AIChE journal* **48**(9), 1981–1990.
- Zayas, J.R., C. P. van Dam, R. Chow, J.P. Baker and E.A. Mayda (2006). Active aerodynamic load control for wind turbine blades. In *Proceedings of the European Wind Energy Conference (EWEC)*, Athens, Greece.
- Zeng, Y and S. N. Singh (1998). Output feedback variable structure adaptive control of an aeroelastic system. *Journal of Guidance, Control and Dynamics* **21**(6), 830–837.

Zhou, K., J.C. Doyle and K. Glover (1996). *Robust and optimal control*. New Jersey: Upper Saddle River.

List of Abbreviations

1P	once-per-revolution
2-D	2-dimensional
3-D	3-dimensional
3P	three-times-per-revolution
ADC	Analog to Digital Converter
ARX	Auto-Regressive with eXogenous input
CCA	Canonical Correlation Analysis
CFD	Computational Fluid Dynamics
DAC	Digital to Analog Converter
FP	First Principles
GCV	Generalized Cross Validation
HAWT	Horizontal Axis Wind Turbine
IO	Input-Output
IPC	Individual Pitch Control
LFT	Linear Fractional Transformation
LPV	Linear Parameter-Varying
LS	Least Squares
LTI	Linear Time-Invariant
LTV	Linear Time-Varying
MEM-tabs	Micro-Electro-Mechanical translational tabs
MIMO	Multi-Input Multi-Output
OE	Output Error
PBSID	Predictor-Based Subspace IDentification
PSD	Power Spectral Density
PZT	Piezoelectric patches

SISO	Single-Input Single-Output
SLS	Separable Least Squares
SMI	Subspace Model Identification
SNR	Signal-to-Noise Ratio
SSARX	State-Space Auto-Regressive with eXogenous input
SVD	Singular Value Decomposition
SVM	Support Vector Machines
Tikh	Tikhonov Regularization
VAF	Variance-Accounted-For

Summary

Control of Wind Turbines with ‘Smart’ Rotors: Proof of Concept & LPV Subspace Identification

Jan-Willem van Wingerden

Active control is becoming more and more important for the wind energy community. If we compare the ‘old’ stall regulated turbines with today’s individual pitch controlled turbines we see that the loads can be considerably reduced, leading to lighter or larger turbines. However, limited actuator bandwidth and component fatigue impose significant constraints on the pitch system. Furthermore, with the trend to go offshore it is of interest to increase the rotor diameter as much as possible because the foundation costs of offshore wind turbines amount to a large part of the total costs. Due to the increasing size of wind turbines and the limitations of individual pitch control, it is thus necessary to look ahead to new control concepts which can impose a force profile matching the distributed nature of turbulence, and guarantee an economic lifetime of 20 years for the next generation of offshore wind turbines (diameter over 150 meters).

One novel concept is to use multiple control devices that locally change the force profile on the wind turbine blade to copy the spatially distributed nature of turbulence. The success of distributed load reduction greatly depends on the selection of appropriate sensors that measure the loads and a controller that manipulates the measured signals and generates an actuation signal. This overall combination of sensors, actuators, and control is defined as the ‘smart’ rotor concept. We contribute in this thesis to the development of this new concept by showing the feasibility of the ‘smart’ rotor under realistic wind turbine conditions (*i.e.* realistic disturbances, feedback control, and load measurements). In the wind tunnel we showed, on our designed experimental setup, that when the disturbance is known, perfect cancellation of the disturbance can be realized. However, under realistic circumstances the disturbance is not known and feedback control is required. For this situation we showed the broadband load reduction capabilities of the ‘smart’ rotor for different load cases (*e.g.* with a sinusoidal disturbance we showed that we can reduce the amplitudes by 90% at the first eigenfrequency of the blade). The success of this proof of concept should help aerodynamicists and structural experts to embrace control engineering much earlier in their design cycle, and utilize it as a lever to create additional design freedom.

Besides demonstrating the *proof of concept of the 'smart' rotor*, we developed tools to obtain experimental models by introducing novel *subspace Linear Parameter-Varying system identification* algorithms. The motivation for this is that in industry it is common practice to use first principles models to optimize controllers. However, such models are directly calibrated with measured data which is often of very limited use for model based controller design. It is well known that first principles models often have a large number of tunable parameters leading to a nonconvex, time-consuming, calibration process. Furthermore, the obtained models may be too complex for controller synthesis (high-order, nonlinear, etc.). One of the most promising solutions is the application of system identification.

In our experiment the dynamics can be considered to be Linear Time-Invariant (LTI) due to the constant wind speed. However, 'real' wind turbines are nonlinear systems and their dynamics are time-varying. This time-variation is characterized by: rotor speed, pitch position, and rotor position, etc., which are all measurable quantities. For the identification of such systems, computationally efficient schemes are still lacking. In this thesis we addressed this problem by developing novel algorithms for the identification of parameter-dependent dynamics. We developed linear parameter-varying identification frameworks that cope with either periodically varying dynamics or arbitrarily parameter-varying dynamics. Both frameworks can handle data generated in open-loop and closed-loop. Although this work is introduced from a wind turbine perspective, it can also be used for other applications.

Samenvatting

Het Regelen van Windturbines met een 'Slimme' Rotor: Proof of Concept & LPV Subspace Identificatie

Jan-Willem van Wingerden

Het actief regelen van windturbines wordt steeds belangrijker voor de wind-energiegemeenschap. Als we bijvoorbeeld de 'oude' op overtrek geregelde turbines vergelijken met de huidige bladhoekgeregelde turbines, zien we dat de belastingen aanzienlijk verminderd zijn. Dit resulteert in lichtere en grotere turbines. Echter, door de gelimiteerde bandbreedte van de benodigde actuatoren en de vermoeiingsbelastingen op de verschillende componenten zijn de grenzen van de bladhoekregeling bereikt. Toch wil men naar grotere windturbines, vanwege het feit dat steeds meer windturbines in zee gezet worden. De kostprijs van deze turbines wordt in grote mate bepaald door de funderingskosten. Door de toenemende grootte van windturbines, waardoor ook de belastingen toenemen, en de beperkingen van de bladhoekregelingen is het noodzakelijk om andere concepten te onderzoeken. Met een nieuw concept moet een economische levensduur van 20 jaar gewaarborgd worden voor de nieuwe generatie windturbines (met een diameter groter dan 150 m).

Het meest veelbelovende concept maakt gebruik van een aantal regelbare flapjes die lokaal op het blad de liftkrachten kunnen regelen en zodoende het krachterspel op de windturbine kunnen beïnvloeden. Dit dient op een zodanige manier te gebeuren dat de turbine zich aanpast aan het gedistribueerde gedrag van de verstoringen (b.v. turbulentie). Het succes van dit concept hangt grotendeels af van de keuze voor de sensor die de verstoring waarneemt en van de regelaar die het gemeten signaal manipuleert om vervolgens de flapjes uit te sturen. Een rotor uitgerust met dergelijke sensoren, flapjes en regelaars valt onder de noemer 'slimme' rotor. In dit proefschrift wordt bijgedragen aan de ontwikkeling van een 'slimme' rotor, door experimenteel aan te tonen dat onder realistische omstandigheden dit concept werkt. In de windtunnel hebben we laten zien dat als de verstoring bekend is, deze compleet kan worden onderdrukt. Onder realistische omstandigheden is dit niet het geval en terugkoppelregelingen dienen dan te worden gebruikt. Voor deze situatie zijn een aantal belastingsscenario's gedefinieerd en er wordt getoond dat de verstoringen breedbandig kunnen worden onderdrukt. (b.v. bij een sinusverstoring met dezelfde frequentie als de eigen

trillingfrequentie van het rotorblad, konden de amplitudes met 90% gereduceerd worden). Tevens laat deze studie zien dat het van essentieel belang is om regeltechnici in een vroeg stadium te betrekken bij het ontwerp van een dergelijk multidisciplinair concept.

Naast het demonstreren van de haalbaarheid van het 'slimme' rotor concept, zijn er ook nieuwe methodes ontwikkeld om uit meetdata wiskundige modellen te schatten met variërende parameters. De motivering voor de ontwikkeling van een dergelijke methode komt voort uit het feit dat in de windindustrie deze modellen worden opgesteld aan de hand van fysische wetten. Deze modellen worden gekalibreerd met gemeten data en zijn meestal niet direct bruikbaar voor modelgebaseerd regelen. Deze modellen bevatten namelijk een hoog aantal onbekende parameters en het kalibreren is uiterst tijdrovend en is een niet-convex optimalisatieproces. Verder zijn deze modellen doorgaans toegespitst op belastingsberekeningen, in plaats van regelaarontwerp, waardoor de modellen te complex zijn (niet lineair, hoge orde, etc.). Eén van de veelbelovende oplossingen is het gebruik maken van modelleringstechnieken gebaseerd op meetdata: systeemidentificatie.

Tijdens de windtunnelexperimenten waren de stromingscondities constant, waardoor er gebruik kon worden gemaakt van lineaire tijdinvariante systeemidentificatietechnieken. Echter, in werkelijkheid zijn deze stromingscondities tijdsvariant. Deze tijdsvariatie kan geparameteriseerd worden in de rotorsnelheid, rotorpositie en bladhoek. Voor de identificatie van dergelijke systemen bestaan er nog geen efficiënte identificatietechnieken. In dit proefschrift zijn er efficiënte identificatietechnieken ontwikkeld, gebaseerd op subspace identificatie, die om kunnen gaan met dergelijke Lineair Parameter Variërende (LPV) dynamica. Door het uitbuiten van de structuur in de parametervariatie en de datamatrices zijn er twee nieuwe identificatiemethodes ontwikkeld. Beide methodes kunnen omgaan met data verkregen uit zowel open- als gesloten-lus experimenten. Hoewel dit werk uit het perspectief van de windenergie is gemotiveerd, hebben de identificatiemethodes een veel breder toepassingsgebied.

Curriculum Vitae

Jan-Willem van Wingerden was born on December 9, 1980 in Ridderkerk, The Netherlands. He started his pre-university school (VWO) in 1993 at the Farel College in Ridderkerk. In 1999 he started his study Mechanical Engineering at the Delft University of Technology, where he graduated cum laude in December 2004 at the Control Engineering group, nowadays part of the Delft Center for Systems and Control (DCSC). His graduation project was carried out at Philips Applied Technologies in Eindhoven. For his master thesis entitled: *Control of motion systems using piezoelectric over-actuation* he received two graduate awards: *Betaafs Genootschap der Proefondervindelijke Wijsbegeerte graduate award* and the *Universiteitsfonds best graduate award*.

In January 2005, he started his Ph.D. project entitled: *'Smart' dynamic rotor control for large offshore wind turbines* within the Delft Center for Systems and Control (DCSC) and the Delft University Wind Energy Research Institute (DUWIND) under the supervision of Michel Verhaegen. Within this project he mainly worked on subspace LPV system identification, and 'Smart' rotor control. During his Ph.D. he supervised 10 Master students and received his DISC certificate.

