Identification of Linear Parameter-Varying Input-Output Models

Vom Promotionsausschuss der
Technischen Universität Hamburg-Harburg
zur Erlangung des akademischen Grades

Doktor-Ingenieur

genehmigte Dissertation

von

Sudchai Boonto

aus Samutprakarn, Thailand

2011

Abstract

This thesis focuses on the identification of *Linear Parameter-Varying* (LPV) input-output models. Three problems are considered: the approximation of the nonlinear scheduling function of LPV systems, the unbiased identification of LPV systems, and the state-space realization of LPV input-output models.

The approximation of the nonlinear scheduling function of LPV systems is conducted by using cubic spline basis functions which are more easily tunable and less oscillatory than polynomial functions. A *Separable Least-Squares* (SLS) algorithm is proposed to reduce the number of parameters by separating them into linear and nonlinear parameters. Moreover for unstable systems, models are identified in closed-loop by using a two-step method with a neural network as a noise filter. Simulation and experimental results are given; they demonstrate the efficiency of the presented approach.

Concerning the unbiased identification of LPV systems, the bias error is reduced by using the "*instrumental variable with auxiliary model*" method. To improve the performance while still maintaining simplicity, an auxiliary model with output error structure, which is estimated by the output error method, is used. The proposed approach gives a significant improvement in terms of bias error and variance. A comparison with an existing method is illustrated with several simulation examples.

A state-space realization of LPV input-output models using a *Linear Time-Invariant* (LTI) framework introduces undesired dynamic dependence on the parameters. To solve this problem, the identification of LPV systems in a *Left Polynomial Representation* (LPR) is proposed. By using skew polynomials, a systematic realization procedure for LPV systems is presented. The resulting state-space models involve only static dependence on the scheduling parameters and are in observable form and minimal. Therefore standard LPV controller synthesis techniques can be used without loosing closed-loop performance. The efficiency of the proposed method is demonstrated by applying it to a laboratory scale magnetic levitation plant and an arm-driven inverted pendulum.

Acknowledgements

Writing this thesis at the Institute of Control Systems (ICS), Hamburg University of Technology could not be accomplished without different kinds of support and guidance from others. This page is dedicated to those who contributed to this work.

First of all, I would like to thank my advisor, Prof. Dr. Herbert Werner, for his support, guidance, patience and encouragement. He has given me the freedom to pursue my own interests. I am also thankful for numerous discussions, critical comments, constructive suggestions, and careful readings of this thesis.

I would like to thank Prof. Dr. Wolfgang Mackens for serving at the oral examination and the evaluation committee. I am also thankful to Prof. Dr.-Ing. Günter Ackermann for his kindness on being the chairman of the examination committee.

Many thanks go to colleagues who have given technical help while this research was going on. In particular, I am much obliged to Ahsan Ali for proofreading this thesis and many interesting discussions. Special thanks go to Dr.-Ing. Gerwald Lichtenberg for proofreading this thesis. His patience and gentleness are sincerely appreciated. I am also thankful to Andrey Popov, Andreas Kominek, Seyed Mahdi Hashemi, and Dr.-Ing. Hossam Seddik Abbas for their fruitful discussions. Special thank, to Dr.-Ing Adel Omran Farag, Dr.-Ing Juma Wady Bader, Dr.-Ing Christian Schmidt, Sven Pfeiffer and Qin Liu who shared an office with me generation after generation. Working in this room would be unpleasant and joyless if they had not allowed Pannawish Boonto, my son, to use some area of the room like his playground. I would like to thank Dr.-Ing. Saadia Faisal, Dr.-Ing. Martyn Durrnat, Dr.-Ing Andreas Kwiatkowski, Ulf Pilz, Long van Doan, Dr.-Ing. Mukhtar Ali, Dr. Sault Chughtai, Mochammad Sahal, Esteban Rosero, Annika Eichler and Georg Pangalos for their friendship and assistance. My thanks go to Herwig Meyer, Klaus Baumgart and Uwe Jahns for their cordiality and technical support. I also acknowledge the contributions from my graduate students notably Jonas Witt and Ireneus Wior.

Living aboard would never have been so easy had it not been for Birthe von Dewitz, the secretary of the ICS. I would like to express my deepest gratitude to her and her family who have made Harburg and Winsen as my home away from home. Without their support, encouragement and endless love, me and my family could not have passed the difficult time during this long stay in Germany.

I am indebted to my parents, my parents in law, my brothers, my sisters and my sisters in law for their endless support throughout my Ph.D studies.

Many professors and staff in the Department of Control Systems and Instrumentation Engineering at King Mongkut's University of Technology Thonburi, Thailand, ex-

tended their kind help towards me during hard times. My entire study at Hamburg was partially supported by the Royal Thai Government, which is gratefully acknowledged. I would like to express my gratitude to all staff members of the office of educational affairs, Royal Thai Embassy, Germany. I would also like to thank many of Thai people in Hamburg for their friendship and for making my life in Hamburg so enjoyable.

I am foremost grateful to my wife, Tipaporn Boonto. Without her support and motivation, this research could not be finished. Actually, I am inestimably thankful for all that she has added to my life. Finally I would like to thank my son, Pannawish Boonto. The only way that I could relax during my writing was when I was with him. The young boy could not understand the difficulty of my research, he just wanted to ask me many of his own created quizzes and play fighting with me.

Hamburg, May 2011
Sudchai Boonto

To my parents, wife and son.

Contents

Abstract **iii**

Acknowledgements **v**

Contents **ix**

List of Figures **xiii**

List of Tables **xv**

1 Introduction **1**
- 1.1 Motivation and Objectives 3
  - 1.1.1 Approximation of Nonlinear Scheduling Functions 3
  - 1.1.2 Closed-Loop System Identification 3
  - 1.1.3 Unbiased LPV Input-Output Model Identification 4
  - 1.1.4 Realization of LPV Input-Ouput Models 4
  - 1.1.5 Model Validation Method for LPV models 5
- 1.2 Contribution of This Thesis 5
- 1.3 Thesis Overview 6

2 LPV System Representations and Background Material **9**
- 2.1 LPV State-Space Representation 9
- 2.2 LPV Input-Output Representation 10
- 2.3 LPV and Quasi-LPV Systems 11
- 2.4 Multi-Layer Perceptron Networks 12
- 2.5 Summary 13

3 Identification of LPV Input-Output Models Using Cubic Splines **15**
- 3.1 LPV-ARX Input-Output Models 16
- 3.2 Polynomial Functions and Cubic Spline Functions 19
- 3.3 Separable Least-Squares Algorithm 21
  - 3.3.1 Levenberg-Marquardt Algorithm 23
  - 3.3.2 Recursive Levenberg-Marquardt Algorithm 24
- 3.4 Closed-Loop LPV Input-Output System Identification 25
- 3.5 Validation in The Prediction Error Setting 28
- 3.6 Application to Open Loop Identification of A Two-Tank System 29

3.6.1 System Setup 29
3.6.2 Experimental Results 30
3.7 Application to Closed-Loop Identification 32
3.7.1 Experimental Setup 32
3.7.2 Excitation Signal Design 33
3.7.3 Experimental Results 34
3.8 Conclusions 38

4 Unbiased Identification of LPV Input-Output Models 41
4.1 System Description 42
4.2 Model Structure 45
4.2.1 LPV-ARX Representation 45
4.2.2 LPV-OE Representation 46
4.3 Bias Error of Least Squares Method 46
4.4 Unbiased Identification Methods 47
4.4.1 Output Error Method 47
4.4.2 Output Error Method for LPV Input-Output Model Identification 49
4.4.3 Instrumental Variable Method 50
4.5 Simulation Examples of LPV Systems 51
4.5.1 White Noise Case 52
4.5.2 Colored Noise Case 54
4.6 Identification of Quasi-LPV Input-Output Models 58
4.7 Conclusion 61

5 State-Space Realization of LPV Input-Output Models 63
5.1 Algebraic Preliminaries 65
5.1.1 Skew Polynomials 65
5.1.2 Noncommutative Algebra 66
5.2 LPV State-Space Representation 67
5.3 Realization of SISO LPV Input-Output Models 68
5.3.1 Observable Form 68
5.3.2 Reachable Form 71
5.4 Extension to MIMO Systems 73
5.4.1 Skew Polynomial Matrices 74
5.4.2 MIMO Observable Form 74
5.4.3 Minimal Realization 78
5.5 Application to LPV Input-Output System Identification 80
5.5.1 Identifiability of Model Structures 81
5.5.2 LPV Input-Output Models Identification in Left Polynomial Representation 82
5.6 Conclusion 85

6 Identification for LPV Control of Nonlinear Plants – Applications 87
6.1 ν-Gap and Model Validation for Control 88
6.2 Application to Magnetic Levitation System 90
6.2.1 Physical Plant Model 90
6.2.2 Experiment Setup 91
6.2.3 Identification Result and Controller Synthesis 91
6.3 Arm-Driven Inverted Pendulum (ADIP) 95
6.3.1 Model Validation 95
6.3.2 LPV Controller Synthesis 96
6.4 Conclusion 100

7 Conclusions and Outlook 101

A Derivation of Equations in Chapter 3 103

B Properties of the Kronecker Product 105

C Rings, Fields 107

Bibliography 109

List of Symbols and Abbreviations 119

List of Refereed Publications 123

Curriculum Vitae 125

List of Figures

2.1 LPV system (a) and quasi-LPV system (b) 12
2.2 Two-layer perceptron network 12
2.3 NNARX model structure 13

3.1 The scheduling function is estimated by using the cubic spline function $g_{\mathrm{sp}}(\theta(k))$, and compared it with a 7^{th} order polynomial function $g_{\mathrm{poly}}(\theta(k))$; $S_1 = c_0 + c_1\theta(k)$, $S_2 = c_2|\theta(k) + 4|^3$, $S_3 = c_3|\theta(k) + 1|^3$, $S_4 = c_4|\theta(k)|^3$, $S_5 = c_5|\theta(k) - 1|^3$, and $S_6 = c_6|\theta(k) - 4|^3$ 21
3.2 Closed-loop configuration for identification 27
3.3 NNARX model structure for predicting noise free $u(k)$ 27
3.4 NNARX model structure for predicting noise free $y(k)$ 27
3.5 Configuration for the direct identification method. 28
3.6 Two tanks system 30
3.7 Comparison between measured data (blue) and simulation results for y of the linear model (green), LPV model with polynomial dependence (red) and LPV model with cubic spline dependence (black) 31
3.8 Histogram of VAF values (%) on a cross-validation output for the 100 random initial Cubic Spline knots. 32
3.9 Arm-Driven Inverted Pendulum (ADIP) 33
3.10 Block diagram for closed-loop identification 33
3.11 Spectrum of both normalized outputs of ADIP: upper for φ_1, lower for φ_2. 35
3.12 Comparison between measured data (blue) and simulation results for φ_1 of the linear model (green), LPV model with polynomial dependence (red) and LPV model with cubic spline dependence (black) 36
3.13 Comparison between measured data (blue) and simulation results for φ_2 of the linear model (green), LPV model with polynomial dependence (red) and LPV model with cubic spline dependence (black) 36
3.14 Comparison between measured data (blue) and simulation results for φ_1 of an LPV model with cubic spline dependence with NNARX (black) to generate a noise free scheduling signal and without NNARX (red) . . . 37
3.15 Comparison between measured data (blue) and simulation results for φ_2 of an LPV model with cubic spline dependence with NNARX (black) to generate a noise free scheduling signal and without NNARX (red). . . . 37

3.16 Histogram of VAF values (%) on a cross-validation output φ_1 for the 100 random initial Cubic Spline knots. 39
3.17 Histogram of VAF values (%) on a cross-validation output φ_2 for the 100 random initial Cubic Spline knots. 39

4.1 LPV model for system identification 43
4.2 LPV model structure 43
4.3 Comparison between the recursive least squares and the output error method. 48
4.4 Cross validation test of IV-OE at SNR = 10 dB when the model structure is not correct. 57
4.5 A quasi-LPV data generating system: the dashed box is a nonlinear system represented as a quasi-LPV model. 58
4.6 Cross validation test of IV-OE for a quasi-LPV at SNR = 15 dB when the model structure is not correct. 61

6.1 Closed-loop system 88
6.2 2-DOF control configuration 89
6.3 Magnetic levitation System 91
6.4 Magnetic levitation block diagram for identification. 92
6.5 Cross-validation of the magnetic levitation model. Black: measured output, red: quasi-LPV model, green: ARX model 93
6.6 Generalized plant for LPV controller design 93
6.7 Tracking response of magnetic disks with $\mathcal{H}_\infty$ (green) and LPV (blue) controller. Left hand side: lower disk y_1, right hand side: upper disk y_2. 94
6.8 Block diagram for closed-loop identification 95
6.9 Bode magnitude plot of G_{id} compared with $\hat{G}_{lpv}$ at each frozen parameter where $-0.96 \leq \theta \leq 0.96$ rad and $\theta = \sin(\varphi_1)$: (a) from u to φ_1, (b) from u to φ_2. 96
6.10 Closed-loop simulation of $T(G_{id}, K_{id})$ and $T(\hat{G}_{lpv}, K_{id})$ 97
6.11 Comparison between ν-gap metric of $\hat{G}_{lpv}(\theta)$ and G_{id} with generalized stability margin $b_{\hat{G}_{lpv},K_{id}}$, here $-0.96 \leq \theta \leq -0.96$ where $\theta = \sin(\varphi_1)$. . 98
6.12 Experimental results of ADIP system with LPV controller ($\varphi_1^t = \pm 70°$) 99

List of Tables

2.1 Some special cased of the generalized LPV input-output model. 11

3.1 Model validation 31
3.2 Model validation 38

4.1 Estimation bias 2-norm and its variance at different SNR : LPV with white measurement noise 53
4.2 Mean and standard deviation (Std) of the estimated $A(q^{-1}, \theta(k))$ polynomial parameters at SNR 15 dB : LPV with white measurement noise 54
4.3 Mean and standard deviation (Std) of the estimated $B(q^{-1}, \theta(k))$ polynomial parameters at SNR 15 dB : LPV with white measurement noise 54
4.4 Estimation bias 2-norm and its variance at different SNR : LPV with colored measurement noise 55
4.5 Mean and standard deviation (Std) of the estimated $A(q^{-1}, \theta(k))$ polynomial parameters at SNR 15 dB : LPV with colored measurement noise 56
4.6 Mean and standard deviation (Std) of the estimated $B(q^{-1}, \theta(k))$ polynomial parameters at SNR 15 dB : LPV with colored measurement noise 56
4.7 The properties of each algorithm when use to identify LPV model with colored measurement noise. 57
4.8 Estimation bias 2-norm and its variance at different SNR: quasi-LPV with white measurement noise 59
4.9 Mean and standard deviation (Std) of the estimated $A(q^{-1}, \theta(k))$ polynomial parameters at SNR 20 dB : quasi-LPV with white measurement noise 60
4.10 Mean and standard deviation (Std) of the estimated $B(q^{-1}, \theta(k))$ polynomial parameters at SNR 20 dB : quasi-LPV with white measurement noise 60

6.1 Control performance achieved on ADIP 98

Chapter 1

Introduction

In real life many systems to be controlled are nonlinear. One technique that can be used to deal with nonlinear systems is *gain-scheduling* control. The classical gain-scheduling approach consists of two steps: one designs local linear controllers based on linearization of the nonlinear system at several different operating points; the overall nonlinear controller is then obtained by *interpolating* or *scheduling* among the designed local operation points. The main drawback of the linearization-based gain-scheduling controller design methods is that the controlled system may exhibit a poor response or go unstable when operating points are not at the equilibrium [Rugh and Shamma, 2000].

To overcome the limitations of the classical gain-scheduling technique, the gain-scheduling controller design method based on *Linear Parameter-Varying* (LPV) systems was introduced to the control community in the 1990s, see [Shamma and Athans, 1991; Packard, 1994]. An LPV system has the form of a linear system in which the system matrices in state-space representation or the system transfer operator matrices in input-output representation depend on static functions of the measured scheduling parameters. In contrast to the classical gain-scheduling method, the gain-scheduling controller design based on LPV systems can guarantee stability and optimal performance over the entire range of operation of the system. This design technique has gained a lot of attention since LPV gain scheduling based on $\mathcal{H}_\infty$ and $\mathcal{H}_2$ synthesis using *linear matrix inequalities* (LMIs) optimization was proposed by Apkarian et al. [1995]; Apkarian and Gahinet [1995]. The LPV framework has been applied to a wide area of application, for example; wind turbines [Bianchi et al., 2007], aircraft [Marcos and Balas, 2004], Diesel engines [Kwiatkowski, 2007], etc.

Notwithstanding the above attractive features of LPV systems, it has to be mentioned that the performance of the closed-loop control system depends strongly on the quality of the LPV model being used for controller synthesis. Assuming that the nonlinear dynamic equations of the plant are available, there are several methods to construct LPV models from them. The nonlinear terms can be made linear in parameters by linearization, substituting with functions and considered as linear varying parameters, etc. (see [Kwiatkowski, 2007]). However, with these techniques the models may end up with too many scheduling parameters. Models that have many scheduling parameters are too complicated and unsuitable for LPV controller design [Kwiatkowski, 2007].

Instead of constructing LPV models from nonlinear equations, experimental system identification can be considered as an alternative. In system identification, the aim is to estimate dynamic models directly from measured input and output data. The identification framework of *Linear Time-invariant* (LTI) systems is now firmly established and well known [Ljung, 1999] while the identification of LPV models is still developing. Several problems and challenges have not yet been solved [Tóth, 2010].

The identification of LPV systems has a long history. Here we discuss just some aspects related to the thesis, for the rest the reader is referred to [Tóth, 2010]. LPV identification was first discussed in [Nemani et al., 1995], by assuming that the system has only one varying parameter and the state vector can be measured. The problem can be shown to be equivalent to a linear regression problem and solved by a *Least-Squares* (LS) method. Keeping this track, the most general and widely used technique of LPV input-output model identification, was introduced in the work of Bamieh and Giarré [2002], where the method can be used not only for *Single-Input Single-Output* (SISO) models but also for *Multi-Input Multi-Output* (MIMO) models. The approach uses a linear parameterization of function dependent coefficients with polynomials of scheduling parameters. Then, LS methods and *Recursive Least Square* (RLS) methods can be applied to estimate the models [Bamieh and Giarré, 2002; Wei, 2006]. However, noise issues in the identification were neglected. Due to an interest in applications which are operated under noisy conditions, some attempts have been made to deal with the identification of an LPV input-output model when the measured data is corrupted by noise. The *Instrumental Variable* (IV) method and its variations have been introduced to solve this problem by using auxiliary models in LPV *Auto-Regressive with Exogenous* form (LPV-ARX), LPV *Output Error* (LPV-OE) and LPV *Box-Jenkins* (LPV-BJ) structures [Butcher et al., 2008; Abbas and Werner, 2007; Laurain et al., 2010]. In [Abbas et al., 2010a], this issue is also considered by using a *Linear Recurrent Neural Network* (LRNN). Nevertheless, all LPV input-output models share the same disadvantage as they are not in a structure that is ready to be used by advanced controller synthesis methods. The models have to be converted to state-space forms.

The closed-loop system identification of LPV input-output models is also becoming an active research area. In [Boonto and Werner, 2008] a closed-loop identification technique is proposed for the estimation of LPV models in ARX form by directly extending the *Output Error Method* for the LTI case [Landau and Karimi, 1997]. Use of IV method to the closed-loop problem is also considered in [Abbas and Werner, 2007].

In parallel, a subspace identification approach for LPV state-space models has been developed in [Verdult and Verhaegen, 2000; Verdult, 2002]. This branch of methods employs the concept of LTI subspace identification using the *Multivariable Output-Error State-Space* (MOESP) algorithm (see [Verdult, 2002] and reference therein). LPV descriptions with affine dependence on the time-varying parameters are considered without full state measurements. A recent development of this technique depends on only one single least squares problem and a *Singular Value Decomposition* (SVD) [van Wingerden and Verhaegen, 2009]. The main advantage of subspace-based techniques is that the resulting model can be directly used for MIMO state-space controller synthesis methods. However, in this technique, the number of rows in the data matrices grows exponentially with the system order, which might be not suitable for identifying high-order LPV models. Some significant improvements against this drawback have been presented in [Felici et al., 2007]. The subspace method is also extended to tackle closed-loop identification issues by tailoring the data generated in closed-loop [Felici et al., 2007].

Apart from the two methods mentioned above, there are a few more approaches that can be seen as separate categories of LPV identification, for examples an orthogonal basis function approach [Tóth, 2008], set membership [Belforte et al., 2005; Cerone and Regruto, 2008], nonlinear optimization [Lee and Poolla, 1999].

In this thesis, we aim to improve the LPV input-output identification framework by solving some open problems that have not been adequately treated so far. This will be discussed in the next Section.

1.1 Motivation and Objectives

This thesis deals with identification and control of nonlinear and time-varying systems based on the concept of LPV input-output models. One of the main objectives is to bridge the gap between the LPV identification framework and LPV controller synthesis, with particular emphasis on the following.

1.1.1 Approximation of Nonlinear Scheduling Functions

To identify accurate LPV input-output models (2.6), one crucial issue is to approximate a nonlinear scheduling function. In the initial work in this field [Bamieh and Giarré, 2002], simple polynomials of scheduling variables are used. However, their capabilities are limited and they are difficult to tune.

In the literature, there are few reports that consider this problem. In [Previdi and Lovera, 2003, 2004], nonlinear scheduling functions are identified through a neural network model, while the linear part of the model is estimated in ARX form. Similarly, in [Hsu et al., 2008], a non-parametric method is applied by using a *dispersion function* to estimate the scheduling functions in terms of piece-wise linear functions.

One objective of this thesis is to develop an approximation algorithm which is fairly simple but yet more effective than the polynomial approach.

1.1.2 Closed-Loop System Identification

The *two-step* method is a powerful technique for closed-loop identification [Leskens et al., 2002]. Unfortunately, the method is limited to closed-loops composed of an LTI plant and an LTI controller. In [Forssell and Ljung, 2000], a projection method has been proposed and the restriction to LTI controllers has been removed. An extension to LPV systems is however not considered.

Since the two-step method is attractive in the sense of simplicity, we extend the idea to LPV system identification in closed-loop.

1.1.3 Unbiased LPV Input-Output Model Identification

In the past, the identification of LPV input-output models has been considered as a least squares or recursive least squares problem, see [Bamieh and Giarré, 2002; Wei, 2006; Qin and Wang, 2007a,b]. However, it is well known that these methods are biased due to the correlation between output data and measured noise data [Butcher et al., 2008; Landau and Zito, 2006]. This problem becomes important if the *Signal to Noise Ratio* (SNR) is low. In [Butcher et al., 2008], this problem has been solved by using an *instrumental variable with auxiliary model* [Landau and Zito, 2006]. The proposed method simply uses an LPV-ARX[†] model as an auxiliary model. It turns out that the method is applicable only when the corrupting noise is white. For the colored noise case, a *Refined Instrumental Variable* (RIV) method with LPV-OE and LPV-BJ have been proposed. However, this method requires knowledge of the noise model.

Another way to deal with this issue is a method using a LRNN proposed by Abbas et al. [2010a]. By using an LPV-OE structure, a recurrent neural network is used for searching the model parameters. Since the LPV model is estimated by employing a predicted output which is free of noise, the identified model is unbiased. The main drawback of this method is the large computation time. Moreover, since each recurrent neural network is restricted to one LPV input-output model structure, the overall network has to be reconstructed if the model order is changed.

This thesis extends the instrumental variable with auxiliary model method to LPV identification problems when the corrupting noise is not white and the SNR is low.

1.1.4 Realization of LPV Input-Ouput Models

Since most of the LPV controller synthesis methods require LPV state-space models, the realization of LPV input-output models is essential. It has been recently observed that the realization of LPV input-output representations is not straightforward as in the LTI case [Tóth et al., 2007], due to the fact that the shift operator and the time-varying coefficient functions do not commute. In the LPV literature this problem has been largely ignored [Wassink et al., 2005; Giarré et al., 2006] while it is well known in the *Linear Time-Varying* (LTV) literature, see for example [Kamen, 1976; Verriest, 1993]. However, for the LPV case, the realization issue is more involved than the LTV case because of the infinite dimensional space of the scheduling parameters. Using a recently developed behavioral framework to solve such realization problems [Tóth, 2008], the equivalence between LPV state-space and LPV input-output representations can be obtained by allowing a dynamic mapping between the scheduling parameters and the system matrices. However, this increases the complexity of the resulting state-space models.

In [Abbas et al., 2010b], three LPV input-output model structures for identification, which can be transformed into LPV state-space models without any dynamic dependence of the scheduling parameters, have been introduced. The resulting LPV state-space models are also equivalent to the identified LPV input-output model.

[†] See Chapter 2

In this thesis, we establish a practical and systematic framework for LPV state-space realizations. Compared with the work in [Tóth et al., 2011], this technique should be accessible to control engineers and can be used with both SISO and MIMO systems.

1.1.5 Model Validation Method for LPV models

In closed-loop system identification of unstable models, the cross-validation between simulated outputs of the identified models and the measured data may not give sufficient information about the suitability of a model for controller design. This test cannot show whether or not the model can be stabilized by the controller used for identification. In [Codrons, 2005] and references therein, the ν-gap metric [Vinnicombe, 1993a] is suggested as a validation tool for LTI models. The ν-gap metric can provide a sufficient condition for closed-loop stability. If the ν-gap is larger than the generalized stability margin, one can ensure that the model can be stabilized by the controller used for identification. Moreover, the models can be used to design a better controller than the one being used. In [Fujimori and Ljung, 2007; Wood, 1995] this idea has been extended to LPV models by freezing the scheduling parameters at different values and analyzing the ν-gap for each resulting LTI model. However, there is so far no connection between the system identification and LPV controller design.

One of the objectives of this work is to develop a model validation procedure that can be used as a tool to test whether or not the LPV models identified in closed-loop are suitable for LPV controller design.

1.2 Contribution of This Thesis

The main contributions of this thesis are as follows:

1. An algorithm to approximate nonlinear scheduling functions based on cubic splines is introduced. The algorithm is combined with the LPV input-output identification method by using a *Separable Least-Squares* (SLS) framework and the Levenberg-Marquardt Algorithm. The proposed algorithm gives better results than a polynomial-based method in terms of accuracy. Moreover, a recursive version of the algorithm is also given to be used when the amount of data is too large.

2. An extension of a two-step method for closed-loop identification to LPV models is proposed. Since the plant is no longer assumed to be linear, a linear sensitivity function cannot be used anymore. An NNARX network[†] is introduced to serve as a noise filter. The network can be used not only to remove noise from an input signal but from the scheduling signal also. The latter ability can reduce the bias error of the identified models.

[†] See Chapter 2.

3. An alternative IV method for an unbiased LPV input-output identification is introduced. The method is based on the output error identification algorithm which can construct a better auxiliary model than the traditional LPV-ARX model. The proposed method provides with a much lower complexity a performance comparable to methods such as *Refined Instrumental Variable* (RIV) and *Simplified* RIV (SRIV) [Laurain et al., 2010], which require a larger computational effort.

4. Constructing a state-space realization of an LPV input-output model is a crucial problem. Due to the noncommutativity of the shift operator with a parameter dependent coefficient function of the LPV model, an LTI realization procedure cannot be employed. In this thesis, we propose a realization method which takes the noncommutativity into account by using a skew polynomial concept. The resulting technique preserves the simplicity of the LTI realization and is suitable for used in engineering applications.

5. A general LPV input-output model structure for SISO and MIMO identification is given. Together with the proposed realization method, this model structure can be directly transformed into an LPV state-space model which involves only static dependence.

6. The proposed techniques of this thesis are applied to real plants: a magnetic levitation system and an *Arm Driven Inverted Pendulum* (ADIP). Experimental results show the efficiency of the proposed approaches.

1.3 Thesis Overview

This thesis is organized as follows:

Chapter 2 describes the preliminary background material that will be used throughout the thesis. The LPV representations related to the controller synthesis and identification method are given. The difference between LPV and quasi-LPV systems is also discussed. Finally, the neural network model structures, which are required for LPV closed-loop identification, are provided.

In Chapter 3, we deal with the LPV input-output identification method. This method is focused on a way to approximate nonlinear scheduling functions. The chapter starts by giving a general review of the LPV input-output identification [Bamieh and Giarré, 2002] approach. Then, the difference between a polynomial and a cubic splines based approach is explained. A separable least-squares algorithm is used as a main tool to estimate the parameters of the cubic splines. Recursive and non-recursive techniques are also proposed. Moreover, the extension of the two-step method to identify LPV models in closed-loop using neural networks is introduced. Application to the model of an arm-driven inverted pendulum with experiment data illustrates the effectiveness of the techniques presented in the chapter.

Chapter 4 begins with an overview of LPV identification techniques used to identify LPV models when the data is collected in noisy environments. A combination of the output error and instrumental variable methods is proposed for unbiased identification. The proposed method is compared with results in the literature. Finally, the extension to quasi-LPV models is also given with numerical examples.

In Chapter 5, a realization method for LPV input-output models is developed. The proposed method is based on the LTI realization method. The chapter starts with introducing algebraic tools for the algorithm. The realizations to observable and reachable forms are shown. Then, an LPV input-output model structure, that can be directly converted into an LPV state-space model in observable form, is provided.

In Chapter 6, model validation of LPV models for controller design is discussed. The application of the methods in Chapter 3 and 5 are shown. They include identification and real-time control of a magnetic levitation system, and an arm-driven inverted pendulum. The first plant is a nonlinear open-loop stable MIMO plant. The second one is a nonlinear open-loop unstable SIMO plant. Both results are compared with results in the literature.

Chapter 7 draws conclusions and gives an outlook on further research.

Chapter 2

LPV System Representations and Background Material

In this chapter, some material is presented concerning the representation, control and identification of discrete-time LPV systems, that will be used throughout the thesis. The representations of LPV systems in state-space and input-output structures are described in Section 2.1 and 2.2. In Section 2.3, a brief review of the difference between LPV and quasi-LPV systems is given. The *Multi-Layer Perceptron* (MLP) network which is necessary for the two-step closed-loop LPV identification approach is given in Section 2.4.

2.1 LPV State-Space Representation

In general, a discrete-time LPV state-space system is defined by (see e.g. [Verdult, 2002])

$$\mathcal{S}(\theta)\begin{cases} x(k+1) = A(\theta(k))x(k) + B(\theta(k))u(k) \\ \quad\;\; y(k) = C(\theta(k))x(k) + D(\theta(k))u(k) \end{cases}, \tag{2.1}$$

where $x(k) \in \mathbb{R}^{n}$ is the state variable, $u(k) \in \mathbb{R}^{n_u}$ is the input signal, $y(k) \in \mathbb{R}^{n_y}$ is the output signal, $A(\theta(k))$, $B(\theta(k))$, $C(\theta(k))$, and $D(\theta(k))$ are the parameter-varying system matrices, and $\theta \in \mathbb{R}^{n_\theta}$ is the time-dependent parameter vector. To make the notation of the LPV model more compact, the model can be represented as

$$S(\theta) = \begin{bmatrix} A(\theta(k)) & B(\theta(k)) \\ C(\theta(k)) & D(\theta(k)) \end{bmatrix} \in \mathbb{R}^{(n+n_y)\times(n+n_u)}. \tag{2.2}$$

The time-dependent parameter vector $\theta(k)$ depends on a vector of measurable scheduling signals

$$\rho(k) = \begin{bmatrix} \rho(k) & \rho_2(k) & \cdots & \rho_{n_\rho}(k) \end{bmatrix}^T \in \mathbb{R}^{n_\rho}, \tag{2.3}$$

and is generated as

$$\theta(k) = f(\rho(k)), \tag{2.4}$$

where $f(\rho(k)) : \mathbb{R}^{n_\rho} \to \mathbb{R}^{n_\theta}$ is a continuous function.

The scheduling parameter θ is taken from a compact set $\mathbb{P}_\theta \subset \mathbb{R}^{n_\theta}$ which is defined as

$$\mathbb{P}_\theta := \left\{\theta \in \mathbb{R}^{n_\theta} \middle| \, \underline{\theta}_i \leq \theta_i \leq \overline{\theta}_i, \; \forall i \in \{1, \dots, n_\theta\}\right\}, \tag{2.5}$$

where

$$\underline{\theta}_i = \min_{k \in \mathbb{Z}_+} \theta_i(k), \qquad \overline{\theta}_i = \max_{k \in \mathbb{Z}_+} \theta_i(k).$$

2.2 LPV Input-Output Representation

The LPV input-output models, that are used in the LPV identification framework, are commonly defined in a transfer operator form by separating a process and a noise part. The general LPV MIMO model is defined as

$$\mathcal{S}(\theta) \begin{cases} A(q^{-1}, \theta(k))y(k) = B(q^{-1}, \theta(k))q^{-d}u(k) \\ \qquad\qquad \tilde{y}(k) = y(k) + D^{-1}(q^{-1}, \theta(k))C(q^{-1}, \theta(k))e(k), \end{cases} \tag{2.6}$$

where $e(k)$ is a white signal, q^{-1} denotes the backward shift operator, $y(k) \in \mathbb{R}^{n_y}$ and $u(k) \in \mathbb{R}^{n_u}$ are output and input signals, d is the delay time between the input and the output signals, and

$$A(q^{-1}, \theta(k)) = I + \sum_{i=1}^{n_a} A_i(\theta(k))q^{-i}, \tag{2.7a}$$

$$B(q^{-1}, \theta(k)) = \sum_{i=0}^{n_b} B_i(\theta(k))q^{-i}, \tag{2.7b}$$

$$C(q^{-1}, \theta(k)) = I + \sum_{i=1}^{n_c} C_i(\theta(k))q^{-i}, \tag{2.7c}$$

$$D(q^{-1}, \theta(k)) = I + \sum_{i=1}^{n_d} D_i(\theta(k))q^{-i} \tag{2.7d}$$

where the coefficients $A_i(\theta(k))$, $B_i(\theta(k))$, $C_i(\theta(k))$ and $D_i(\theta(k))$ are matrices where each element is analytic function with static dependence on the scheduling parameters θ, and I is an identity matrix of appropriate dimension.

The special cases of (2.6) are defined similar to linear models case [Ljung, 1999] and are presented in Table 2.1.

It should be noted that in general $C(q^{-1}, \theta(k))$ and $D(q^{-1}, \theta(k))$ are not parameter dependent matrices. In many cases, we can define them like LTI transfer operators as $C(q^{-1})$ and $D(q^{-1})$.

The coefficients of the LPV input-output model are functions of a varying parameter θ. In [Bamieh and Giarré, 2002], it has been shown that the coefficient functions of LPV system can be parameterized as a linear combination of the basis functions

Table 2.1: Some special cased of the generalized LPV input-output model.

Name of model structure	Special defined polynomials
LPV-FIR (Finite Impulse Response)	$A(q^{-1},\theta(k)) = C(q^{-1},\theta(k)) = I$ $D(q^{-1},\theta(k)) = I$
LPV-ARX (AutoRegressive with eXternal input)	$C(q^{-1},\theta(k)) = D(q^{-1},\theta(k)) = I$
LPV-ARMA (AutoRegressive Moving Average)	$B(q^{-1},\theta(k)) = 0, D(q^{-1},\theta(k)) = I$
LPV-ARMAX (LPV-ARMA with eXternal input)	$D(q^{-1},\theta(k)) = I$
LPV-OE (Output Error)	$C(q^{-1},\theta(k)) = D(q^{-1},\theta(k)) = I$
LPV-BJ (Box-Jenkins)	None

of the scheduling parameters and the resulting model is linear-in-parameters. For example, consider the case of SISO models, the matrix coefficient functions are reduced to scalar coefficient functions. Define them as linear combinations of the polynomial of a scheduling parameter as follows,

$$a_i(q^{-1},\theta(k)) = a_{i0} + \sum_{l=1}^{n} a_{il}\theta^l(k), \tag{2.8}$$

where $a_{il} \in \mathbb{R}$. The estimation of the coefficient a_{il} can be obtained by linear regression method [Bamieh and Giarré, 2002; Wei, 2006]. As a consequence, IV method and its variations can be applied to the identification problem in a noisy environment [Laurain et al., 2010; Butcher et al., 2008].

2.3 LPV and Quasi-LPV Systems

In the literature and this thesis, an LPV system is a system, that can be viewed as an LTI system depending on the external scheduling signals, which is independent of the state of the system. A vector of external scheduling signals is denoted by $\rho_{\text{ext}} \in \mathbb{R}^{n_{\rho_{\text{ext}}}}$. The real systems, which can be included in this class are, for example, a robot manipulator or a crane system in which the models of the plants depend on the external mass load, a spark-ignited (SI) engine, which depends on the engine speed and the air mass flow [Kwiatkowski et al., 2006a] etc.

Another class of systems is referred to as a quasi-LPV system. In this case, the scheduling signals are not limited to external scheduling signals ρ_{ext} only, but include also the internal signals $\rho_{\text{int}} \in \mathbb{R}^{n_{\rho_{\text{int}}}}$ that depend on inputs, states or output of the system. The class of the quasi-LPV systems can be used to describe, of course, a wider range of nonlinear systems than LPV system. The difference between the LPV system and the quasi-LPV system is shown in Figure 2.1.

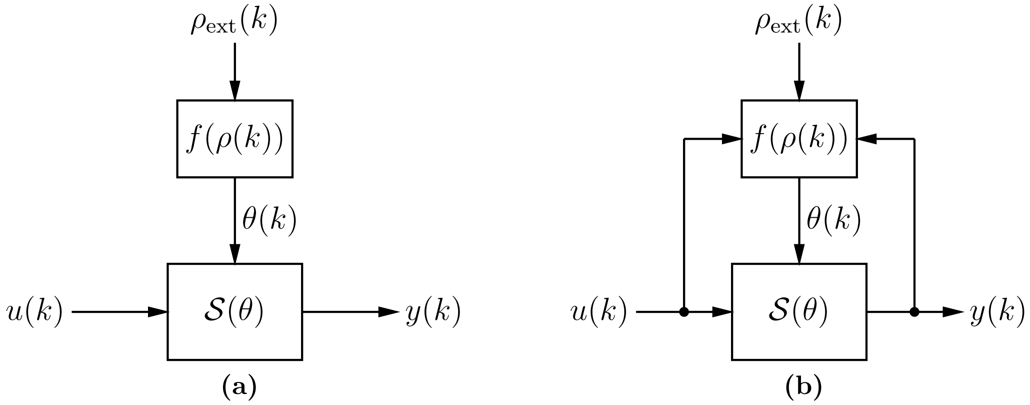


Figure 2.1: LPV system (a) and quasi-LPV system (b)

2.4 Multi-Layer Perceptron Networks

This section provides a short introduction to the MLP Network, that can be trained to approximate nonlinear functions. Throughout the Thesis, use is made of the networks just to improve the quality of the identified LPV input-output models. The networks do not identify the LPV models but the sensitivity functions which are used to generate noise free signals, (see Chapter 3). For more details of using the neural network to approximate nonlinear model, the reader is referred to [Nørgaard et al., 2003] and references therein.

The MLP network which is widely used in control is a two-layer perceptron; one hidden layer and one output layer. Each layer consist of several neurons, and one neuron represents an operation weight sum of its input signals and a bias term. The output of each neuron is fed through an activation function $\sigma(\cdot)$ to generate the output of the perceptron. The network has sigmoidal activation functions in the hidden layer, and linear activation functions in the output layer. The block diagram representation of a two-layer perceptron network is shown in Figure 2.2.

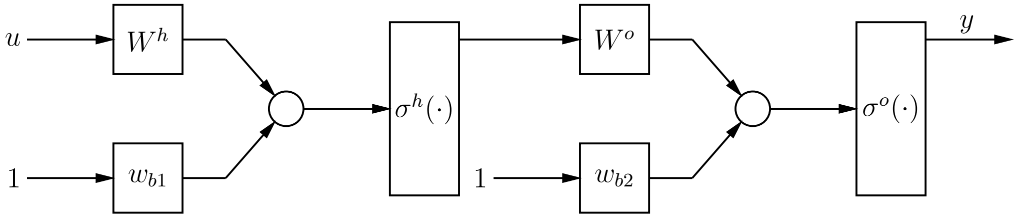


Figure 2.2: Two-layer perceptron network

In this case, the network output is

$$y = \sigma^o \left(W^o \sigma^h \left(W^h u + w_{b1} \right) + w_{b2} \right), \tag{2.9}$$

where $u \in \mathbb{R}^{n_u}$ and $y \in \mathbb{R}^{n_y}$ are the input and output vector of the network, respectively $W^o \in \mathbb{R}^{n_y \times n_u}$ and $W^h \in \mathbb{R}^{n_h \times n_u}$ include output and hidden layer weights, respectively, and n_h is the number of hidden neurons. The vector $w_{b1} \in \mathbb{R}^{n_h}$ and $w_{b2} \in \mathbb{R}^{n_y}$ are vectors containing bias weight of hidden and output layers. This two-layer perceptron network with sufficient hidden neurons can approximate any real continuous function to any desired accuracy. This property is called *universal approximation capability* [Suykens et al., 1996].

To identify nonlinear system using MLP, one can use a nonlinear regressor model

$$y(k) = g(\varphi(k), \mathfrak{p}) + e(k), \tag{2.10}$$

where $y(k)$ represents measured system outputs, the regressor vector $\varphi(k) \in \mathbb{R}^{n_r}$ contains samples of measured input and output signals, $e(k)$ is a white noise sequence, and $\mathfrak{p} \in \mathbb{R}^{n_\mathfrak{p}}$ are vectors whose elements are the weights and bias of the MLP network to be trained. In this case $g(\cdot)$ is a nonlinear function mapping the neural network inputs into outputs. The predictor form of this model is

$$\hat{y}(k|k-1) = g(\varphi(k), \mathfrak{p}). \tag{2.11}$$

Similar to the linear model, we can use structure with similar assumptions on the noise model for neural network based predictors. A neural network based ARX structure (NNARX) is shown in Figure 2.3. Other types of neural networks, such as NNARMAX, NNOE and NNBJ, can be constructed in the same way.

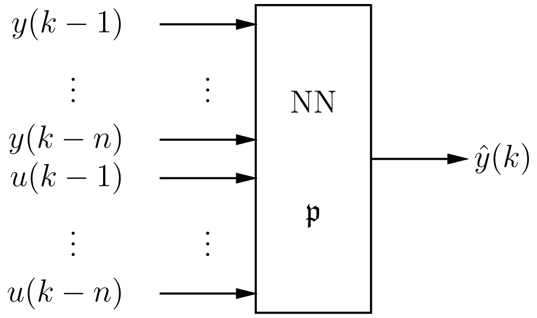


Figure 2.3: NNARX model structure

2.5 Summary

This chapter summarizes some important material that will be used in this thesis. The LPV representations of discrete-time LPV systems in various forms necessary for identification and controller have been discussed in Section 2.1 and 2.2. The difference between LPV and quasi-LPV is shown in 2.3. Finally, the multi-layer perceptron network and the neural network model structures which are used for the closed-loop LPV identification is summarized in Section 2.4.

Chapter 3

Identification of LPV Input-Output Models Using Cubic Splines

In the literature, e.g. [Belforte and Gay, 2002; Wei, 2006; Bamieh and Giarré, 2002], on identification of LPV input-output models, the functional dependence on scheduling parameters usually is approximated as a linear combination of basis functions. The most frequently used basis function is a polynomial in the scheduling variables.

Using a polynomial function for approximation is rather simple but restricted and difficult to tune. The only tuning parameter of this function is the order of the polynomial; however high order polynomials may cause oscillation of the model output. Instead of using a polynomial function, several other functions can be used. Among them, the function which is considered in this thesis is a *Cubic Spline* function. This function was used by Dempsey and Westwick [2004] and Zhu [2002] to identify a nonlinear system in block-structure form, and by Zhu and Xu [2008] as an interpolation function for several local linear models to construct a global LPV model.

There are several advantages in using the cubic spline function instead of the polynomial function. One of them is that cubic splines are defined by the position of their *knots* and react locally; moving a single knot does not effect the whole function. However, unlike polynomial basis functions, cubic spline functions are nonlinear in their parameters, therefore nonlinear optimization techniques have to be used to determine the knot positions. In this work we use a *Separable Least Squares* (SLS) algorithm proposed by Golub and Pereyra [1973] to improve the numerical condition of the nonlinear optimization by reducing the number of searching parameters in the nonlinear optimization step.

In this work we not only consider an open-loop LPV input-output identification but also identification in closed-loop. This is because there are many situations in which identification in open-loop is difficult or even impossible, for example for unstable plants and plants with integral behavior. To identify an LPV input-output model in closed-loop, a *two-step* method, proposed by Van den Hof and Schrama [1993] for LTI systems, will be extended to LPV systems. This can be done by using a neural network as a noise filter. The neural network is also used to reduce the noise disturbance correlated with the scheduling signals for the case of output-dependent LPV models (quasi-LPV models).

The proposed method is applied to two examples. The first one is a simulation result on an open-loop two-tanks system. The system is a nonlinear system identification demo example of the System Identification Toolbox of Matlab [Ljung, 2010]. This plant is a SISO open loop stable system. The second example is a real experiment with an Arm-Driven Inverted Pendulum (ADIP) [Kajiwara et al., 1999] by Quanser Consulting Inc. [Quanser Consulting Inc., 1993]. This plant is a *single-input multiple-output* (SIMO) system and open-loop unstable. In this case, a linear controller is used to stabilize the system in an initial range. Multi-sine signals are then used to excite all possible input-output levels of both outputs of the system. Applications of the proposed method and simulations for both examples are shown.

This Chapter is organized as follows. In Section 3.1, the model class considered is defined and the estimation method is presented. In Section 3.2, the structure of basis functions used in this Chapter is explained and the separable least square algorithm is discussed in Section 3.3. Section 3.4 presents the closed-loop system identification method using a two-step method, and Section 3.5 gives the validation criteria. Section 3.6 and 3.7 show the results and discussion of the simulation results of the two-tanks system and the experimental results of ADIP, respectively. Conclusions are drawn in Section 3.8. Some results in this chapter have been presented in [Boonto and Werner, 2010].

3.1 LPV-ARX Input-Output Models

In this chapter, we consider the LPV-ARX MIMO model structure with one scheduling parameter

$$A(q^{-1},\theta(k))y(k) = B(q^{-1},\theta(k))u(k-d) + e(k), \tag{3.1}$$

where q^{-1} is the backward shift operator, $\theta(k) \in \mathbb{R}$ represents a scheduling parameter, n_θ is the number of scheduling parameters, $y(k) \in \mathbb{R}^{n_y}$ and $u(k) \in \mathbb{R}^{n_u}$ are the system output and input signals at time k and n_u and n_y are number of input and output signals respectively. The polynomial matrices $A(q^{-1},\theta(k))$ and $B(q^{-1},\theta(k))$ are defined by

$$\begin{aligned} A(q^{-1},\theta(k)) &= I_{n_y} + A_1(\theta(k))q^{-1} + \cdots + A_{n_a}(\theta(k))q^{-n_a} \\ B(q^{-1},\theta(k)) &= B_1(\theta(k))q^{-1} + \cdots + B_{n_b}(\theta(k))q^{-n_b}, \end{aligned} \tag{3.2}$$

where $A_i(\theta(k)) \in \mathbb{R}^{n_y \times n_y}$ and $B_i(\theta(k)) \in \mathbb{R}^{n_y \times n_u}$ are coefficient matrices. The elements of the coefficient matrices $A_i(\theta(k))$ and $B_i(\theta(k))$ are continuous functions of the scheduling parameter θ, which can be expressed as a linear combination of a set of known fixed basis functions of scheduling parameters $\psi(\theta(k))$. These functions usually are collected in a vector form as:

$$\psi^T(\theta(k)) = \begin{bmatrix} 1 & \psi_1(\theta(k)) & \cdots & \psi_{n_p-1}(\theta(k)) \end{bmatrix}, \tag{3.3}$$

where $\psi_i(\theta(k)) : \mathbb{R} \to \mathbb{R}$, $i = 1, 2, \ldots, n_p-1$ are basis functions of the online measurable scheduling parameter $\theta(k)$, and n_p is the number of basis functions. In this thesis, we consider only the case of one scheduling parameter ($\theta(k)$ is a scalar); the extension to multiple scheduling parameters system is straightforward.

The LPV-ARX model (3.1) can be rewritten in linear regressor form. For MIMO systems, we form the predictor output as

$$\hat{y}(k) = \mathfrak{P}^T \phi(k), \tag{3.4}$$

where $\phi(k) \in \mathbb{R}^{n_p(n_a+n_b)}$ is a regressor vector given by

$$\phi(k) = \varphi(k) \otimes \psi(\theta(k)). \tag{3.5}$$

Here $\otimes$ denotes the Kronecker product[‡], and $\varphi(k) \in \mathbb{R}^{n_a+n_b}$ is defined as

$$\begin{aligned}
\varphi(k) &= \big[y^T(k-1) \ \cdots \ y^T(k-n_{a1}) \ \cdots \ y^T(k-n_{an_y}) \\
&\qquad\qquad u^T(k-d) \ \cdots \ u^T(k-d-n_{b1}) \ \cdots \ u^T(k-d-n_{bn_u})\big]^T, \\
n_a &= \sum_{i=1}^{n_y} n_{ai}, \qquad n_b = \sum_{i=1}^{n_u} n_{bi}, \\
\psi^T(\theta(k)) &= \begin{bmatrix} 1 & \psi_1(\theta(k)) & \cdots & \psi_{n_p-1}(\theta(k)) \end{bmatrix},
\end{aligned}$$

and $\mathfrak{P} \in \mathbb{R}^{n_p(n_a+n_b)\times n_y}$ is a matrix containing all coefficients to be identified as:

$$\mathfrak{P}^T = \begin{bmatrix} a_{1111} & \cdots & a_{111n_p} & \cdots & a_{1n_yn_an_p} & b_{1111} & \cdots & b_{111n_p} & \cdots & b_{1n_un_bn_p} \\ \vdots & & & & & & & & & \\ a_{n_y111} & \cdots & a_{n_y11n_p} & \cdots & a_{n_yn_yn_an_p} & b_{n_y111} & \cdots & b_{n_y11n_p} & \cdots & b_{n_yn_un_bn_p} \end{bmatrix}.$$

The subscript vector contains the output channel, the correlation output and input channel, the past output and input in a regressor vector $\varphi(k)$ and the position of an element in $\psi(\theta(k))$, respectively.

Example 3.1 [*input-output MIMO LPV*] Consider a two-input two-output system with one scheduling variable ρ of order two given as

$$\begin{aligned}
y_1(k) &= \left[a_{1111} + a_{1112}\rho(k) + a_{1113}\rho^2(k)\right] y_1(k-1) + \left[a_{1121} + a_{1122}\rho(k) + a_{1123}\rho^2(k)\right] y_1(k-2) \\
&+ \left[a_{1211} + a_{1212}\rho(k) + a_{1213}\rho^2(k)\right] y_2(k-1) + \left[b_{1111} + b_{1112}\rho(k) + b_{1113}\rho^2(k)\right] u_1(k-1) \\
&+ \left[b_{1121} + b_{1122}\rho(k) + b_{1123}\rho^2(k)\right] u_1(k-2) + \left[b_{1211} + b_{1212}\rho(k) + b_{1213}\rho^2(k)\right] u_2(k-1) \\
y_2(k) &= \left[a_{2111} + a_{2112}\rho(k) + a_{2113}\rho^2(k)\right] y_1(k-1) + \left[a_{2121} + a_{2122}\rho(k) + a_{2123}\rho^2(k)\right] y_1(k-2) \\
&+ \left[a_{2211} + a_{2212}\rho(k) + a_{2213}\rho^2(k)\right] y_2(k-1) + \left[b_{2111} + b_{2112}\rho(k) + b_{2113}\rho^2(k)\right] u_1(k-1) \\
&+ \left[b_{2121} + b_{2122}\rho(k) + b_{2123}\rho^2(k)\right] u_1(k-2) + \left[b_{2211} + b_{2212}\rho(k) + b_{2213}\rho^2(k)\right] u_2(k-1).
\end{aligned}$$

Then the regressor vector and vector of the scheduling functions are

$$\begin{aligned}
\varphi(k) &= \begin{bmatrix} y_1^T(k-1) & y_1^T(k-2) & y_2^T(k-1) & u_1^T(k-1) & u_1^T(k-2) & u_2^T(k-1) \end{bmatrix}^T, \\
\psi^T(\theta) &= \begin{bmatrix} 1 & \theta(k) & \theta^2(k) \end{bmatrix} = \begin{bmatrix} 1 & \rho(k) & \rho^2(k) \end{bmatrix}.
\end{aligned}$$

[‡] See the Appendix B.

In this case, we have

$$n_{a1} = 2, \qquad n_{a2} = 1, \qquad n_{b1} = 2, \qquad n_{b_2} = 1.$$

The problem then becomes the problem of finding the coefficient matrix which is given as

$$\mathfrak{P} = \begin{bmatrix} a_{1111} & a_{2111} \\ a_{1112} & a_{2112} \\ a_{1113} & a_{2113} \\ a_{1121} & a_{2121} \\ a_{1122} & a_{2122} \\ a_{1123} & a_{2123} \\ a_{1211} & a_{2211} \\ a_{1212} & a_{2212} \\ a_{1213} & a_{2213} \\ b_{1111} & b_{2111} \\ b_{1112} & b_{2112} \\ b_{1113} & b_{2113} \\ b_{1121} & b_{2121} \\ b_{1122} & b_{2122} \\ b_{1123} & b_{2123} \\ b_{1211} & b_{2211} \\ b_{1212} & b_{2212} \\ b_{1213} & b_{2213} \end{bmatrix}.$$

The number of coefficients that have to be determined is 36.

■

Given a data set

$$Z_N = \{y(k), u(k), \rho(k),\ k = 1, 2, \ldots, N\}$$

the *least-squares* (LS) parameter estimate [Ljung, 1999] is given by:

$$\hat{\mathfrak{P}}_N = \arg\min_{\mathfrak{P}} V_N(\mathfrak{P}), \tag{3.6}$$

where

$$V_N(\mathfrak{P}) = \frac{1}{2N} \sum_{k=1}^{N} \|y(k) - \mathfrak{P}^T \phi(k)\|_2^2, \tag{3.7}$$

and the solution of the problem is

$$\hat{\mathfrak{P}}_N = \left[\frac{1}{N} \sum_{k-1}^{N} \phi(k) \phi^T(k) \right]^{-1} \frac{1}{N} \sum_{k=1}^{N} \phi(k) y^T(k), \tag{3.8}$$

or in short matrix form:

$$\hat{\mathfrak{P}}_N = (\Phi\Phi^T)^{-1}\Phi Y^T \triangleq \Phi^{\dagger} Y^T, \tag{3.9}$$

where

$$Y = [y(1) \quad y(2) \quad \dots \quad y(N)] \in \mathbb{R}^{n_y \times N} \tag{3.10}$$

$$\Phi = [\phi(1) \quad \phi(2) \quad \dots \quad \phi(N)] \in \mathbb{R}^{n_p(n_a+n_b)\times N}, \tag{3.11}$$

and $\phi(k)$ is a column vector of the corresponding regressor and $\Phi^{\dagger} = (\Phi\Phi^T)^{-1}\Phi$. The proof of (3.8) and (3.9) can be seen in Appendix A.

Remark: The structure of the linear MIMO regressor form in (3.4) has the advantage that all output channels share the same regressor vector. Another structure that represents the measured data in regressor matrix form is also possible, see [Ljung, 1999].

3.2 Polynomial Functions and Cubic Spline Functions

In general, the exact nonlinear scheduling function of an LPV system, denoted by $g(\theta(k))$, is not known in advance. However, this function can be approximated by using a linear combination of basis functions: polynomials, cubic spline functions etc. In LPV identification [Belforte and Gay, 2002; Wei, 2006; Bamieh and Giarré, 2002], polynomials of the scheduling parameter $\theta(k)$ are used. It is well known that any continuous function on a closed interval can be approximated as closely as desired by a polynomial function if the order of the polynomial is high enough. However, polynomial functions with a given order are not easily tunable, very sensitive to data variation when the order is high, and oscillatory. Moreover, changing one coefficient of the polynomial will change the shape of the entire function.

As an alternative, one can use cubic spline functions, which have the following advantages [de Boor, 1978]:

i. They can represent nonlinear functions with significantly less oscillation than polynomials;

ii. cubic splines react locally to a parameter change;

iii. cubic splines can capture sharp corners better than polynomials.

The shape of cubic spline functions can be tuned by changing the tuning parameters called *knots* as function parameters.

A given nonlinear $\theta(k)$-dependent function can be approximated in terms of a cubic spline function depending on $\theta(k)$ and a set of knots $\{\eta_1, \eta_2, \dots, \eta_{n_\eta}\}$, where $\eta_i \in \mathbb{R}$. Order the knot positions as

$$\eta_1 < \eta_2 < \dots < \eta_{n_\eta}, \tag{3.12}$$

then the cubic spline function associated with $\theta(k)$ [Lancaster and Šalkauskas, 1986] is given by

$$g(\theta(k)) = c_0 + c_1\theta(k) + \sum_{j=1}^{n_\eta} c_{j+1}|\theta(k) - \eta_j|^3, \tag{3.13}$$

where c_i are coefficients and η_i are tuning knots. In this case, we call c_i a *linear parameter* and η_i a *nonlinear parameter.*

Given a data set generated by $g(\theta(k))$ and fixing the values of the knots beforehand, we can solve for the linear coefficients c_i by using the LS method. The following shows an example of using a cubic spline function to estimate a nonlinear function.

Example 3.2 [*Cubic Spline Function*] Consider a scheduling parameter $g(\theta(k)) = \frac{1}{1+(\theta(k))^2}$. This function is estimated by using the following cubic spline function :

$$\begin{aligned} g_{\text{sp}}(\theta(k)) = c_0 + c_1\theta(k) + c_2|\theta(k) + \eta_1|^3 + c_3|\theta(k) + \eta_2|^3 + c_4|\theta(k) + \eta_3|^3 \\ + c_5|\theta(k) + \eta_4|^3 + c_6|\theta(k) + \eta_5|^3, \end{aligned}$$

where $\eta_1 = -4$, $\eta_2 = -1$, $\eta_3 = 0$, $\eta_4 = 1$ and $\eta_5 = 4$. The coefficient parameters c_i are determined by using the linear LS method. The result is shown in Figure 3.1 and compared with a 7^{th} order polynomial in $\theta(k)$.

■

This type of cubic spline function (3.13) has been used in [Zhu, 2002] to identify a Hammerstein–Wiener model and in [Zhu and Xu, 2008] to interpolate several local linear models to construct an LPV model. Actually there are several types of spline functions that can be used for approximation; for example, cardinal natural cubic splines, B-splines etc., see [de Boor, 1978; Lancaster and Šalkauskas, 1986]. However to estimate the knot positions of these spline functions is more complicated than the cubic spline (3.13) used in this thesis.

To use the cubic spline function (3.13) as an approximated scheduling function, one can define the vector $\psi(\theta(k))$ in (3.5) as

$$\psi(\theta(k)) = \begin{bmatrix} 1 & \theta(k) & |\theta(k) - \eta_1|^3 & \cdots & |\theta(k) - \eta_{n_\eta}|^3 \end{bmatrix}. \tag{3.14}$$

However, this regressor vector contains nonlinear parameters η which are not known in advance. The problem of identification of an LPV system is changed to a nonlinear optimization problem. Instead of considering all linear and nonlinear parameters as variables that have to be determined by nonlinear optimization, we use a separable least-square method to reduce the number of nonlinear variables by separating the problem into linear and nonlinear part.

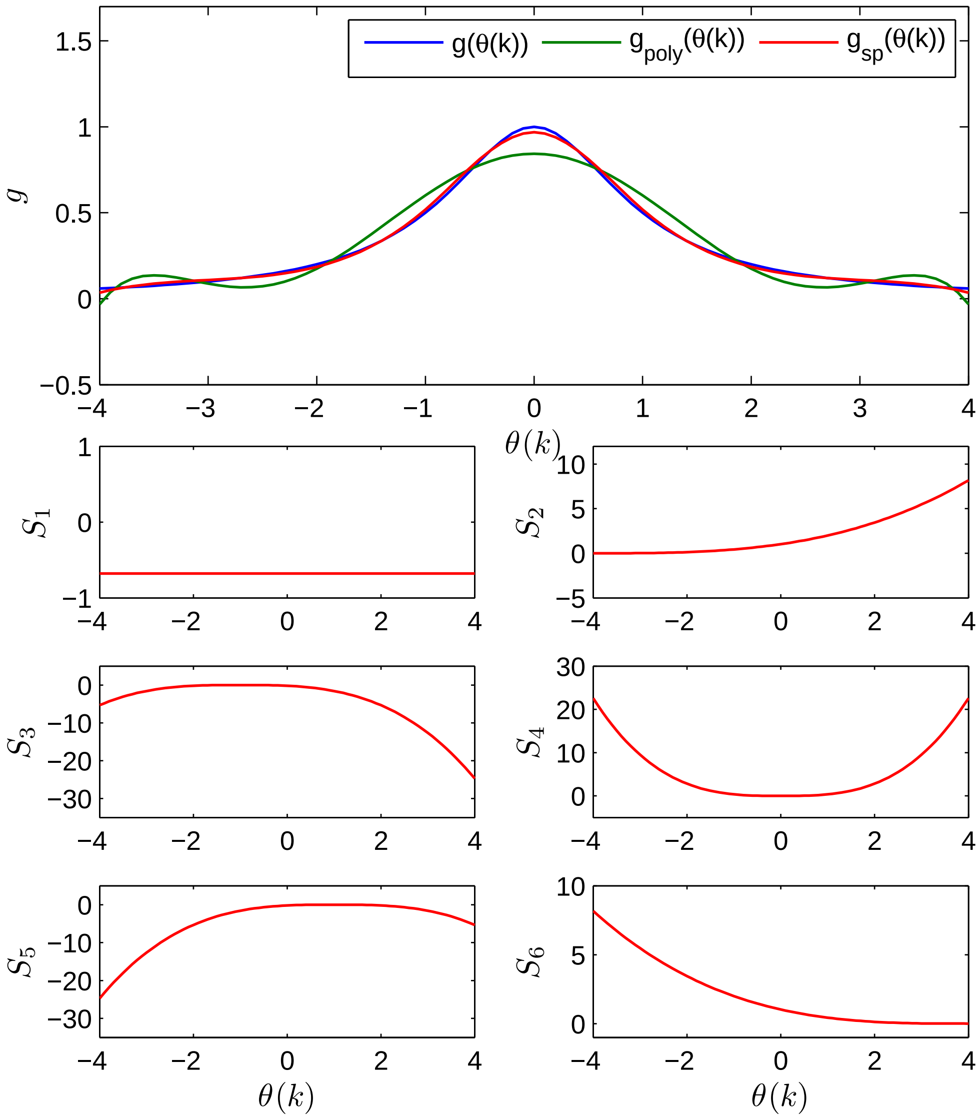


Figure 3.1: The scheduling function is estimated by using the cubic spline function $g_{\text{sp}}(\theta(k))$, and compared it with a 7th order polynomial function $g_{\text{poly}}(\theta(k))$; $S_1 = c_0 + c_1\theta(k)$, $S_2 = c_2|\theta(k) + 4|^3$, $S_3 = c_3|\theta(k) + 1|^3$, $S_4 = c_4|\theta(k)|^3$, $S_5 = c_5|\theta(k) - 1|^3$, and $S_6 = c_6|\theta(k) - 4|^3$

3.3 Separable Least-Squares Algorithm

An LPV model with a cubic spline functional dependence on scheduling parameters is represented by linear coefficients $\mathfrak{P}$ and cubic spline knot positions η. Given a set of knots, the linear coefficients can be simply calculated by the linear LS method. On the other hand, given the linear coefficients the knot values can be obtained by a nonlinear optimization. Thus the estimation problem can be cast as a Separable Least-Squares (SLS) problem [Golub and Pereyra, 1973]. The performance index we are interested in here is the sum of squared prediction errors of a MIMO system similar to (3.7). The cost function depends on both the linear parameters $\mathfrak{P}$ and nonlinear parameters η, and is defined in matrix form as

$$\tilde{V}_N(\eta, \mathfrak{P}) = \frac{1}{2N}||Y^T - \Phi^T(\eta)\mathfrak{P}||_F^2 \tag{3.15}$$

where Y is the matrix of measured output data (3.10), $\Phi(\eta)$ is a regressor matrix (3.11), $\Phi(\eta)^T\mathfrak{P}$ is a matrix of predicted outputs and $\|X\|_F$ denotes the Frobenius norm defined by $\|X\|_F = \sqrt{\text{trace}(X^TX)}$. The objective is then to find the minimizing value

$$\left[\hat{\eta}, \hat{\mathfrak{P}}\right] = \arg\min_{\eta, \mathfrak{P}} \tilde{V}_N(\eta, \mathfrak{P}). \tag{3.16}$$

For the above problem, if η is known, the corresponding $\mathfrak{P}$ that minimizes $\tilde{V}_N$ can be computed (3.8) as

$$\hat{\mathfrak{P}} = \Phi^\dagger(\eta)Y^T. \tag{3.17}$$

Substituting the solution $\hat{\mathfrak{P}}$ back into (3.15) changes the cost function to

$$\tilde{V}_N(\eta) = \frac{1}{2N}||(I - \Phi^T(\eta)\Phi^\dagger(\eta))Y^T||_F^2. \tag{3.18}$$

Following above algorithm recursively, the overall problem can be solved in two steps: the first step involves finding $\Phi^\dagger(\eta)Y^T$ with given initial value of η; the second step is finding η that minimizes (3.18). The original nonlinear problem can indeed be solved by the above two-step procedure and we have the following theorem:

Theorem 3.1 (Golub and Pereyra [1973])**.** Assume that the matrix $\Phi(\eta)$ has constant rank over an open set $\Omega \in \mathbb{R}^{n_\eta}$.

i. If $\hat{\eta} \in \Omega$ is a minimizer of $\tilde{V}(\eta)$ and $\hat{\mathfrak{P}} = \Phi(\hat{\eta})^\dagger Y^T$, then $\hat{\mathfrak{P}}, \hat{\eta}$ is also a minimizer of $\tilde{V}(\mathfrak{P}, \eta)$.

ii. If $(\hat{\eta}, \hat{\mathfrak{P}})$ is a minimizer of $\tilde{V}(\eta, \mathfrak{P})$ for $\eta \in \Omega$, then $\hat{\eta}$ is a minimizer of $\tilde{V}(\eta)$ in Ω and $\tilde{V}(\hat{\eta}) = \tilde{V}(\hat{\eta}, \hat{\mathfrak{P}})$. Furthermore, if there is a unique $\mathfrak{P}$ among the minimizing pairs of $\tilde{V}(\eta, \mathfrak{P})$, then $\mathfrak{P}$ must satisfy $\mathfrak{P} = \Phi(\eta)^\dagger Y^T$.

Proof. See [Golub and Pereyra, 1973]. □

This means that we can solve the original nonlinear problem with only n_η variables – the number of knot positions – which is much less than the original problem $n_\theta + n_\eta$, where n_θ is the number of linear coefficient parameter. The SLS algorithm used in this Chapter is based on the projection method Golub and Pereyra [1973]; Sjöberg and Viberg [1997].

3.3.1 Levenberg-Marquardt Algorithm

In this work, we use the Levenberg-Marquardt algorithm [Ljung, 1999; Nørgaard et al., 2003] to search for the nonlinear parameters η. Given a fixed value of $\hat{\mathfrak{P}}$ the cost function (3.18) can be rewritten as

$$\tilde{V}_N(\eta, \hat{\mathfrak{P}}) = \frac{1}{2N} \sum_{k=1}^{N} \varepsilon_k^T(\eta, \hat{\mathfrak{P}}) \varepsilon_k(\eta, \hat{\mathfrak{P}}) = \frac{1}{2N} E^T(\eta, \hat{\mathfrak{P}}) E(\eta, \hat{\mathfrak{P}}), \tag{3.19}$$

where $E(\eta, \hat{\mathfrak{P}}) \in \mathbb{R}^{n_y \cdot N}$ and

$$E(\eta, \hat{\mathfrak{P}}) = \begin{bmatrix} \varepsilon_1^T(\eta, \hat{\mathfrak{P}}) & \varepsilon_2^T(\eta, \hat{\mathfrak{P}}) & \cdots & \varepsilon_N^T(\eta, \hat{\mathfrak{P}}) \end{bmatrix}^T,$$

$\varepsilon_i(\eta, \hat{\mathfrak{P}}) \in \mathbb{R}^{n_y}, \ i = 1, \dots, N$.

A second order Taylor series approximation of the above cost function (3.19) is given by

$$\tilde{V}_N(\eta_{l+1} + \Delta\eta_l, \hat{\mathfrak{P}}) = \tilde{V}_N(\eta_l, \hat{\mathfrak{P}}) + \nabla \tilde{V}_N(\eta_l, \hat{\mathfrak{P}})^T \Delta\eta_l + \frac{1}{2} \Delta\eta_l^T \nabla^2 \tilde{V}_N(\eta_l, \hat{\mathfrak{P}}) \Delta\eta_l, \tag{3.20}$$

where $\nabla \tilde{V}_N(\eta_l, \hat{\mathfrak{P}}) \in \mathbb{R}^{n_\eta}$ denotes the gradient vector of $\tilde{V}_N(\eta_l, \hat{\mathfrak{P}})$, $\Delta\eta_l = \eta_{l+1} - \eta_l$ and $\nabla^2 \tilde{V}_N(\eta_l, \hat{\mathfrak{P}}) \in \mathbb{R}^{n_\eta \times n_\eta}$ denotes the Hessian matrix of $\tilde{V}_N(\eta_l, \hat{\mathfrak{P}})$.

Approximating the Hessian with the Gauss-Newton Hessian, the Levenberg-Marquardt update step is

$$\eta_{l+1} = \eta_l - (J_S^T J_S + \mu_l I)^{-1} J_S^T E(\eta_l, \hat{\mathfrak{P}}), \tag{3.21}$$

where $J_S \in \mathbb{R}^{n_\eta \times (n_y \cdot N)}$ is the Jacobian of the prediction error vector $E(\eta_l, \hat{\mathfrak{P}})$ with respect to η_l, μ_l is a time-varying parameter that is used for tuning the search direction. If μ_l is large, the search direction is closed to the *steepest descent* direction. In a different manner, if μ_l is small the search direction turns into the *Gauss-Newton* direction. The update step of the algorithm can be started by selecting a positive constant $\gamma > 1$ (e.g. $\gamma = 10$) and a small positive value μ_0 (e.g. $\mu_0 = 0.01$), and repeat the following:

- At iteration step l update η_l according to (3.21) and compute $\tilde{V}_N(\eta_{l+1})$
- If $\tilde{V}_N(\eta_{l+1}) \geq \tilde{V}_N(\eta_l)$, replace μ_l by $\gamma\mu_l$, repeat iteration step l
- If $\tilde{V}_N(\eta_{l+1}) < \tilde{V}_N(\eta_l)$, set $\mu_{l+1} = \mu/\gamma$, go to iteration step $l+1$.

Since

$$\varepsilon_i(\eta, \hat{\mathfrak{P}}) = \begin{bmatrix} y_{i,1} & \dots & y_{i,n_y} \end{bmatrix}^T - \begin{bmatrix} \hat{y}_{i,1}(\eta, \hat{\mathfrak{P}}) & \dots & \hat{y}_{i,n_y}(\eta, \hat{\mathfrak{P}}) \end{bmatrix}^T,$$

where $y_{i,j}, \ i = 1, \dots, N, j = 1, \dots, n_y$ denotes an output channel j at sample time i, $\hat{y}_{i,j}(\eta, \hat{\mathfrak{P}})$ is a predicted output in channel j at sample time i, and $J_S \in \mathbb{R}^{n_y \cdot N \times n_\eta}$ denotes the Jacobian of the prediction error vector $E(\eta, \hat{\mathfrak{P}})$ and is computed as

$$J_S^T = \begin{bmatrix} -\frac{\partial \hat{y}_{1,1}(\eta,\hat{\mathfrak{P}})}{\partial \eta_1} & \cdots & -\frac{\partial \hat{y}_{1,n_y}(\eta,\hat{\mathfrak{P}})}{\partial \eta_1} & \cdots & -\frac{\partial \hat{y}_{N,1}(\eta,\hat{\mathfrak{P}})}{\partial \eta_1} & \cdots & -\frac{\partial \hat{y}_{N,n_y}(\eta,\hat{\mathfrak{P}})}{\partial \eta_1} \\ \vdots & \ddots & \vdots & \ddots & \vdots & \ddots & \vdots \\ -\frac{\partial \hat{y}_{1,1}(\eta,\hat{\mathfrak{P}})}{\partial \eta_{n_\eta}} & \cdots & -\frac{\partial \hat{y}_{1,n_y}(\eta,\hat{\mathfrak{P}})}{\partial \eta_{n_\eta}} & \cdots & -\frac{\partial \hat{y}_{N,1}(\eta,\hat{\mathfrak{P}})}{\partial \eta_{n_\eta}} & \cdots & -\frac{\partial \hat{y}_{N,n_y}(\eta,\hat{\mathfrak{P}})}{\partial \eta_{n_\eta}} \end{bmatrix}. \tag{3.22}$$

Note that J_S also depends on the linear parameters $\mathfrak{P}$ since the optimal value of $\mathfrak{P}$ in (3.17) is changed when η is changed in the numerical derivative calculation step. Hence, the dependence of J_S on $\mathfrak{P}$ must be taken into account. This can be done by first computing the Jacobian J_L of $E(\eta, \hat{\mathfrak{P}})$ with respect only to $\mathfrak{P}$, and the Jacobian J_{NL} of $E(\eta, \hat{\mathfrak{P}})$ with respect only to the nonlinear parameters η, separately. Then J_S can be obtained [Sjöberg and Viberg, 1997; Westwick and Kearney, 2001] by

$$J_S = (I - \Xi) J_{NL} \tag{3.23}$$

where $\Xi \in \mathbb{R}^{n_y \cdot N \times n_y \cdot N}$ denotes $J_L (J_L^T J_L)^{-1} J_L^T$, which is an orthogonal projection onto the columns of the linear Jacobian J_L, where $J_L \in \mathbb{R}^{n_y \cdot N \times n_\mathfrak{p}}$ is the Jacobian of $E(\eta)$ with respect to $\mathfrak{P}$, and $n_\mathfrak{p}$ is the number of all elements of $\mathfrak{P}$.

3.3.2 Recursive Levenberg-Marquardt Algorithm

The algorithm discussed in the previous section refers to the fact that each iteration on the estimated parameters require an evaluation of the entire data set Z_N. In some situations when the data set Z_N is too large, the dimension of Ξ may exceed the capacity of 32-bit computer systems. To avoid this, one can use a recursive algorithm [Ljung and Söderström, 1983; Ngia and Sjöberg, 2000]. In this case, the cost function at time l is augmented with a exponential forgetting weight as

$$\tilde{V}_l(\eta_l, \hat{\mathfrak{P}}) = \frac{1}{2} \sum_{\tau=1}^{l} \lambda^{l-\tau} \varepsilon_l^T(\eta_l, \hat{\mathfrak{P}}) \varepsilon_l(\eta_l, \hat{\mathfrak{P}}), \tag{3.24}$$

where $0 < \lambda \leq 1$ is a *forgetting factor*, and $\varepsilon_l(\eta_l, \hat{\mathfrak{P}}) \in \mathbb{R}^{n_y}$ is the prediction error at iterative step l. The estimation update is given by

$$\eta_{l+1} = \eta_l + [\nabla^2 \tilde{V}_{l+1}(\eta_l, \hat{\mathfrak{P}})]^{-1} (\nabla \tilde{V}_l(\eta_l, \hat{\mathfrak{P}}))^T. \tag{3.25}$$

A second order approximation of the cost function (3.24) at iteration step $l+1$ is

$$\tilde{V}_{l+1}(\eta_{l+1}, \hat{\mathfrak{P}}) = \tilde{V}_l(\eta_l, \hat{\mathfrak{P}}) + \nabla \tilde{V}_l(\eta_l, \hat{\mathfrak{P}}) \Delta\eta_l + \frac{1}{2} \Delta\eta_l^T \nabla^2 \tilde{V}_l(\eta_l, \hat{\mathfrak{P}}) \Delta\eta_l,$$

where $\nabla \tilde{V}_l(\eta_l, \hat{\mathfrak{P}}) \in \mathbb{R}^{n_\eta}$ denotes the gradient vector of $\tilde{V}_l(\eta_l, \hat{\mathfrak{P}})$ and $\nabla^2 \tilde{V}_l(\eta_l, \hat{\mathfrak{P}}) \in \mathbb{R}^{n_\eta \times n_\eta}$ denotes the Hessian matrix of $\tilde{V}_l(\eta_l, \hat{\mathfrak{P}})$.

From (3.24), we have

$$\begin{aligned} \nabla \tilde{V}_l(\eta_l, \hat{\mathfrak{P}}) &= \sum_{\tau=1}^{l} \lambda^{l-\tau} \nabla(\varepsilon_l^T(\eta_l, \hat{\mathfrak{P}}) \varepsilon(\eta_l, \hat{\mathfrak{P}})) \\ &= \lambda \nabla \tilde{V}_{l-1}(\eta_l, \hat{\mathfrak{P}}) + \tilde{J}_{S,l}^T \varepsilon_l(\eta_l, \hat{\mathfrak{P}}) \end{aligned} \tag{3.26}$$

$$\nabla^2 \tilde{V}_l(\eta_l, \hat{\mathfrak{P}}) = \lambda \nabla^2 \tilde{V}_{l-1}(\eta_l, \hat{\mathfrak{P}}) + \tilde{J}_{S,l}^T \tilde{J}_{S,l} + (\nabla^2 \varepsilon(\eta_l, \hat{\mathfrak{P}})) \varepsilon_l^T(\eta_l, \hat{\mathfrak{P}}) \tag{3.27}$$

where

$$\tilde{J}_{S,l}^T = \begin{bmatrix} \frac{\partial E_{l,1}(\eta_l,\hat{\mathfrak{P}})}{\partial \eta_1} & \cdots & \frac{\partial E_{l,1}(\eta_l,\hat{\mathfrak{P}})}{\partial \eta_{n_\eta}} \\ \vdots & \ddots & \vdots \\ \frac{\partial E_{l,n_y}(\eta_l,\hat{\mathfrak{P}})}{\partial \eta_1} & \cdots & \frac{\partial E_{l,n_y}(\eta_l,\hat{\mathfrak{P}})}{\partial \eta_{n_\eta}} \end{bmatrix}. \tag{3.28}$$

Since η_l is the optimal estimate at time l, then we have $\nabla \tilde{V}_l(\eta_l, \hat{\mathfrak{P}}) = 0$. By approximating $(\nabla^2 \varepsilon_l^T(\eta_l, \hat{\mathfrak{P}}))\varepsilon_l(\eta_l) = 0$, the update of the recursive Levenberg-Marquardt algorithm is

$$\eta_{l+1} = \eta_l + [\nabla^2 \tilde{V}_{l+1}(\eta_l, \hat{\mathfrak{P}})]^{-1} \tilde{J}_{S,l}^T \varepsilon_l(\eta_l) \tag{3.29a}$$

$$\nabla^2 \tilde{V}_{l+1}(\eta_l, \hat{\mathfrak{P}}) = \lambda \nabla^2 \tilde{V}_l(\eta_l, \hat{\mathfrak{P}}) + (\tilde{J}_{S,l}^T \tilde{J}_{S,l} + \mu I), \tag{3.29b}$$

where $\tilde{J}_{S,l}^T \tilde{J}_{S,l}$ is replaced by $\tilde{J}_{S,l}^T \tilde{J}_{S,l} + \mu I$.

Similar to the off-line algorithm, $\tilde{J}_S$ depends also on $\mathfrak{P}$. For the same reason, $\tilde{J}_{S,l}$ can be determined using

$$\tilde{J}_{S,l} = (I - \tilde{\Xi}_l)\tilde{J}_{NL,l}, \tag{3.30}$$

where $\tilde{J}_{NL,l} \in \mathbb{R}^{n_y \times n_\eta}$ is the Jacobian of $E_l(\eta_l, \hat{\mathfrak{P}})$ with respect to η_l, $\tilde{\Xi}_l \in \mathbb{R}^{n_y \times n_y}$ denotes $\tilde{J}_{L,l}(\tilde{J}_{L,l}^T \tilde{J}_{L,l})^{-1} \tilde{J}_{L,l}^T$, which is an orthogonal projection onto the columns of the linear Jacobian $\tilde{J}_{L,l}$, where $\tilde{J}_{L,l} \in \mathbb{R}^{n_y \times n_{\mathfrak{p}}}$ is the Jacobian of $\varepsilon_l(\eta_l, \hat{\mathfrak{P}})$ with respect to $\mathfrak{P}$, and $n_{\mathfrak{p}}$ is the number of all elements of $\mathfrak{P}$.

Finally the Levenberg-Marquardt estimate can be updated in each sample in the same manner as in the nonrecursive algorithm. Then the recursive algorithm for a LPV cubic spline model can be summarized as Algorithm 3.1 below.

Although Algorithm 3.1 contains nonlinear optimization, it is not too sensitive to the initial values. Extensive trials show that the knot positions focused by the algorithm with random initial values, usually converge to a narrow value range. This will be illustrated by experimental results in the next section.

3.4 Closed-Loop LPV Input-Output System Identification With A Two-Step Method

In this section, we discuss a method that can be used to identify an LPV input-output model of a nonlinear unstable system which has to be stabilized in closed-loop form. The LPV model considered here is an output-dependent system. Two main issues are discussed: bias and consistency of identification. The first issue comes from the correlation between input signal and disturbance noise via the feedback loop [Van den Hof and Schrama, 1993; de Callafon, 1998]. The latter issue arises because the scheduling signal is not noise free [Butcher et al., 2008].

Algorithm 3.1 SLS algorithm for LPV-SP

```
Require: $\lambda, \gamma, R_0, \mu_0$, $\hat{\eta}_0$, ischange = 1
for $l = 1$ to $N$ do
    Use LS to find $\hat{\mathfrak{P}}_l$
    if ischange = 1 then
        Compute Jacobian $\tilde{J}_S$ using (3.30)
    end if
    Update $\nabla^2\tilde{V}_l(\eta_l, \hat{\mathfrak{P}})$ by(3.29b)
    Estimate $\hat{\eta}_{l+1}$ using (3.29a)
    Calculate $\hat{\mathfrak{P}}_{l+1}$ with $\hat{\eta}_{l+1}$
    Calculate the cost function, $\tilde{V}_l(\eta_{l+1}, \mathfrak{P}_{l+1})$
    if $\tilde{V}_l(\eta_{l+1}, \mathfrak{P}_{l+1}) < \tilde{V}_l(\eta_l, \mathfrak{P}_l)$ then
        ischange = 1
        if The change of $\tilde{V}_l(\eta_{l+1}, \mathfrak{P}_{l+1})$ is greater than a threshold then
            $\mu_{l+1} = \mu_l/\gamma$
        else
            Stop the loop and return $\hat{\mathfrak{P}}_l$ and $\hat{\eta}_l$
        end if
    else
        ischange = 0
        Restore old values of $\hat{\eta}_{l+1} = \hat{\eta}_l$, $\hat{\mathfrak{P}}_{l+1} = \hat{\mathfrak{P}}_l$ and
        $\tilde{V}_l(\eta_{l+1}, \hat{\mathfrak{P}}_{l+1}) = \tilde{V}_l(\eta_l, \hat{\mathfrak{P}}_l, )$
        $\mu_{l+1} = \gamma\mu_l$
    end if
end for
```

The closed-loop system is shown in Figure 3.2; $r_2(k)$ is an external excitation signal designed by the user, $u(k)$ and $y(k)$ are input and output signals measured in an experiment, $\mathcal{NL}$ is a nonlinear plant while K can be either an LTI or an LPV controller which stabilizes the closed-loop system. If we use a direct method to estimate the model, directly on the basis of the measured input and output data set, this may result in a biased model [Van den Hof and Schrama, 1993; de Callafon, 1998]. To avoid this bias error, we will extend the two-step method [Van den Hof and Schrama, 1993] to LPV models. From Figure 3.2, the cause for a biased model is the correlation between the inputs u_k and the disturbances v_k acting on the output, which also appears in the input via feedback.

For a linear system, we can use a linear sensitivity function to filter the noise. In this work, we propose to use a nonlinear filter, in this case a neural network [Nørgaard et al., 2003], to remove the disturbance term from the input for the first step. To do this we can use a *Neural Network AutoRegressive with eXogenous input* (NNARX) [Nørgaard et al., 2003] structure as a predictor model to predict the noise free input $u(k)$ by using $r_2(k)$ and $\tilde{u}(k)$ as inputs to the network as shown in Figure 3.3.

The consistency of LPV input-output model identification was investigated in [Butcher et al., 2008]. To get a consistent identification, the scheduling parameters have to be

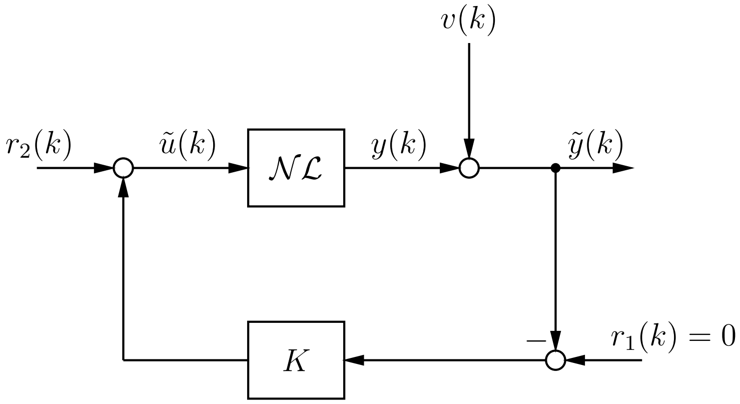


Figure 3.2: Closed-loop configuration for identification

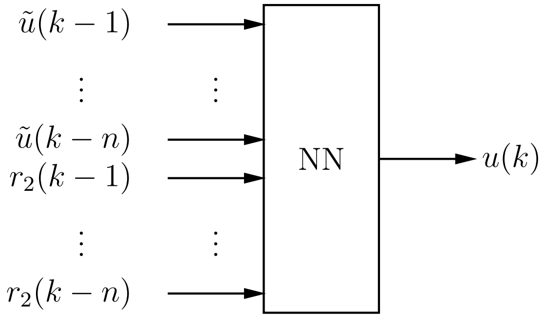


Figure 3.3: NNARX model structure for predicting noise free $u(k)$

noise-free for the case of an output dependent LPV model. In [Butcher et al., 2008] an *instrumental variable* (IV) method is proposed to solve this problem. Instead of using an IV method, we can also use a neural network to remove the disturbance noise from the output as well since the model structure is not known beforehand. As above, we can use an NNARX network to identify the whole closed-loop system by using $r_2(k)$ and $\tilde{y}(k)$ as an input signal and $y(k)$ as a predicted output signal as shown in Figure 3.4.

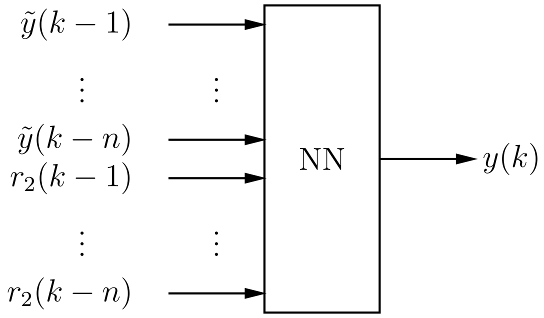


Figure 3.4: NNARX model structure for predicting noise free $y(k)$

Remark 1: We can use more complex neural network structures, for example NNARMAX, when a more flexible disturbance model is required. Moreover we can take the NNARX network as a MIMO model which can be used to predict both noise free input and noise free scheduling signals at the same time. However, the computation time will be increased.

Remark 2: Since the objective of using an NNARX network is to generate noise-free signals, we can use a high-order NNARX model. Similar to an ARX model, a high order NNARX model can approximate any type of noise model [Nørgaard et al., 2003]. However, the network should not be too complex to avoid over-fitting. This can be prevented by reducing the number of hidden neurons.

Finally we can use the direct method with the noise free data set to get the input-output quasi-LPV model. The configuration for the direct method is shown in Figure 3.5. The synthesis procedure is summarized in Algorithm 3.2.

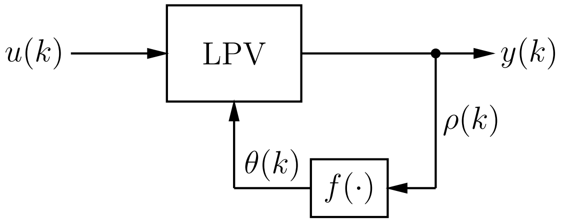


Figure 3.5: Configuration for the direct identification method.

Algorithm 3.2 Two-step method with NNARX

Step 1. Use a high order NNARX model to regenerate the noise free input $u(k)$ and the noise free scheduling signals $\theta(k)$ of the closed-loop system.

Step 2. Identify an LPV model using the data set of $y(k)$ and noise free signals $u(k)$ and $\theta(k)$.

3.5 Validation in The Prediction Error Setting

To validate the accuracy of the model, we check the difference between the simulation output $\hat{y}(k)$ of the model and the measured output $y(k)$ using a different data set. The measures used in this chapter are the following:

- **Mean squared error [Ljung, 1999]** The Mean Squared Error (MSE) is the expected value of the squared estimation error

$$\text{MSE} := \frac{1}{2N}\sum_{k=0}^{N-1}(y(k)-\hat{y}(k))^2. \tag{3.31}$$

- **Best fit percentage [Ljung, 1999]** The Best Fit (BFT) percentage is defined as

$$\text{BFT} := 100\% \cdot \max\left(1 - \frac{\|y(k) - \hat{y}(k)\|_2}{\|y(k) - \bar{y}\|_2}, 0\right), \tag{3.32}$$

  where $\bar{y}$ is the mean value of $y(k)$.

- **Variance accounted for [Verdult and Verhaegen, 2002]** The Variance Accounted For (VAF) percentage is the percentage of the output variation that is explained by the model:

$$\text{VAF} := 100\% \cdot \max\left(1 - \frac{\text{var}(y(k) - \hat{y}(k))}{\text{var}(y(k))}\right). \tag{3.33}$$

The MSE is the same as the LS criterion and a high value indicates invalidity of the model. The BFT percentage is used in the identification toolbox of Matlab [Ljung, 2010] and a low value indicates invalidity of the model. The last measure VAF gives the variation between the measured and the estimated model output, disregarding possible bias of the estimates.

3.6 Application to Open Loop Identification of A Two-Tank System

In this section, we aim to show the performance of the method described before by applying it to the identification of an open-loop nonlinear system in a quasi-LPV input-output model structure. The nonlinear system considered here is a two-tank system; the example is borrowed from the system identification toolbox of Matlab [Ljung, 2010].

3.6.1 System Setup

The two-tank system is shown in Figure 3.6. It is composed of an upper tank and a lower tank connected with an internal pipe. However for identification purpose, we assume that the dynamic model equation is not known in the first place and will be identified in LPV-ARX structure. In this example the data set is the input voltage $u(k)$ applied to the valve of the upper tank and the output level of the lower tank $x_2(k)$ collected at each sample time k. The sampling time is set to 0.2 second. To make sure that the excitation input signal is rich enough we use an *N-samples-constant* signal [Söderström and Stoica, 1989; Nørgaard et al., 2003] to excite the system for 6000 samples. In simulation, a white noise signal is added to the output to make the simulation setup more realistic. The first 3000 samples are used for identification of the system and the next 3000 samples for validation. In this case the output level $x_2(k)$ is used as a feedback scheduling signal.

To reduce the noise correlated to the output signal and the scheduling signal, we use an NNARX as a nonlinear noise filter.

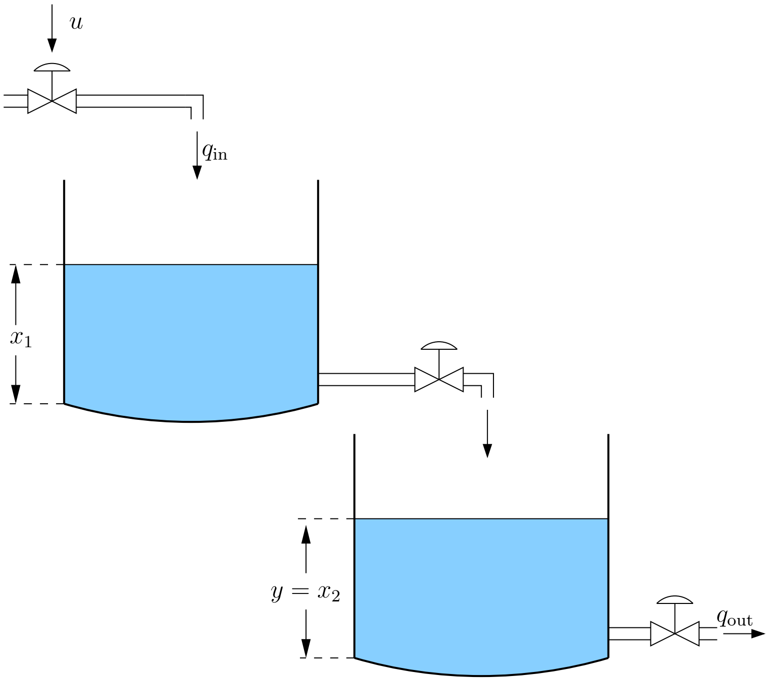


Figure 3.6: Two tanks system

3.6.2 Experimental Results

There are three model types used to identify this two-tanks system. For comparisons, a linear ARX model is identified. For LPV-ARX models, an LPV-ARX model with cubic spline basis functions and an LPV-ARX model with polynomial basis functions are used. scheduling signal of both LPV-ARX models is the previous sample of the output level of the second tank $x_2(k-1)$.

To make a fair comparison, the order of each model is chosen to get the best fit between the measured validation data and simulation output of each model. For the best ARX model $n_a = 3$, $n_b = 3$ with one step delay, while both the LPV-ARX with cubic spline (LPV-SP) and the LPV-ARX with polynomial functional dependence (LPV-Poly) have the same structure $n_a = 3$, $n_b = 3$ with two steps delay. All scheduling function structures are selected by heuristic methods. The scheduling function in vector form for the LPV-Poly is given by

$$\psi_{\text{Poly}} = \begin{bmatrix} 1 & x_2(k-1) & x^2(k-1) \end{bmatrix}$$

and for cubic splines by

$$\psi_{\text{sp}} = \begin{bmatrix} 1 & x_2(k-1) & |x_2(k-1) - \eta_1|^3 \end{bmatrix}.$$

The comparison of the simulation output (open-loop model output) and the measured output is shown in Figure 3.7. Both LPV-ARX models give much better results than the linear ARX model, especially when the output is close to the upper and lower limits of the tank levels. Moreover the LPV-SP performs better than LPV-Poly. This is because the LPV-SP can be tuned to approximate the nonlinearity of the system better than LPV-poly. In this test we also use the validation test given in Section 3.5 and the results are shown in Table 3.1. Again in all tests, the LPV-SP outperforms the other models.

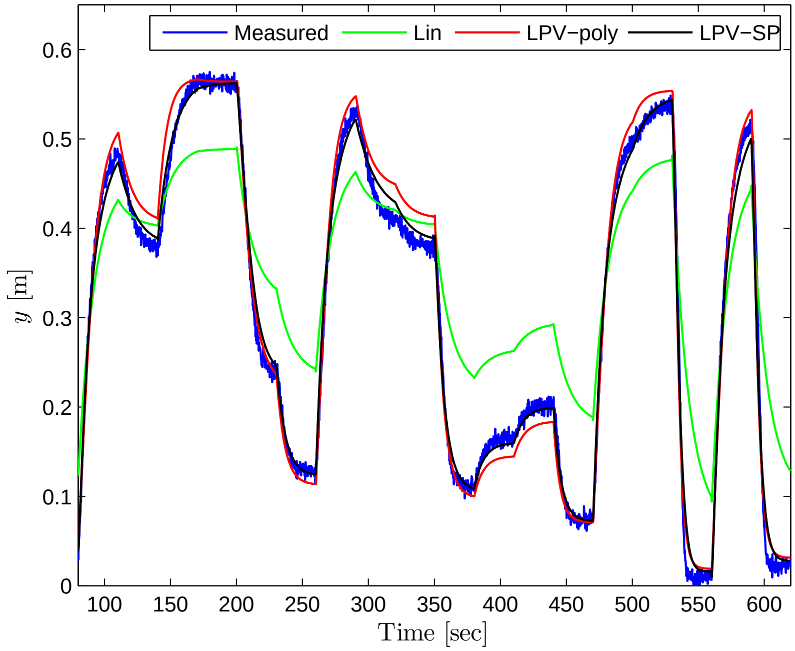


Figure 3.7: Comparison between measured data (blue) and simulation results for y of the linear model (green), LPV model with polynomial dependence (red) and LPV model with cubic spline dependence (black)

Table 3.1: Model validation

Model	MSE	% BFT	% VAF
Linear	0.00392	51.7590	79.9776
LPV-Poly	0.00023	88.2247	98.7247
LPV-SP	0.00012	91.4254	99.2666

Figure 3.8 shows the convergence of the algorithm using 100 random initial cubic spline knots between 0.2 and 0.6 to construct LPV-SP models. The accuracy of each model is measured in term of VAF. It is clear that even if the initial values of knots are different the VAF test results are close.

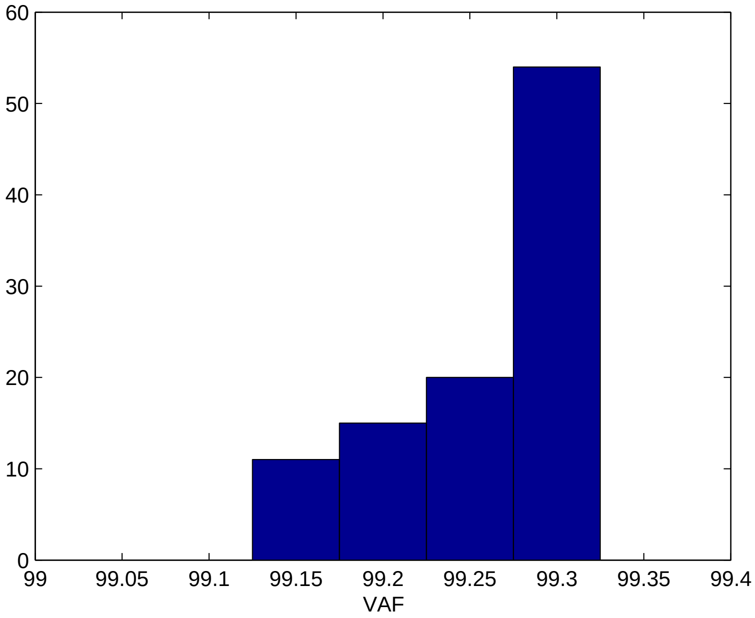


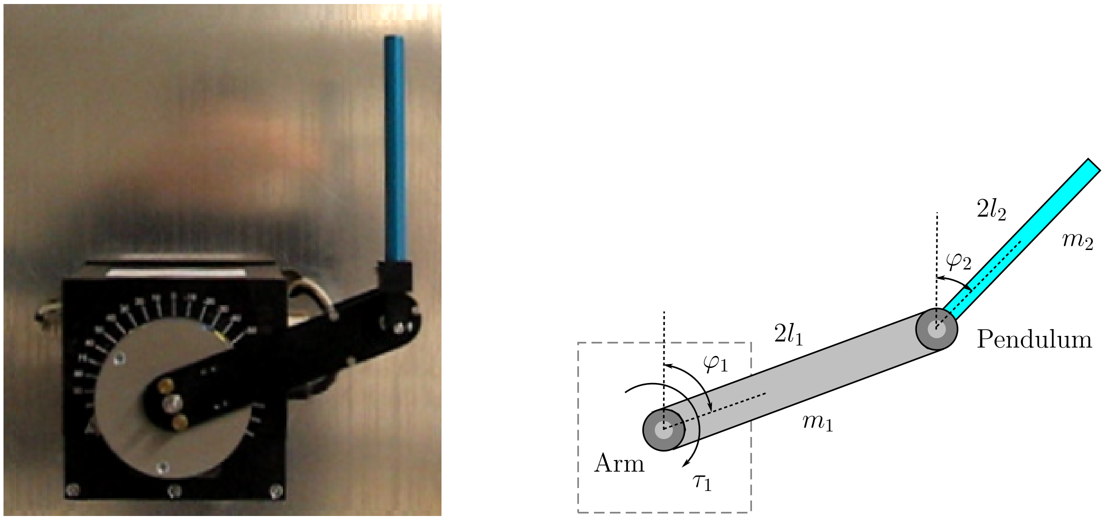
Figure 3.8: Histogram of VAF values (%) on a cross-validation output for the 100 random initial Cubic Spline knots.

3.7 Application to Closed-loop Identification of An Arm-Driven Inverted Pendulum

The second example gives an application of the proposed method to a real experimental system, which is more realistic than a simulation example. To illustrate the improvement of the MIMO LPV model with cubic splines (LPV-SP) over the polynomial model (LPV-Poly) and LTI model, experiments on an arm-driven inverted pendulum (ADIP) are carried out. This system has been used to test the performance of LPV control by [Kajiwara et al., 1999] before. In the experiments, a laboratory version of this plant manufactured by Quanser Inc. [Quanser Consulting Inc., 1993] is used. Its structure is shown in Figure 3.9. The pendulum is the top link hinged on the rotated arm – the bottom link – which is driven by a DC motor. The plant input is the DC voltage applied to the motor that drives the arm; controlled outputs are the angular position φ_1 of the arm, and the position of the pendulum φ_2 which is to be held at zero degree position. When one operates this plant in a wide range, i.e. φ_1 is large, the plant will display a stronger nonlinearity. In this situation, an LTI model is not sufficient to identify the system.

3.7.1 Experimental Setup

The closed-loop configuration is shown in Figure 3.10, where K is a stabilizing discrete-time LTI $\mathcal{H}_\infty$ controller based on a nominal model provided by Quanser Inc. [Quanser Consulting Inc., 1993], and an inner PI loop is used to control the angular velocity of



(a) Real System **(b)** Free body diagram

Figure 3.9: Arm-Driven Inverted Pendulum (ADIP)

the arm. The angular velocity $\dot{\varphi}_1$ is measured by means of a differentiator filter FD. Then the plant to be identified here includes the PI controller and the filter FD, shown as the dashed box in Figure 3.10.

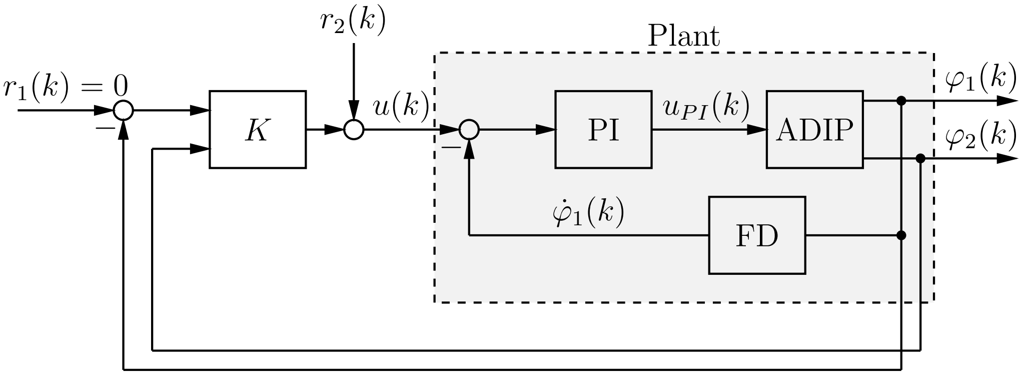


Figure 3.10: Block diagram for closed-loop identification

3.7.2 Excitation Signal Design

An important step in MIMO LPV system identification is the excitation signal design. For this purpose the persistency of excitation condition has to be satisfied in order to guarantee the consistency of the algorithm [Bamieh and Giarré, 2002; Wei, 2006]. In practice, the following multi-sine signal with adequate harmonics can be used to guarantee this condition [Ljung, 1999]

$$r_2(k) = \beta \sum_{i=1}^{n_s} \alpha_i \cos(\omega_i k T + \phi_i), \tag{3.34}$$

where the power spectrum of this multi-sine signal can be directly specified by the user through the selection of the scaling factor β, the Fourier coefficients α_i, the number of harmonics n_s, the signal length N, and the sampling time T. For selecting these parameters, more details can be found in [Ljung, 1999; Lee, 2006].

The outputs φ_1 and φ_2 have different frequency bands and amplitudes. The excitation signal should cover both bands but with different amplitudes in each frequency band. In this case, we use

$$r_2(k) = \beta_1 \sum_{i=1}^{n_s} \alpha_i \cos(\omega_i kT + \phi_i) + \beta_2 \sum_{j=1}^{N/2} \alpha_j \cos(\omega_j kT + \phi_i). \tag{3.35}$$

Following the guidelines in [Lee, 2006], α is chosen as

$$\alpha_g = \begin{cases} 1, & g = 1, \dots, n_s \\ h_f, & g = n_s + 1, \dots, N/2 \end{cases} \tag{3.36}$$

where g is either i or j in (3.35), and h_f is the high frequency coefficient magnitude.

According to the power spectrum of both outputs of the ADIP in closed-loop shown in Figure 3.11, the multi-sine input is selected to cover two frequency ranges. The first part is between 0.063 rad/sec and 3.77 rad/sec, and the second part is between 3.77 rad/sec and 37 rad/sec. Each frequency range contains 40 harmonics to guarantee the richness of the excitation signal. Moreover the amplitude of the second part is 10 times smaller than the first part since this part is aimed to excite the angle φ_2, which is more sensitive than φ_1, and that angle should not move far away from the zero degree vertical position.

3.7.3 Experimental Results

The ADIP is excited by a multi-sine input signal with 2 periods and 24,000 samples, with a sampling time of 10 ms, as described in the previous section. The first half of the data set is used for identification and the second half is for validation. For the LPV-SP model, linear LS is used in the linear step, and the recursive Levenberg-Marquardt algorithm is used in the nonlinear step of the SLS algorithm. In this thesis we use a trial-and-error method to select the structure and the order of the model for all cases; a more systematic method has been developed in [Tóth et al., 2009a]. The chosen linear model for our sampled data set is $n_a = \left[\begin{smallmatrix} 2 & 5 \\ 5 & 2 \end{smallmatrix}\right]$, $n_b = 3$ and one step delay, while both LPV-poly and LPV-SP have the same structure $n_a = \left[\begin{smallmatrix} 2 & 5 \\ 5 & 2 \end{smallmatrix}\right]$, $n_b = 5$ and 2 steps delay.

The scheduling signal is a previous sample of the angular position φ_1, which is the main source of the nonlinearity of this plant. The scheduling function structure, obtained heuristically, for the polynomial case is given in vector form by

$$\psi_{\text{Poly}} = \begin{bmatrix} 1 & \varphi_1(k-1) & \varphi_1^2(k-1) & \varphi_1^3(k-1) \end{bmatrix}$$

and for cubic spline by

$$\psi_{\text{sp}} = \begin{bmatrix} 1 & \varphi_1(k-1) & |\varphi_1(k-1) - \eta_1|^3 & |\varphi_1(k-1) - \eta_2|^3 \end{bmatrix}.$$

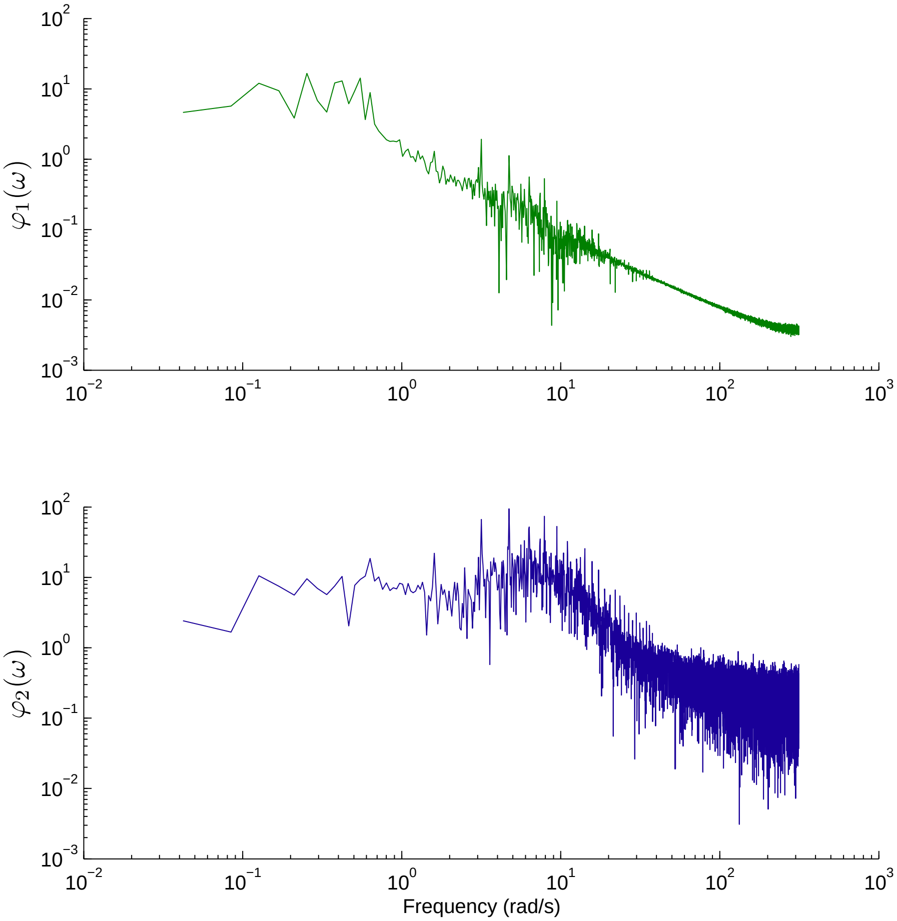


Figure 3.11: Spectrum of both normalized outputs of ADIP: upper for φ_1, lower for φ_2.

For both cases, one step delay of the scheduling variable is necessary to avoid an algebraic loop. The nonlinear part of the spline function contains only two terms to reduce the dimension of the initial value space. The search of all model parameters is initialized randomly, and the best model is taken as solution. The value of λ in the Levenberg-Marquardt algorithm step is 0.999, and the initial value of $\eta \in [0.01 \quad 0.9]$.

Figures 3.12 and 3.13 show the comparison of the validation data and simulated (open-loop) output of all models. For the output φ_1, both LPV models outperform the linear model while the output φ_2 is difficult to judge from the plot. The LPV-SP model is slightly better than LPV-Poly over the whole range of φ_2 – this is because the cubic spline function has more freedom to fit the nonlinear scheduling function. The price to pay for this improvement is the computation time in the nonlinear step of the SLS algorithm – the increase in computation time depends on the number of nonlinear parameters. The validation results are shown in Table 3.2. Table 3.2 confirms again that an LPV-SP model gives better results for the output φ_1 in all tests, while the output φ_2 is nearly the same as for the LPV-Poly model.

Figures 3.14 and 3.15 show the difference between the identified models using a NNARX network as a noise filter to reproduce a noise free scheduling signal and the one

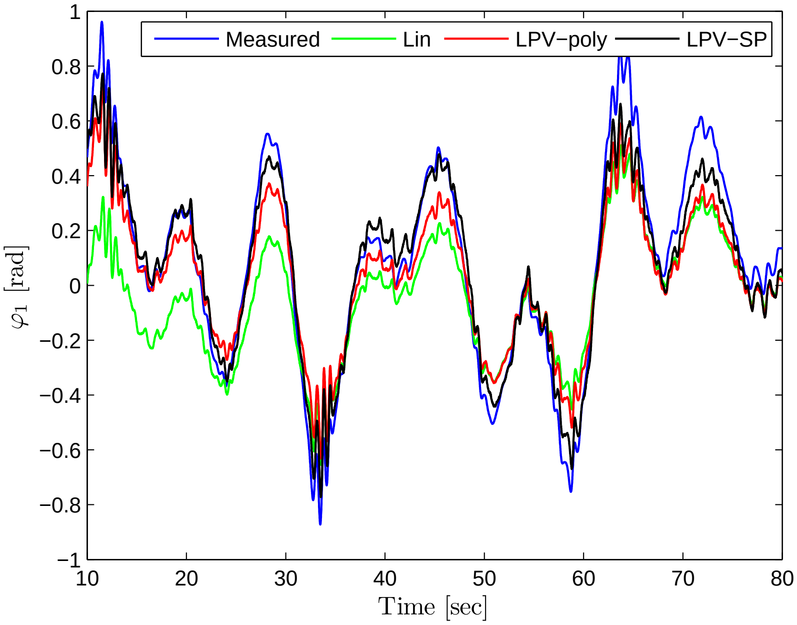


Figure 3.12: Comparison between measured data (blue) and simulation results for φ_1 of the linear model (green), LPV model with polynomial dependence (red) and LPV model with cubic spline dependence (black)

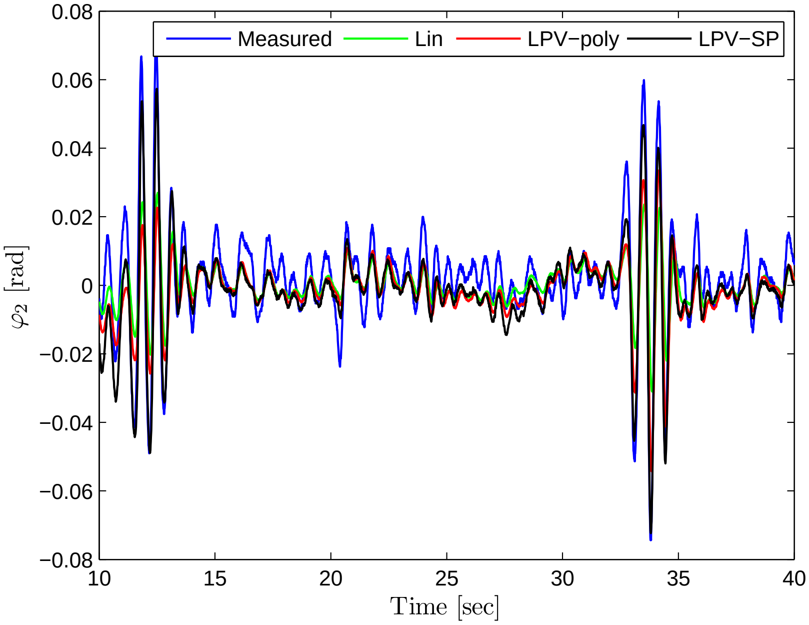


Figure 3.13: Comparison between measured data (blue) and simulation results for φ_2 of the linear model (green), LPV model with polynomial dependence (red) and LPV model with cubic spline dependence (black)

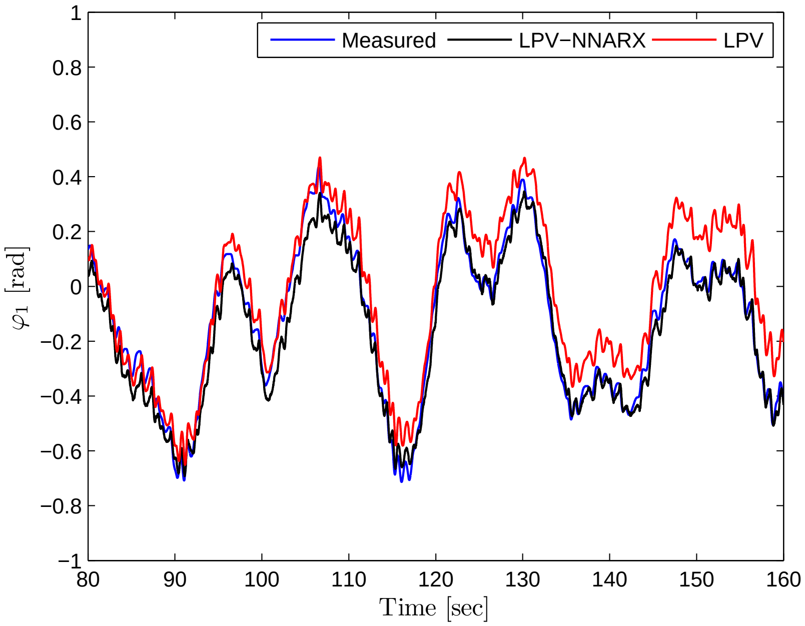


Figure 3.14: Comparison between measured data (blue) and simulation results for φ_1 of an LPV model with cubic spline dependence with NNARX (black) to generate a noise free scheduling signal and without NNARX (red) .

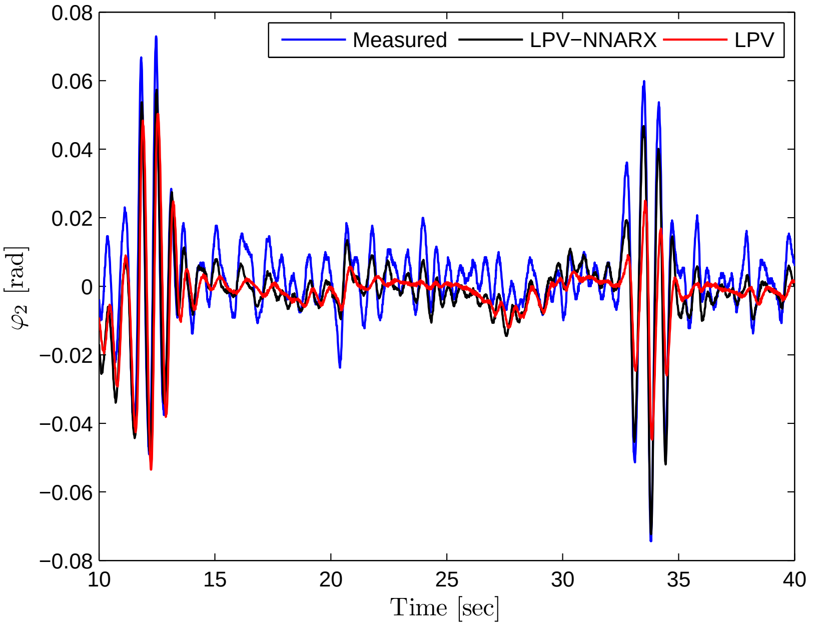


Figure 3.15: Comparison between measured data (blue) and simulation results for φ_2 of an LPV model with cubic spline dependence with NNARX (black) to generate a noise free scheduling signal and without NNARX (red).

Table 3.2: Model validation

Output	Model	MSE	% BFT	% VAF
φ_1	Linear	0.0416	40.5387	65.2372
	LPV-Poly	0.0184	60.5190	84.4756
	LPV-SP	0.0066	76.3063	94.5185
φ_2	Linear	6.1780×10^{-5}	25.3413	47.6410
	LPV-Poly	4.3792×10^{-5}	37.1582	68.5193
	LPV-SP	4.4426×10^{-5}	36.6924	70.3316

using raw data. For output φ_1 considered as a high *Signal to Noise ratio* (SRN) signal, the LPV model constructed by using raw data has less accuracy than the LPV model constructed by using filtered data (LPV-NNARX) in terms of bias drift. However, for output φ_2, since the correlation of the measured noise signal with the former is quite high, using raw data the LPV model cannot capture the dynamics of the real system.

Finally, to see the convergence of the algorithm for the MIMO system, again we use 100 random initial values of $\eta_1 \in [-1\ ,\ 0]$ and $\eta_2 \in [0\ ,\ 1]$ and compare the simulation output of the model with the measured data in terms of VAF. Figures 3.16 and 3.17 show the histogram of VAF. Since the results of VAF are quite close to each other the x-axis of each plot shows only a small range of VAF values of the results. From both plots in 3.16 and 3.17, one can see that the VAF tests converges to a narrow range.

3.8 Conclusions

In this chapter we have shown that LPV models with cubic spline dependence on scheduling parameters can be identified using a separable least squares method in closed loop. The identification of quasi-LPV input-output models is carried out using a two-step approach with an NNARX network as a noise filter. This model class has advantages over LPV models with polynomial dependence, such as the tunability via the knot positions. Simulation and experimental results show successful applications of this approach to the simulation of a two-tank model and to a real ADIP system, where the proposed method outperforms LTI as well as LPV-Poly models. The convergence of the algorithm is tested by Monte-Carlo tests with random initial conditions.

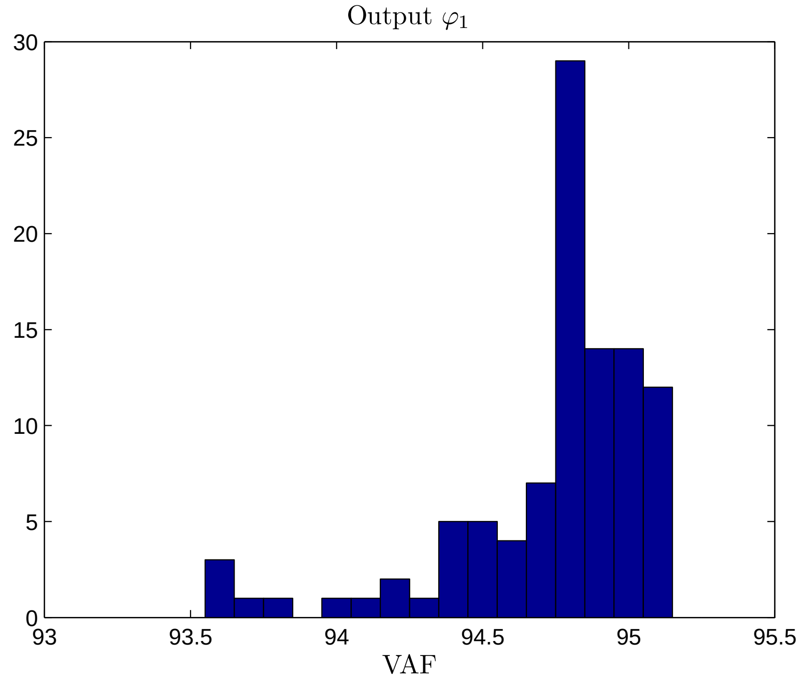


Figure 3.16: Histogram of VAF values (%) on a cross-validation output φ_1 for the 100 random initial Cubic Spline knots.

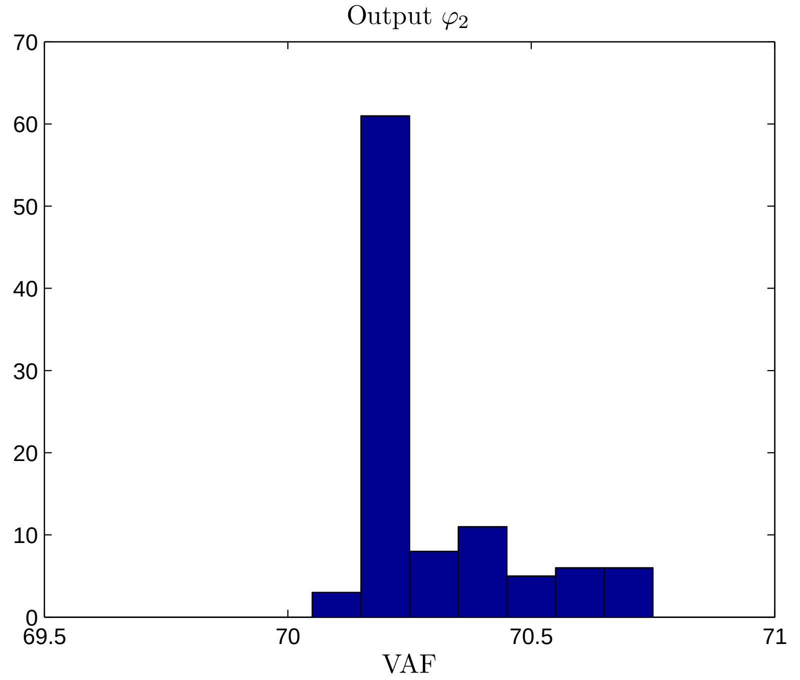


Figure 3.17: Histogram of VAF values (%) on a cross-validation output φ_2 for the 100 random initial Cubic Spline knots.

Chapter 4

Unbiased Identification of LPV Input-Output Models

In Chapter 3, the bias error due to measurement noise in closed-loop identification is reduced by using NNARX as a noise filter. It is well known that NNARX is suitable only when the noise is white [Janczak, 2005]. This assumption is, in general, not realistic. Of practical relevance when the measured data is corrupted with colored noise are NNARMAX, NNOE and NNBF, however those require much more computation time due to the recurrent network structure [Nørgaard et al., 2003]. To train such networks for LPV models may take hours or even days depending on the length of the data set and the order of the identified models. Instead of using a neural network, less computing power is required by methods like *Instrumental Variable* (IV) and its variation.

The bias error of LPV identification was first considered by [Butcher et al., 2008] and has received more attention during the last years. In Butcher et al. [2008], the IV method with LPV-ARX auxiliary model denoted by IV-ARX was introduced to solve the biased problem when scheduling signals are noise free and output signals of the system are corrupted with white noise. The main drawback of this method is that the variance of the estimated parameters is very high when the noise is colored [Laurain et al., 2010]. Then the method is not a good choice to use for practical problems since many experiments have to be conducted to get unbiased estimated parameters. The high variance problem has been solved by Laurain et al. [2010] by using a *Refined Instrumental Variable* (RIV) algorithm [Söderström and Stoica, 1983; Laurain et al., 2010] and has also been extended to LPV-BJ models by assuming that a noise model structure is known. A simplified version of this algorithm is also introduced and known as *Simplified Refined Instrumental Variable* (SRIV) algorithm used in case the noise model is neglected [Laurain et al., 2010]. However these methods require iterative algorithms and more computation time. The *Linear Recurrent Neural Network* (LRNN) introduced by Abbas et al. [2010a] is also a powerful technique since the estimated results have low bias and acceptable low variance. However, the computation time to train the recurrent network is high even for a low-order model. Moreover, when the order of the estimated model has to be changed, which is normal in identification procedures, the structure of the network has to be redesigned by the user, which is not trivial.

Here we improve the IV-ARX method which is used by [Butcher et al., 2008]. The IV-ARX method is started with identifying an auxiliary model using LPV-ARX structure and estimating the parameters using a LS method. This auxiliary model is used to generate an estimated output signal which is assumed to be noise free. However the model estimated by the LPV-ARX model is always biased [Söderström and Stoica, 1983; Landau and Zito, 2006] due to the incorrect noise model. Instead of using LPV-ARX, the LPV-OE model with *Output Error* (OE) method [Landau et al., 1998; Landau and Zito, 2006] is introduced here. The LPV-OE model is used as the first step of the IV to obtain a regressor vector which is correlated to the noise free output signal but not to the noise signals. This method is referred to as IV-OE. Since the method uses a predicted output signal to form a regressor vector instead of using a measured output signal, the output signal obtained from this method is always uncorrelated with measured noise. By using IV-OE, we can preserve the two simple steps of the IV-ARX method but significantly improve the performance in both bias and variance terms. To illustrate the improvement of the proposed method, simulation results of the estimated model when the systems are corrupted by white noise and colored noise are given.

Finally, we show that the introduced method not only works for LPV systems but can also be applied to quasi-LPV systems. The identification of quasi-LPV systems is normally considered as a more difficult problem than that of normal LPV systems. This is because in this problem noise not only corrupts the output signal but also the scheduling signal via the measured output. Again, since the predicted output by the LPV-OE model only depends on measured input signals which are uncorrelated with noise, we can use the identified model to generate a noise free predicted output signal for both output regressor vector and a sequence of the scheduling signal. This leads to an unbiased identification when we use the IV-OE estimation.

For comparison purposes, in this chapter we consider only the identification of LPV and quasi-LPV models in the SISO case. The extension to the MIMO case can be done in the same way as in Chapter 3.

This chapter is organized as follow. The preliminaries of the method are presented in terms of model structure, and algorithm. Then the proposed method is compared with other methods using Monte-Carlo simulation examples in both LPV and quasi-LPV model structure. Some results in this chapter have been presented in [Boonto and Werner, 2008].

4.1 System Description

Consider the SISO LPV system shown in Figure 4.1, with input signal $u(k) \in \mathbb{R}$, scheduling input signal $\rho(k) \in \mathbb{R}$, scheduling parameter $\theta(k) \in \mathbb{R}$ and measured output $y(k) \in \mathbb{R}$, which is corrupted by noise while $\tilde{y}(k)$ is a noise free signal. The signal $v(k) \in \mathbb{R}$ represents all external factors like disturbances or measurement noise that corrupt the output signal.

Assume that all LPV systems considered here can be represented by the structure

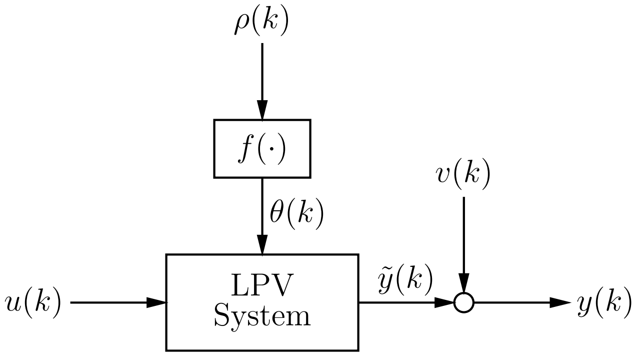


Figure 4.1: LPV model for system identification

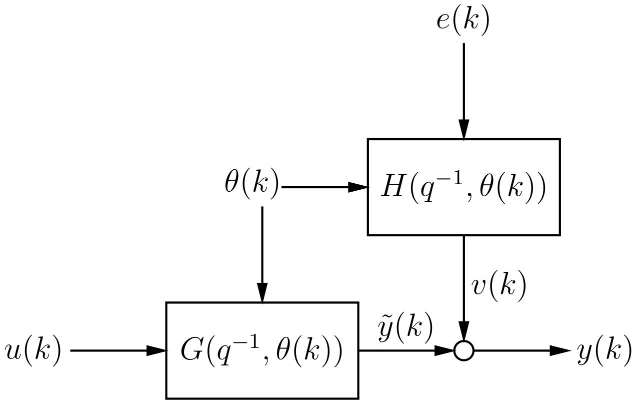


Figure 4.2: LPV model structure

of a discrete-time LPV model shown in Figure 4.2. For SISO LPV models, the output signal is given by

$$y(k) = G(q^{-1},\theta(k))u(k-d) + H(q^{-1},\theta(k))e(k), \tag{4.1}$$

where $\theta(k) \in \mathbb{R}$ is the measurable scheduling parameter at time k, d is the delay between input and output signal, $u(k) \in \mathbb{R}$ is the input signal, $y(k) \in \mathbb{R}$ is the output signal, $e(k)$ is a white noise signal and q is the time-shift operator, i.e. $q^{-i}u(k) = u(k-i)$. $G(q^{-1},\theta(k))$ and $H(q^{-1},\theta(k))$ are called the transfer operators [Ljung, 1999] which are the ratio of two polynomials in the operator q. For LPV systems the coefficients of the polynomial depend on a scheduling parameter $\theta(k)$[†] while for LTI systems these parameters are constant. With this parameter dependence, the multiplication of the θ-dependent polynomials with the shift operator q is not commutative over the θ-dependent coefficients, i.e., $q^{-1}A(q^{-1},\theta(k))y(k) = A(q^{-1},\theta_{k-1})q^{-1}y(k)$ which is not equal to $A(q^{-1},\theta(k))q^{-1}y(k)$ like in LTI systems. Moreover, the inverse of the θ-dependent polynomials of LPV systems are defined as a left-inverse[†] and denoted by

---

[†] For simplicity, we consider only systems that depend on a single scheduling parameter. The extension to multiple scheduling parameters is straightforward, see e.g. [Abbas et al., 2010b].

[†] This will be explained in details in Chapter 5.

$(\cdot)^{-1}$, e.g.,

$$\frac{B(q^{-1},\theta(k))}{A(q^{-1},\theta(k))} = A^{-1}(q^{-1},\theta(k))B(q^{-1},\theta(k)),$$

which is in general not equal to $B(q^{-1},\theta(k))A^{-1}(q^{-1},\theta(k))$.

Then, $G(q^{-1},\theta(k))$ in (4.1) is defined as:

$$G(q^{-1},\theta(k)) = A^{-1}(q^{-1},\theta(k))B(q^{-1},\theta(k)), \tag{4.2a}$$

$$A(q^{-1},\theta(k)) = 1 + \sum_{i=1}^{n_a} a_i(\theta(k))q^{-i}, \tag{4.2b}$$

$$B(q^{-1},\theta(k)) = \sum_{i=0}^{n_b} b_j(\theta(k))q^{-i}, \tag{4.2c}$$

where the coefficients a_i and b_j are real meromorphic functions[‡] with static dependence on $\theta(k)$ – the functions depend on θ only at time k – and $H(q^{-1},\theta(k))$ is defined as:

$$H(q^{-1},\theta(k)) = D^{-1}(q^{-1},\theta(k))C(q^{-1},\theta(k))), \tag{4.3a}$$

$$C(q^{-1},\theta(k)) = 1 + \sum_{i=1}^{n_c} c_i(\theta(k))q^{-i}, \tag{4.3b}$$

$$D(q^{-1},\theta(k)) = 1 + \sum_{i=1}^{n_d} d_i(\theta(k))q^{-i}, \tag{4.3c}$$

which represents a filter that is used to generate a model of the disturbance $v(k)$. The noise models considered here do not always depend on $\theta(k)$. Generally only LPV-ARX and LPV-ARMAX require the noise model in LPV form, i.e., $H(q^{-1},\theta(k)) = A^{-1}(q^{-1},\theta(k))$ and $H(q^{-1},\theta(k)) = A^{-1}(q^{-1},\theta(k))C(q^{-1},\theta(k))$ respectively. Other models such as, LPV-OE and LPV-BJ, can be considered to have noise models in LTI form.

As in Chapter 3, we assume that the functions $a_i(\theta(k))$, $b_i(\theta(k))$, $c_i(\theta(k))$ and $d_i(\theta(k))$ are linear combinations of a set of known fixed basis function; for example a polynomial basis function, cubic spline function etc. In this chapter, we focus only on polynomial basis function for simplicity. Hence, the polynomial coefficient functions are defined as

$$\begin{aligned} a_i(\theta(k)) &= a_{i0} + a_{i1}\theta(k) + \cdots + a_{in_p}(\theta(k))^{n_p-1}, \\ b_i(\theta(k)) &= b_{i0} + b_{i1}\theta(k) + \cdots + b_{in_p}(\theta(k))^{n_p-1}, \\ c_i(\theta(k)) &= c_{i0} + c_{i1}\theta(k) + \cdots + c_{in_p}(\theta(k))^{n_p-1}, \\ d_i(\theta(k)) &= d_{i0} + d_{i1}\theta(k) + \cdots + d_{in_p}(\theta(k))^{n_p-1}, \end{aligned} \tag{4.4}$$

where n_p is the number of monomial terms of the basis function.

The true model structure can be written in linear regression form as

$$y(k) = \varphi^T(k)\mathfrak{p}_0 + v(k), \tag{4.5}$$

[‡] $f : \mathbb{R}^n \mapsto \mathbb{R}$ is a real meromorphic function if $f = g/h$ with g, h analytic and $h \neq 0$.

where the θ-dependent regressor vector is given by

$$\phi(k) = \varphi(k) \otimes \psi(\theta(k)) \tag{4.6}$$

where $\otimes$ is the Kronecker product[†] and

$$\varphi(k) = [y(k-1)\ y(k-2)\ \dots\ y(k-n_a)\ u(k-d)\ u(k-d-1)\ \dots\ u(k-d-n_b)]\,,$$
$$\psi(\theta(k)) = \begin{bmatrix}1 & \theta(k) & \theta^2(k) & \dots & \theta^{n_p-1}(k)\end{bmatrix}$$

and the true parameter vector is

$$\mathfrak{p}_0^T = \begin{bmatrix}a_{0,1,0}\ a_{0,1,1}\ \dots\ a_{0,1,n_\rho}\ a_{0,n_a,1}\ \dots\ a_{0,n_a,n_\rho}\ b_{0,0,0}\ b_{0,0,1}\ \dots b_{0,0,n_\rho}\ b_{0,n_b,1}\ \dots\ b_{0,n_b,n_\rho}\end{bmatrix}.$$

4.2 Model Structure

The LPV model structures which are considered here are LPV-ARX and LPV-OE. LPV-ARX model parameters are estimated by using the *Prediction Error Method* (PEM) as in [Butcher et al., 2008; Bamieh and Giarré, 2002] and LPV-OE model parameters are estimated by using the OE method see [Landau et al., 1998; Landau and Zito, 2006; Boonto and Werner, 2008]. Both methods have been successfully applied to LTI systems; however the extension of these methods to the LPV case is more involved.

4.2.1 LPV-ARX Representation

The LPV-ARX model structure is written as

$$A(q^{-1}, \theta(k))y(k) = B(q^{-1}, \theta(k))u(k-d) + e(k), \tag{4.7}$$

with noise model

$$H(q^{-1}, \theta(k)) = A^{-1}(q^{-1}, \theta(k)). \tag{4.8}$$

Define $A^*(q^{-1}, \theta(k)) = (1 - A(q^{-1}, \theta(k)))$, then it is possible to show that the *one-step ahead predictor* [Tóth, 2008] of $y(k)$ is

$$\hat{y}(k) = -A^*(q^{-1}, \theta(k))y(k) + B(q^{-1}, \theta(k))u(k-d), \tag{4.9}$$

or

$$\hat{y}(k) = \phi^T \hat{\mathfrak{p}}, \tag{4.10}$$

where $\tilde{\phi}$ is a θ-dependent regressor vector containing past measured output and past measured input data as

$$\phi = \varphi(k) \otimes \psi(\theta(k)),$$

where

$$\varphi(k) = [y(k-1)\ y(k-2)\ \dots\ y(k-n_a)\ u(k-d)\ u(k-d-1)\ \dots\ u(k-d-n_b)]\,,$$
$$\psi(\theta(k)) = \begin{bmatrix}1 & \theta(k) & \theta^2(k) & \dots & \theta^{n_p-1}(k)\end{bmatrix}$$

and $\hat{\mathfrak{p}}$ is the estimated parameters vector.

[†] See Appendix B.

4.2.2 LPV-OE Representation

In LPV-OE model, the model structure is written as

$$y(k) = A^{-1}(q^{-1}, \theta(k))B(q^{-1}, \theta(k))u(k) + e(k), \tag{4.11}$$

with the noise model

$$H(q^{-1}, \theta(k)) = 1, \tag{4.12}$$

and the one-step ahead predictor of $y(k)$ is

$$\hat{y}(k) = -A^{*}(q^{-1}, \theta(k))^{*}\hat{y}(k) + B(q^{-1}, \theta(k))u(k-d), \tag{4.13}$$

or

$$\hat{y}(k) = \tilde{\phi}^{T}\hat{\mathfrak{p}}, \tag{4.14}$$

where $\tilde{\phi}$ is a θ-dependent regressor vector containing past predicted output and past measured input data as

$$\tilde{\phi} = \tilde{\varphi}(k) \otimes \psi(\theta(k)),$$

where

$$\tilde{\varphi}(k) = \left[\hat{y}(k-1)\ \hat{y}(k-2)\ \ldots\ \hat{y}(k-n_a)\ u(k-d)\ u(k-d-1)\ \ldots\ u(k-d-n_b)\right],$$
$$\psi(\theta(k)) = \begin{bmatrix} 1 & \theta(k) & \theta^{2}(k) & \ldots & \theta^{n_p-1}(k) \end{bmatrix}$$

and $\hat{\mathfrak{p}}$ is the estimated parameter vector. The main difference between LPV-ARX and LPV-OE models is that the latter regressor vector contains the past predictor output $\hat{y}(k)$ instead of the past measured output $y(k)$. Since the regressor vector of the LPV-OE model also depends on the previous estimated parameters and the estimated parameter vector $\hat{\mathfrak{p}}$, an iterative search for the parameter estimate $\hat{\mathfrak{p}}$ is required. This algorithm will be explained in Section 4.4.1.

4.3 Bias Error of Least Squares Method

Assume the data set $Z_N = \{y(k), u(k), \theta(k), k = 1, 2, \ldots, N\}$ generated by the system and a model structure $\mathcal{M}(\mathfrak{p})$ are given. For the SISO case, the problem is the estimation of parameter $\hat{\mathfrak{p}}$ based on the minimization of the mean square error

$$V_N(Z_N, \mathfrak{p}) = \frac{1}{2N}\sum_{k=1}^{N}\varepsilon^{2}(k) = \frac{1}{2N}\sum_{k=1}^{N}(y(k) - \hat{y}(k))^{2}, \tag{4.15}$$

such that the estimated parameter is

$$\hat{\mathfrak{p}}_N = \arg\ \min_{\mathfrak{p}} V_N(Z_N, \mathfrak{p}). \tag{4.16}$$

This problem can be solved by using the LS method for batch data, given by

$$\hat{\mathfrak{p}}_N^{\mathrm{LS}} = \left[\frac{1}{N} \sum_{k=1}^{N} \phi(k)\phi^T(k) \right]^{-1} \frac{1}{N} \sum_{k=1}^{N} \phi(k)y(k). \tag{4.17}$$

Substituting $y(k)$ in the above equation from (4.5), we have

$$\hat{\mathfrak{p}}_N^{\mathrm{LS}} = \mathfrak{p}_0 + \left[\frac{1}{N} \sum_{k=1}^{N} \phi(k)\phi^T(k) \right]^{-1} \frac{1}{N} \sum_{k=1}^{N} \phi(k)v(k). \tag{4.18}$$

For the estimate $\hat{\mathfrak{p}}_N^{\mathrm{LS}}$ to be consistent, it is well known that it is necessary that:

i. $\lim_{N\to\infty} \left[\frac{1}{N} \sum_{k=1}^{N} \phi(k)\phi^T(k) \right]^{-1}$ is nonsingular,

ii. $\lim_{N\to\infty} \frac{1}{N} \sum_{k=1}^{N} \phi(k)v(k) = 0.$

The second term of (4.18) will be zero only when $v(k)$ is a zero mean white noise sequence and the elements in $\phi(k)$ are not correlated with $v(k)$. When this term is not identical zero, it is called *bias error* [Ljung, 1999]. It is clear that the estimated parameters of LPV-ARX models obtained by the LS method are always giving biased results because the noise model is assumed to be $A^{-1}(q^{-1}, \theta(k))$.

In the literature, the bias error is considered in terms of its 2-norm defined by

$$\mathrm{Bias} = \|\mathfrak{p}_0 - \mathbf{E}\{\hat{\mathfrak{p}}\}\|_2, \tag{4.19}$$

where $\mathbf{E}$ is the expectation operator, approximated via Monte-Carlo simulations. This term indicates the difference of the mean of the estimated parameters and the true parameters. Its variance is defined by

$$\mathrm{Variance} = \|\mathbf{E}\{(\hat{\mathfrak{p}} - \mathbf{E}\{\hat{\mathfrak{p}}\})^2\|_2. \tag{4.20}$$

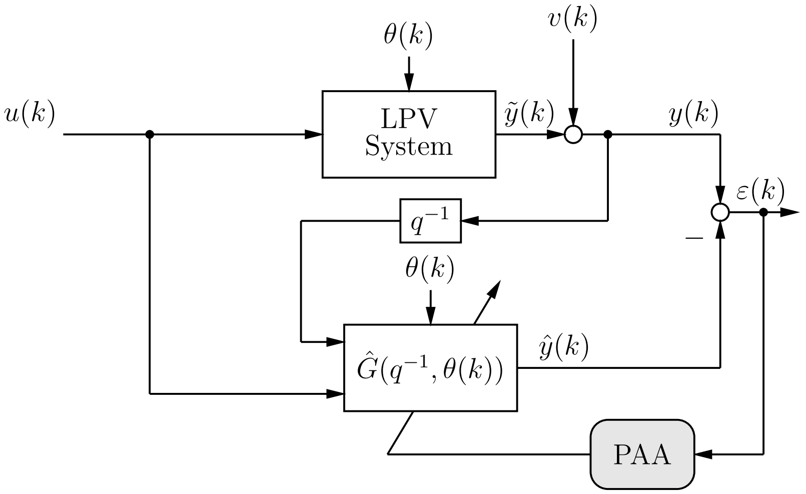
A good estimation algorithm must have both a low bias error and a low variance.

4.4 Unbiased Identification Methods

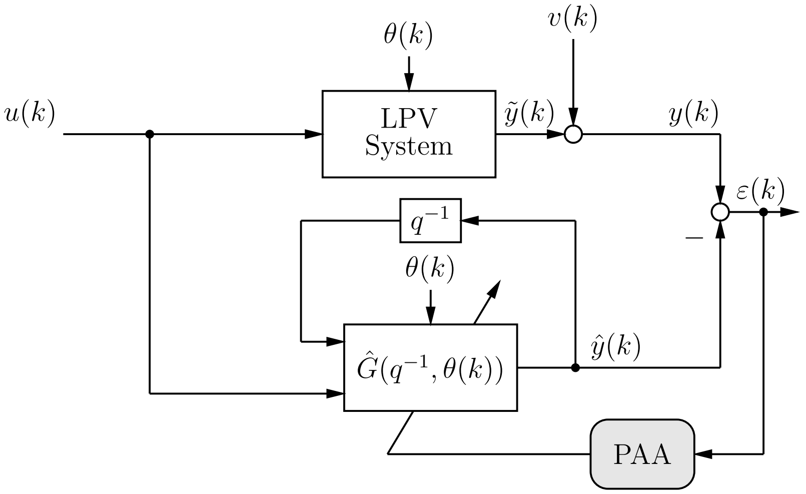
In this Chapter we consider two methods: an output error method and instrumental variable method [Landau and Zito, 2006; Ljung, 1999].

4.4.1 Output Error Method

The output error method discussed here is based on a *Recursive PseudoLinear Regressions* (RPLR) method for OE models proposed by Landau [1976], see also [Landau et al., 1998; Landau and Zito, 2006]. The basic idea of this method is to use a prediction $\hat{y}(k)$ instead of a measured $y(k)$ in the regressor vector. If past predicted output $\hat{y}(k)$ is employed in the regressor and estimated equation, the current predicted output will only depend on $u(k)$, which is independent of the disturbance signal $v(k)$. With a suitable *Parameter Adaptation Algorithm* (PAA) an unbiased estimated plant model will be obtained. To extend this idea to LPV systems, one can adapt the regressor vector as a function of the scheduling parameters, see [Boonto and Werner, 2008]. The difference between this method and a general *Recursive Least Squares* (RLS) algorithm can be seen in Figure 4.3.



(a) Recursive least squares method (RLS)



(b) Output error method (OE)

Figure 4.3: Comparison between the recursive least squares and the output error method.

4.4.2 Output Error Method for LPV Input-Output Model Identification

Consider the true model structure in (4.5) and define the adjustable predictor output as:

$$\hat{y}^o(k) = \tilde{\phi}^T(k-1)\hat{\mathfrak{p}}(k-1) \qquad \textit{a priori}, \tag{4.21}$$

$$\hat{y}(k) = \tilde{\phi}^T(k-1)\hat{\mathfrak{p}}(k) \qquad \textit{a posteriori}, \tag{4.22}$$

where

$$\begin{aligned}\tilde{\phi}(k) &= \tilde{\varphi}(k) \otimes \psi(\theta(k)),\\ \tilde{\varphi}^T(k) &= \left[\hat{y}(k-1)\ \hat{y}(k-2)\ \ldots\ \hat{y}(k-n_a)\ u(k-d)\ u(k-d-1)\ \ldots\ u(k-d-n_b)\right],\\ \psi^T(\theta(k)) &= \begin{bmatrix}1 & \theta(k) & \theta^2(k) & \ldots & \theta^{n_p-1}(k)\end{bmatrix},\end{aligned}$$

where $\hat{y}^o(k)$ and $\hat{y}(k)$ represent the *a priori* and the *a posteriori* output of the predictor, respectively, $u(k)$ is the input signal and $\hat{\mathfrak{p}}(k)$ is the estimated parameter at sample time k.

The prediction error is given by

$$\varepsilon^o(k) = y(k) - \hat{y}^o(k) \qquad \textit{a priori}, \tag{4.23}$$

$$\varepsilon(k) = y(k) - \hat{y}(k) \qquad \textit{a posteriori}. \tag{4.24}$$

In order to estimate the parameters of the system, the PAA for LPV input-output models is given as follows [Landau et al., 1998; Landau and Zito, 2006]:

$$\hat{\mathfrak{p}}(k) = \hat{\mathfrak{p}}(k-1) + \varepsilon(k)\tilde{\phi}^T(k)F(k), \tag{4.25a}$$

$$F^{-1}(k-1) = \lambda_1 F^{-1}(k) + \lambda_2\tilde{\phi}(k)\tilde{\phi}^T(k), \tag{4.25b}$$

$$0 < \lambda_1 \leq 1,\ 0 \leq \lambda_2 < 2,$$

$$F(0) > 0, F^{-1}(k) > \alpha F^{-1}(0), 0 < \alpha < \infty,$$

$$F(k+1) = \frac{1}{\lambda_1}\left[F(k) - \frac{F(k)\tilde{\phi}(k)\tilde{\phi}^T(k)F(k)}{\frac{\lambda_1}{\lambda_2} + \tilde{\phi}^T(k)F(k)\tilde{\phi}(k)}\right], \tag{4.25c}$$

$$\varepsilon(k) = \frac{\varepsilon^o(k)}{1 + \tilde{\phi}^T(k)F(k)\tilde{\phi}(k)}, \tag{4.25d}$$

where $F(0)$ is the initial value of the adaptation matrix gain, normally given by a nonsingular constant matrix, and λ_1 and λ_2 are constant factors of the adaptation gain $F(k)$. These three parameters are considered tuning parameters of PAA.

The usage of $\hat{y}(k)$ instead of $y(k)$ has a clear benefit in the presence of disturbances. With an asymptotically decreasing adaptation gain, $\hat{y}(k)$ will only depends on $u(k)$ and this will lead to $\lim_{N\to\infty}\frac{1}{N}\sum_{k=0}^{N}\tilde{\phi}(k)v(k) = 0$ asymptotically. As a result a plant model with unbiased parameter estimation will be achieved.

A sufficient (but not necessary) condition for the convergence of the algorithm for LTI systems [Landau et al., 1998] is that

$$\frac{1}{A(z^{-1})} - \frac{\lambda_2}{2} \tag{4.26}$$

is a *strictly positive real* transfer function. The strictly positive real concept can be extended to LPV system by assuming that any LPV system $G(q^{-1}, \theta(k))$ has the following properties:

i. $G(q^{-1}, \theta(k))$ is globally asymptotically stable for any trajectory of scheduling parameter $\theta \in \mathbb{P}_\theta$

ii. the real part of $G(q^{-1}, \theta)$ is positive at all frequencies for all values $\theta \in \mathbb{P}_\theta$.

Then the sufficient convergence condition of the PAA can be extended to LPV systems by replacing $A(z^{-1})$ with $A(z^{-1}, \theta)$, where $A(z^{-1}, \theta)$ is a frequency response function of $A(q^{-1}, \theta)$ at each freezing parameter θ. To relax this condition, the *extended output error method* can be used instead see [Landau et al., 1998]. However, since we use this algorithm in an off-line manner, instead of checking the infinite dimensional sufficient condition, one can simply check the convergence of the algorithm by plotting the estimated parameters. If the parameters do not converge, three tuning parameters $F(0)$, λ_1 and λ_2 can be adapted. A selection guideline for these parameter can be seen in [Landau and Zito, 2006].

4.4.3 Instrumental Variable Method

The basic idea behind the instrumental variable method and its variations is to generate a new regressor vector which is highly correlated with uncorrupted measured data but uncorrelated with the noise disturbance [Söderström and Stoica, 1983; Ljung, 1999]. The IV estimator equation for LPV systems is given by

$$\hat{\mathfrak{p}}_N^{\mathrm{IV}} = \left[\frac{1}{N} \sum_{k=1}^{N} \zeta(k) \phi^T(k) \right]^{-1} \frac{1}{N} \sum_{k=1}^{N} \zeta(k) y(k). \tag{4.27}$$

where $\zeta(k)$ is the *instrumental variable* regressor vector which is generated from an auxiliary LPV model. Actually there are several ways to select the instrument variable, see [Söderström and Stoica, 1983], but in this thesis, we concentrate on a IV with auxiliary model [Söderström and Stoica, 1983; Landau and Zito, 2006; Butcher et al., 2008].

Instrumental Variable With An Auxiliary Model Method

One instrumental variable technique is based on an auxiliary model to generate the IV regressor. This can be done by choosing an instrumental variable generator equal to the true system, which is clearly not known. However, we can let the instrumental variable depend on the parameters in an explicit way.

In [Butcher et al., 2008], the IV regressor vector $\zeta(k)$ which is generated by the LPV-ARX model was used and denoted as IV-ARX. The algorithm gives reasonable results when the measurement output is corrupted with white noise. For colored noise, for example with LPV-BJ as data generating system, the signal generated by an LPV-ARX estimated model is still highly correlated with noise due to the limitation of the noise model structure of LPV-ARX models. Moreover it has been shown in [Laurain et al., 2010] that for LPV systems, it is not possible to use a LS algorithm to filter the measured signals when it is corrupted by colored noise. This is because the noncommutativity of shift operator q^{-1} in the polynomials of LPV models; there is no way to write the predicted output of LPV-BJ in a least square problem format [Laurain et al., 2010].

To retain the attractive simplicity of the above IV-ARX methods, we introduce the IV method with an auxiliary model estimated with the OE method, denoted as IV-OE. The motivation to use the OE method is simple. For a white noise corrupted system, we know that the OE method is an unbiased model estimation algorithm. The OE method alone is not satisfactory when the system is dealing with colored noise. However the method is designed such that whether or not it converges to the true parameters value, it does not depend on the colored noise model [Moore, 1982]. By this observation, the auxiliary model estimated by the OE method should give a predicted output $\hat{y}(k)$ which is free of colored noise and better than using a LS technique. The algorithm of IV-OE is summarized as follows:

Algorithm 4.1 IV-OE Algorithm

Step 1 Estimate an OE model by the OE method using the PAA step in (4.25a)-(4.25d).

Step 2 Generate an estimate of $\hat{y}(k)$ based on the result of the previous step. Build an instrument regressor vector using $\hat{y}(k)$ and $u(k)$ and then estimate $\hat{\mathfrak{p}}$ using the IV method.

4.5 Simulation Examples of LPV Systems

In this section we show simulation results with the proposed method. The simulations are carried out by generating data sets from the true LPV model and then comparing the true parameters with the estimated parameters. The comparisons of each method are given in term of the bias 2-norm (4.19) and the variance (4.20) of the estimated parameters. We separate the test into two categories: LPV model corrupted by white noise and LPV model corrupted by colored noise. We also give a comparison with the method of Butcher et al. [2008] and with the methods of Abbas et al. [2010a] and Laurain et al. [2010].

Model Structures

For comparison purposes, the LS method, the OE method and the IV-ARX method used in Butcher et al. [2008], are compared to the proposed IV-OE method. The

IV-ARX, IV-OE and LS have the following LPV-ARX model structure:

$$\mathcal{M}_{\mathfrak{p}}^{\text{LPV-ARX}} \begin{cases} A(q^{-1},\theta(k)) = 1 + a_1(\theta(k))q^{-1} + a_2(\theta(k))q^{-2} \\ B(q^{-1},\theta(k)) = b_0(\theta(k))q^{-1} + b_1(\theta(k))q^{-2} \\ H(q^{-1},\theta(k)) = A^{-1}(q^{-1},\theta(k)) \end{cases}, \tag{4.28}$$

where

$$a_1(\theta(k)) = a_{1,0} + a_{1,1}\theta(k) + a_{1,2}\theta^2(k), \tag{4.29a}$$
$$a_2(\theta(k)) = a_{2,0} + a_{2,1}\theta(k) + a_{2,2}\theta^2(k), \tag{4.29b}$$
$$b_0(\theta(k)) = b_{0,0} + b_{0,1}\theta(k) + b_{0,2}\theta^2(k), \tag{4.29c}$$
$$b_1(\theta(k)) = b_{1,0} + b_{1,1}\theta(k) + b_{1,2}\theta^2(k). \tag{4.29d}$$

This is in contrast with the OE method, as in the latter the regressor vector is constructed by an estimated noise free output (predicted output) data set. Then, the model of the OE method has the following LPV-OE model structure:

$$\mathcal{M}_{\mathfrak{p}}^{\text{LPV-OE}} \begin{cases} A(q^{-1},\theta(k)) &= 1 + a_1(\theta(k))q^{-1} + a_2(\theta(k))q^{-2} \\ B(q^{-1},\theta(k)) &= b_0(\theta(k))q^{-1} + b_1(\theta(k))q^{-2} \\ H(q^{-1},\theta(k)) &= 1 \end{cases} \tag{4.30}$$

These model structures will be used for all tests.

Remark: In the case of IV-OE, the auxiliary model uses the LPV-OE model structure while the IV step uses the LPV-ARX structure.

4.5.1 White Noise Case

Data Generating System

In the first example, we consider a white noise disturbance $H(q^{-1},\theta(k)) = 1$. This example is borrowed from [Butcher et al., 2008]. The true system is given by

$$\mathcal{S}_0 \begin{cases} A(q^{-1},\theta(k)) &= 1 + a_1(\theta(k))q^{-1} + a_2(\theta(k))q^{-2} \\ B(q^{-1},\theta(k)) &= b_0(\theta(k))q^{-1} + b_1(\theta(k))q^{-2} \\ H(q^{-1}) &= 1. \end{cases} \tag{4.31}$$

The coefficients dependence on the scheduling parameter θ is chosen as follows

$$a_1(\theta(k)) = 1 - 0.5\theta(k) + 0.2\theta^2(k), \tag{4.32a}$$
$$a_2(\theta(k)) = 0.5 - 0.7\theta(k) - 0.1\theta^2(k), \tag{4.32b}$$
$$b_0(\theta(k)) = 0.5 - 0.4\theta(k) + 0.01\theta^2(k), \tag{4.32c}$$
$$b_1(\theta(k)) = 0.2 - 0.3\theta(k) - 0.02\theta^2(k). \tag{4.32d}$$

The scheduling parameter $\theta(k)$ is a periodic function of time as $\theta(k) = 0.5\sin(0.35\pi k) + 0.5$. The input $u(k)$ is taken as white noise with uniform distribution in the interval

[-1,1], and with length $N = 5000$ to generate the data set Z_N of the system. The noise on the output is taken such that $v(k)$ is zero-mean, normally distributed white noise with different power. The experiments are done by using Monte-Carlo simulation of 100 runs for three different *Signal to Noise Ratio* (SNR), i.e. 5dB, 10dB and 15dB, defined as

$$\text{SNR} = 10 \log \left(\frac{P_y}{P_e} \right), \tag{4.33}$$

where P_e is the average power of the zero mean additive on the system output and P_y represents the average power of the noise-free output signal.

In Table 4.1, the comparison of all methods in terms of the bias error and the variance is displayed. The IV-OE presents the best performance when the SNR are 15dB and 10dB while at 5dB the bias error is equivalent to IV-ARX but the variance is a little bit lower.

Table 4.1: Estimation bias 2-norm and its variance at different SNR : LPV with white measurement noise

Method		15dB	10dB	5dB
LS	Bias	1.2212	1.2495	1.2229
	Variance	0.0004	0.0007	0.0009
OE	Bias	0.0118	0.0456	0.2200
	Variance	0.0007	0.0023	0.0067
IV-ARX	Bias	0.0233	0.0294	0.0318
	Variance	0.0008	0.0034	0.0104
IV-OE	Bias	0.0129	0.0180	0.0351
	Variance	0.0007	0.0017	0.0062

Table 4.2 and 4.3 show the comparison result of each method in terms of mean and standard variation of the estimated parameters at 15dB SNR. It can be concluded that the LS method gives absolute mean estimated parameters far away from the true values, especially those of the coefficients in $A(q^{-1}, \theta(k))$. This is not surprising since the coefficients in $A(q^{-1}, \theta(k))$ are estimated by using the contaminated output data. In contrast with OE, IV-ARX and IV-OE methods, they estimate parameters very close to their true values.

In conclusion, if the output signal is corrupted by white noise and the scheduling signal is free of noise, the unbiased error method for the LPV system identification can be selected as follows:

i. if the SNR is not too low, e.g., SNR > 10dB, the OE method should be selected since the estimation contains only one step.

ii. if the SNR is lower than 10dB either IV-ARX or IV-OE have to be used to get an unbiased result.

Table 4.2: Mean and standard deviation (Std) of the estimated $A(q^{-1}, \theta(k))$ polynomial parameters at SNR 15 dB : LPV with white measurement noise

		$a_{1,0}$	$a_{1,1}$	$a_{1,2}$	$a_{2,1}$	$a_{2,2}$	$a_{2,3}$
Method	True Value	1	-0.5	0.2	0.5	-0.7	-0.1
LS	Mean	0.7330	-1.1504	0.9307	0.3758	-1.1785	0.3020
	Std	0.0199	0.1064	0.1014	0.0119	0.0769	0.0774
OE	Mean	0.9957	-0.5047	0.2078	0.4976	-0.6997	-0.0991
	Std	0.0241	0.1387	0.1547	0.0127	0.1075	0.1400
IV-ARX	Mean	0.9985	-0.5101	0.2129	0.4995	-0.7108	-0.0879
	Std	0.0224	0.1462	0.1422	0.0094	0.1013	0.1158
IV-OE	Mean	1.0010	-0.5042	0.1968	0.4998	-0.6955	-0.1096
	Std	0.0165	0.1204	0.1288	0.0077	0.0878	0.1118

Table 4.3: Mean and standard deviation (Std) of the estimated $B(q^{-1}, \theta(k))$ polynomial parameters at SNR 15 dB : LPV with white measurement noise

		$b_{1,0}$	$b_{1,1}$	$b_{1,2}$	$b_{2,1}$	$b_{2,2}$	$b_{2,3}$
Method	True Value	0.5	-0.4	0.01	0.2	-0.3	-0.02
LS	Mean	0.4993	-0.3960	0.0059	0.0577	-0.3546	0.1645
	Std	0.0199	0.1064	0.1014	0.0119	0.0769	0.0774
OE	Mean	0.4998	-0.4014	0.0115	0.1982	-0.3030	-0.0161
	Std	0.0041	0.0239	0.0235	0.0133	0.0526	0.0490
IV-ARX	Mean	0.4997	-0.3988	0.0084	0.1986	-0.2984	-0.0202
	Std	0.0042	0.0213	0.0206	0.0131	0.0545	0.0487
IV-OE	Mean	0.4992	-0.3960	0.0069	0.2002	-0.3021	-0.0187
	Std	0.0041	0.0205	0.0197	0.0095	0.0463	0.0425

4.5.2 Colored Noise Case

For the colored noise case, all model parameters are as same as in the white noise case, but the noise model is changed to

$$H(q^{-1}) = \frac{1}{1 - q^{-1} + 0.2q^{-1}}.$$

Again a Monte-Carlo simulation of 100 runs is carried out at SNR of 15dB, 10dB and 5dB. This example has been used in [Abbas et al., 2010a] and [Laurain et al., 2010]. The values of the bias error and variance from those references are repeated here for the purpose of comparison.

Table 4.4 shows the 2-norms of bias error and variance of the estimated parameters using all methods as in the white noise case and also gives a comparison with the SRIV, RIV [Laurain et al., 2010] and LRNN methods [Abbas et al., 2010a]. As shown in the table, all IV methods except IV-ARX give unbiased results along with LRNN that also gives an unbiased result. The improvement of IV-OE over the IV-ARX is considerable in this colored noise case. In IV-ARX, the first step using LS is not satisfactory enough to exclude the colored noise from the measured data; the generated output data is still highly correlated with noise and read bias results. In this aspect, the IV-OE method uses the same idea and can do a better job than IV-ARX since the IV-OE method uses $\hat{y}(k)$ instead of $y(k)$ which is less correlated with the colored noise. Compared to the recursive methods, i.e. SRIV and RIV – they use LPV-OE and LPV-BJ structure respectively – IV-OE gives also comparable results in terms of bias errors and better results in terms of variance. Compared to the LRNN method when SNR is 15dB and 10dB the bias errors are similar but IV-OE has much less variance. For 5 dB SNR, IV-OE gives a ten times higher bias error but the variance is ten time lower than LRNN. From a practical point of view, if the bias error is not too high a lower variance would be preferred because in real life the number of experiments should be as low as possible to reduce the computation time. Moreover, in each identification process in this simple example, LRNN requires almost one day to get this result–Monte-Carlo simulation or 100 runs–while IV-OE uses only 5 minutes for the same case on the same computer.

Table 4.4: Estimation bias 2-norm and its variance at different SNR : LPV with colored measurement noise

Method		15dB	10dB	5dB
LS	Bias	2.3108	1.9532	2.0846
	Variance	0.0008	0.0005	0.0005
OE	Bias	0.1619	0.7421	1.4727
	Variance	0.0018	0.0057	0.0198
IV-ARX	Bias	0.5506	1.7576	2.2505
	Variance	0.0648	2.8951	20.5136
IV-OE	Bias	0.0219	0.0446	0.2361
	Variance	0.0020	0.0086	0.0606
RIV[a]	Bias	0.0068	0.0184	0.0408
	Variance	0.0063	0.0219	0.0696
SRIV[a]	Bias	0.0072	0.0426	0.2988
	Variance	0.0149	0.0537	0.4425
LRNN	Bias	0.0184	0.0347	0.0866
	Variance	0.0853	0.1997	0.5861

---

[a] The data is taken from [Laurain et al., 2010].

Table 4.5 and 4.6 show the mean value and the standard variation of the estimated parameters at 15dB SNR. The results are consistent with the white noise case.

In conclusion, when we deal with the colored noise problem only IV-OE, SRIV, RIV and LRNN methods are recommended to obtain unbiased results. If we consider only the variance, one should select either IV-OE or RIV. However the latter method requires at least 30 iterations [Laurain et al., 2010] as shown in Table 4.4, and the noise model is assumed to be known. The properties of each algorithm are summarized in Table 4.7.

Table 4.5: Mean and standard deviation (Std) of the estimated $A(q^{-1}, \theta(k))$ polynomial parameters at SNR 15 dB : LPV with colored measurement noise

		$a_{1,0}$	$a_{1,1}$	$a_{1,2}$	$a_{2,1}$	$a_{2,2}$	$a_{2,3}$
Method	True Value	1	-0.5	0.2	0.5	-0.7	-0.1
LS	Mean	0.1733	-1.6268	1.3448	0.0629	-1.1608	0.4577
	Std	0.0432	0.1592	0.1326	0.0278	0.1109	0.0947
OE	Mean	1.0032	-0.6295	-0.1796	0.5002	-0.7438	-0.1178
	Std	0.0302	0.2384	0.2369	0.0095	0.1330	0.1757
IV-ARX	Mean	1.0062	-0.1151	-0.4878	0.5017	-0.5189	-0.1923
	Std	0.1178	1.4850	1.1202	0.0489	0.6916	0.4674
IV-OE	Mean	0.9978	-0.5177	-0.1963	0.4991	-0.7084	-0.1045
	Std	0.0287	0.2500	0.0090	0.1525	0.1835	0.0104

Table 4.6: Mean and standard deviation (Std) of the estimated $B(q^{-1}, \theta(k))$ polynomial parameters at SNR 15 dB : LPV with colored measurement noise

		$b_{1,0}$	$b_{1,1}$	$b_{1,2}$	$b_{2,1}$	$b_{2,2}$	$b_{2,3}$
Method	True Value	0.5	-0.4	0.01	0.2	-0.3	-0.02
LS	Mean	0.5001	-0.3992	0.0092	-0.2320	-0.1904	0.2757
	Std	0.0066	0.0250	0.0218	0.0211	0.0686	0.0522
OE	Mean	0.4993	-0.3926	0.0025	0.2013	-0.3569	-0.0232
	Std	0.0059	0.0361	0.0357	0.0215	0.1044	0.0904
IV-ARX	Mean	0.5012	-0.4042	0.0135	0.2077	-0.1756	-0.1436
	Std	0.0101	0.0467	0.0421	0.0667	0.5387	0.4956
IV-OE	Mean	0.5002	-0.4011	0.0113	0.1988	-0.3054	-0.0150
	Std	0.0104	0.0480	0.0427	0.0235	0.1013	0.0859

System and Model Structure are mismatched

It is worth to illustrate the performance of the proposed method when selecting a wrong model structure. In this experiment all settings of the true system are the same as the previous experiment but the selected identification model is incorrect, i.e. $n_a = 3$ instead of 2.

The result is shown in Figure 4.4 where the comparison of a cross validation between the simulation output of the identified model and the original noise-free and the noisy

Table 4.7: The properties of each algorithm when use to identify LPV model with colored measurement noise.

Method	Bias	Variance	Computation Time
LS	high (-)	very low (+)	very fast (+)
OE	low (+)	low(+)	very fast (+)
IV-ARX	high (-)	very high (-)	very fast (+)
IV-OE	very low (+)	very low (+)	very fast (+)
RIV	lowest (++)	very low(+)	slow (-)[a]
SRIV	very low (+)	low (+)	slow (-)[a]
LRNN	very low (+)	low (+)	very slow (- -)

---

[a] Depends on the number of iterations.

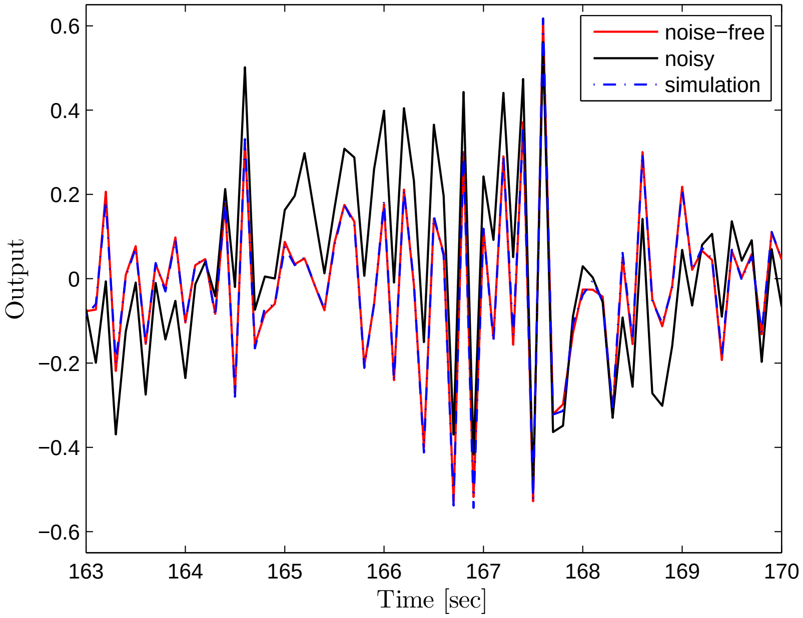


Figure 4.4: Cross validation test of IV-OE at SNR = 10 dB when the model structure is not correct.

output at SNR = 10 dB are depicted. The plot of the simulation output is almost identical to the noise free output. This means that the introduced method works well although the the selected model structure is not correct.

4.6 Identification of Quasi-LPV Input-Output Models

In practice, the nonlinear systems are approximated by a quasi-LPV structure like the one given in Chapter 3. The identification of the quasi-LPV model in which an output signal is considered as a scheduling signal, is more difficult than that of the normal LPV model. This is because the noise signal is correlated not only with the output signal but also with the scheduling signal. As discussed in [Butcher et al., 2008], consistent estimation occurs only when the scheduling signal is noise free. Using the IV-OE method described in the previous section is clearly more useful in this case because the estimation procedure of the auxiliary model relies on $\hat{y}(k)$ instead of $y(k)$, which is uncorrelated with noise. The simulation output from the model estimated by an OE step, can be used for both a noise free scheduling signal and a noise free output. In the following example, we show that if the noise signal is white and SNR is reasonably high, the IV-OE can be used for quasi-LPV system identification.

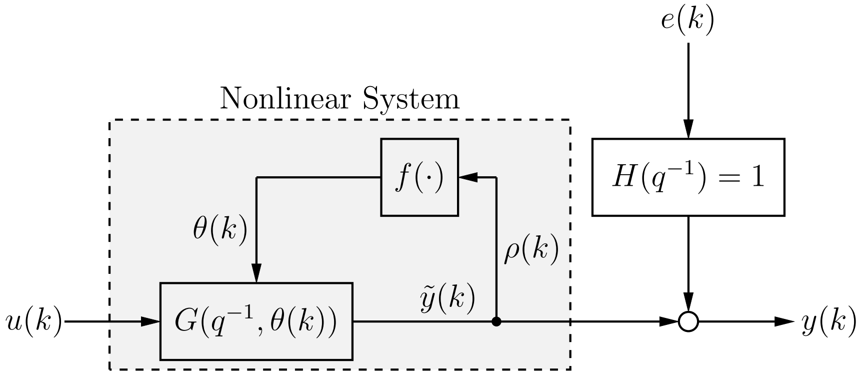


Figure 4.5: A quasi-LPV data generating system: the dashed box is a nonlinear system represented as a quasi-LPV model.

The data generating system is shown in Figure 4.5, it is assumed that a nonlinear system is represented as a quasi-LPV model shown in the shaded box. The aim is to estimate the quasi-LPV model in the shaded box by using the input data $u(k)$ and the contaminated output data $y(k)$. The noise signal is white noise and $H(q^{-1}) = 1$. The model structure used in this section is the same as in the previous cases. The true model is given by

$$\mathcal{S}_0 \begin{cases} A(q^{-1}, \theta(k)) & = 1 + a_1(\theta(k))q^{-1} + a_2(\theta(k))q^{-2} \\ B(q^{-1}, \theta(k)) & = b_0(\theta(k))q^{-1} + b_1(\theta(k))q^{-2} \\ H(q^{-1}) & = 1, \end{cases} \tag{4.34}$$

The coefficient dependence on the scheduling parameter is chosen as follow:

$$a_1(\theta(k)) = 1 - 0.5\theta(k) + 0.2\theta^2(k), \tag{4.35a}$$

$$a_2(\theta(k)) = 1 - 0.7\theta(k) - 0.1\theta^2(k), \tag{4.35b}$$

$$b_0(\theta(k)) = 0.5 - 0.4\theta(k) + 0.01\theta^2(k), \tag{4.35c}$$

$$b_1(\theta(k)) = 0.2 - 0.3\theta(k) - 0.02\theta^2(k), \tag{4.35d}$$

where $\rho(k) = y(k-1)$, and $\theta(k) = 0.5\cos(\rho(k))$.

The input signal $u(k)$ in Figure 4.5 is taken as a white noise like in the previous cases. Then, Monte-Carlo simulation of 100 runs is carried out for three different values of SNR: 30dB, 20dB and 10 dB to compare the proposed IV-OE method with LS, OE and IV-ARX methods.

The comparison in terms of bias error and variance is shown in Table 4.8. Both instrumental variable methods provide unbiased results but it is clear that IV-OE gives the better results in both statistical indicators. In Table 4.9 and 4.10 the mean and standard deviation of the estimated parameters at 20 dB are displayed in details. Again it is clear that the IV-OE method outperforms the other methods.

Table 4.8: Estimation bias 2-norm and its variance at different SNR: quasi-LPV with white measurement noise

Method		30dB	20dB	15dB
LS	Bias	0.1645	1.1554	2.3984
	Variance	0.0029	0.0271	0.0295
OE	Bias	0.1061	1.0311	1.0819
	Variance	0.0125	0.1516	0.1576
IV-ARX	Bias	0.0102	0.0459	0.0656
	Variance	0.0031	0.1211	0.7175
IV-OE	Bias	0.0097	0.0294	0.0456
	Variance	0.0034	0.0391	0.1934

System and Model Structure are mismatched

In this section we show the performance of the proposed algorithm when the model structure is not correct. The true system is the same as the previous experiment but the selected model is different, e.g. $n_a = 3$ instead of 2.

The result is shown in Figure 4.6, in which the comparisons of a cross validation between the simulation output of the identified model and the original noise-free and the noisy output at SNR = 15 dB are displayed. Again the plot of the simulation output is nearly identical to the noise-free output. It is confirming that the IV-OE method works well even when the true model structure is not in the selected model set.

Table 4.9: Mean and standard deviation (Std) of the estimated $A(q^{-1}, \theta(k))$ polynomial parameters at SNR 20 dB : quasi-LPV with white measurement noise

		$a_{1,0}$	$a_{1,1}$	$a_{1,2}$	$a_{2,1}$	$a_{2,2}$	$a_{2,3}$
Method	True Value	1	-0.5	0.2	1.0	-0.7	-0.1
LS	Mean	0.7321	-0.2524	0.1708	0.3047	0.1119	-0.3188
	Std	0.0471	0.0693	0.0277	0.1079	0.1339	0.0415
OE	Mean	0.7860	-0.2128	0.0972	0.4707	-0.0391	-0.3108
	Std	0.1654	0.2698	0.1085	0.2401	0.3248	0.1118
IV-ARX	Mean	1.0121	-0.5288	0.2135	1.0154	-0.7223	-0.0932
	Std	0.0887	0.2494	0.1281	0.2811	0.2564	0.0586
IV-OE	Mean	1.0011	-0.5138	0.2087	1.0119	-0.7183	-0.0944
	Std	0.0752	0.1122	0.0423	0.1327	0.1776	0.0586

Table 4.10: Mean and standard deviation (Std) of the estimated $B(q^{-1}, \theta(k))$ polynomial parameters at SNR 20 dB : quasi-LPV with white measurement noise

		$b_{1,0}$	$b_{1,1}$	$b_{1,2}$	$b_{2,1}$	$b_{2,2}$	$b_{2,3}$
Method	True Value	0.5	-0.4	0.01	0.2	-0.3	-0.02
LS	Mean	0.4335	-0.3441	0.0025	0.1530	-0.2761	-0.0143
	Std	0.0341	0.0502	0.0181	0.0346	0.0496	0.0176
OE	Mean	0.3371	-0.1547	-0.0785	0.0559	-0.0937	-0.0911
	Std	0.1158	0.1739	0.0628	0.1284	0.1882	0.0667
IV-ARX	Mean	0.4956	-0.3940	0.0073	0.1943	-0.2921	-0.0231
	Std	0.0623	0.0930	0.0335	0.0694	0.1169	0.0462
IV-OE	Mean	0.4951	-0.3946	0.0080	0.2031	-0.3048	-0.0187
	Std	0.0416	0.0622	0.0225	0.0488	0.0707	0.0250

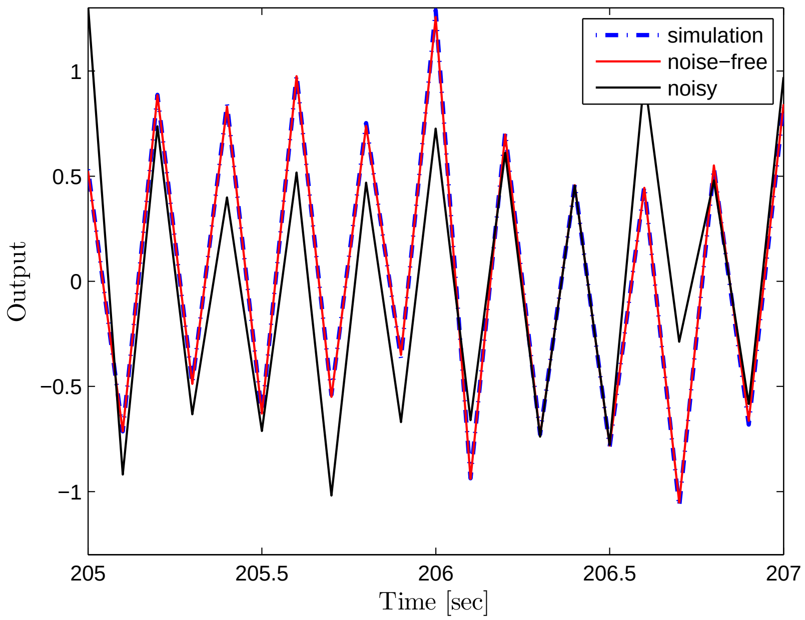


Figure 4.6: Cross validation test of IV-OE for a quasi-LPV at SNR = 15 dB when the model structure is not correct.

4.7 Conclusion

This chapter has shown the improvement of the model dependent instrumental variable method by using an LPV-OE model instead of LPV-ARX in the first steps of the IV algorithm. The new IV-OE algorithm shows significant improvement over the original method in terms of bias error and variance, by maintaining the simple two steps of the original algorithm. Moreover the method can be used in both LPV and quasi-LPV cases. The performance of the method in the colored noise case is also compared with more complicated methods like SRIV, RIV and LRNN. Simulation examples have shown that the proposed method is comparable to the best existing methods while using much less computation power, since it requires only one iteration of the IV procedure.

Chapter 5

State-Space Realization of LPV Input-Output Models

If we classify system identification methods based on system structures, we can separate them into two subgroups. One is an input-output representation based technique, e.g. [Bamieh and Giarré, 2002; Wei, 2006], which is based on the *Prediction-Error Method* (PEM). Another one is a state-space representation technique, e.g. [van Wingerden and Verhaegen, 2009], which is based on the *subspace method*. Both subgroups have their own advantages and disadvantages. The input-output based method does not suffer from the curse of dimensionality like the subspace based method [Verdult and Verhaegen, 2001], however the state-space realization of the LPV input-output model is still developing. The realization is very important when the models are used for controller design purposes, because all high-performance controller design techniques are based on state-space models. In the LTV literature, state-space realizations of LTV models with time-varying coefficients have been considered for a long time, e.g. [Kamen et al., 1985; Khargonekar and Poolla, 1986; Poolla and Khargonekar, 1987; Verriest, 1993; Guidorzi and Diversi, 2003] and more recently [Zerz, 2007]. In contrast with LPV systems, the problems of time dependence of the scheduling signals are always neglected and the transformation of input-output models into state-space form is done using an LTI realization method, see for example [Giarré et al., 2006; Qin and Wang, 2007a,b]. The first investigation of this problem has been done in Tóth et al. [2007]. The transformation of LPV input-output models to state-space LPV models without considering the time dependence of the scheduling signals typically lead to inequivalent state-space models. These inequivalent models weaken the performance of the closed-loop system.

To illustrate the above problem, a simple example introduced by Tóth et al. [2009b] shows the inequivalence between a state-space realization and the original LPV input-output model.

Example 5.1 [*Realization of LPV model using LTI framework*] A LPV input-output system is given by

$$y(k) = a_1(\theta(k))y(k-1) + a_2(\theta(k))y(k-2) + b_1(\theta(k))u(k-1) + b_2(\theta(k))u(k-2). \tag{5.1}$$

Constructing a realization without considering the time-dependence of the scheduling signal $\theta(k)$, we get the following second-order state-space representation:

$$\begin{aligned} \begin{bmatrix} x_1(k+1) \\ x_2(k+1) \end{bmatrix} &= \begin{bmatrix} 0 & a_2(\theta(k)) \\ 1 & a_1(\theta(k)) \end{bmatrix} \begin{bmatrix} x_1(k) \\ x_2(k) \end{bmatrix} + \begin{bmatrix} b_2(\theta(k)) \\ b_1(\theta(k)) \end{bmatrix} u(k), \\ y(k) &= x_2(k). \end{aligned} \tag{5.2}$$

However, with simple manipulations the above system (5.2) can be transformed into an equivalent input-output representation

$$y(k) = a_1(\theta(k-1))y(k-1) + a_2(\theta(k-2))y(k-2) \tag{5.3}$$

$$+ b_1(\theta(k-1))u(k-1) + b_2(\theta(k-2))u(k-2). \tag{5.4}$$

It is clear that the LPV input-output model (5.3) is not equivalent to the original one (5.1). ■

In Tóth [2010], this problem is addressed by using a behavioral approach. The study concentrates on an analysis framework of LPV systems. However synthesis tools are still missing. In contrast to this work, we will use an approach based on algebraic concepts, in particular a ring of skew polynomials as skew polynomial matrices [Wolovich, 1974; Callier and Desoer, 1982; Antsaklis and Michel, 2006] and extend to LTV systems see [Verriest, 1993; Ylinen and Zenger, 1992], to build a practical framework for LPV system identification and realization.

Inspired by the algebraic framework for LTV system realization of Verriest [1993]; Zerz [2006]; Ylinen and Zenger [1992], we extend it to LPV systems. Firstly we propose a method using a *skew polynomial* concept to construct a realization framework for LPV input-output systems. This can be done by separating the LPV input-output representation with polynomials in a *Left Polynomial Representation* (LPR) and a *Right Polynomial Representation* (RPR) form. With this separation, we can modify the realization framework of LTI systems to transform LPV input-output models into state-space models without violating the noncommutative rule. A state-space system in reachable form can be directly obtained from an RPR, while the observable form can be derived from an LPR. This modification is not limited to SISO systems but can also be used for MIMO systems. Finally we show that when an LPV input-output model is identified in LPR, a state-space model in observable form can be directly obtained which is minimal.

The chapter is organized as follows. A review of some required algebraic concepts is given in Section 5.1. All necessary definitions related to the skew polynomial theory are also given. Section 5.2 gives a brief review of LPV state-space representations and their properties. In Section 5.3 we draw the modification of the LTI realization framework to use SISO LPV systems and also show examples. The extension to MIMO systems is given in Section 5.4. Section 5.5 presents the application of the idea to LPV input-output system identification, including an example. Finally conclusion are drawn in the last section.

5.1 Algebraic Preliminaries

Some material which is required for the proposed method, is introduced in this section. To make it simpler, here we restrict the discussion to SISO systems, while the material necessary for MIMO systems will be explained when it is needed. The material introduced in this section is mostly adapted from [Verriest, 1993; Ylinen and Zenger, 1992] and references therein, and has been converted from continuous-time to discrete-time systems.

5.1.1 Skew Polynomials

Throughout this chapter, let $\mathcal{K}$ denote a field† generated by real-rational functions of a real scheduling parameter θ. These rational functions are allowed to depend on a set of time-shifted instances of $\theta(k)$ denoted by Θ, i.e. $\Theta = \cdots, \theta(k-1), \theta(k), \theta(k+1), \cdots$. $\mathcal{K}[q]$ denotes the noncommutative ring of polynomials in q with coefficients $\boldsymbol{a}(\theta(k)) \in \mathcal{K}$ where q is the time shift operator. Polynomials which are members of this ring are called skew polynomials. A general operator in the ring $\mathcal{K}[q]$ is denoted by $a(q, \theta(k))$. Now an element of $\mathcal{K}[q]$ not equal zero, can be represented either in a left or a right form as follows:

i. Right Polynomial Representation (RPR)

$$a(q, \theta(k)) = \sum_{i=0}^{n} \boldsymbol{a}(\theta(k)) q^{n-i}, \tag{5.5}$$

ii. Left Polynomial Representation(LPR)‡

$$\tilde{a}(q, \theta(k)) = \sum_{i=0}^{\bar{n}} q^{\bar{n}-i} \boldsymbol{a}(\theta(k)). \tag{5.6}$$

The $a(q, \theta(k))$ and $\tilde{a}(q, \theta(k))$ are related by the commutation rule,

$$q\boldsymbol{a}(\theta(k)) = \boldsymbol{a}(\theta(k+1))q, \tag{5.7}$$

for example

$$\boldsymbol{a}(\theta(k)) + \boldsymbol{a}(\theta(k))q^2 = \boldsymbol{a}(\theta(k)) + q^2 \boldsymbol{a}(\theta(k-2)).$$

---

† For rings and fields see Appendix C.

‡ In [Abbas et al., 2010b] this representation has been called *shifted form*

A degree function for the skew polynomials is defined by the highest value of the power of non-zero monomial terms. The degree of the zero polynomial is $-\infty$. With this degree function, it can be shown that the ring $\mathcal{K}[q]$ is a noncommutative Euclidean domain and it satisfies the so-called *Ore condition*.

Lemma 5.1 (Ore condition [Ore, 1931; Cohn, 1961])**.** For all non-zero $a(q,\theta(k))$, $b(q,\theta(k)) \in \mathcal{K}[q]$, there exist non-zero $a_1(q,\theta(k))$,$b_1(q,\theta(k)) \in \mathcal{K}[q]$ such that

$$a_1(q,\theta(k))b(q,\theta(k)) = b_1(q,\theta(k))a(q,\theta(k)).$$

The left Ore condition means that any two elements of $\mathcal{K}[q]$ have a *Common Left Multiple* (CLM), see Section 5.1.2. Then $\mathcal{K}[q]$ can be embedded in a noncommutative fraction field [Ore, 1931; Cohn, 1961].

Let a_1, a_2, b_1, b_2, c_1, c_2, d_1, d_2 be in $\mathcal{K}[q]$, then $\mathcal{K}[q]$ can be embedded in a noncommutative fraction field by defining each fraction [Ylinen and Zenger, 1992; Halás and Kotta, 2007] as

$$\frac{a_1}{b_1} = b_1^{-1}a_1, \tag{5.8}$$

where $b_1 \neq 0$. Addition and multiplication are defined as

$$\frac{a_1}{b_1} + \frac{a_2}{b_2} = \frac{c_2a_1 + c_1a_2}{c_2b_1} \tag{5.9}$$

where $c_2b_1 = c_1b_2$ by the Ore condition and

$$\frac{a_1}{b_1}\frac{a_2}{b_2} = \frac{d_1a_2}{d_2a_1} \tag{5.10}$$

where $d_2a_1 = d_1b_2$ by the Ore condition. The resulting fraction field of skew polynomials is denoted by $\mathcal{K}(q)$ [Halás, 2009]; $\mathcal{K}[q] \subset \mathcal{K}(q)$ and $\frac{1}{a_1} \notin \mathcal{K}[q]$ but $\frac{1}{a_1} \in \mathcal{K}(q)$. These addition and multiplication rules are necessary for the row and column operations of the skew polynomial matrices in a subsequent section.

5.1.2 Noncommutative Algebra

To construct the polynomials c_1, c_2, d_1, d_2 mentioned in the previous section, a division algorithm for skew rings is required. All material here is adopted from [Verriest, 1993]. Parameters (q, θ) are omitted for simplicity.

Definition 5.1. Let $a, b \in \mathcal{K}[q]$, then

i. $c \in \mathcal{K}[q]$ is a *left common divisor* (LCD) of a and b if there exist a_1 and $b_1 \in \mathcal{K}[q]$ such that $a = ca_1$ and $b = cb_1$.

ii. a and b are *left coprime* (LC) if there is no LCD of a and b, except for the units in $\mathcal{K}$.

iii. $d \in \mathcal{K}[q]$ is a *left common multiple* (LCM) of a and b if $\exists a_1, b_1 \in \mathcal{K}[q]$ such that $d = a_1 a = b_1 b$.

Similarly we can define a *right common divisor* (RCM), a *right common multiple* (LCM) and a *right coprime* (RC).

Theorem 5.1 (Left Division Algorithm)**.** $\forall a, b \in \mathcal{K}[q]$ such that $\deg(a) \geq \deg(b) \geq 1$, there exist $g \in \mathcal{K}$; $c, d \in \mathcal{K}[q]$, such that:

$$\begin{aligned} ag &= bc + d \\ \deg(d) &< \deg(b) \\ g, c &\text{ right coprime .} \end{aligned}$$

Definition 5.2 (Greatest common left divisor)**.** The *Greatest Common Left Divisor* (GCLD) of two skew polynomials a and $b \in \mathcal{K}[q]$ is the skew polynomial $l \in \mathcal{K}[q]$ such that $\exists a_1, b_1 \in \mathcal{K}[q]$ with $a = la_1$ and $b = lb_1$; and if $m \in \mathcal{K}[q]$ is any other element of $\mathcal{K}[q]$ such that $a = ma_1$ and $b = mb_1$ then there exists $l_1 \in \mathcal{K}[q]$ such that $l = ml_1$.

5.2 LPV State-Space Representation

The LPV state-space representation discussed in this chapter is standard and defined as

$$\begin{aligned} x(k+1) &= \mathcal{A}(\theta(k))x(k) + \mathcal{B}(\theta(k))u(k) \\ y(k) &= \mathcal{C}(\theta(k))x(k) + \mathcal{D}(\theta(k))u(k) \end{aligned} \tag{5.11}$$

where $\mathcal{A}(\theta(k))$, $\mathcal{B}(\theta(k))$, $\mathcal{C}(\theta(k))$ and $\mathcal{D}(\theta(k))$ are matrices with elements in $\mathcal{K}$.

For LTI systems, three important properties of state-space representations are *reachability*, *observability* and *minimality*, see [Kailath, 1980; Antsaklis and Michel, 2006]. For LPV systems, however, these properties are more complicated than for LTI systems, because they have to hold for all values of the scheduling variables θ in the considered range. The problem becomes an infinite dimensional problem because one has to check for all possible values of θ. These properties for LPV systems can be defined by extending the properties of LTI systems [Wolovich, 1974] by taking the scheduling signals into account [Tóth, 2010].

Definition 5.3 (Complete LPV state-reachability)**.** An LPV state-space system is said to be complete state reachable, if for any given two state x_1, x_2 and any value of scheduling signals $\theta \in \mathbb{R}^{n_\theta}$, there exist an input signal u such that $x(k_1) = x_1$ and $x(k_2) = x_2$ for some $k_1, k_2 \in \mathbb{T}$.

Definition 5.4 (Complete LPV state-observability)**.** The LPV state-space system is said to be complete state observable if and only if the entire state $x(k)$ can be determined over any finite time interval $[k_0, k_1]$ from complete knowledge of the system input and output over the time interval $[k_0, k_1]$ with $k_1 > k_0 \geq 0$ for any value of scheduling parameters $\theta \in \mathbb{R}^{n_\theta}$.

Definition 5.5 (Minimality)**.** An LPV state-space system defined through the matrices $\{\mathcal{A}(\theta(k)), \mathcal{B}(\theta(k)), \mathcal{C}(\theta(k)), \mathcal{D}(\theta(k))\}$ with order n is said to be complete minimal if there exists no equivalent LPV state-space system with order $\hat{n} < n$.

5.3 Realization of SISO LPV Input-Output Models

A state-space representation is one of the most commonly used methods to describe a dynamic system. Moreover most advanced controller design methods are based on the state-space models. Due to this reason the construction of a state-space LPV representation that is equivalent to a given LPV input-output representation is an important problem. This problem is know as the *realization* problem. In this part, we will consider only the SISO case. The extension to MIMO systems is not straightforward and will be discussed in a later section.

An LPV input-output systems in the operator form

$$a(q,\theta(k))y(k) = b(q,\theta(k))u(k), \tag{5.12}$$

can be viewed as an *operational transfer function*

$$y(k) = h(q,\theta(k))u(k), \tag{5.13}$$

where $h(q,\theta(k)) = a^{-1}(q,\theta(k))b(q,\theta(k))$ and $h(q,\theta(k)), a(q,\theta(k))$, and $b(q,\theta(k))$ are in $\mathcal{K}(q)$.

The realization frameworks used for LTI [Antsaklis and Michel, 2006] and LTV [Verriest, 1993] systems can be adapted to be used for LPV systems. According to [Verriest, 1993], the noncommutativity of q operators must be separated between a *Right Fraction Description* (RFD) and a *Left Fraction Description* (LFD) of the operational transfer function. For an LPV system,

RFD:

$$\begin{aligned} a(q,\theta(k))x(k) &= u(k) \\ y(k) &= b(q,\theta(k))x(k) \\ &= b(q,\theta(k))a^{-1}(q,\theta(k))u(k) \end{aligned} \tag{5.14}$$

LFD:

$$\begin{aligned} \tilde{a}(q,\theta(k))x(k) &= \tilde{b}(q,\theta(k))u(k) \\ y(k) &= x(k) \\ &= \tilde{a}^{-1}(q,\theta(k))\tilde{b}(q,\theta(k))u(k) \end{aligned} \tag{5.15}$$

The LFD is directly obtained from the input-output representation (5.12) while the RFD is obtained via the GCLD. This is because in the LPV case, even for SISO systems, the operational transfer functions $b(q,\theta(k))a^{-1}(q,\theta(k))$ and $a^{-1}(q,\theta(k))b(q,\theta(k))$ do not render the same system due to the noncommutativity.

5.3.1 Observable Form

A SISO discrete-time LPV input-output system given in LFD form by

$$\tilde{h}(q,\theta(k)) = \tilde{a}^{-1}(q,\theta(k))\tilde{b}(q,\theta(k)), \tag{5.16}$$

where

$$\begin{aligned}\tilde{a}(q,\theta(k)) &= q^n + q^{n-1}\tilde{a}_{n-1}(\theta(k)) + \cdots + q\tilde{a}_1(\theta(k)) + \tilde{a}_0(\theta(k))\\ \tilde{b}(q,\theta(k)) &= q^n\tilde{b}_n(\theta(k)) + \cdots + q\tilde{b}_1(\theta(k)) + \tilde{b}_0(\theta(k))\end{aligned}$$

is in LPR structure.

A state-space realization in observable form is then given by

$$\begin{aligned}x(k+1) &= \tilde{\mathcal{A}}_o(\theta(k))x(k) + \tilde{b}_o(\theta(k))u(k)\\ &= \begin{bmatrix} 0 & \cdots & 0 & -\tilde{a}_0(\theta(k))\\ 1 & \cdots & 0 & -\tilde{a}_1(\theta(k))\\ \vdots & \ddots & \vdots & \vdots\\ 0 & \cdots & 1 & -\tilde{a}_{n-1}(\theta(k))\end{bmatrix} x(k) + \begin{bmatrix} \tilde{b}_0(\theta(k)) - \tilde{b}_n(\theta(k))\tilde{a}_0(\theta(k))\\ \tilde{b}_1(\theta(k)) - \tilde{b}_n(\theta(k))\tilde{a}_1(\theta(k))\\ \vdots\\ \tilde{b}_{n-1}(\theta(k)) - \tilde{b}_n(\theta(k))\tilde{a}_{n-1}(\theta(k))\end{bmatrix} u(k)\\ y(k) &= \tilde{c}_o(\theta(k))x(k) + \tilde{d}_o(\theta(k))u(k)\\ &= \begin{bmatrix}0 & 0 & \cdots & 0 & 1\end{bmatrix} x(k) + \tilde{b}_n(\theta(k))u(k).\end{aligned} \tag{5.17}$$

Without loss of generality we can assume that the system is strictly proper or $\tilde{b}_n(\theta(k))$ in $\tilde{b}(q,\theta(k))$ and $\tilde{d}_o(\theta(k))$ are zero. Then, one can show that the state-space representation in (5.17) is equivalent to the input-output system (5.16). Let $\tilde{s}(q) = \begin{bmatrix}1 & q & \cdots & ,q^{n-1}\end{bmatrix}$. We have

$$\begin{aligned}\tilde{s}(q)x(k+1) &= \tilde{s}(q)\tilde{\mathcal{A}}_o(\theta(k))x(k) + \tilde{s}(q)\tilde{b}_o(\theta(k))u(k)\\ qx_1(k) + q^2x_2(k) + \cdots + q^nx_n(k) &= \big[qx_1(k) + q^2x_2(k) + \cdots + q^{n-1}x_{n-1}(k)\\ &\quad - \tilde{a}_0(\theta(k))x_n(k) - \ q\tilde{a}_1(\theta(k))x_n(k) - \cdots\\ &\quad -q^{n-1}\tilde{a}_{n-1}(\theta(k))x_n(k)\big] + \tilde{b}(q,\theta(k))u(k),\end{aligned}$$

then

$$\begin{aligned}q^nx_n(k) + q^{n-1}\tilde{a}_{n-1}(\theta(k))x_n(k) + \cdots + q\tilde{a}_1(\theta(k))x_n(k) + \tilde{a}_0(\theta(k))x_n(k)&\\ = \tilde{b}(q,\theta(k))u(k),&\end{aligned} \tag{5.18}$$

where it is clear that $\tilde{s}(q)\tilde{b}_o(\theta(k)) = \tilde{b}(q,\theta(k))u(k)$.

The last equation (5.18) is exactly $\tilde{a}(q,\theta(k))x_n(k) = \tilde{b}(q,\theta(k))u(k)$ when $\tilde{b}_n(\theta(k)) = 0$. By defining $y(k) = x_n(k)$, we will have the equivalent input-output representation. Conversion from the input-output representation to the state-space representation in observable form, can be done simply by defining

$$\begin{aligned}y(k) &= x_n(k)\\ x_1(k+1) &= -\tilde{a}_0(\theta(k))x_n(k) + \tilde{b}_0(\theta(k))u(k)\\ x_2(k+1) &= x_1(k) - \tilde{a}_1(\theta(k))x_n(k) + \tilde{b}_1(\theta(k))u(k)\\ &\vdots\\ x_n(k+1) &= x_{n-1}(k) - \tilde{a}_{n-1}(\theta(k))x_n(k) + \tilde{b}_{n-1}(\theta(k))u(k).\end{aligned}$$

Hence the state-space representation in observable form is constructed by (5.17). Since the state-variables can be constructed by using only the sampled data of input and output signals with any values of $\theta(k)$, the representation is observable. We can summarize this as the following lemma.

Lemma 5.2. The LPV state-space model (5.17) is a realization of $\tilde{h}(q,\theta(k))$ (5.16) and observable.

Since the order of the state-space system in an observable form is equal to the order of $\tilde{a}(q,\theta(k))$, the minimal order of the state-space representation is reached when $\tilde{a}(q,\theta(k))$ and $\tilde{b}(q,\theta(k))$ are left coprime, i.e. when there is no left common factor between $\tilde{a}(q,\theta(k))$ and $\tilde{b}(q,\theta(k))$.

Lemma 5.3. The realization of $\tilde{h}(q,\theta(k)) = \tilde{a}^{-1}(q,\theta(k))\tilde{b}(q,\theta(k))$ in the observable form is minimal if and only if $\tilde{a}(q,\theta(k))$ and $\tilde{b}(q,\theta(k))$ are left coprime.

The following example shows the realization in observable form of a discrete-time LPV SISO system. This example is adapted from an LTV example in [Ylinen and Zenger, 1992].

Example 5.2 [*Observable form*] Consider the input-output system described by

$$(q^2 + e^{-k}q + k^2)y(k) = (kq+1)u(k),$$

where k is the discrete-time variable. In this case the functional dependence is $\theta_1(k) = e^{-k}$ and $\theta_2(k) = k$. By converting the polynomials into LPR form, we have

$$\Big[q^2 + qe^{-(k-1)} + (k-1)^2\Big]y(k) = \Big[q(k-1)+1\Big]u(k)$$

and

$$q\Big[qy(k) + e^{-(k-1)}y(k) - (k-1)u(k)\Big] + \Big[(k-1)^2y(k) - u(k)\Big] = 0.$$

Let

$$x_2(k) = y(k), x_1(k+1) = -(k-1)^2x_2(k) + u(k),$$

and

$$x_2(k+1) = x_1(k) - e^{-(k-1)}x_2(k) + (k-1)u(k).$$

We obtain a state-space representation as

$$q\begin{bmatrix}x_1(k)\\x_2(k)\end{bmatrix} = \begin{bmatrix}0 & -(k-1)^2\\1 & -e^{-(k-1)}\end{bmatrix}x(k) + \begin{bmatrix}1\\k-1\end{bmatrix}u(k)$$

$$y(k) = \begin{bmatrix}0 & 1\end{bmatrix}\begin{bmatrix}x_1(k)\\x_2(k)\end{bmatrix},$$

which is in observable form.

■

From the above example, it is not difficult to see that an LPV input-output model with all polynomials in LPR form can be transformed to an LPV state-space model in observable form with static dependence, by using the introduced method.

5.3.2 Reachable Form

The realization of an input-output representation to a state-space representation in *reachable form* is more difficult than the observable form. In contrast to the observable form, we start with an input-output model with polynomials in RPR, as

$$h(q,\theta(k)) = a^{-1}(q,\theta(k))b(q,\theta(k)) \tag{5.19}$$

where

$$\begin{aligned}
a(q,\theta(k)) &= q^n + \mathscr{a}_{n-1}(\theta(k))q^{n-1} + \cdots + \mathscr{a}_1(\theta(k))q + \mathscr{a}_0(\theta(k)),\\
b(q,\theta(k)) &= \mathscr{b}_n(\theta(k))q^n + \cdots + \mathscr{b}_1(\theta(k))q + \mathscr{b}_0(\theta(k)).
\end{aligned}$$

A state-space realization in a reachable form is given by the equation

$$\begin{aligned}
x(k+1) &= \mathscr{A}_c(\theta(k))x(k) + \mathscr{b}_c(\theta(k))u(k)\\
&= \begin{bmatrix} 0 & 1 & \cdots & 0\\ \vdots & \vdots & \ddots & \vdots\\ 0 & 0 & \cdots & 1\\ -\mathscr{e}_0(\theta(k)) & -\mathscr{e}_1(\theta(k)) & \cdots & -\mathscr{e}_{n-1}(\theta(k)) \end{bmatrix} x(k) + \begin{bmatrix} 0\\ 0\\ \vdots\\ 1 \end{bmatrix} u(k) \qquad (5.20)\\
y(k) &= \mathscr{c}_c(\theta(k))x(k) + \mathscr{d}_c(\theta(k))u(k)\\
&= \begin{bmatrix} \mathscr{m}_0(\theta(k)) & \mathscr{m}_1(\theta(k)) & \cdots & \mathscr{m}_{n-1}(\theta(k)) \end{bmatrix} x(k) + \mathscr{d}_c(\theta(k))u(k).
\end{aligned}$$

In this structure the elements of $\mathscr{A}_c(\theta(k))$ and $\mathscr{c}_c(\theta(k))$ cannot be directly taken from the coefficient parameters of $a(q,\theta(k))$ and $b(q,\theta(k))$. In the reachable form, we must first represent a transfer operator function in RFD form. Due to the noncommutativity, one could not simply swap the polynomials $a^{-1}(q,\theta(k))$ and $b(q,\theta(k))$. One has to do it via the GCLD of the polynomials.

Again we assume that the system is strictly proper or $\mathscr{b}_n(\theta(k))$ in $b(q,\theta(k))$ and $\mathscr{d}_c(\theta(k))$ are zero. In reachable form (5.20), we know that

$$\begin{aligned}
x(k) &= \begin{bmatrix} x_1(k), & \ldots\ x_n(k) \end{bmatrix}^T\\
&= \begin{bmatrix} 1 & q & \ldots & q^{n-1} \end{bmatrix}^T x_1(k)\\
&= s(q)x_1(k),
\end{aligned}$$

where $s(q)$ denotes a vector $\begin{bmatrix} 1 & q & \ldots & q^{n-1} \end{bmatrix}^T$. Moreover from the last row of $\mathscr{A}_c(\theta(k))$ and $\mathscr{b}_c(\theta(k))$ in (5.20) we also know that

$$\begin{aligned}
x_n(k+1) &= -\mathscr{e}_0(\theta(k))x_1(k) - \mathscr{e}_1(\theta(k))x_2(k) - \cdots - \mathscr{e}_{n-1}(\theta(k))x_n(k) + u(k)\\
q^n x_1(k) &= -\mathscr{e}_0(\theta(k))x_1(k) - \mathscr{e}_1(\theta(k))qx_1(k) - \cdots - \mathscr{e}_{n-1}(\theta(k))q^{n-1}x_1(k) + u(k),
\end{aligned}$$

where $x_2(k) = qx_1(k)$, …, $x_n(k) = q^{n-1}x_1(k)$. Then

$$\begin{aligned}
e(q,\theta(k))x_1(k) &= u(k)\\
x_1(k) &= e^{-1}(q,\theta(k))u(k),
\end{aligned}$$

where $e(q,\theta(k)) = q^n + \mathscr{e}_{n-1}(\theta(k))q^{n-1} + \cdots + \mathscr{e}_1(\theta(k))q + \mathscr{e}_0(\theta(k))$. Altogether, we have

$$x(k) = s(q)e^{-1}(q,\theta(k))u(k) = (qI - \mathscr{A}_c(\theta(k)))^{-1}\mathscr{b}_c(\theta(k))u(k),$$

from (5.20) or

$$s(q)e^{-1}(q,\theta(k)) = (qI - \mathscr{A}_c(\theta(k)))^{-1}\mathscr{b}_c(\theta(k)).$$

Hence

$$\mathscr{c}_c(\theta(k))(qI - \mathscr{A}_c(\theta(k)))^{-1}\mathscr{b}_c(\theta(k)) = \mathscr{c}_c(\theta(k))s(q)e^{-1}(q,\theta(k)) = m(q,\theta(k))e^{-1}(q,\theta(k)), \tag{5.21}$$

where $m(q,\theta(k)) = \mathscr{m}_{n-1}(\theta(k))q^{n-1} + \cdots + \mathscr{m}_1(\theta(k))q + \mathscr{m}_0(\theta(k))$.

To get the equivalent state-space representation and input-output representation, it is clear that the condition has to be satisfied:

$$m(q,\theta(k))e^{-1}(q,\theta(k)) = a^{-1}(q,\theta(k))b(q,\theta(k)). \tag{5.22}$$

This can be done by constructing a GCLD in the form

$$\begin{bmatrix} a(q,\theta(k)) & -b(q,\theta(k)) \end{bmatrix} \begin{bmatrix} p_1(q,\theta(k)) & p_2(q,\theta(k)) \\ p_3(q,\theta(k)) & p_4(q,\theta(k)) \end{bmatrix} = \begin{bmatrix} l(q,\theta(k)) & 0 \end{bmatrix},$$

where $l(q,\theta(k))$ is the GCLD. Then

$$a(q,\theta(k))p_2(q,\theta(k)) = b(q,\theta(k))p_4(q,\theta(k)).$$

Since $p_4(q,\theta(k))$ is invertible, we have that $m(q,\theta(k)) = p_2(q,\theta(k))$ and $e(q,\theta(k)) = p_4(q,\theta(k))$ satisfied the condition (5.22).

Because, $e(q,\theta(k))$ is monic, we can move $x(k+1)$ to everywhere in the space using $x(k)$ and $u(k)$ with any values of $\theta(k)$. Then the system representation in (5.20) is reachable and the result is summarized in the following lemma.

Lemma 5.4. The LPV state-space model (5.20) is a realization of $h(q,\theta(k))$ (5.19) and reachable.

Moreover, since the order n of the reachable form is equal to the maximum order of $e(q,\theta(k))$, the minimal order of the state-space representation is attained when $e(q,\theta(k))$ and $m(q,\theta(k))$ are right coprime, i.e. there exists no right common factor between $e(q,\theta(k))$ and $m(q,\theta(k))$.

Lemma 5.5. The realization of

$$h(q,\theta(k)) = a^{-1}(q,\theta(k))b(q,\theta(k)) = m(q,\theta(k))e^{-1}(q,\theta(k))$$

in reachable form is minimal if and only if $e(q,\theta(k))$ and $m(q,\theta(k))$ are right coprime.

Example 5.3 [*Reachable form*] Consider the same input-output system as in Example 5.2:

$$(q^2 + e^{-k}q + k^2)y(k) = (kq + 1)u(k).$$

Using (5.22), we must find $p_2(q, \theta(k))$ and $p_4(q, \theta(k))$ such that

$$a(q, \theta(k))p_2(q, \theta(k)) = b(q, \theta(k))p_4(q, \theta(k)).$$

In this example we have

$$\begin{aligned} e(q, \theta(k)) &= e_2(\theta(k))q^2 + e_1(\theta(k))q + e_0(\theta(k)) \\ m(q, \theta(k)) &= m_1(\theta(k))q + m_0(\theta(k)) \end{aligned}$$

with

$$\begin{aligned} e_0(\theta(k)) &= \frac{(k^3 - k^2)\left[-k^4 + 7k^3 - 17k^2 + (17k + e^{-(k-2)})k - 7 - 3e^{-(k-2)}\right]}{(k-3)\left[-k^4 + 3k^3 + 2k^2 + e^{-(k-1)}k - 2e^{-(k-1)} - 1\right]} \\ e_1(\theta(k)) &= \frac{(k-1)\left[-e^{-(k-1)}k^4 + (4 + 3e^{-(k-1)})k^2 + (1 + 2e^{-(k-1)} + e^{-k}e^{-(k-1)})k - e^{-k} - 2e^{-k}e^{-(k-1)}\right]}{(k-2)\left[-k^4 - k^3 + k^2 + (1 + e^{-k})k - 1 - e^{-k}\right]} \\ e_2(\theta(k)) &= 1 \\ m_0(\theta(k)) &= \frac{k^2\left[-k^4 + 7k^3 - 17k^2 + (17k + e^{-(k-2)})k - 7 - 3e^{-(k-2)}\right]}{-k^5 + 6k^4 - 11k^3 + (6 + e^{-(k-1)})k^2 + (-1 - 5e^{-(k-1)})k + 3} \\ m_1(\theta(k)) &= 1 \end{aligned}$$

The state-space representation is then

$$\begin{aligned} \begin{bmatrix} x_1(k+1) \\ x_2(k+1) \end{bmatrix} &= \begin{bmatrix} 0 & 1 \\ -e_0(\theta(k)) & -e_1(\theta(k)) \end{bmatrix} \begin{bmatrix} x_1(k) \\ x_2(k) \end{bmatrix} + \begin{bmatrix} 0 \\ 1 \end{bmatrix} u(k) \\ y(k) &= \begin{bmatrix} m_0(\theta(k)) & m_1(\theta(k)) \end{bmatrix} \begin{bmatrix} x_1(k) \\ x_2(k) \end{bmatrix} \end{aligned}$$

It has to be mentioned that in the above example, the system matrices $\mathcal{A}(\theta(k))$ and $c(\theta(k))$ contain dynamic dependence on scheduling parameters even if the original LPV input-output system has static dependence. This happens because of the noncommutative property of the operator q in the transformation process.

5.4 Extension to MIMO Systems

To extend the ideas in the previous section to MIMO systems, all concepts concerning the skew ring have to be extended to matrices.

5.4.1 Skew Polynomial Matrices

A skew polynomial matrix is a matrix with skew polynomial elements in $\mathcal{K}(q)$. The definitions below are adopted from [Ylinen and Zenger, 1992].

Definition 5.6 (Unimodular matrix)**.** A skew polynomial matrix $P(q, \theta(k))$ is *unimodular* if $P^{-1}(q, \theta(k))$ exists and is a skew polynomial matrix.

Definition 5.7 (row (column) equivalent)**.** Two skew polynomial matrices $A(q, \theta(k))$, $B(q, \theta(k))$ are *row(column) equivalent* if there is a unimodular matrix $P(q, \theta(k))$ such that $A(q, \theta(k)) = P(q, \theta(k))B(q, \theta(k))$ $A(q, \theta(k)) = B(q, \theta(k))P(q, \theta(k))$.

Definition 5.8 (greatest common left (right) divisor)**.** A skew polynomial matrix $L(q, \theta(k))$ is a greatest common left divisor (GCLD) of $A(q, \theta(k)), B(q, \theta(k))$ if

$$A(q, \theta(k)) = L(q, \theta(k))A_1(q, \theta(k)), B(q, \theta(k)) = L(q, \theta(k))B_1(q, \theta(k))$$

and if

$$A(q, \theta(k)) = M(q, \theta(k))A_2(q, \theta(k)), B(q, \theta(k)) = M(q, \theta(k))B_2(q, \theta(k)).$$

Then $L(q, \theta(k)) = M(q, \theta(k))L_1(q, \theta(k))$. A *greatest common right divisor* (GCRD) is defined correspondingly.

Definition 5.9. Two skew polynomial matrices $A(q, \theta(k)), B(q, \theta(k))$ are *left (right) coprime* if I is a GCLD (GCRD) of $A(q, \theta(k))$ and $B(q, \theta(k))$, where I denotes the identity matrix with an appropriate dimension.

Elementary Operations

Skew polynomial matrices can be brought to row and column equivalent forms using the *elementary operations*:

i. $T_{ij}^{r(c)}(a(q, \theta(k))) \triangleq$ addition of the j^{th} row (column) multiplied from the left (right) by $a(q, \theta(k))$ to the i^{th} row (column).

ii. $U_{ij}^{r(c)} \triangleq$ interchange of the i^{th} and j^{th} rows (columns)

iii. $V_i(c(q, \theta(k)))^{r(c)} \triangleq$ multiplication of the i^{th} row (column) from the left (right) by $c(q, \theta(k))$.

$V_i(c(q, \theta(k)))$ is unimodular if $c(q, \theta(k))$ is an invertible skew polynomial, i.e. if $c(q, \theta(k)) = c_0$ a nonzero constant value. Then $V_i^{-1}(c(q, \theta(k))) = V_i(c^{-1}(q, \theta(k)))$.

5.4.2 MIMO Observable Form

In Examples 5.3 and 5.2 , it is clear that transforming a system into the state-space observable form is much easier than into the state-space reachable form. From practical point of view the observable form is more preferred because it does not contain any dynamic dependence on the scheduling parameters. Moreover, the transfer operator obtained from the LPV state-space model in observable form is in *Left Matrix Fractional Description* (LMFD)[†] as in the LPV input-output model, i.e. $\tilde{A}^{-1}(q, \theta(k))\tilde{B}(q, \theta(k))$.

[†] Matrix form of LFD form.

For this reason, in MIMO cases, we consider only the observable form.

A MIMO LPV input-output system given by

$$H(q,\theta(k)) = \tilde{A}^{-1}(q,\theta(k))\tilde{B}(q,\theta(k)) \tag{5.23}$$

and

$$\tilde{A}(q,\theta(k)) = \begin{bmatrix} \tilde{A}_{11}(q,\theta(k)) & \tilde{A}_{12}(q,\theta(k)) & \cdots & \tilde{A}_{1n_y}(q,\theta(k)) \\ \vdots & \vdots & \ddots & \vdots \\ \tilde{A}_{n_y1}(q,\theta(k)) & \tilde{A}_{n_y2}(q,\theta(k)) & \cdots & \tilde{A}_{n_yn_y}(q,\theta(k)) \end{bmatrix}$$
$$\tilde{B}(q,\theta(k)) = \begin{bmatrix} \tilde{B}_{11}(q,\theta(k)) & \cdots & \tilde{B}_{1n_u}(q,\theta(k)) \\ \vdots & \ddots & \vdots \\ \tilde{B}_{n_y1}(q,\theta(k)) & \cdots & \tilde{B}_{n_yn_u}(q,\theta(k)) \end{bmatrix},$$

where

$$\tilde{A}_{ij}(q,\theta(k)) = \begin{cases} q^{n_{ai}} + q^{n_{ai}-1}\tilde{a}_{ij,n_{ai}-1}(\theta(k)) + \cdots + q\tilde{a}_{ij,1}(\theta(k)) + \tilde{a}_{ij,0}(\theta(k)) & ,i=j \\ q^{n_{aj}-1}\tilde{a}_{ij,n_{aj}-1}(\theta(k)) + \cdots + q\tilde{a}_{ij,1}(\theta(k)) + \tilde{a}_{ij,0}(\theta(k)) & ,i\neq j \end{cases}$$
$$\tilde{B}_{ij}(q,\theta(k)) = q^{n_{bj}}\tilde{b}_{ij,n_{bj}}(\theta(k)) + q^{n_{bj}-1}\tilde{b}_{ij,n_{bj-1}}(\theta(k)) + \cdots + q\tilde{b}_{ij,1}(\theta(k)) + \tilde{b}_{ij,0}(\theta(k))$$

Using a realization framework for LTI systems [Antsaklis and Michel, 2006], the $\mathcal{A}_o(\theta(k))$, $\mathcal{B}_o(\theta(k))$, and $\mathcal{C}_o(\theta(k))$ matrices of the MIMO state-space model in observable form are given by

$$\begin{aligned} x(k+1) &= \tilde{\mathcal{A}}_o(\theta(k))x(k) + \tilde{\mathcal{B}}_o(\theta(k))u(k), \\ y(k) &= \tilde{\mathcal{C}}_o(\theta(k))x(k), \end{aligned} \tag{5.24}$$

and

$$\tilde{\mathcal{A}}_o(\theta(k)) = \bar{A}_o + \mathcal{A}_p(\theta(k))\bar{C}_o, \qquad \tilde{\mathcal{C}}_o(\theta(k)) = \tilde{\mathcal{C}}_p(\theta(k))\bar{C}_o, \tag{5.25}$$

where $\bar{A}_o = $ block diag$[A_1, A_2, \ldots, A_{n_y}]$ with

$$A_i = \begin{bmatrix} 0 & \cdots & 0 & 0 \\ & & & 0 \\ & I_{n_{ai}-1} & & \vdots \\ & & & 0 \end{bmatrix} \in \mathbb{R}^{n_{ai}\times n_{ai}},$$
$$\bar{C}_0 = \text{ block diag}([0,\ldots,0,1] \in \mathbb{R}^{1\times n_{ai}}, i = 1,\ldots,n_y),$$

where $\tilde{\mathcal{B}}_o(\theta(k))$ is the matrix of the coefficients of the lower row degree terms of $\tilde{B}(q,\theta(k))$ corresponding to the highest row degree coefficient matrix of $\tilde{A}(q,\theta(k))$. The matrices $\tilde{\mathcal{A}}_p(\theta(k))$, $\tilde{\mathcal{C}}_p(\theta(k))$ are defined by

$$\tilde{\mathcal{C}}_p(\theta(k)) = \mathcal{A}_h^{-1}(\theta(k)), \qquad \tilde{\mathcal{A}}_p(\theta(k)) = -\mathcal{A}_l(\theta(k))\mathcal{A}_h^{-1}(\theta(k)) \tag{5.26}$$

where $\mathcal{A}_h(\theta(k))$ is the highest row degree coefficient matrix of $\tilde{A}(q,\theta(k))$ and $\mathcal{A}_l(\theta(k))$ is the matrix of coefficients of the lower row degree terms of $\tilde{A}(q,\theta(k))$.

Example 5.4 [*Row degree coefficient matrix*] If

$$\tilde{A}(q,\theta(k)) = \begin{bmatrix} q^2(3\theta(k)) + \theta(k) & q(2\theta(k)) \\ q(2\theta(k)) & q \end{bmatrix}, \quad \tilde{B}(q,\theta(k)) = \begin{bmatrix} q(2\theta(k)) & \theta(k) \\ 3\theta(k) & 1 \end{bmatrix}$$

then we have

$$\mathscr{A}_h(\theta(k)) = \begin{bmatrix} 3\theta(k) & 0 \\ 2\theta(k) & 1 \end{bmatrix}, \qquad \mathscr{A}_l^T(\theta(k)) = \begin{bmatrix} \theta(k) & 0 & 0 \\ 0 & 2\theta(k) & 0 \end{bmatrix},$$

and

$$\tilde{\mathscr{B}}_o^T(\theta(k)) = \begin{bmatrix} 0 & 2\theta(k) & 3\theta(k) \\ \theta(k) & 0 & 1 \end{bmatrix}.$$

■

To see that the matrices defined above are the observable realization matrices of the system (5.23), one can start by defining

$$\breve{S}(q) = \text{ block diag}\{[1, q, \ldots, q^{n_{ai}-1}], \quad i = 1, \ldots, n_y\}, \tag{5.27}$$

and let $q^{\bar{n}_{ai}}$ be the row ith highest degree of $\tilde{A}(q,\theta(k))$. Next define

$$\tilde{\Lambda}(q) = \text{diag}\{q^{\bar{n}_{a1}}, q^{\bar{n}_{a2}}, \ldots, q^{\bar{n}_{an_y}}\}. \tag{5.28}$$

With $\breve{S}(q)$ and $\tilde{\Lambda}(q)$, we then have

$$\tilde{A}(q,\theta(k)) = \tilde{\Lambda}(q)\mathscr{A}_h(\theta(k)) + \breve{S}(q)\mathscr{A}_l(\theta(k)) \tag{5.29}$$

and

$$\tilde{B}(q,\theta(k)) = \breve{S}(q)\tilde{\mathscr{B}}_o(\theta(k)). \tag{5.30}$$

By multiplying $\breve{S}(q)$ from the right with $\left[qI - \tilde{\mathscr{A}}_o(\theta(k))\right]$, we obtain

$$\begin{aligned} \breve{S}(q)\left[qI - \tilde{\mathscr{A}}_o(\theta(k))\right] &= \breve{S}(q)\left[qI - \bar{A}_o\right] - \breve{S}(q)\mathscr{A}_p(\theta(k))\bar{C}_o \\ &= \left[\tilde{\Lambda}(q) - \breve{S}(q)\mathscr{A}_p(\theta(k))\right]\bar{C}_o \\ &= \left[\tilde{\Lambda}(q) + \breve{S}(q)\mathscr{A}_l(\theta(k))\mathscr{A}_h^{-1}(\theta(k))\right]\bar{C}_o \\ &= \left[\tilde{\Lambda}(q)\mathscr{A}_h(\theta(k)) + \breve{S}(q)\mathscr{A}_l(\theta(k))\right]\mathscr{A}_h^{-1}(\theta(k))\bar{C}_o \\ &= \tilde{A}(q,\theta(k))\mathscr{A}_h^{-1}(\theta(k))\bar{C}_o = \tilde{A}(q,\theta(k))\tilde{\mathcal{C}}_o(\theta(k)). \end{aligned}$$

The last line comes from the fact that $\tilde{\mathcal{C}}_o(\theta(k)) = \mathcal{A}_h^{-1}(\theta(k))\bar{C}_o$.

Rewriting the above equation, we have

$$\tilde{A}^{-1}(q,\theta(k))\breve{S}(q) = \tilde{\mathcal{C}}_o(\theta(k))\left[qI - \tilde{\mathcal{A}}_o(\theta(k))\right]^{-1}. \tag{5.31}$$

Multiply $\tilde{\mathcal{B}}(\theta(k))$ from the right of both sides to obtain

$$\tilde{A}^{-1}(q,\theta(k))\tilde{B}_o(q,\theta(k)) = \tilde{\mathcal{C}}_o(\theta(k))\left[qI - \tilde{\mathcal{A}}_o(\theta(k))\right]^{-1}\tilde{\mathcal{B}}(\theta(k)). \tag{5.32}$$

The equation (5.32) shows the equivalence of a MIMO LPV input-output representation and the state-space realization in observable form. One can summarize the above result in the following lemma.

Lemma 5.6. The system $\{\tilde{\mathcal{A}}_o(\theta(k)), \tilde{\mathcal{B}}_o(\theta(k)), \tilde{\mathcal{C}}_o(\theta(k)), \tilde{\mathcal{D}}_o(\theta(k))\}$ defined above is an n_a $(= \sum_{i=1}^{n_y} n_{ai})$-th-order observable realization of $H(q,\theta(k))$ with $(\mathcal{A}_o(\theta(k)), \mathcal{C}_o(\theta(k)))$ in observable form.

The following example is adapted from an LTI example in [Kailath, 1980, pp. 416].

Example 5.5 [*MIMO Observable form*] Suppose we have an LPV input-output model in LPR form described in term of a transfer operator

$$\tilde{H}(q,\theta(k)) = \tilde{A}^{-1}(q,\theta(k))\tilde{B}(q,\theta(k)),$$

where

$$\begin{aligned}
\tilde{A}(q,\theta(k)) &= \begin{bmatrix} q^3 + q^2(4\theta(k)) + q(5\theta(k)) + (2\theta(k)) & q + (2\theta(k)) \\ 0 & q^2 + q(4\theta(k)) + (4\theta(k)) \end{bmatrix} \\
\tilde{B}(q,\theta(k)) &= \begin{bmatrix} 0 & -q^2 \\ -q & -q \end{bmatrix}.
\end{aligned}$$

We can see that $n_{a1} = 3$ and $n_{a2} = 2$. Then the order of the state-space model is 5. The highest-row-degree coefficient matrix is

$$\mathcal{A}_h(\theta(k)) = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} \text{ and } \mathcal{A}_h^{-1}(\theta(k)) = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}$$

while

$$\begin{aligned}
\tilde{\Lambda}(q) &= \begin{bmatrix} q^3 & 0 \\ 0 & q^2 \end{bmatrix}, \qquad \breve{S}(q) = \left[\begin{array}{ccc:cc} 1 & q & q^2 & 0 & 0 \\ \hdashline 0 & 0 & 0 & 1 & q \end{array}\right], \\
\mathcal{A}_l^T(\theta(k)) &= \left[\begin{array}{ccc:cc} 2\theta(k) & 5\theta(k) & 4\theta(k) & 0 & 0 \\ \hdashline 2\theta(k) & 1 & 0 & 4\theta(k) & 4\theta(k) \end{array}\right], \\
\tilde{\mathcal{B}}_o^T(\theta(k)) &= \left[\begin{array}{ccc:cc} 0 & 0 & 0 & 0 & -1 \\ \hdashline 0 & 0 & -1 & 0 & -1 \end{array}\right].
\end{aligned}$$

Hence

$$\tilde{\mathscr{A}}_p^T(\theta(k)) = [-\mathscr{A}_l(\theta(k))\mathscr{A}_h^{-1}(\theta(k))]^T = \left[\begin{array}{ccc|cc} -2\theta(k) & -5\theta(k) & -4\theta(k) & 0 & 0 \\ \hline -2\theta(k) & -1 & 0 & -4\theta(k) & -4\theta(k) \end{array}\right].$$

Using the above construction, we obtain

$$\tilde{\mathscr{A}}_o(\theta(k)) = \left[\begin{array}{ccc|cc} 0 & 0 & -2\theta(k) & 0 & -2\theta(k) \\ 1 & 0 & -5\theta(k) & 0 & -1 \\ 0 & 1 & -2\theta(k) & 0 & 0 \\ \hline 0 & 0 & 0 & 0 & -4\theta(k) \\ 0 & 0 & 0 & 1 & -4\theta(k) \end{array}\right], \qquad \tilde{\mathscr{B}}_o(\theta(k)) = \left[\begin{array}{c|c} 0 & 0 \\ 0 & 0 \\ 0 & -1 \\ \hline 0 & 0 \\ -1 & -1 \end{array}\right]$$

$$\tilde{\mathscr{C}}_o(\theta(k)) = \left[\begin{array}{ccc|cc} 0 & 0 & 1 & 0 & 0 \\ \hline 0 & 0 & 0 & 0 & 1 \end{array}\right].$$

■

As in the SISO case, the minimal order of the state-space realization occurs when $\tilde{A}(q,\theta(k))$ and $\tilde{B}(q,\theta(k))$ are left coprime.

5.4.3 Minimal Realization

Similar to an LTI case discussed in [Antsaklis and Michel, 2006], if $\tilde{A}(q,\theta(k))$ and $\tilde{B}(q,\theta(k))$ are not left coprime, the minimal realization cannot be guaranteed. In order to obtain a minimal realization the GCLD has to be extracted from $\tilde{A}(q,\theta(k))$ and $\tilde{B}(q,\theta(k))$. If $\tilde{A}(q,\theta(k))$ and $\tilde{B}(q,\theta(k))$ are not left coprime, they can be considered as

$$\tilde{A}(q,\theta(k)) = L(q,\theta(k))\tilde{A}^*(q,\theta(k))$$

and

$$\tilde{B}(q,\theta(k)) = L(q,\theta(k))\tilde{B}^*(q,\theta(k)),$$

where $L(q,\theta(k))$ is a GCLD of $\tilde{A}(q,\theta(k))$ and $\tilde{B}(q,\theta(k))$.

Then we have

$$\begin{aligned} H(q,\theta(k)) &= (L(q,\theta(k))\tilde{A}^*(q,\theta(k)))^{-1}(L(q,\theta(k))\tilde{B}^*(q,\theta(k))) \\ &= \tilde{A}^{*^{-1}}(q,\theta(k))\tilde{B}^*(q,\theta(k)) \end{aligned}$$

and the degree $n = \sum_i^{n_y} n_{ai}$ of $\tilde{A}^*(q,\theta(k))$ is definitely less than or equal to that of $\tilde{A}(q,\theta(k))$ and the realization is minimal.

The GCLD of two skew polynomial matrices $\tilde{A}(q,\theta(k))$ and $\tilde{B}(q,\theta(k))$ can be obtained using row and column operations to transform the matrix $\left[\tilde{A}(q,\theta(k)) \quad \tilde{B}(q,\theta(k))\right]$ into *Hermite form* or *Jacobson form* – the time-varying version of the *Smith form* [Zerz, 2007] – and using the following lemma which is extended from the LTI version in [Antsaklis and Michel, 2006].

Lemma 5.7. Let $P_1(q,\theta(k)) \in \mathcal{K}(q)^{m\times n_1}$ and $P_2(q,\theta(k)) \in \mathcal{K}(q)^{m\times n_2}$ with $n_1+n_2 \geq m$. Let the unimodular $U(q,\theta(k))$ be such that

$$\begin{bmatrix} P_1(q,\theta(k)) & P_2(q,\theta(k)) \end{bmatrix} U(q,\theta(k)) = \begin{bmatrix} L(q,\theta(k)) & 0 \end{bmatrix}$$

Then $L(q,\theta(k))$ is a GCLD of $P_1(q,\theta(k))$, $P_2(q,\theta(k))$. Moreover if $L(q,\theta(k))$ is a unimodular matrix, then $P_1(q,\theta(k))$ and $P_2(q,\theta(k))$ are left coprime.

Proof. The proof is similar to in the LTI case [Antsaklis and Michel, 2006] and therefore omitted. □

The following example shows how to use the above algorithm to reduce the maximum order of the polynomials in the elements of the system matrices.

Example 5.6 [*Minimal Realization*] Consider an LPV input-output system described by the following two matrices

$$\tilde{A}(q,\theta(k)) = \begin{bmatrix} q+\theta(k) & -1 \\ -\theta(k+1)q - \theta(k+1)\theta(k) - \theta(k) & q+1+\theta(k+1) \end{bmatrix}, \quad \tilde{B}(q,\theta(k)) = \begin{bmatrix} -1 \\ 0 \end{bmatrix},$$

then

$$\begin{bmatrix} \tilde{A}(q,\theta(k)) & \tilde{B}(q,\theta(k)) \end{bmatrix} = \begin{bmatrix} q+\theta(k) & -1 & -1 \\ -\theta(k+1)q - \theta(k+1)\theta(k) - \theta(k)\frac{1}{k} & q+1+\theta(k+1) & 0 \end{bmatrix}.$$

Using column operations, the Jacobson form is

$$\begin{bmatrix} \tilde{A}(q,\theta(k)) & \tilde{B}(q,\theta(k)) \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & q+1+\theta(k+1) & 0 \end{bmatrix}.$$

Hence a GCLD is

$$L(q,\theta(k)) = \begin{bmatrix} 1 & 0 \\ 0 & q+1+\theta(k+1) \end{bmatrix},$$

and a unimodular matrix which can be used to transform the original above matrix to the Jacobson form is

$$U(q,\theta(k)) = \begin{bmatrix} 0 & 0 & 1 \\ 0 & 1 & \theta(k) \\ -1 & -1 & q \end{bmatrix}.$$

Left coprime matrices $\tilde{A}^*(q,\theta(k))$, $\tilde{B}^*(q,\theta(k))$ can be found from the corresponding rows of

$$U^{-1}(q,\theta(k)) = \begin{bmatrix} q+\theta(k) & -1 & -1 \\ -\theta(k) & 1 & 0 \\ 1 & 0 & 0 \end{bmatrix}$$

and finally we have

$$\tilde{A}^*(q, \theta(k)) = \begin{bmatrix} q + \theta(k) & -1 \\ -\theta(k) & 1 \end{bmatrix}, \qquad \tilde{B}^*(q, \theta(k)) = \begin{bmatrix} -1 \\ 0 \end{bmatrix}.$$

It is clear that the maximum order of the polynomial in the second row of $\tilde{A}(q, \theta(k))$ is reduced by one. If we transform this system by using $\tilde{A}^*(q, \theta(k))$ and $\tilde{B}^*(q, \theta(k))$ into state-space form, the order of the system according to the second row output will be less than using the original input-output system matrices.

■

From the above simple example, it seems that the procedure in Section 5.4.3 can be implemented practically. However, in practice, the higher the maximum row degree, the more complicated the unimodular matrix. This happens even when using a symbolic computing program like `Maple`. The unimodular matrix which is used to extract the left coprime matrix of $\tilde{A}(q, \theta(k))$ and $\tilde{B}(q, \theta(k))$ is too intricate when the maximum row degree of $\tilde{A}(q, \theta(k))$ is greater than 1.

Fortunately, the realization method described in the previous section can be applied to the identification input-output models. If the identifiability of the input-output model is given, the left coprimeness of $\tilde{A}(q, \theta(k))$ and $\tilde{B}(q, \theta(k))$ is automatically guaranteed [Ljung, 1999].

5.5 Application to LPV Input-Output System Identification

In the LTI case, most of input-output model structures are constructed using polynomials in the RPR form. This is also widely used for LPV systems, e.g. [Bamieh and Giarré, 2002; Wei, 2006]. With this configuration, the realization always contains dynamic parameter dependence in the LPV state-space model. This issue has been investigated by Tóth et al. [2007]. The problem is revisited by Tóth et al. [2011], where three types of input-output model structures for identification are proposed to solve the problem. Here we extend the idea by considering the structure of the so called *shifted form* in [Abbas et al., 2010b], which is equivalent to the LPR in the previous section. The left- and right representations are preferred since they can be used to categorize representations better than the shifted form. By using the skew polynomial concept and the realization procedures described before, we can develop a framework for state-space realization based on LPR structure, and the equivalence between a state-space realization and a input-output representation can be shown. Furthermore, if the order of the skew polynomials in $A(q, \theta(k))$ and $B(q, \theta(k))$ follow the criteria of identifiability given in [Ljung, 1999], one can expect that there are no common factors between $A(q, \theta(k))$ and $B(q, \theta(k))$, i.e. $A(q, \theta(k))$ and $B(q, \theta(k))$ are left coprime. Therefore, an observable form is always minimal.

5.5.1 Identifiability of Model Structures

In this section we discuss the equivalence of the left coprimeness of $\tilde{A}(q, \theta(k))$ and $\tilde{B}(q, \theta(k))$ and the identifiability concept. In the following, $G(q, \theta(k), \mathfrak{P})$ denotes the transfer operator of the plant and $H(q, \theta(k), \mathfrak{P})$ denotes the transfer operator of the noise model. The following definition is general for all SISO and MIMO systems.

Definition 5.10 (Identifiability [Gevers et al., 2009; Ljung, 1999]). A model structure $\{G(q, \theta(k), \mathfrak{P}), H(q, \theta(k), \mathfrak{P})\}$ with a parameter domain $\boldsymbol{\mathfrak{P}} \subset \mathbb{R}^{n_{\mathfrak{P}}}$ is called locally identifiable at a parameter value $\mathfrak{P}_1 \in \boldsymbol{\mathfrak{P}}$, if $\exists \delta > 0$ such that for all $\mathfrak{P} \in \boldsymbol{\mathfrak{P}}$ with $\|\mathfrak{P} - \mathfrak{P}_1\| \leq \delta$, the corresponding one-step-ahead predictors are distinguishable:

$$H(q, \theta(k), \mathfrak{P}) = H(q, \theta(k), \mathfrak{P}_1) \tag{5.33}$$

and

$$\begin{aligned} H^{-1}(q, \theta(k), \mathfrak{P}) G(q, \theta(k), \mathfrak{P}) &= H^{-1}(q, \theta(k), \mathfrak{P}_1) G(q, \theta(k), \mathfrak{P}_1) \\ &\Rightarrow \mathfrak{P} = \mathfrak{P}_1. \end{aligned} \tag{5.34}$$

The model structure $\{G(q, \theta(k), \mathfrak{P}), H(q, \theta(k), \mathfrak{P})\}$ is called globally identifiable at $\mathfrak{P}_1$ if it is locally identifiable at $\mathfrak{P}$ with $\delta \to \infty$. Moreover,

$$\{G(q, \theta(k), \mathfrak{P}), H(q, \theta(k), \mathfrak{P})\}$$

is called globally identifiable if it is globally identifiable at all $\mathfrak{P} \in \boldsymbol{\mathfrak{P}}$.

For an LPV-ARX model structure, we have

$$G(q, \theta(k), \mathfrak{P}) = A^{-1}(q, \theta(k), \mathfrak{P}) B(q, \theta(k), \mathfrak{P}), \qquad H(q, \theta(k), \mathfrak{P}) = A^{-1}(q, \theta(k), \mathfrak{P}).$$

In this case, the equality of $H(q, \theta(k), \mathfrak{P})$ and $H(q, \theta(k), \mathfrak{P}_1)$ implies that $A(q, \theta(k), \mathfrak{P})$ and $A(q, \theta(k), \mathfrak{P}_1)$ are equivalent and $B(q, \theta(k), \mathfrak{P})$ and $B(q, \theta(k), \mathfrak{P}_1)$ must coincide to make $G(q, \theta(k), \mathfrak{P})$ equal. This means that (5.33) holds for all $\mathfrak{P}_1$ in the LPV-ARX model structure. Moreover, since a common factor between $A(q, \theta(k), \mathfrak{P})$ and $B(q, \theta(k), \mathfrak{P})$ must not exist [Ljung, 1999], the left coprimeness of $A(q, \theta(k), \mathfrak{P})$ and $B(q, \theta(k), \mathfrak{P})$ is obtained.

Other model structures, e.g. LPV-OE, LPV-BJ, are not always identifiable. To satisfy the identifiability condition, the orders of all polynomials inside the model matrices have to be selected with care, see [Ljung, 1999].

5.5.2 LPV Input-Output Models Identification in Left Polynomial Representation

SISO Case

An LPV-ARX input-output model in LPR form is described as

$$y(k) + \sum_{i=1}^{n_a} q^{-i}\tilde{a}_{n_a-i}(\theta(k))y(k) = \sum_{j=0}^{n_b} q^{-(d+j)}\tilde{b}_{n_b-j}(\theta(k))u(k), \tag{5.35}$$

where $\tilde{a}_i(\theta(k))$ and $\tilde{b}_i(\theta(k))$ are defined on $\mathcal{K}$, and $n_a \geq n_b + d$.

Without loss of generality assume $d = 1$ and $n_a = n_b + d$. Multiplying both sides from the left with q^{n_a}, we obtain

$$q^{n_a}y(k) + \sum_{i=1}^{n_a} q^{n_a-i}\tilde{a}_{n_a-i}(\theta(k))y(k) = \sum_{j=0}^{n_a-1} q^{n_a-(j+1)}\tilde{b}_{n_a-1-j}(\theta(k))u(k),$$

hence

$$\begin{aligned}\left[q^{n_a}y(k) + q^{n_a-1}\tilde{a}_{n_a-1}(\theta(k)) + \cdots + q\tilde{a}_1(\theta(k)) + \tilde{a}_0(\theta(k))\right] y(k) = \\ \left[q^{n_a-1}\tilde{b}_{n_a-1}(\theta(k)) + q^{n_a-2}\tilde{b}_{n_a-2}(\theta(k)) + \cdots + q\tilde{b}_1(\theta(k)) + q\tilde{b}_0(\theta(k))\right],\end{aligned}$$

and

$$\tilde{a}(q,\theta(k))y(k) = \tilde{b}(q,\theta(k))u(k).$$

It is clear that $\tilde{b}_{n_a}(\theta(k)) = 0$ since the system is strictly proper. Moreover there is no common factor between $\tilde{a}(q,\theta(k))$ and $\tilde{b}(q,\theta(k))$ since we identify it in LPV-ARX structure. Now using a procedure described in Section 5.3.1 we can obtain a state-space LPV representation in observable form as

$$\begin{aligned} x_k(k+1) &= \begin{bmatrix} 0 & \cdots & 0 & -\tilde{a}_0(\theta(k)) \\ 1 & \cdots & 0 & -\tilde{a}_1(\theta(k)) \\ \vdots & \ddots & \vdots & \vdots \\ 0 & \cdots & 1 & -\tilde{a}_{n_a-1}(\theta(k)) \end{bmatrix} x(k) + \begin{bmatrix} \tilde{b}_0(\theta(k)) \\ \tilde{b}_1(\theta(k)) \\ \vdots \\ \tilde{b}_{n_a-1}(\theta(k)) \end{bmatrix} u(k) \\ y(k) &= \begin{bmatrix} 0 & 0 & \cdots & 0 & 1 \end{bmatrix} x(k). \end{aligned} \tag{5.36}$$

and this state-space representation is minimal.

MIMO case

An LPV-ARX input-output model in LPR form is described as

$$\begin{aligned} y_1(k) = &-\sum_{i=1}^{n_{a1}} q^{-i}\tilde{\mathfrak{a}}_{1,n_{a1}-i}(\theta(k))y_1(k) - \sum_{i=1}^{n_{a2}} q^{-i}\tilde{\mathfrak{a}}_{1,n_{a2}-i}(\theta(k))y_2(k) - \cdots \\ &- \sum_{i=1}^{n_{any}} q^{-i}\tilde{\mathfrak{a}}_{1,n_{any}-i}(\theta(k))y_{n_y}(k) + \sum_{j=0}^{n_{b1}} q^{-(d+j)}\tilde{\mathfrak{b}}_{1,n_{a1}}(\theta(k))u_1(k) + \cdots \\ &+ \sum_{j=0}^{n_{bny}} q^{-(d+j)}\tilde{\mathfrak{b}}_{1,n_{bny}}(\theta(k))u_{n_u}(k) \\ &\vdots \\ y_{n_y}(k) = &-\sum_{i=1}^{n_{a1}} q^{n_{a1}-i}\tilde{\mathfrak{a}}_{n_y,n_{a1}-i}(\theta(k))y_1(k) - \sum_{i=1}^{n_{a2}} q^{n_{a2}-i}\tilde{\mathfrak{a}}_{n_y,n_{a2}-i}(\theta(k))y_2(k) - \cdots \\ &- \sum_{i=1}^{n_{any}} q^{n_{any}-i}\tilde{\mathfrak{a}}_{n_y,n_{any}-i}(\theta(k))y_{n_y}(k) + \sum_{j=0}^{n_{b1}} q^{n_{b1}}\tilde{\mathfrak{b}}_{n_y,n_{a1}}(\theta(k))u_1(k) + \cdots \\ &+ \sum_{j=0}^{n_{bny}} q^{n_{bny}}\tilde{\mathfrak{b}}_{n_y,n_{bny}}(\theta(k))u_{n_u}(k), \end{aligned} \tag{5.37}$$

where $\tilde{\mathfrak{a}}_i(\theta(k))$ and $\tilde{\mathfrak{b}}_i(\theta(k))$ are defined on $\mathcal{K}$, and $n_{ai} \geq n_{bi} + d$.

Again without lost of generality, assume $d = 1$, $n_{a_i} = n_a$, $n_{b_i} = n_b$ and $n_a = n_b + d$. Rewrite (5.37) in matrix form and multiply both sides with

$$\tilde{\Lambda}(q) = \begin{bmatrix} q^{n_{a1}} & \cdots & 0 \\ 0 & \ddots & 0 \\ 0 & \cdots & q^{n_{any}} \end{bmatrix}.$$

Then we have

$$\tilde{H}(q,\theta(k)) = \tilde{A}^{-1}(q,\theta(k))\tilde{B}(q,\theta(k)), \tag{5.38}$$

where

$$\tilde{A}(q,\theta(k)) = \begin{bmatrix} \tilde{A}_{11}(q,\theta(k)) & \tilde{A}_{12}(q,\theta(k)) & \cdots & \tilde{A}_{1n_y}(q,\theta(k)) \\ \vdots & \vdots & \ddots & \vdots \\ \tilde{A}_{n_y1}(q,\theta(k)) & \tilde{A}_{n_y2}(q,\theta(k)) & \cdots & \tilde{A}_{n_yn_y}(q,\theta(k)) \end{bmatrix}$$

$$\tilde{B}(q,\theta(k)) = \begin{bmatrix} \tilde{B}_{11}(q,\theta(k)) & \cdots & \tilde{B}_{1n_u}(q,\theta(k)) \\ \vdots & \ddots & \vdots \\ \tilde{B}_{n_y1}(q,\theta(k)) & \cdots & \tilde{B}_{n_yn_u}(q,\theta(k)) \end{bmatrix},$$

and

$$\tilde{A}_{ij}(q,\theta(k)) = \begin{cases} q^{n_{ai}} + q^{n_{ai}-1}\tilde{\mathscr{a}}_{ij,n_{ai}-1}(\theta(k)) + \cdots + q\tilde{\mathscr{a}}_{ij,1}(\theta(k)) + \tilde{\mathscr{a}}_{ij,0}(\theta(k)) & ,i=j \\ q^{n_{aj}-1}\tilde{\mathscr{a}}_{ij,n_{aj}-1}(\theta(k)) + \cdots + q\tilde{\mathscr{a}}_{ij,1}(\theta(k)) + \tilde{\mathscr{a}}_{ij,0}(\theta(k)) & ,i\neq j \end{cases}$$
$$\tilde{B}_{ij}(q,\theta(k)) = q^{n_{bj}}\tilde{\mathscr{b}}_{ij,n_{bj}}(\theta(k)) + q^{n_{bj}-1}\tilde{\mathscr{b}}_{ij,n_{bj-1}}(\theta(k)) + \cdots + q\tilde{\mathscr{b}}_{ij,1}(\theta(k)) + \tilde{\mathscr{b}}_{ij,0}(\theta(k)).$$

With the method in Section 5.4, we can transform the LPV input-output model into a state-space form that is minimal, i.e. $\tilde{A}(q,\theta(k))$ and $\tilde{B}(q,\theta(k))$ are left coprime.

Example 5.7 [*MIMO Realization*] Consider an input-output model identified in LPR form

$$\begin{aligned} y_1(k) &= q^{-1}\tilde{\mathscr{a}}_{111}(\theta(k))y_1(k) + q^{-2}\tilde{\mathscr{a}}_{112}(\theta(k))y_1(k) + q^{-1}\tilde{\mathscr{a}}_{121}(\theta(k))y_2(k) + q^{-2}\tilde{\mathscr{a}}_{122}(\theta(k))y_2(k) \\ &\qquad + q^{-1}\tilde{\mathscr{b}}_{11}(\theta(k))u_1(k) + q^{-2}\tilde{\mathscr{b}}_{12}(\theta(k))u_2(k) \\ y_2(k) &= q^{-1}\tilde{\mathscr{a}}_{211}(\theta(k))y_1(k) + q^{-1}\tilde{\mathscr{a}}_{221}(\theta(k))y_2(k) + q^{-1}\tilde{\mathscr{b}}_{21}(\theta(k))u_1(k). \end{aligned}$$

Rewrite the above system in matrix form and multiply both sides with

$$\tilde{\Lambda}(q) = \begin{bmatrix} q^2 & 0 \\ 0 & q \end{bmatrix}.$$

We have

$$\begin{aligned} &\begin{bmatrix} q^2 - q\tilde{\mathscr{a}}_{111}(\theta(k)) - \tilde{\mathscr{a}}_{112}(\theta(k)) & -q\tilde{\mathscr{a}}_{121}(\theta(k)) - \tilde{\mathscr{a}}_{122}(\theta(k)) \\ -\tilde{\mathscr{a}}_{211}(\theta(k)) & q - \tilde{\mathscr{a}}_{221}(\theta(k)) \end{bmatrix} \begin{bmatrix} y_1(k) \\ y_2(k) \end{bmatrix} \\ &\qquad\qquad = \begin{bmatrix} q\tilde{\mathscr{b}}_{11}(\theta(k)) & \tilde{\mathscr{b}}_{12}(\theta(k)) \\ \tilde{\mathscr{b}}_{21}(\theta(k)) & 0 \end{bmatrix} \begin{bmatrix} u_1(k) \\ u_2(k) \end{bmatrix}. \end{aligned}$$

With,

$$\mathscr{A}_h(\theta(k)) = \mathscr{A}_h^{-1}(\theta(k)) = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}$$

and,

$$\breve{S}(q) = \left[\begin{array}{cc:c} 1 & q & 0 \\ \hdashline 0 & 0 & 1 \end{array}\right], \qquad \mathscr{A}_l^T(\theta(k)) = \left[\begin{array}{cc:c} -\tilde{\mathscr{a}}_{112}(\theta(k)) & -\tilde{\mathscr{a}}_{111}(\theta(k)) & -\tilde{\mathscr{a}}_{211}(\theta(k)) \\ \hdashline -\tilde{\mathscr{a}}_{122}(\theta(k)) & -\tilde{\mathscr{a}}_{121}(\theta(k)) & -\tilde{\mathscr{a}}_{221}(\theta(k)) \end{array}\right]$$
$$\tilde{\mathscr{B}}_o^T(\theta(k)) = \left[\begin{array}{cc:c} 0 & \tilde{\mathscr{b}}_{11}(\theta(k)) & \tilde{\mathscr{b}}_{21}(\theta(k)) \\ \hdashline \tilde{\mathscr{b}}_{21}(\theta(k)) & 0 & 0 \end{array}\right], \qquad \tilde{\mathscr{C}}_o(\theta(k)) = \left[\begin{array}{cc:c} 0 & 1 & 0 \\ \hdashline 0 & 0 & 1 \end{array}\right]$$

hence

$$\tilde{\mathscr{A}}_p^T(\theta(k)) = [-\mathscr{A}_l(\theta(k))\mathscr{A}_h^{-1}(\theta(k))]^T = -\mathscr{A}_l^T(\theta(k))$$

and using the above construction, we get

$$\tilde{\mathcal{A}}_o(\theta(k)) = \left[\begin{array}{cc|c} 0 & -\tilde{a}_{112}(\theta(k)) & -\tilde{a}_{122}(\theta(k)) \\ 1 & -\tilde{a}_{111}(\theta(k)) & -\tilde{a}_{121}(\theta(k)) \\ \hline 0 & -\tilde{a}_{211}(\theta(k)) & -\tilde{a}_{221}(\theta(k)) \end{array}\right], \qquad \tilde{B}_o(\theta(k)) = \left[\begin{array}{c|c} 0 & \tilde{b}_{21}(\theta(k)) \\ \tilde{b}_{11}(\theta(k)) & 0 \\ \hline \tilde{b}_{21}(\theta(k)) & 0 \end{array}\right]$$

$$\tilde{\mathcal{C}}_o(\theta(k)) = \left[\begin{array}{cc|c} 0 & 1 & 0 \\ \hline 0 & 0 & 1 \end{array}\right].$$

■

It should also be mentioned that, by doing the identification with all polynomials in LPR structure and using the proposed realization algorithm, the resulting state-space model will not contain any dynamic dependence. This is a clear benefit of this method because the more dynamically dependent parameters the state-space model contains, the more scheduling parameters have to be defined [Kwiatkowski et al., 2006b], i.e. $\theta(k)$ and $\theta(k+1)$ are considered as different parameters, and the model is not suitable for LPV controller synthesis methods.

5.6 Conclusion

The noncommutativity issue of the shift operator acting on the scheduling signal plays an important role in the realization theory of LPV systems. Ignoring this property, the resulting state-space model may not be equivalent to the original LPV input-output model. With the help of skew polynomials, this problem can be solved by separating the LPV input-output structure into RPR and LPR. We show that with a slight modification, we can use a LTI realization procedure to transform an LPV input-output representation into reachable or observable state-space form, without violating the commutative rule. Furthermore, using the LPV identification in LPR form with an identifiable model structure like LPV-ARX, a minimal state-space model in observable form is obtained that does not contain any dynamic dependence. The proposed algorithm can be used for both SISO and MIMO systems.

Finally, the identification of LPV-ARX models with polynomial in LPR structure and the observable state-space model construction are systematic. This framework has been developed as a Malab toolbox and it is used in experimental applications will be shown in the next chapter.

Chapter 6

Identification for LPV Control of Nonlinear Plants – Applications

The identification of global LPV input-output models has been introduced by Bamieh and Giarré [2002]. In spite of the advantages of this method, there are until now only very few reports of applications to controller design. Practical LPV control design typically relies on different techniques: e.g. a neural network based LPV state-space model applied to an arm-driven inverted pendulum plant [Lachhab et al., 2008], interpolation of the frequency response of local linear models applied to wafer steppers [Steinbuch et al., 2003; Wassink et al., 2005], grey-box identification applied to the boost pressure of a diesel engine [Wei and del Re, 2007], etc. One reason for the rare application of LPV input-output identification was revealed by [Tóth et al., 2007]: the conversion from an LPV input-output model to an LPV state-space model may lead to an incorrect model.

This chapter shows that the identification and realization techniques proposed in Chapter 3 and Chapter 5 can be used in real applications. For this purpose, a Matlab toolbox LIDT was developed [‡]. The toolbox is applied to two experimental plants: a magnetic levitation system [Parks, 1999] and the arm driven-inverted pendulum (ADIP) [Quanser Consulting Inc., 1993] considered in Chapter 3. Both models are identified in the quasi-LPV input-output structure with all polynomials in LPR form. The models are then transformed into LPV state-space models in observable form, which is then used for LPV controller design. The LPV controller design method used in this chapter is a technique based on dilated *Linear Matrix Inequalities* (LMIs) described for LTI systems in [de Oliveira et al., 2002] and for LPV systems in [Ali and Werner, 2011].

The magnetic levitation plant is operated at an open-loop stable equilibrium. Therefore the open-loop identification scheme can be used. The obtained model can be directly used for designing an LPV controller. On the other hand, the ADIP system is open-loop unstable as explained in Chapter 3. The closed-loop identification method has to be used. Closed-loop identification with the objective of designing a controller

---

[‡] LIDT stands for *Left polynomial representation LPV input-output system IDentification Toolbox.* The interested reader can contact the author via `boonto@tu-harburg.de`

is more involved, especially when an LPV controller is to be designed. In this case, the estimated model must be stabilized by the controller that is used for identification, otherwise it cannot be used for the design of a better controller [Codrons, 2005]. The estimated LPV input-output model can be evaluated by the closed-loop time domain response and the ν-gap metric. The latter is a framework for assessing model uncertainty of LTI systems in terms of closed-loop performance [Vinnicombe, 1993a]. However, there is no LPV version of the ν-gap metric. To use this tool for LPV systems, one can freeze the system at suitable scheduling values and calculate the ν-gap. This idea has been used in [Fujimori and Ljung, 2005; Wood, 1995], but no connection was made between system identification and controller design. Here we measure the quality of the identified LPV models by comparing the ν-gap and the *generalized stability margin* of the identified model (see [Vinnicombe, 1993b]). The ν-gap between the frozen LPV models and the model used to design a controller for identification has to be less than the generalized stability margin of the identified model for all values of the scheduling parameters over the considered range. An LPV model which satisfies this criterion, can be used to design a controller that gives better results than the one used for identification.

The contents of this chapter are as follows. In Section 6.1 some basic concepts of model validation using the ν-gap metric are reviewed. Application to experimental identification and closed-loop control of a magnetic levitation system and an arm driven inverted pendulum are discussed in Section 6.2 and Section 6.3, respectively.

6.1 ν-Gap and Model Validation for Control

The ν-gap metric and related analysis tools are briefly reviewed in this section.

Consider a closed-loop system made up of the feedback interconnection of a LTI system G and a LTI controller K; see Figure 6.1. The closed-loop transfer function matrix $T(G,K)$ from $[r_1^T(k)\ \ r_2^T(k)]^T$ to $[y^T(k)\ \ u^T(k)]^T$ is

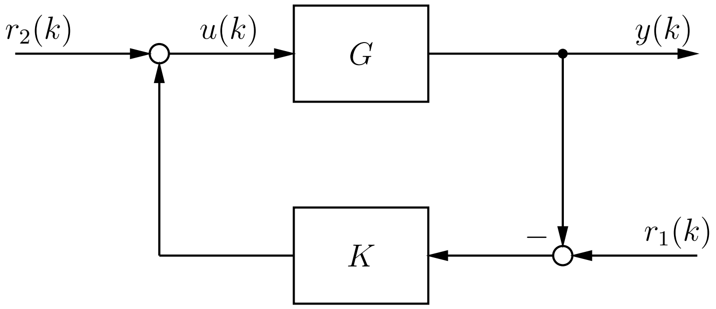


Figure 6.1: Closed-loop system

$$T(G,K) = \begin{bmatrix} K(I+GK)^{-1}G & K(I+GK)^{-1} \\ (I+GK)^{-1}G & (I+GK)^{-1}, \end{bmatrix}. \tag{6.1}$$

Definition 6.1 (generalized stability margin [Vinnicombe, 1993a])**.** The generalized stability margin of the closed-loop system in Figure 6.1 is defined as

$$b_{GK} = \begin{cases} \|T(G,K)\|_\infty^{-1} & \text{if } T(G,K) \text{ is stable,} \\ 0 & \text{otherwise.} \end{cases} \tag{6.2}$$

Thus, the generalized stability margin of the closed-loop system takes its values between 0 and 1; the higher this value, the better the stability margin.

In the case of a two-degrees-of-freedom (2-DOF) control configuration as shown in Figure 6.2, a controller is split into two parts K_1 and K_2. K_1 acts as a feedforward prefilter of the reference signal while K_2 ensures robust stability. To compute the generalized stability margin (6.2), the feedforward part is excluded since it is not related to stability.

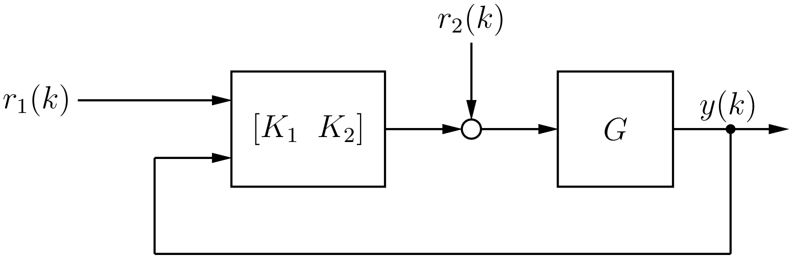


Figure 6.2: 2-DOF control configuration

The ν-gap metric has been proposed by Vinnicombe [1993a,b] to measure the distance between two continuous-time transfer matrices G_1 and G_2, and is defined as follows.

Definition 6.2 (ν-gap metric)**.** The ν-gap between two transfer matrices G_1 and G_2, denoted by δ_ν, is defined as

$$\delta_\nu(G_1, G_2) = \begin{cases} \|\kappa(G_1(e^{j\omega}), G_2(e^{j\omega}))\|_\infty & \text{if } \det \Xi(j\omega) \neq 0 \;\forall\omega \\ & \text{and } \mathrm{wno}(\det(\Xi(s))) = 0 \\ 1 & \text{otherwise ,} \end{cases} \tag{6.3}$$

where

- $\Xi(s) \triangleq N_2^*(s)N_1(s) + M_2^*(s)M_1(s)$;
- $\kappa(G_1(j\omega), G_2(j\omega) \triangleq -\tilde{N}_2(j\omega)M_1(j\omega) + \tilde{M}_2(j\omega)N_1(j\omega)$ is called the *chordal distance* between G_1 and G_2 at frequency ω;
- $G_1(s) = N_1(s)M^{-1}(s)$ and $G_2(s) = N_2(s)M_2^{-1} = \tilde{M}_2^{-1}(s)\tilde{N}_2(s)$ are *normalized coprime factorizations* of G_1 and G_2;
- $\mathrm{wno}(G(s)) = \eta(G^{-1}(s)) - \eta(G(s))$ is called the *winding number* of the transfer function $G(s)$ and is defined as the number of *counterclockwise encirclements* around the origin of the Nyquist contour of $G(s)$ indented to the right of any imaginary axis pole of $G(s)$;

- $\eta(G(s))$ denotes the number of poles of $G(s)$ in $\mathbb{C}_0^+$.

The ν-gap has a value between 0 and 1; the smaller the value, the closer the two models. If the models are close then a controller designed for one system will work with the other one as well. The definition can be applied to discrete-time system by means of the bilinear transformation $s = \frac{z-1}{z+1}$.

For closed-loop system identification, the data is collected in closed-loop in which the unknown true system G_0 is stabilized by a controller K_{id}, and an estimate $\hat{G}$ of G_0 is obtained. It is obvious that the closed-loop system $(\hat{G}, K_{id})$ must be stable since the actual closed-loop system (G_0, K_{id}) is stable. What is required is that G_0 and $\hat{G}$ are "close to each other" in a *closed-loop sense*, i.e., $\|T(G_0, K_{id}) - T(\hat{G}, K_{id})\|_\infty$ must be small, where T is the closed-loop transfer matrix. Instead of directly computing the norm value, it is more convenient to use the ν-gap [Codrons, 2005]. In this case, the ν-gap metric between G_0 and $\hat{G}$ must be less than the stability margin $b_{\hat{G},K_{id}}$ or

$$\delta_\nu(G_0, \hat{G}) < b_{\hat{G},K_{id}}. \tag{6.4}$$

If the distance between $\delta_\nu(G_0, \hat{G})$ and $b_{\hat{G},K_{id}}$ is small, it will be very difficult or impossible to design a new controller K based on $\hat{G}$ with a better performance on G_0. In practice, $b_{G_0,K_{id}}$ is unknown. One possible solution is to replace it by an estimated $b_{G_{id},K_{id}}$ where G_{id} is a nominal model used to design K_{id} to stabilized G_0 [Codrons, 2005].

6.2 Application to Magnetic Levitation System

In this Section, we show an application of LPV input-output identification using the LPV-ARX model structure with polynomials in LPR structure to a magnetic levitation plant. The magnetic levitation plant used in this chapter is manufactured by Education Control Product (ECP) [Parks, 1999]. The plant is shown in Figure 6.3: two magnetic coils can be used to control the position of two magnetic disks. Control inputs are the currents in the upper and lower coil, respectively, and the feedback signals are the positions of the upper and lower disk, measured by laser sensors [Parks, 1999].

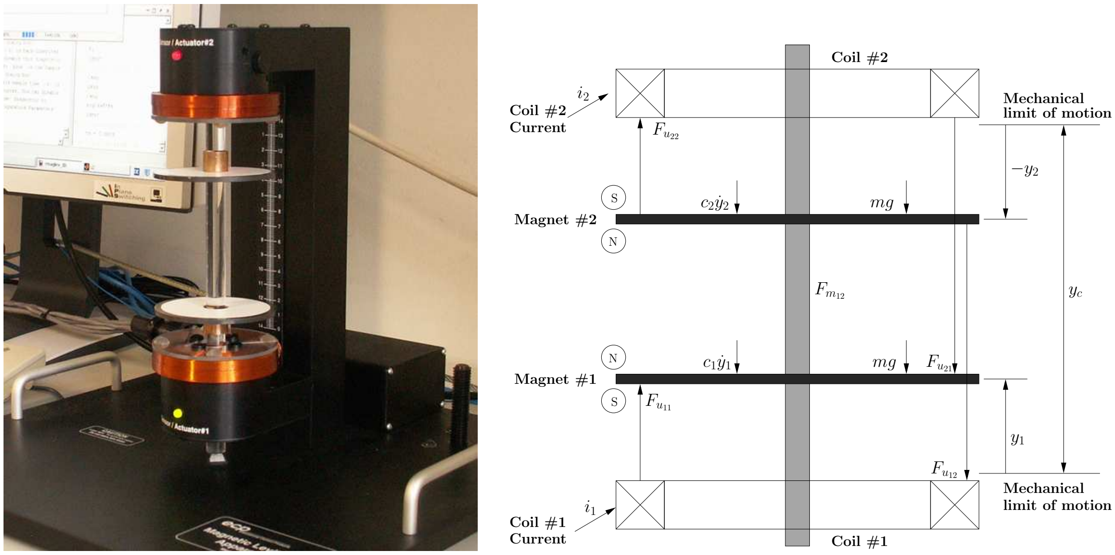
6.2.1 Physical Plant Model

Figure 6.3b shows a free body diagram of the two suspended magnetic disks, from which the equations of motion of the system can be derived as

$$m\ddot{y}_1 + c_1\dot{y}_1 = F_{u_{11}}(y_1, i_i) - F_{m_{12}}(y_1, y_2) - F_{u_{21}}(y_1, i_2) - mg \tag{6.5a}$$

$$m\ddot{y}_2 + c_2\dot{y}_2 = F_{u_{22}}(y_2, i_2) - F_{u_{12}}(y_1, i_1) - mg, \tag{6.5b}$$

where y_1, y_2 are the disk positions; y_c is considered fixed, $F_{u_{11}}$ and $F_{u_{22}}$ are the magnetic forces generated by the input current i_1 acting on the lower disk, and by i_2 acting on the upper disk, respectively. $F_{m_{12}}$ is the magnet-to-magnet force between the disks, and $F_{u_{21}}$, $F_{u_{12}}$ represent cross-coupling, i.e. the coil-to-magnet forces between lower coil and



(a) Real plant **(b)** Free body diagram

Figure 6.3: Magnetic levitation System

upper disk, and upper coil and lower disk, respectively c_1 and c_2 are friction coefficients. The system is nonlinear due to the nonlinearity in the actuator characteristics, in the magnet-to-magnet characteristics and also to the nonlinearity of the input and output sensors.

6.2.2 Experiment Setup

In this experiment, we consider a tracking problem to move the disks up and down to several positions. The nonlinearity of the plant is reduced by using the approximated inverse characteristic model [Parks, 1999]. To obtain the data for identification, the upper and lower disks are initially set up at 3 cm from the upper coil and 1.5 cm from the lower coil, respectively. This is done by passing offset currents to both upper and lower coils. With this offset current, the plant can be considered as locally open loop stable when the disks move up and down a small distance around the operating points. Multi-sine signals are added to both coils as excitation signals. A data set is collected at 2 msec sampling time and the data is filtered by a 60 Hz low pass filter to exclude undesired high frequency noise. A block diagram of the identified plant is shown in the dashed box in Figure 6.4.

6.2.3 Identification Result and Controller Synthesis

The LPV input-output model in the LPR structure is used to identify the plant model where $n_a = \left[\begin{smallmatrix}6&6\\6&6\end{smallmatrix}\right]$, $n_b = \left[\begin{smallmatrix}6&6\\6&6\end{smallmatrix}\right]$ and one sample delay. The scheduling signal is y_2 and the scheduling function is given in the vector form by $\psi(y_2) = \begin{bmatrix}1 & y_2 & y_2^2\end{bmatrix}$. This selection

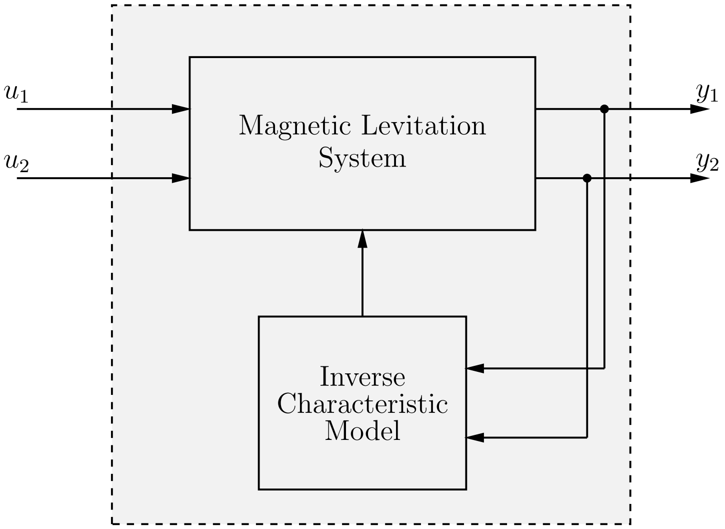


Figure 6.4: Magnetic levitation block diagram for identification.

is done by heuristic search and from the fact that the dynamics of the upper disk is more nonlinear than that of the lower one. A more systematic method to do this is a *principal component analysis* (PCA) method [Kwiatkowski and Werner, 2008].

The obtained model is converted into LPV observable state-space form with the minimal 12^{th} order. Figure 6.5 shows the cross-validation of the LPV-ARX model compared with measured output data and a linear ARX model; the ARX model uses same structure and delay time as the LPV-ARX model. It is clear that the LPV-ARX model with only two scheduling parameters – y_2 and y_2^2 – gives a more accurate model than the linear ARX model.

Since y_2 is the only scheduling signal, the resulting polytopic model has two vertices; one at $y_2 = 1$ cm and the other one at $y_2 = -1$ cm from the operating point. In this case, system matrices are in observable form and the $A(\theta(k))$ and $B(\theta(k))$ matrices are parameter dependent in affine form. To make $B(\theta(k))$ satisfy assumptions (A3) in [Apkarian et al., 1995], an input filter W_B is added to the input channels [Apkarian et al., 1995]. A mixed sensitivity controller design approach is used as shown in Figure 6.6. Weighting filters are chosen at each vertex and given below;

$$\begin{aligned} W_B &= \frac{0.9901z + 0.9901}{z + 0.9802} \\ W_{S,y_1} &= \frac{0.02(z+1)}{z-1}, \qquad W_{S,y_2} = \frac{0.2(z+1)}{z-1} \\ W_{K,y_1} &= \frac{9.901 \times 10^{-5} z}{z + 0.9802}, \qquad W_{K,y_2} = \frac{9.901 \times 10^{-4} z}{z + 0.9802} \end{aligned}$$

where W_B is an input signal filter and is equal for both channels. W_{K,y_1}, W_{K,y_2} are used for shaping the control sensitivity and W_{S,y_1}, W_{S,y_2} for the sensitivity at the target positions. The weighting filters related to position y_1 are selected to be low-pass filters at a lower frequency than that of y_2. This is because the movement of the lower disk is slower than the upper disks by physical construction. A resulting 18^{th} order controller

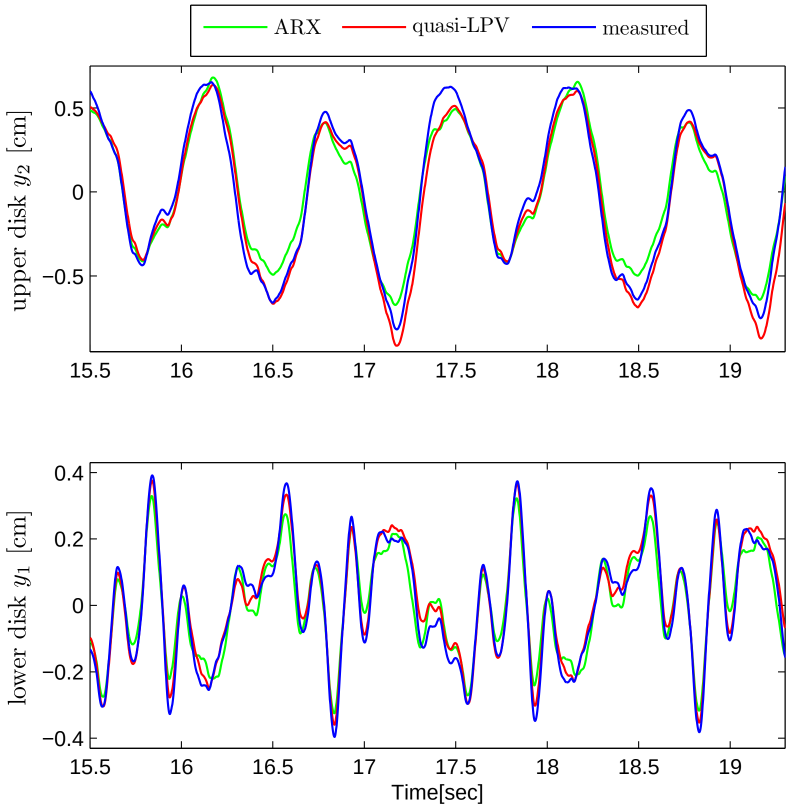


Figure 6.5: Cross-validation of the magnetic levitation model. Black: measured output, red: quasi-LPV model, green: ARX model

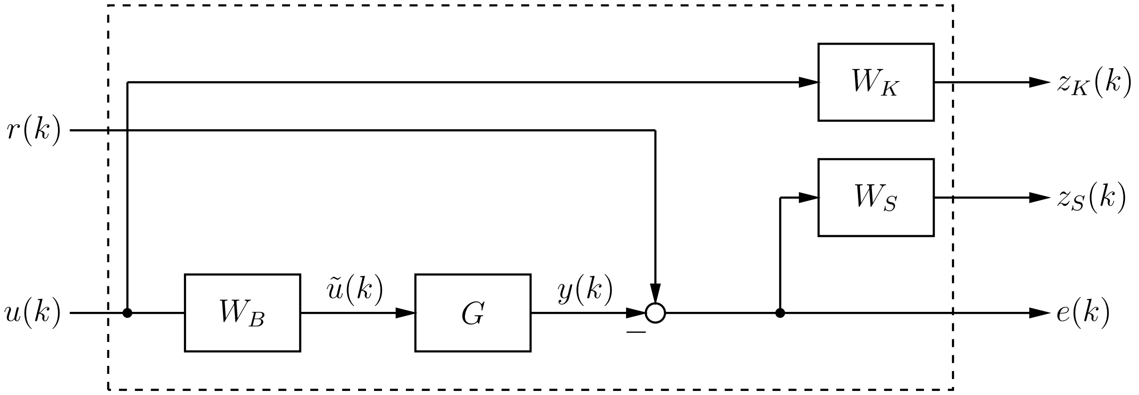


Figure 6.6: Generalized plant for LPV controller design

is designed using the dilated LMIs approach described for LTI systems in [de Oliveira et al., 2002] and for LPV systems in [Ali and Werner, 2011]; it achieves an induced ℓ_2 gain of 1.6744. Experimental results with this LPV discrete-time controller are shown in Figure 6.7. The result is compared with a discrete-time LTI $\mathcal{H}_\infty$ of order 11 which is the best controller that can be designed by using a linear model, see [Gürcüğğlu, 2010]. Since the main source of the nonlinearity of the magnetic levitation system has been compensated by using an inverse nonlinear function, both $\mathcal{H}_\infty$ and LPV controller give satisfactory results. However, the LPV controller gives better performance than $\mathcal{H}_\infty$, especially when we look at each peak of the reference signals, see three bottom rows of Figure 6.7. The LPV controller has clearly a smoother responses and less cross-coupling than the LTI controller.

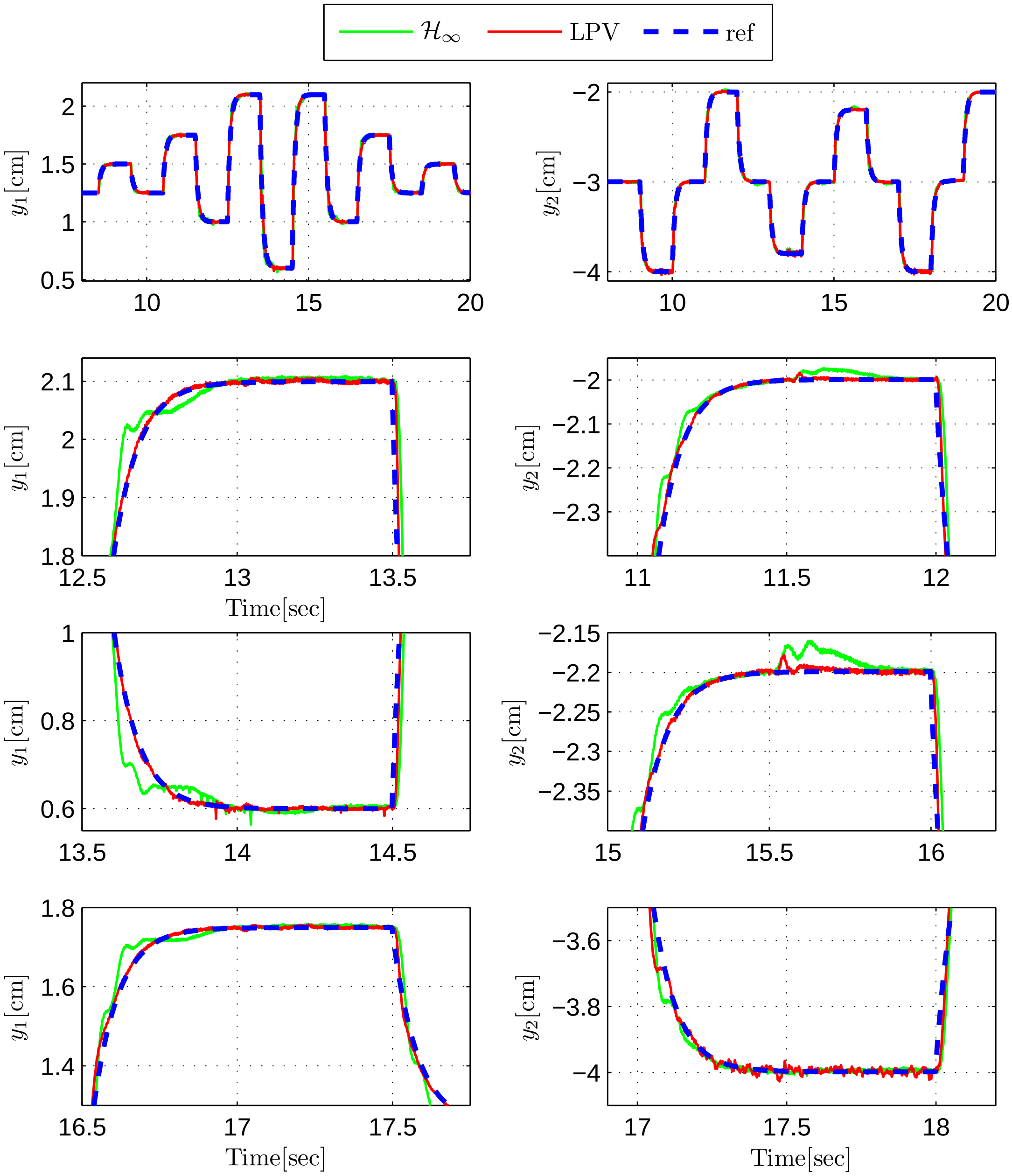


Figure 6.7: Tracking response of magnetic disks with $\mathcal{H}_\infty$ (green) and LPV (blue) controller. Left hand side: lower disk y_1, right hand side: upper disk y_2.

6.3 Arm-Driven Inverted Pendulum (ADIP)

In this section, the identification of an LPV-ARX model in LPR structure is applied to the Arm-Driven Inverted Pendulum (ADIP) plant shown in Figure 3.9, which is the same plant as used in Chapter 3. It should be noted that the LPV-ARX model of this plant identified in Chapter 3 was obtained for practical reasons with the polynomials in the RPR format and therefore can not be transformed to an LPV state-space model having static dependence. The model is not suitable for LPV controller synthesis even though it can be used for other purposes, for example as a simulation model etc.

In this chapter, the objective of the identification is to use the identified LPV input-output model $\hat{G}_{lpv}$ to design an LPV controller that can stabilize the ADIP for a wider operation range than that of the LTI controller used for identification. For this objective, we design a the 2-DOF controller to improve the tracking performance of the closed-loop system [Apkarian et al., 1995]. The closed-loop system configuration is shown in Figure 6.8. The discrete-time $\mathcal{H}_\infty$ controller denoted by K_{id} is designed by using a linear model G_{id} taken from [Lachhab et al., 2008] which can guarantee the stability of ADIP up to $\pm 51°$. The multi-sine excitation signal $r_2(k)$ adds to the input $u(k)$. The whole system to be identified is marked as a dashed box. The data $y(k)$ and $u(k)$ is collected at 5 msec sampling time and is filtered by a 10 Hz low pass filter. The model is identified by using the two-step method described in Chapter 3. A high order NARX model is used as a noise filter in the first step; and the second step is the identification of LPV-ARX model with all polynomials in the LPR form.

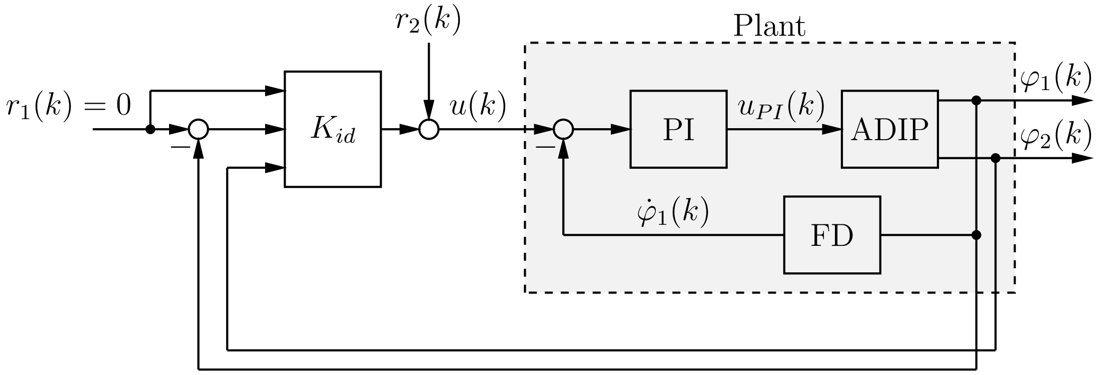


Figure 6.8: Block diagram for closed-loop identification

6.3.1 Model Validation

In this experiment, the resulting LPV input-output model structure is selected as $n_a = [\begin{smallmatrix} 2 & 2 \\ 2 & 2 \end{smallmatrix}]$, $n_b = 2$ and 1 step delay. The system is then converted to LPV state-space model with minimal 4^{th} order.

As discussed in Section 6.1, an identified LPV model with a good BFT is not necessarily suitable for LPV controller synthesis. For controller design, the LPV model $\hat{G}_{lpv}$ must be stabilized by K_{id}, or $\|T(G_{id}, K_{id}) - T(\hat{G}_{lpv}, K_{id})\|_\infty$ is small where $T(\cdot, \cdot)$

are closed-loop transfer functions[†]. For this purpose, the bode magnitude plot of the frequency response of $\hat{G}_{lpv}$ at each frozen parameter θ is plotted. Moreover, the ν-gap metric is used to compare the distance between $\hat{G}_{lpv}$ and G_{id}. This gap has to be smaller than the generalized stability margin $b_{\hat{G}_{lpv}}$ otherwise the model cannot be used for LPV controller design.

The bode magnitude plot of the identified model is shown in Figure 6.9. In the frequency range of interest around the closed-loop bandwidth of ~ 10 rad/sec, for both outputs the difference between the magnitude of G_{id} and $\hat{G}_{lpv}(\theta)$ are very small.

This means that in frequency domain $\hat{G}_{lpv}$ is similar to G_{id} and the model should be stabilized by K_{id}. The stability test of $T(\hat{G}_{lpv}, K_{id})$ is implemented by a closed-loop simulation and shown in Figure 6.10.

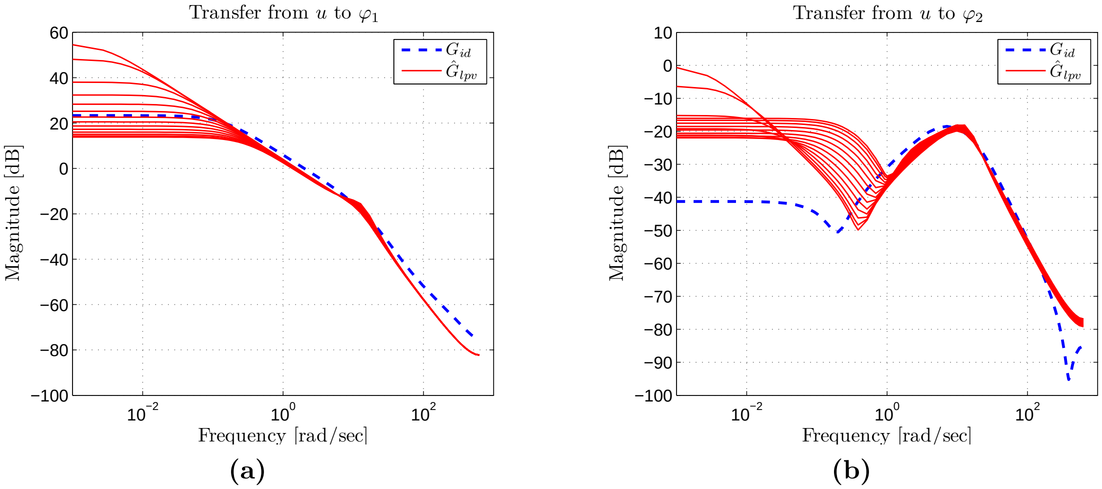


Figure 6.9: Bode magnitude plot of G_{id} compared with $\hat{G}_{lpv}$ at each frozen parameter where $-0.96 \leq \theta \leq 0.96$ rad and $\theta = \sin(\varphi_1)$: (a) from u to φ_1, (b) from u to φ_2.

This test confirms that $\hat{G}_{lpv}$ can be stabilized by K_{id}. The ν-gap $\delta_\nu(G_{id}, \hat{G}_{lpv})$ and the generalized stability margin $b_{\hat{G}_{lpv},K_{id}}$ of $\hat{G}_{lpv}$ at each frozen parameter θ are shown in Figure 6.10. As $b_{\hat{G}_{lpv},K_{id}} > \delta_\nu(G_{id}, \hat{G}_{lpv})$ in the scheduling parameter range, it follows that this LPV model is suitable for LPV controller design.

6.3.2 LPV Controller Synthesis

After the LPV model is identified, which is affine in the parameters, it is converted into the polytopic form. The full-order discrete-time polytopic LPV controller is designed based on dilated LMIs approach described for LTI systems in [de Oliveira et al., 2002] and for LPV systems in [Ali and Werner, 2011] to satisfy the following objectives:

- stabilize the plant in a wide-range,

[†] As mentioned before, since the true system G_0 is unknown, G_{id} is used for test instead.

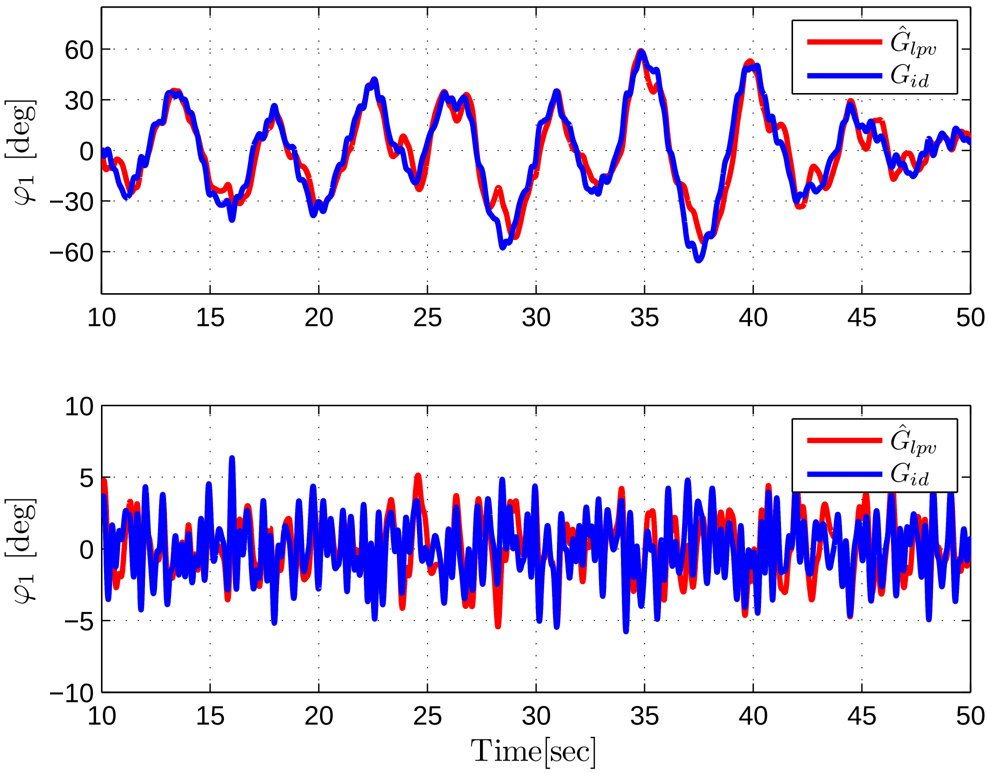


Figure 6.10: Closed-loop simulation of $T(G_{id}, K_{id})$ and $T(\hat{G}_{lpv}, K_{id})$

- control input is in the range of actuator limits,
- small overshoot, rise time (t_s), and steady state error.

Due to the sinusoidal scheduling function ($\theta = \sin(\varphi_1)$), the resulting polytopic model has three vertices: zero position and $\pm\varphi_1^t$ (target angular position). The weighting filters for the output φ_1 are designed to have the same value for each vertex. The output φ_2 is more difficult to stabilize at φ_1^t, so the weighting filters are selected to be slower than at the zero position. Moreover, since the real plant does not have ideal symmetry, the designed weighting filters of this output at positive and negative angles are different. The weighting filters are selected as

$$W_{S1}=\begin{bmatrix} \frac{0.0125z+0.0125}{z-1} & 0 \\ 0 & \frac{0.5553z+0.05553}{z-0.999} \end{bmatrix}, \quad W_{S2}=\begin{bmatrix} \frac{0.0125z+0.0125}{z-1} & 0 \\ 0 & \frac{0.0040z+0.0040}{z-0.999} \end{bmatrix},$$

$$W_{S3}=\begin{bmatrix} \frac{0.0125z+0.0125}{z-1} & 0 \\ 0 & \frac{0.0050z+0.0050}{z-0.999} \end{bmatrix}, \quad W_K=\frac{0.1013z+0.09867}{z+1},$$

where W_{S1}, W_{S2} and W_{S3} are for shaping the sensitivity function at zero, positive and negative target positions respectively, while W_K is for shaping the control sensitivity

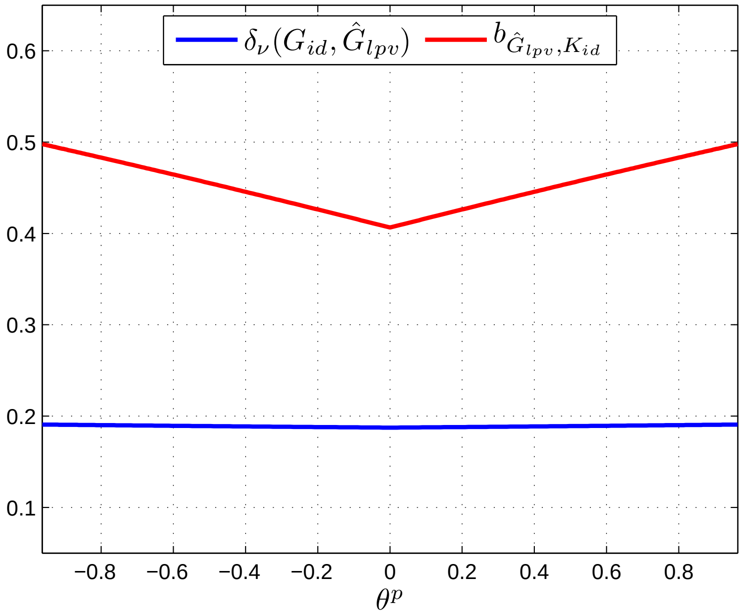


Figure 6.11: Comparison between ν-gap metric of $\hat{G}_{lpv}(\theta)$ and G_{id} with generalized stability margin $b_{\hat{G}_{lpv},K_{id}}$, here $-0.96 \leq \theta \leq -0.96$ where $\theta = \sin(\varphi_1)$.

function and is equal for all vertices. The resulting 8^{th}-order polytopic LPV controller achieves an induced ℓ_2 gain of $\gamma = 1.927$. The scheduling parameter θ is bounded between $\pm\sin(1.3)$. The resulting LPV controller has been tested on a real ADIP plant and experimental results are shown in Figure 6.12. It should be highlighted that these experimental results are the best among results achieved with this plant published so far. An overview is given in Table 6.1.

Table 6.1: Control performance achieved on ADIP

Method	φ_1^t	t_s	Overshoot	Oscillation
Polytopic LPV with NN state-space model [Lachhab et al., 2008]	±60°	1.5 sec	0°	$\leq \pm 2°$
Fixed-structure LFR with NN state-space model [Lachhab et al., 2008]	±66°	1.5 sec	0°	$\leq \pm 5°$
Dilated LMI with NN state-space model [Ali and Werner, 2011]	±67°	2.5 sec	5°	0°
Dilated LMI with Proposed method	±70°	1.5 sec	0°	$\leq \pm 1°$

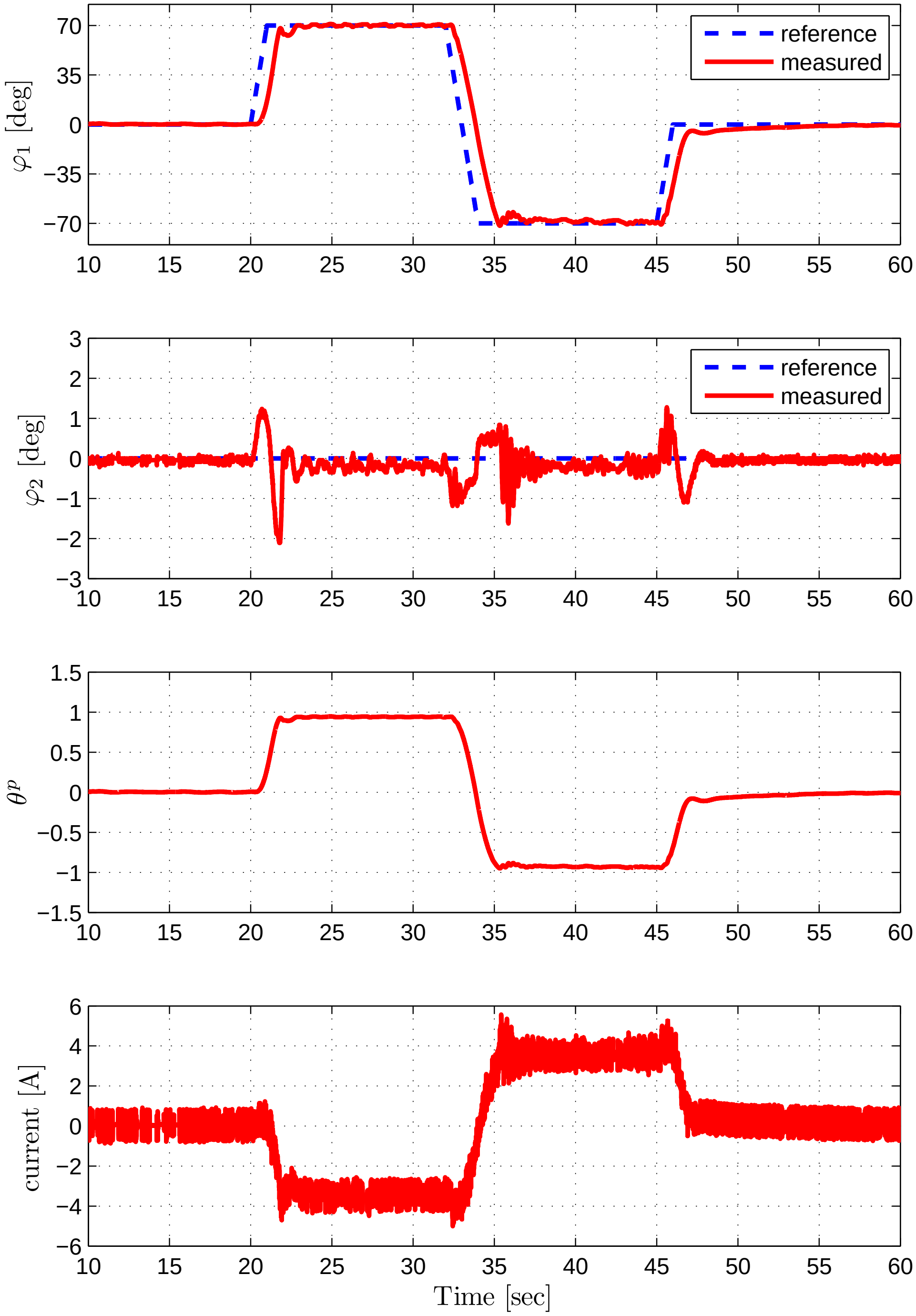


Figure 6.12: Experimental results of ADIP system with LPV controller ($\varphi_1^t = \pm 70°$)

6.4 Conclusion

This chapter reports practical applications of LPV input-output identification in LPR form. For open-loop identification, the model can be identified using an open-loop method and validated by time-domain criteria. Users have to take care only of the model structure and the estimation of the nonlinear scheduling function. For closed-loop system identification, time-domain criteria are not suitable because they do not guarantee the stability of the feedback system under the controller being used in the identification experiment. The latter can be achieved with the ν-gap metric. A procedure is proposed and shown to be suitable for model based LPV controller design.

Chapter 7

Conclusions and Outlook

In this thesis several problems associated with the identification of LPV input-output models have been investigated. The contributions are focused on three main problems: the approximation of the nonlinear scheduling function of LPV systems, the unbiased identification of LPV systems, and the construction of state-space realizations of input-output LPV models. Both open-loop and closed-loop system identification have been considered. Moreover, the efficiency of the proposed methods is demonstrated by applying them to several plants in simulation and experiment.

In Chapter 3, the approximation of the nonlinear scheduling function of LPV systems is conducted by using cubic spline functions, which have more easily tunable parameters than polynomial functions. A separable least-squares algorithm is proposed to reduce the number of parameters by separating them into linear and nonlinear parts. Given initial values of the nonlinear parameters, one can solve for the linear parameters by using a linear least squares method. On the other hand, by fixing the linear parameters the Levenberg-Marquardt algorithm can be applied to solve for nonlinear parameters. A recursive version of the algorithm is also discussed for the case when the dimension of the data exceed the computing capacity. Moreover, for unstable systems, a model is identified in a closed-loop by using a two-step method with a neural network as a noise filter. Simulated and experimental results are given, demonstrating the efficiency of the presented approach.

Concerning the unbiased identification of LPV systems, which is presented in Chapter 4, the bias error is reduced by using instrumental variables with auxiliary model method. To improve the performance while still maintaining simplicity, an auxiliary model in LPV-OE structure, which is estimated by an OE method, has been proposed. Models estimated by the OE method give a predicted output which is free of noise. Thus they represent better auxiliary models than LPV-ARX. The proposed approach gives a significant improvement in terms of bias error and variance. A comparison with an existing method is illustrated by several simulation examples. This technique has been applied to both LPV and quasi-LPV systems.

A state-space realization of LPV input-output models, typically required for control synthesis, is difficult because of the noncommutativity of the shift operator and the coefficient functions which depend on scheduling parameters. Using an LTI procedure, the resulting state-space models will contain dynamic dependence on the scheduling parameters. This problem has been solved by using skew polynomials in Chapter 5. LPV models can be represented as RFD and LFD forms. The former can be converted to LPV state-space models in a reachable form while the latter can be transformed to LPV state-space models in an observable form. It has been argued that the observable form is more practical since the obtained models contain only static dependence. Furthermore, the link between identification and realization has been discussed. In identification, the model structures should be in LPR form. The estimated model can then be directly transformed into an LPV state-space model in observable form. LPV controller synthesis techniques can be used without losing closed-loop performance. The efficiency of the proposed methods is demonstrated by applying them to a laboratory scale magnetic levitation plant and an ADIP system.

Even though the method for solving the realization problem yields reasonable results, the identified models may not be ready to be used when they are identified in the closed-loop. This issue has been explored in Section 6.1. In closed loop, it is obvious that the identified models should be stabilized by the controller which is used for identification. The use of the ν-gap metric to LPV systems has been proposed to measure the quality of models. In this case, the ν-gap of the identified model and the model used for collecting the data in the closed loop, must be less than the generalized stability margin for all values of the scheduling parameters in a considered range. This idea has been applied to the identification and control of ADIP system.

There are a number of important issues that have not been considered in this thesis and which are worthy of being explored further. They are outlined briefly as follows:

- An algorithm utilized in Chapter 3, which combines LS and cubic splines to calculate the coefficients of LPV input-output models has not yet been shown to converge.

- Persistency of excitation is an important issue in system identification. Conditions for persistency of excitation for LPV systems have been formulated in [Bamieh and Giarré, 2002; Wei, 2006]. However, there are no implementation methods to design such signals to meet the condition. This topic can be explored in order to design a sufficiently rich excitation signal for both open and closed-loop identification settings.

- The unbiased LPV identification method proposed in Chapter 4 is tested by simulation examples. This method would be even more valuable if experimental results can be shown. Moreover, an extension of the method to closed-loop identification should be studied.

- The ν-gap metric was designed for LTI systems. A time-varying version of this tool has not been considered so far in the literature. In Chapter 6, an ad-hoc method has been applied to LPV systems by freezing them at several values of the scheduling parameters. Establishing a worst-case bound on the ν-gap for LPV systems would give a better insight to the quality models.

Appendix A

Derivation of Equations in Chapter 3

This appendix shows the derivation of equations (3.8) and (3.9). The idea of this proof is adopted from [Ljung, 1999]. Consider the cost function

$$V_N(\mathfrak{P}, Z^N) = \frac{1}{N} \sum_{k=1}^{N} \| y(k) - \mathfrak{P}^T \phi(k) \|_2^2, \tag{A.1}$$

where $\mathfrak{P} \in \mathbb{R}^{n_y \times n_r}$, n_y and $n_{\mathfrak{p}}$ are the number of outputs and number of estimated parameters for each output, $y \in \mathbb{R}^{n_y \times 1}$ and $\phi \in \mathbb{R}^{1 \times n_r}$ is the regressor vector.

The cost function (A.1) can be rewritten as

$$\begin{aligned} V_N(\mathfrak{P}, Z^N) &= \text{trace} \frac{1}{N} \sum_{k=1}^{N} [y(k) - \mathfrak{P}^T \phi(k)][y(k) - \mathfrak{P}^T \phi(k)]^T \\ &= \frac{1}{N} \sum_{k=1}^{N} \Big\{ [y_1(k) - \mathfrak{p}_1^T \phi(k)]^T [y_1(k) - \mathfrak{p}_1^T \phi(k)] \\ &\qquad + \cdots + [y_{n_y}(k) - \mathfrak{p}_{n_y}^T \phi(k)]^T [y_{n_y}(k) - \mathfrak{p}_{n_y}^T \phi(k)] \Big\} \\ &= \frac{1}{N} \sum_{k=1}^{N} \Big\{ [y_1^2(k) - 2\phi_k(k) \mathfrak{p}_1 y_1(k) + \mathfrak{p}_1^T \phi(k) \phi^T(k) \mathfrak{p}_1] \\ &\qquad + \cdots + [y_{n_y}^2(k) - 2\phi(k) \mathfrak{p}_{n_y} y_{n_y}(k) + \mathfrak{p}_{n_y}^T \phi(k) \phi^T(k) \mathfrak{p}_{n_y}] \Big\}. \end{aligned}$$

Computing the gradient with respect to each $\mathfrak{p}_i$ we have

$$\nabla_{\mathfrak{p}^T} V_N = \begin{bmatrix} \frac{\partial V_N}{\partial \mathfrak{p}_1^T} & \cdots & \frac{\partial V_N}{\partial \mathfrak{p}_{n_y}^T} \end{bmatrix}^T,$$

then

$$\hat{\mathfrak{P}}_N^{\text{LS}} = \left[\frac{1}{N} \sum_{k=1}^{N} \phi(k) \phi^T(k) \right]^{-1} \left[\frac{1}{N} \sum_{k=1}^{N} \phi(k) y^T(k) \right].$$

We use a more compact representation

$$Y = \mathfrak{P}^T \Phi, \tag{A.2}$$

where $Y = [y_1 \quad \dots \quad y_N]$ and $\Phi = [\phi_1 \quad \dots \quad \phi_N]$. Then

$$\begin{aligned} Y^T &= \Phi^T \mathfrak{P} \\ \Phi Y^T &= \Phi \Phi^T \mathfrak{P} \\ \hat{\mathfrak{P}}_N^{\mathrm{LS}} &= [\Phi \Phi^T]^{-1} \Phi Y^T. \end{aligned}$$

Appendix B

Properties of the Kronecker Product

All materials in this Appendix are taken from [Steeb, 2006].

Let A be an $m \times n$ matrix and B an $r \times s$ matrix. The Kronecker product of A and B is defined as the $(mr) \times (ns)$ matrix

$$A \otimes B = \begin{bmatrix} a_{11}B & a_{12}B & \cdots & a_{1n}B \\ a_{21}B & a_{22}B & \cdots & a_{2n}B \\ \vdots & \vdots & \ddots & \vdots \\ a_{m1}B & a_{m2}B & \cdots & a_{mn}B \end{bmatrix}. \tag{B.1}$$

The Kronecker product has the following properties:

Bilinearity and Associativity

$$\begin{aligned} A \otimes (B + C) &= A \otimes B + A \otimes C, \\ (A + B) \otimes C &= A \otimes C + B \otimes C, \\ (kA) \otimes B &= A \otimes (kB) = k(A \otimes B), \\ (A \otimes B) \otimes C &= A \otimes (B \otimes C), \end{aligned}$$

where A, B and C are matrices with appropriate dimensions and k is a scalar.

The mixed-product property

Let A, B, C and D be matrices with appropriate dimensions. Then

$$(A \otimes B)(C \otimes D) = (AC) \otimes (BD).$$

Inverse

$A \otimes B$ is invertible if and only if A and B are invertible. The inverse of the product is given by

$$(A \otimes B)^{-1} = A^{-1} \otimes B^{-1}.$$

Transpose

The operation of transposition is distributive over the Kronecker product:

$$(A \otimes B)^T = A^T \otimes B^T.$$

Appendix C

Rings, Fields

All materials in this Appendix are taken from [Vidyasagar, 1985] if no other references are given.

Definition C.1 (Ring)**.** A *ring* is a nonempty set $\mathbb{R}$ together with two binary operations $+$ (addition) and $\cdot$ (multiplication) such that the following axioms are satisfied:

(R1) $\mathcal{R}[+]$ is a commutative group. This means that

$$\begin{gathered} a+(b+c)=(a+b)+c, \quad \forall a,b,c \in \mathcal{R}, \\ a+b=b+a, \quad \forall a,b \in \mathcal{R}. \end{gathered}$$

There exists an element $0 \in \mathcal{R}$ such that

$$a+0=0+a=a, \quad \forall a \in \mathcal{R}.$$

For every element $a \in \mathcal{R}$, there exists a corresponding element $-a \in \mathcal{R}$ such that $a+(-a)=0$.

(R2) $\mathcal{R}[\cdot]$ is a semigroup. This means that

$$a \cdot (b \cdot c) = (a \cdot b) \cdot c, \quad \forall a,b,c \in \mathcal{R}.$$

(R3) Multiplication is distributive over addition. This means that

$$\begin{aligned} a \cdot (b+c) &= a \cdot b + a \cdot c, \\ (a+b) \cdot c &= a \cdot c + b \cdot c, \quad a,b,c \in \mathcal{R}. \end{aligned}$$

As is customary, $a \cdot b$ is denoted as ab, and $a+(-b)$ is denoted as $a-b$.

Definition C.2 (Commutative)**.** A ring $\mathcal{R}$ is said to be *commutative* if $ab = ba \; \forall a,b \in \mathcal{R}$, and is said to have an *identity* if there exists an element $1 \in \mathcal{R}$ such that $1 \cdot a = a \cdot 1 = a \; \forall a \in \mathcal{R}$.

Definition C.3 (Unit)**.** An element in x in a ring $\mathcal{R}$ is called a *unit* of $\mathcal{R}$ if there is a $y \in \mathcal{R}$ such that $xy = yx = 1$. It can be shown that such a y is unique; y is called the *inverse* of x and is denoted by x^{-1}.

Definition C.4 (Field)**.** A *field* is a commutative ring $\mathcal{F}$ with an identity, satisfying two additional assumptions:

(F1) $\mathcal{F}$ contains at least two elements.

(F2) Every nonzero element of $\mathcal{F}$ is a unit.

Bibliography

H. Abbas and H. Werner. An instrumental variable technique for open-loop and closed-loop identification of input-output LPV models. In *Proceedings of the European Control Conference*, pages 2646–2651, Kos, Greece, July, 2–5 2007. – **One citation in pages** 2.

H. S. Abbas, M. Ali, and H. Werner. Linear recurrent neural networks for open- and closed-loop consistent identification of LPV models. In *Proceedings of the 49th IEEE Conference on Decision and Control*, pages 6851–6856, Atlanta, GA, USA, December 15–17 2010a. Accepted. – **5 citations in pages** 2, 4, 41, 51, and 54.

H. S. Abbas, R. Tóth, and H. Werner. State-space realization of LPV input-output models: Practical methods for the user. In *Proceedings of 2010 American Control Conference*, pages 3318–3323, Baltimore, MD, USA, June 30–July 2 2010b. – **4 citations in pages** 4, 43, 65, and 80.

M. Ali and H. Werner. Discrete-time LPV conroller synthesis using dilated LMIs with application to an arm-driven inverted pendulum. In *Proceedings of the 18th IFAC World Congress*, 2011. to appear. – **4 citations in pages** 87, 93, 96, and 98.

P. J. Antsaklis and A. N. Michel. *Linear Systems*. Birkhäuser, 2006. – **6 citations in pages** 64, 67, 68, 75, 78, and 79.

P. Apkarian and P. Gahinet. A convex characterization of gain-scheduled $\mathcal{H}_\infty$ controller. *IEEE Transactions on Automatic Control*, 40(5):853–864, May 1995. – **One citation in pages** 1.

P. Apkarian, P. Gahinet, and G. Becker. Self-scheduled $\mathcal{H}_\infty$ control of linear parameter-varying systems: a design example. *Automatica*, 31(9):1251–1261, 1995. – **3 citations in pages** 1, 92, and 95.

B. Bamieh and L. Giarré. Identification of linear parameter varying models. *International Journal of Robust and Nonlinear Control*, 12(9):841–853, July 2002. – **14 citations in pages** 2, 3, 4, 6, 10, 11, 15, 19, 33, 45, 63, 80, 87, and 102.

G. Belforte and P. Gay. Optimal experiment design for regression polynomial models identification. *International Journal of Control*, 75(15):1178–1189, 2002. – **2 citations in pages** 15 and 19.

G. Belforte, F. Dabbene, and P. Gay. LPV approximation of distributed parameter systems in environmental modelling. *Environmental Modelling & Software*, 20:1063–1070, 2005. – **One citation in pages** 3.

F. D. Bianchi, H. De Battista, and J. M. Mantz. *Wind Turbine Control Systems; Principles, Modelling and Gain Scheduling Design*. Springer, 2007. – **One citation in pages** 1.

S. Boonto and H. Werner. Closed-loop system identification of LPV input-output models – application to an arm-driven inverted pendulum. In *Proceeding of the 47th IEEE Conference on Decision and Control*, pages 2606–2611, Cancun, Mexico, December 9–11 2008. – **4 citations in pages** 2, 42, 45, and 48.

S. Boonto and H. Werner. Closed-loop identification of LPV models using cubic splines with application to an arm-driven inverted pendulum. In *Proceedings of the 2010 American Control Conference – ACC2010*, pages 3100–3105, Baltimore, MD, USA, June 30 – July 02 2010. – **One citation in pages** 16.

M. Butcher, A. Karimi, and R. Longchamp. On the consistency of certain identification methods for linear parameter varying systems. In *Proceedings of the 17th IFAC World Congress*, pages 4018–4023, Seoul, Korea, July 6–11 2008. – **13 citations in pages** 2, 4, 11, 25, 26, 27, 41, 42, 45, 50, 51, 52, and 58.

F. M. Callier and C. A. Desoer. *Multivariable Feedback Systems*. Springer-Verlag, 1982. – **One citation in pages** 64.

V. Cerone and D. Regruto. Set-membership identification of LPV models with uncertain measurements of the time-varying parameter. In *Proceedings of the 47th IEEE Conference on Decision and Control*, pages 4491–4496, Cancun, Mexico, December 9–11 2008. – **One citation in pages** 3.

B. Codrons. *Process Modelling for Control : A Unified Framework Using Standard Black-Box Techniques*. Springer-Verlag, 2005. – **3 citations in pages** 5, 88, and 90.

P. M. Cohn. On the embedding of rings in skew fields. In *Proceedings of The London Mathematical Society*, volume s3-11, pages 511–530, 1961. – **One citation in pages** 66.

C. de Boor. *A Practical Guide to Splines*. Springer-Verlag, 1978. – **2 citations in pages** 19 and 20.

R. A. de Callafon. *Feedback Oriented Identification for Enhanced and Robust Control: a fractional approach applied to a wafer stage*. PhD thesis, Delft University of Technology, 1998. – **2 citations in pages** 25 and 26.

M. de Oliveira, J. Geromel, and J. Bernussou. Extended H_2 and H_∞ norm characterizations and controller parametrizations for discrete-time systems. *International Journal of Control*, 75(9):666–679, 2002. – **3 citations in pages** 87, 93, and 96.

E. J. Dempsey and D. Westwick. Identification of hammerstein models with cubic spline nonlinearities. *IEEE Transactions on Biomedical Engineering*, 51(2):237–245, 2004. − **One citation in pages** 15.

F. Felici, J.-W. van Wingerden, and M. Verhaegen. Subspace identification of MIMO LPV systems using a periodic scheduling sequence. *Automatica*, 43(10):1684–1697, 2007. doi: doi:10.1016/j.automatica.2007.02.027. − **One citation in pages** 3.

U. Forssell and L. Ljung. A projection method for closed-loop identification. *IEEE Transactions on Automatic Control*, 45(11):2101–2105, 2000. − **One citation in pages** 3.

A. Fujimori and L. Ljung. Parameter estimation of polytopic models for a linear parameter varying aircraft system. Technical Report LiTH-ISY-R-2676, Department of Electrical Engineering, Linkping University, SE-581 83 Linkping, Sweden, Jan 2005. − **One citation in pages** 88.

A. Fujimori and L. Ljung. Model identification of linear parameter varying aircraft systems. *Proceedings of the Institution of Mechanical Engineers, Part G: Journal of Aerospace Engineering*, 220(4):337–346, 2007. − **One citation in pages** 5.

M. Gevers, A. S. Bazanella, X. Bombois, and L. Mišković. Identification and the information matrix: How to get just sufficiently rich? *IEEE Transactions on Automatic Control*, 54(12):2828–2830, 2009. − **One citation in pages** 81.

L. Giarré, D. Bauso, P. Falugi, and B. Bamieh. LPV model identification for gain scheduling control: An application to rotating stall and surge control problem. *Control Engineering Practice*, 14(4):351–361, 2006. doi: doi:10.1016/j.conengprac.2005.01.013. − **2 citations in pages** 4 and 63.

G. H. Golub and V. Pereyra. The differentiation of pseudoinverses and nonlinear least squares problems whose variables separate. *SIAM Journal of Numerical Analysis*, 10(2):413–432, April 1973. − **2 citations in pages** 15 and 22.

R. Guidorzi and R. Diversi. Representations of MIMO time-varying systems and realization of cyclostationary models. *Automatica*, 39(11):1903–1914, 2003. doi: 10.1016/S0005-1098(03)00195-X. − **One citation in pages** 63.

U. Gürcügğlu. Model identification and H_∞ controller synthesis fo the ECP magnetic levitator and preparation of educational materials. Studienarbeit, Technische Universität Hamburg-Harburg, 2010. − **One citation in pages** 93.

M. Halás. Nonlinear systems: A polynomial approach. In *Computer Aided Systems Theory – EUROCAST 2009*, pages 595–602, 2009. − **One citation in pages** 66.

M. Halás and Ü. Kotta. Pseudo-linear algebra: A powerful tool in unification of the study of nonlinear control systems. In *Proceedings of 7th IFAC Symposium on Nonlinear Control Systems*, pages 684–689, Pretoria, South Africa, August 22–24 2007. − **One citation in pages** 66.

K. Hsu, T. L. Vincent, and K. Poolla. Nonparametric methods for the identification of linear parameter varying systems. In *Proceedings of the 2008 IEEE International Symposium on Computer-Aided Control System Design Part of the 2008 IEEE Multi-Conference on Systems and Control*, pages 846–850, San Antonio, Texas, USA, September 3–5 2008. – **One citation in pages** 3.

A. Janczak. *Identification of Nonlinear Systems Using Neural Networks and Polynomial Models: A Block-Oriented Approach*. Springer, 2005. – **One citation in pages** 41.

T. Kailath. *Linear Systems*. Prentice-Hall, Inc., Englewood Cliffs, N.J., 1980. – **2 citations in pages** 67 and 77.

H. Kajiwara, P. Apkarian, and P. Gahinet. LPV techniques for control of an inverted pendulum. *IEEE Control Systems Magazine*, 19(1):44–54, February 1999. – **2 citations in pages** 16 and 32.

E. W. Kamen. Representation and realization of operational differential equations with time-varying coefficients. *Journal of the Franklin Institute*, 301(6):559–571, 1976. – **One citation in pages** 4.

E. W. Kamen, P. P. Khargonekar, and K. R. Poolla. A transfer-function approach to linear time-varying discrete-time systems. *SIAM Journal of Cotnrol and Optimization*, 23(4):550–565, 1985. – **One citation in pages** 63.

P. P. Khargonekar and K. R. Poolla. Polynomial matrix-fraction representations for linear time-varying systems. *Linear Algebra and Its Applications*, 80(August 1986): 1–37, 1986. – **One citation in pages** 63.

A. Kwiatkowski. *LPV Modeling and Application of LPV Controllers to SI Engines*. PhD thesis, Institute of Control Systems, Hamburg University of Technology, 2007. – **One citation in pages** 1.

A. Kwiatkowski and H. Werner. PCA-based parameter set mappings for LPV models with fewer parameters and less overbounding. *IEEE Transactions on Control Systems Technology*, 16(4):781–788, 2008. doi: 10.1109/TCST.2007.903094. – **One citation in pages** 92.

A. Kwiatkowski, J. Blath, H. Werner, and M. Schultalbers. Appliation of LPV gain scheduling to charge control of a SI engine. In *Proceedings of the 2006 IEEE Conference on Computer Aided Control Systems Design*, pages 2327–2331, Munich, Germany, October 4-6 2006a. – **One citation in pages** 11.

A. Kwiatkowski, M.-T. Boll, and H. Werner. Automated generation and assessment of affine LPV models. In *Proceedings of 45th IEEE Conference on Decision and Control*, San Diego, CA, USA, December 13-15 2006b. – **One citation in pages** 85.

N. Lachhab, H. Abbas, and H. Werner. Neural-network based technique for modelling and LPV control of an arm-driven inverted pendulum. In *Proceedings of the 47th IEEE Conference on Decision and Control*, pages 3860–3865, Cancun, Mexico, December, 9–11 2008. doi: 10.1109/CDC.2008.4739222. – **3 citations in pages** 87, 95, and 98.

P. Lancaster and K. Šalkauskas. *Curve and Surface Fitting: An Introduction*. Academic Press, 1986. – **One citation in pages** 20.

I. D. Landau. Unbiased recursive identification using model reference adaptive techniques. *IEEE Transactions on Automatic Control*, AC–21(2):194–202, 1976. – **One citation in pages** 47.

I. D. Landau and A. Karimi. An output error recursive algorithm for unbiased identification in closed loop. *Automatica*, 33(5):933–938, 1997. – **One citation in pages** 2.

I. D. Landau and G. Zito. *Digital Control Systems Design, Identification and Implementation*. Springer, 2006. – **7 citations in pages** 4, 42, 45, 47, 48, 49, and 50.

I. D. Landau, R. Lozano, and M. M'Saad. *Adaptive Control*. Springer, 1998. – **5 citations in pages** 42, 45, 47, 49, and 50.

V. Laurain, M. Gilson, R. Tóth, and H. Garnier. Refined instrumental variable methods for identification of LPV Box–Jenkins models. *Automatica*, 46(6):959–967, 2010. – **8 citations in pages** 2, 6, 11, 41, 51, 54, 55, and 56.

H. Lee. *A Plant-Friendly Multivariable System Identification Framework based on Identification Test Monitoring*. PhD thesis, Arizona State University, 2006. – **One citation in pages** 34.

L. H. Lee and K. Poolla. Identification of linear parameter-varying systems using nonlinear programming. *Journal of Dynamic Systems, Measurement, and Control*, 121(1):71–78, March 1999. – **One citation in pages** 3.

M. Leskens, L. Van Kessel, and P. Van den Hof. MIMO closed-loop identification of an MSW incinerator. *Control Engineering Practice*, 10(3):315–326, March 2002. doi: 10.1016/S0967-0661(01)00139-3. – **One citation in pages** 3.

L. Ljung. *System Identification Theory for the User*. Prentice Hall, Upper Saddle River, NJ, 2 edition, 1999. – **15 citations in pages** 2, 10, 18, 19, 23, 28, 29, 33, 34, 43, 47, 50, 80, 81, and 103.

L. Ljung. *System Identification Toolbox 7 User's Guide*. The MathWorks, Inc., 2010. – **2 citations in pages** 16 and 29.

L. Ljung and T. Söderström. *Theory and Practice of Recursive Identification*. MIT Press, 1983. – **One citation in pages** 24.

A. Marcos and G. Balas. Development of linear-parameter-varying models for aircraft. *Journal of Guidance Control, and Dynamics*, 27(2):218–228, 2004. − **One citation in pages** 1.

J. B. Moore. Global convercence of output error recursions in colored noise. *IEEE Transactions on Automatic Control*, AC–27(6):1189–1198, 1982. − **One citation in pages** 51.

M. Nemani, R. Ravikanth, and B. Bamieh. Identification of linear parametrically varying systems. In *Proceedings of the 34th IEEE Decision and Control Conference*, pages 2990–2995, New Orleans, LA, December 1995. − **One citation in pages** 2.

L. S. H. Ngia and J. Sjöberg. Efficient training of neural nets for nonlinear adaptive filtering using a recursive levenberg–marquardt algorithm. *IEEE Transactions on Signal Processing*, 48(7):1915–1927, 2000. − **One citation in pages** 24.

M. Nørgaard, O. Ravn, N. K. Poulsen, and L. K. Hansen. *Neural Networks for Modelling and Control of Dynamic Systems*. Springer-Verlag, London, UK, 2003. − **6 citations in pages** 12, 23, 26, 28, 29, and 41.

O. Ore. Linear equations in non-commutative fields. *The Annals of Mathematics*, 32 (3):463–477, 1931. − **One citation in pages** 66.

A. Packard. Gain scheduling via linear fractional transformations. *Systems and Control Letters*, 22(2):79–92, 1994. − **One citation in pages** 1.

T. R. Parks. *Manual for Model 730 Magnetic Levitation System*. Educational Control Products, 1999. − **3 citations in pages** 87, 90, and 91.

K. Poolla and P. Khargonekar. Stabilizability and stable-proper factorizations for linear time-varying systems. *SIAM Journal on Control and Optimization*, 25(3):723–736, 1987. − **One citation in pages** 63.

F. Previdi and M. Lovera. Indentification of a class of non-linear parametrically varying models. *International Journal of Adaptive Control and Signal Processing*, 17:33–50, 2003. doi: 10.1002/acs.730. − **One citation in pages** 3.

F. Previdi and M. Lovera. Identification of non-linear parametrically varying models using separable least squares. *International Journal of Control*, 77(16):1382–1392, 2004. − **One citation in pages** 3.

W. Qin and Q. Wang. An LPV approximation for admission control of an internet web server: Identification and control. *Control Engineering Practice*, 15(12):1457–1467, December 2007a. doi: doi:10.1016/j.conengprac.2007.02.006. − **2 citations in pages** 4 and 63.

W. Qin and Q. Wang. Modeling and control design for performance management of web servers via an lpv approach. *IEEE Transactions on Control Systems Technology*, 15(2):259–275, March 2007b. doi: 10.1109/TCST.2006.886433. − **2 citations in pages** 4 and 63.

Quanser Consulting Inc. *SRV03 Direct Drive Rotary Plant with Slipring: Position Servo & Planar Self-Erecting IP.* Quanser Consulting Inc., 1993. – **3 citations in pages** 16, 32, and 87.

W. J. Rugh and J. S. Shamma. Research on gain scheduling. *Automatica*, 36(10): 1401–1425, 2000. – **One citation in pages** 1.

J. S. Shamma and M. Athans. Guaranteed properties of gain scheduled control for linear parameter-varying plants. *Automatica*, 27(3):559–564, 1991. – **One citation in pages** 1.

J. Sjöberg and M. Viberg. Separable non-linear least-squares minimization – possible improvements for neural net fitting. In *IEEE Workshop Neural Networks and Signal Processing*, pages 345–354, 1997. – **2 citations in pages** 22 and 24.

T. Söderström and P. Stoica. *Instrumental Variable Methods for System Identification.* Springer-Verlag, 1983. – **3 citations in pages** 41, 42, and 50.

T. Söderström and P. Stoica. *System Identification.* Prentice-Hall, London, UK, 1989. – **One citation in pages** 29.

W.-H. Steeb. *Problems and Solutions in Introductory and Advanced Matrix Calculus.* World Scientific Publishing, 2006. – **One citation in pages** 105.

M. Steinbuch, R. van de Molengraft, and A. van der Voort. Experimental modeling and LPV control of a motion system. In *Proceeding of the American Control Conference*, pages 1374–1379, Denver, Colorado, June 4–6 2003. – **One citation in pages** 87.

J. Suykens, J. Vandewalle, and B. L. R. De Moor. *Artificial Neural Networks for Modelling and Control of Non-Linear Systems.* Kluwer Academic Publishers, 1996. – **One citation in pages** 13.

R. Tóth. *Modeling and Identification of Linear Parameter-Varying Systems – an Orthonormal Basis Function Approach.* PhD thesis, Delft University of Technology, 2008. – **3 citations in pages** 3, 4, and 45.

R. Tóth. *Modeling and Identification of Linear Parameter-Varying Systems.* Springer, 2010. – **3 citations in pages** 2, 64, and 67.

R. Tóth, F. Felici, P. S. C. Heuberger, and P. M. J. Van den Hof. Discrete time LPV I/O and state space representations, differences of behavior and pitfalls of interpolation. In *Proceedings of the European Control Conference*, pages 5418–5425, Kos, Greece, July 2–5 2007. – **4 citations in pages** 4, 63, 80, and 87.

R. Tóth, C. Lyzell, M. Enqvist, P. S. C. Heuberger, and P. M. J. Van den Hof. Order and structural depencence selection of LPV-ARX models using a nonnegative garrote approach. In *Proceedings of the 48th IEEE Conference on Decision and Control*, pages 7406–7409, Shanghai, P. R. China, December 16–18 2009a. – **One citation in pages** 34.

R. Tóth, J. Willems, P. S. C. Heuberger, and P. Van den Hof. A behavioral approach to LPV systems. In *Proceedings of the European Control Conference 2009*, pages 2015–2020, Budapest, Hungary, August 23–26 2009b. – **One citation in pages** 63.

R. Tóth, H. Abbas, and H. Werner. On the state-space realization of LPV input-output models: practical approaches. *IEEE Transactions on Control Systems Technology*, In print, 2011. – **2 citations in pages** 5 and 80.

P. Van den Hof and R. Schrama. An indirect method for transfer function estimation from closed loop data. *Automatica*, 29(6):1523–1527, 1993. – **3 citations in pages** 15, 25, and 26.

J.-W. van Wingerden and M. Verhaegen. Subspace identification of bilinear and LPV systems for open- and closed-loop data. *Automatica*, 45(2):372–381, 2009. doi: doi:10.1016/j.automatica.2008.08.015. – **2 citations in pages** 2 and 63.

V. Verdult. *Nonlinear System Identification: A State-space Approach.* PhD thesis, University of Twente, The Netherlands, 2002. – **2 citations in pages** 2 and 9.

V. Verdult and M. Verhaegen. Identification of multivariable linear parameter-varying systems based on subspace techniques. In *Proceedings of the 39th IEEE Conference on Decision and Control*, pages 1567–1572, Sydney, Australia, December 2000. – **One citation in pages** 2.

V. Verdult and M. Verhaegen. Identification of multivariable bilinear state space systems based on subspace techniques and separable least squares optimization. *International Journal of Control*, 74(18):1824–1836, 2001. doi: 10.1080/00207170110089806. – **One citation in pages** 63.

V. Verdult and M. Verhaegen. Subspace identification of multivariable linear parameter-varying systems. *Automatica*, 38:805–814, 2002. – **One citation in pages** 29.

E. I. Verriest. Algebraic theory for time variant linear systems: Modes, minimality and reachability and observability of interconnected systems. In *Proceedings of the 32nd Conference on Decision and Control*, pages 1349–1354, San Antonio, Texas, December 1993. – **6 citations in pages** 4, 63, 64, 65, 66, and 68.

M. Vidyasagar. *Control System Synthesis: A Factorization Approach.* The MIT Press, 1985. – **One citation in pages** 107.

G. Vinnicombe. Frequency domain uncertainty and the graph topology. *IEEE Tranactions on Automatic Control*, 38(9):1371–1383, 1993a. – **3 citations in pages** 5, 88, and 89.

G. Vinnicombe. *Measuring the Robustness of Feedback Systems.* PhD thesis, Cambridge University, Cambridge, UK., 1993b. – **2 citations in pages** 88 and 89.

M. G. Wassink, M. van de Wal, C. Scherer, and O. Bosgra. LPV control for a wafer stage: Beyond the theoretical solution. *Control Engineering Practice*, 13(2):231–245, 2005. – **2 citations in pages** 4 and 87.

X. Wei. *Advanced LPV Techniques for Diesel Engines.* PhD thesis, Joannes Kepler Universität, Linz, Austria, 2006. – **9 citations in pages** 2, 4, 11, 15, 19, 33, 63, 80, and 102.

X. Wei and L. del Re. Gain scheduled H_{infty} control for air path systems of diesel engines using LPV techniques. *IEEE Transactions on Control Systems Technology*, 15(3):406–415, May 2007. doi: 10.1109/TCST.2007.894633. – **One citation in pages** 87.

D. Westwick and R. E. Kearney. Separable least squares identification of nonlinear hammerstein models: Application to stretch reflex dynamics. *Annals of Biomedical Engineering*, 29(8):707–718, 2001. doi: 10.1114/1.1385806. – **One citation in pages** 24.

W. A. Wolovich. *Linear Multivariable Systems.* Springer-Verlag, 1974. – **2 citations in pages** 64 and 67.

G. D. Wood. *Control of Parameter-Dependent Mechanical Systems.* PhD thesis, Univerisity of Cambridge, 1995. – **2 citations in pages** 5 and 88.

R. Ylinen and K. Zenger. Conmputer aided analysis and design of time–varying systems. In *Computer Aided Systems Theory – EUROCAST '93*, volume 585/1992, pages 73–94. Springer, 1992. doi: 10.1007/BFb0021006. – **5 citations in pages** 64, 65, 66, 70, and 74.

E. Zerz. An algebraic analysis approach to linear time-varying systems. *IMA Journal of Applied Mathematical Control and Information*, 23:113–126, 2006. doi: 10.1093/imamci/dni047. – **One citation in pages** 64.

E. Zerz. State representations of time-varying linear systems. In *Radon Series on computational and Applied Mathematics*, volume 3, pages 235–251, 2007. – **2 citations in pages** 63 and 78.

Y. Zhu. Estimation of and N-L-N Hammerstein–Wiener model. *Automatica*, 38(9): 1607–1614, 2002. doi: 10.1016/S0005-1098(02)00062-6. – **2 citations in pages** 15 and 20.

Y. Zhu and Z. Xu. A method of LPV model idnetification for control. In *Proceedings of the 17th IFAC World Congress*, pages 5018–5023, Seoul, Korea, July 6–11 2008. – **2 citations in pages** 15 and 20.

List of Symbols and Abbreviations

A, B, C, D	State-space matrices, input-output polynomial matrices
A_o, B_o, C_o, D_o	State-space matrices, observable form
A_c, B_c, C_c, D_c	State-space matrices, reachable form
H	Noise model
F	Adaptation matrix gain
G	Plant
I	Identity matrix
J	Jacobian matrix
J_L	Jacobian matrix with respect to linear parameters
J_{NL}	Jacobian matrix with respect to nonlinear parameters
J_S	Jacobian matrix with respect to both linear and nonlinear parameters
K	Controller
N	Number of sample data
P_L	Orthogonal projection onto the column of J_l
$V, \tilde{V}$	Cost function
Z	Data set
a, b, c, d	Input-output representation coefficients
d	delay sample
e	White noise signal
f	Parameter dependent function
k	Sampling instant
l	Recursive algorithm step
max	Maximum
min	Minimum
n	Number of states
n_a	The order of past output polynomial
n_b	The order of past input polynomial
n_h	Number of hidden neurons
n_s	Number of harmonics
n_u	Number of input signals
n_y	Number of output signals
$n_{\mathfrak{p}}$	Number of constant parameters
$n_{\rho_{\text{ext}}}$	Number of external scheduling signals
$n_{\rho_{\text{int}}}$	Number of internal scheduling signals
n_ρ	Number of scheduling signals

n_θ	Number of scheduling parameters
n_η	Number of knots in a cubic spline function
q	Forward time-shift operator
u	Input vector
v	Disturbance signals
x	State vector
y	Output vector
z	z-transform

$\mathbb{C}$	Set of complex numbers
$\mathbf{E}$	Expectation
$\mathcal{H}_2$	Signal 2-norm of impulse response of system
$\mathcal{H}_\infty$	Induced 2-norm of a transfer function
$\mathcal{K}$	Field
$\mathcal{NL}$	Nonlinear system
$\mathbb{P}_\theta$	Scheduling Domain
$\mathfrak{P}$	Constant coefficient parameter matrix
$\mathbb{R}$	Set of real number
$\mathcal{S}$	System set
$\mathbb{Z}$	Integer set
$\mathbb{Z}_+$	Positive integer set
$\mathfrak{p}$	Constant coefficient parameter vector
$\mathfrak{p}_0$	True constant coefficient parameters
$\rho, \rho_{\text{ext}}, \rho_{\text{int}}$	Vector of scheduling signals
θ	Scheduling parameters
λ	forgetting factor
ψ	Vector of scheduling parameters (functions)
φ	Input-Ouput regressor vector
ϕ	LPV Input-Output regressor vector
ψ_{Poly}	Vector of scheduling parameters of polynomial functions
ψ_{sp}	Vector of scheduling parameters of cubic spline functions
η	Cubic spline knot position
ε	Prediction error
δ_ν	ν-gap metric
Ω	Open set

$\lVert \cdot \rVert$	Sequence-2-norm, matrix-2-norm
$\lVert \cdot \rVert_\infty$	Sequence-∞-norm, $\mathcal{H}_\infty$-norm
$\lVert \cdot \rVert_F$	Frobenius norm
$\star$	Transpose of the off-diagonal block matrix
$\otimes$	Kronecker product
$(\cdot)^T$	Transpose
∇	Gradient
∇^2	Hessian

■	End of example
□	End of proof
$\left[\begin{array}{c:c} A & B \\ \hdashline C & D \end{array}\right]$	To represent block partitioned matrix
$\left[\begin{array}{c\|c} A & B \\ \hline C & D \end{array}\right]$	Transfer function in terms of state-space matrices, input-output operator in terms of an LPV stat-space realization

Abbreviations

ADIP	Arm Driven Inverted Pendulum
ARX	Auto Regression with eXogenous input
BFT	Best FiT
BJ	Box-Jenkins
IV	Instrumental Variables
LPR	Left Polynomial Representation
LPV	Linear Parameter-Varying
LTI	Linear Time-Invariant
LTV	Linear Time-Varying
LRNN	Linear Recurrent Neural Network
LS	Least Squares
MIMO	Multi-Input Multi-Output
MOESP	Multivariable Output-Error State-Space
MSE	Mean Square Error
PAA	Parameter Adaptation Algorithm
PEM	Prediction Error Method
OE	Output Error
RIV	Refined Instrumental Variable
RLS	Recursive Least Squares
RPR	Right Polynomial Representation
SI	Spark-Ignited
SISO	Single-Input Single-Output
SIMO	Single-Input Multiple-Output
SLS	Separable Least-Squares
SNR	Signal to Noise Ratio
SRIV	Simplify Refined Instrumental Variable
SVD	Singular Value Decomposition
VAF	The Variance Accounted For

List of Refereed Publications

Published

J. Witt, S. Boonto and H. Werner, "Approximate model predictive control of a 3-DOF helicopter," in *Proceeding of the 46th IEEE Conference on Decision and Control*, New Orleans, LA, USA, December 12–14 2007, pp. 4501–4506.

S. Boonto and H. Werner, "Closed-Loop System Identification of LPV Input-Output Models – Application to an Arm-Driven Inverted pendulum," in *Proceeding of the 47th IEEE Conference on Decision and Control*, Cancun, Mexico,December 9–11 2008, pp. 2606–2611.

S. Boonto and H. Werner, "Closed-Loop Identification of LPV Models Using Cubic Splines with Application to an Arm-Driven Inverted Pendulum," in *Proceedings of the 2010 American Control Conference – ACC2010*, Baltimore, MD, USA ,June 30–July 02 2010, pp. 3100–3105.

I. Wior, S. Boonto, H. Abbas and H. Werner, "Modeling and Control of an Experimental pH Neutralization Plant using Neural Networks based Approximate Predictive Control," in *Proceedings of the 1st Virtual Control Conference*, Aalborg, Denmark, September 22 2010.

R. Körlin, S. Boonto, H. Werner and U. Starossek, "LMI-based Gain Scheduling for Bridge Flutter Control Using Eccentric Rotational Actuators," *Optimal Control Application & Method*, 2011. (accepted)

Submitted

A. Kominek, S. Boonto, M. Garwon, H. Werner and M. Schultalbers, "Low Complexity LPV Input-Output Identification and Control of a Turbocharged Combustion Engine," Submitted to *the 50th IEEE Conference on Decision and Control and European Control Conference (CDC-ECC 2011)*, Florida, USA, 2011.

Q. Liu, G. Kaiser, S. Boonto, H. Werner, F. Holzmann, B. Chretien and M. Korte, "Two-Degree-of-Freedom LPV Controller Design in Torque Vectoring of a through-the-Road Hybrid Electric Vehicle," Submitted to *the IEEE Conference on Decision and Control and European Control Conference (CDC-ECC 2011)*, Florida, USA, 2011.

Curriculum Vitae

Sudchai Boonto, born on 08.02.1972 in Samutprakarn, Thailand

Educational Background

1991–1995	Bachelor of Engineer (B.Eng.) in Electrical Engineering, King Mongkut's University of Technology Thonburi, Bangkok, Thailand
1999–2000	Master of Science (M.Sc) in Advanced Control Systems, UMIST, Manchester, UK.
2003–2010	Ph.D student in the Institute of Control Systems, Hamburg University of Technology, Germany.

Work Experience

1995–1999	Lecturer with the Faculty of Engineering, King Mongkut's University of Technology Thonburi, Bangkok, Thailand
2001–2003	Lecturer with the Faculty of Engineering, King Mongkut's University of Technology Thonburi, Bangkok, Thailand
2003–2010	Teaching Assistant of Neural and Genetic Computing for Control of Dynamic Systems, Hamburg University of Technology, Germany.