LPV System Identification and Order Reduction with Application to Turbocharged SI Engines

Vom Promotionsausschuss der
Technischen Universität Hamburg
zur Erlangung des akademischen Grades
Doktor-Ingenieur (Dr.-Ing.)

genehmigte Dissertation

von
Erik Schulz

aus
Berlin

2022

1. Gutachter: Prof. Dr. Herbert Werner
2. Gutachter: Dr. Hossam Abbas

Datum der mündlichen Prüfung: 14. Mai 2021

Abstract

English

The basis for the model-based design of linear parameter-varying (LPV) controllers is a model of the controlled system. However, models that approximate the system behavior well are often missing in industrial applications. In this thesis, methods for the identification and order reduction of discrete-time LPV models are presented. They are applied to measurements from a real system, the air path of a turbocharged gasoline engine, and show improved computational efficiency and/or provide models of higher accuracy compared to existing methods. Furthermore, the resulting LPV models are suitable for established control design tools.

Deutsch

Grundlage für den modellbasierten Entwurf von linear parameterveränderlichen (LPV) Reglern ist ein Modell von der zu regelnden Strecke. Bei industriellen Anwendungen fehlen jedoch häufig Modelle, die das Streckenverhalten gut annähern. In dieser Arbeit werden Methoden zur Identifikation und Ordnungsreduktion von zeitdiskreten LPV Modellen behandelt. Sie werden auf Messungen von einem realen System, dem Luftpfad eines aufgeladenen Ottomotors, angewendet und weisen im Vergleich zu bestehenden Methoden eine verbesserte Recheneffizienz auf und/oder liefern Modelle mit höherer Genauigkeit. Darüber hinaus sind die erhaltenen LPV Modelle für etablierte Reglerentwurfswerkzeuge geeignet.

Summary

In the present thesis the data-driven identification of discrete-time (DT) linear parameter-varying (LPV) models and the order reduction of the resulting models are investigated. The work is motivated by the fact that linear time-invariant (LTI) controllers may provide insufficient performance for certain nonlinear or operating point dependent systems, which can be represented as (quasi) LPV systems. In order to design an LPV controller for these systems, an LPV model is required that is suitable for control design and that approximates the system to be controlled sufficiently accurately.

The identification of DT LPV models is considered for models in the input-output (IO) or in the state-space (SS) representation. New identification methods for LPV IO model structures are presented. For certain model structures LTI methods can be used to identify LPV IO models and thus off-the-shelf LTI algorithms can be embedded in an LPV identification algorithm. Furthermore, a method for the identification of multi-input multi-output (MIMO) LPV IO models in Box-Jenkins structure is introduced. The performance of the identified models using the different IO model structures is compared in examples. In addition to the IO methods, the subspace identification (SID) method for DT LPV SS models is reviewed. Certain aspects of LPV IO and SID methods are compared such as model structure, scheduling parameter handling, and number of unknowns.

Due to limitations of real-time capable hardware and thus the requirement for computationally efficient controllers, the objective in this thesis is to obtain accurate and low-order LPV SS models, since the controller order depends on the model order in model-based unstructured control design. Furthermore, LPV SS models are required for established control design methods. Unfortunately, SS representations of LPV IO systems have a high order in the MIMO case. For this reason, model order reduction of DT LPV SS models is considered in this thesis using the Ho-Kalman approach. Instead of constructing the exponentially growing Hankel matrix, as it is used in a similar way in the LPV SID method, recursive matrices are used to calculate the (first) Hankel singular values. This improves the computational efficiency and reduces the approximation error in the calculation of the Hankel singular values. The introduced technique is compared to a linear matrix inequality based method, where the generalized Gramians are calculated and used for order reduction.

The introduced methods are tested on an academic example system and on two subsystems of an industrial application, the air path of a turbocharged gasoline engine. The air path describes how the air gets from the ambiance into the cylinder and from there back into the environment. It is equipped with several actuators that (in)directly influence the mass of fresh air in the cylinder and thus the torque. As subsystems of the air path, a MIMO system for modeling the intake manifold pressure and the mass of fresh air in the cylinder and a single-input single-output system for modeling the boost pressure are considered. The results of the identification and model order reduction methods are compared with state-of-the-art LPV identification and order reduction techniques.

Contents

1 Introduction **1**
- 1.1 Related Work . . . 2
- 1.2 Contributions . . . 3
- 1.3 Thesis Outline . . . 4

2 LPV Systems **7**
- 2.1 LPV System Representations . . . 9
  - 2.1.1 Input-Output Representations . . . 9
  - 2.1.2 State-Space Representations . . . 10
- 2.2 Dependency on the Scheduling Parameter . . . 10
  - 2.2.1 External and Internal Dependency . . . 10
  - 2.2.2 Static and Dynamic Dependency . . . 11
  - 2.2.3 General Functional Dependency . . . 12
  - 2.2.4 Affine Dependency . . . 13
  - 2.2.5 Determination of Scheduling Parameters and the Parameter Set . . . 14
- 2.3 LPV System Properties . . . 15
  - 2.3.1 Stability . . . 15
  - 2.3.2 Performance . . . 16
  - 2.3.3 Observability and Reachability . . . 17
- 2.4 Transformation of LPV Systems . . . 21
  - 2.4.1 Transformation between IO and SS Representations . . . 22
  - 2.4.2 Transformation of State-Space Representations . . . 27

3 Identification of LPV Models **31**
- 3.1 Preliminaries for LPV System Identification . . . 33
- 3.2 Review of LPV Identification Methods . . . 38
- 3.3 LPV Input-Output Model Identification . . . 39
  - 3.3.1 Model Structures . . . 39
  - 3.3.2 LPV ARX Model . . . 42
  - 3.3.3 LPV ARMAX Model . . . 46
  - 3.3.4 LPV OE and BJ Model . . . 51
- 3.4 LPV Subspace Identification . . . 55
  - 3.4.1 Model Structures . . . 55
  - 3.4.2 Problem Formulation . . . 56
- 3.5 Comparison of Input-Output and Subspace Identification . . . 58
  - 3.5.1 Relation of Model Structures . . . 58
  - 3.5.2 Noise Handling . . . 60
  - 3.5.3 Structure of Model Coefficient or Matrix Functions . . . 61
  - 3.5.4 Order of the Identified Model . . . 62
  - 3.5.5 Algorithms . . . 62

3.5.6 Scheduling Parameter Handling . . . 63
3.5.7 Number of Unknowns . . . 65
3.5.8 Regularization . . . 67
3.5.9 Summary . . . 68
3.6 Nonlinear Optimization . . . 68

4 Model Order Reduction of LPV Systems 71
4.1 General Procedure in Model Order Reduction . . . 72
4.2 Balanced Realization and Truncation of LPV Systems . . . 74
4.3 Ho-Kalman Reduction . . . 76
4.3.1 Reduction of DT LTI SS Models . . . 76
4.3.2 Reduction of DT LPV SS Models . . . 80
4.3.3 Analysis of the Reduction Scheme . . . 85

5 Application Examples on the Air Path 91
5.1 Modeling of the Turbocharged SI Engine . . . 92
5.2 Identification of the Turbocharged SI Engine . . . 95
5.2.1 LPV MIMO Identification in Suction Operation . . . 95
5.2.2 LPV SISO Identification in Charged Operation . . . 104

6 Summary and Future Work 113

A Appendix 119
A.1 Matrix Operations . . . 119
A.2 From IO to SS Domain . . . 120
A.2.1 Shifted Form . . . 120
A.2.2 Augmented Form . . . 121
A.2.3 Observability Form . . . 122
A.2.4 Extended Augmented Form . . . 123
A.2.5 Initial State . . . 123
A.3 LPV Subspace Identification . . . 124
A.3.1 Problem Formulation . . . 124
A.3.2 Problem Reduction . . . 129

Bibliography 133

Acronyms 143

List of Symbols 145

List of Publications 153

Curriculum Vitae 155

1 Introduction

The control of industrial applications is becoming more and more demanding. On the one hand, industrial applications are being continuously developed, automated, and improved and on the other hand, the system requirements are increasing. A good example of this is the internal combustion engine (ICE). The development of the ICE began with a mainly mechanical system and it is now equipped with a large number of sensors and actuators, which increase the possibilities for processing information and controlling the system. In addition to this increased flexibility, the demands for performance and efficiency of the ICE have also increased, e. g., for a powerful but fuel efficient engine.

In order to meet the increased requirements, a linear time-invariant (LTI) description of the industrial application is often no longer sufficient to adequately describe its nonlinear behavior. Accordingly, the use of LTI controllers becomes too imprecise and thus no longer appropriate so that there is a need for advanced, high-performance control methods, cf. [Dettori 2001; Genç 2002; Kwiatkowski 2007; Mohammadpour & Scherer 2012; Rawlings & Mayne 2016; Shamma 1988; Thiel 2019]. One approach frequently used in industry is classical gain-scheduling, i. e. the technical system is approximated locally at an operating point as an LTI system and individual LTI controllers are designed for the individual operating points, which are then interpolated with so-called scheduling parameters when traversing through the operating range [Leith & Leithead 2000b; Rugh & Shamma 2000]. The scheduling parameters then characterize the operating point. This approach is widely used in the, e. g., automotive and aerospace industry, cf. [Corno et al. 2010; Nichols et al. 1993]. However, stability and performance can be only guaranteed if the dynamics of the scheduling parameter is slow compared to the system dynamics, thus problems can occur for fast operating point changes [Leith & Leithead 1999; Shamma & Athans 1991].

With the introduction of the linear parameter-varying (LPV) system framework linear, operating point dependent systems can be described [Shamma 1988]. Well-known control design methods with stability and performance guarantees for (uncertain) LTI systems can be extended to LPV systems. Within the last 30 years, powerful methods have been developed to design model-based controllers [Apkarian et al. 1996; Scherer 2001; Wu 1995]. Moreover, it has been shown that not only systems that depend on an externally defined operating point can be described as LPV systems, but also systems that depend nonlinearly on their states, inputs, or outputs [Leith & Leithead 2000a]. By extending the LPV systems to such nonlinear systems the applicability of LPV methods has increased. Thus, the LPV approach is promising for the ICE, where the system can be considered in simplified form as operating point dependent system while it has an overall nonlinear behavior.

However, a disadvantage of the LPV control design methods is the need for an LPV model of the industrial application, which is to be controlled. The effort for theoretical modeling a nonlinear or LPV system is high and "at this moment, LPV modeling seems to be the most crucial bottleneck in the LPV controller design procedure" [Wassink et al. 2005]. This is a reason why in industry LPV models are either often missing or inaccurate so that a model-based control design is either not possible at all or – in the case of inaccurate models – leads to good

behavior in simulation but fails in the true system.

An alternative approach to obtain an LPV model is data-driven system identification. Similar to control design, advanced LTI identification methods are practically not suitable. Although local identification can be performed at several operating points, analogous to classical gain-scheduling, this approach suffers from the fact that the models are only locally valid and each has its own local state-space basis [Leith & Leithead 1999], [Theis 2018, Section 2.1] Interpolation between these models is necessary to obtain an LPV model and can lead to a complete misfit. Instead, methods for the identification of LPV models were developed, where commonly either an input-output or state-space model structure is assumed [Cox 2018; Tóth, Heuberger, et al. 2012; Wingerden & Verhaegen 2009].

If an LPV model is available and an LPV controller needs to be implemented, there is commonly the requirement for a low complexity controller or model, either because of computational problems in control design or because of implementation limitations on the real time hardware. Thus, a model of high accuracy with low complexity is desired for the control design and model order reduction becomes relevant, where usually balanced truncation methods for LPV system are studied [Abbas 2010; Theis 2018; Wood 1995].

For this reason, the identification and order reduction of LPV models are investigated in the present thesis with the application to subsystems of a turbocharged ICE. Accordingly, existing approaches are analyzed, compared, and further developed in this work. Therefore, the objective of this work is the computationally efficient, global, data-based identification of LPV state-space models of low order, which are suitable for already developed LPV control synthesis methods. The term computationally efficient refers to a fast computation and low memory requirement. In addition, the identification should scale well in relation to the number of samples and the model dimensions, i. e. the number of inputs, outputs, states, and scheduling parameters. Global means that the data-generating system is excited over its entire operating range, i. e. with varying scheduling parameters that, if possible, cover the entire range of permissible scheduling parameters. Low order means that the state-space representation has few states, which is usually not true in the multi-input multi-output (MIMO) case when it is obtained from an input-output identification approach. Thus, the state order reduction of discrete-time LPV state-space models is essential and a two-step procedure is proposed in this thesis consisting of identifying an LPV input-output model and reducing the order of its state-space representation. Because most LPV control design methods require a state-space model with static dependency on the scheduling parameters, only LPV models with such a state-space representation are considered in identification.

1.1 Related Work

The identification of discrete-time LPV state-space models is dealt with in [Verdult 2002], where the data matrices in the identification problem grow exponentially with the order of the model. Therefore, the method has been improved in [Wingerden & Verhaegen 2009], which significantly reduces the computational effort and increases the practical applicability. In the present thesis, this improved method serves as state-of-the-art of subspace identification method for LPV state-space models and is used as a benchmark for the proposed identification methods. Furthermore, there are nonlinear, non-convex optimization methods to refine identified LPV state-space models, such as in [Gunes 2018], where tensor techniques are used. These methods require a significantly higher computational load and are therefore not considered in detail in

this thesis.

In contrast to the direct identification of LPV state-space models, the identification of LPV input-output models is computationally more efficient. In [Abbas 2010; Bamieh & Giarré 2002; Laurain et al. 2010; Tóth, Heuberger, et al. 2012] the identification of discrete-time LPV input-output models is considered and in [Cox 2018] different methods for identifying LPV input-output and state-space models are presented. The identification methods and the proposed three-step identification scheme in [Cox 2018] are similar to some contributions in this thesis. In contrast to [Cox 2018], a method for the identification of MIMO LPV models in Box-Jenkins structure is introduced in this thesis. Furthermore, the relation to LTI identification methods is shown, e. g., for ARMAX model structures in Section 3.3.3 resulting in good identification results despite the structural model limitations, see Section 5.2.1.

The model order reduction of continuous-time LPV state-space models was introduced in [Wood 1995], where balanced truncation is applied using the generalized Gramians. This approach is adopted in [Abbas 2010] to the discrete-time case, which serves as state-of-the-art benchmark in this thesis using scheduling parameter independent generalized Gramians. Furthermore, the Ho-Kalman reduction scheme for discrete-time LPV systems is introduced in [Abbas 2010]. This scheme is also used in [Cox 2018] in a computationally more efficient way. However, it only becomes practically applicable for LPV systems (with many scheduling parameters) with the simplifications presented in Section 4.3.2. Another model order reduction approach is introduced in [Theis 2018], where continuous-time LPV state-space models are reduced with (partially) parameter-varying oblique projection matrices that are based on the local Gramians.

The model-based control of subsystems in the ICE is considered in, e. g., [Beckmann 2015]. In that thesis, the nonlinear model used for control design is based on a simplified version of a semi-physical model and not on system identification. A similar approach is followed in [Genç 2002; Kwiatkowski 2007; Laurain 2017], where (quasi) LPV models are obtained from a theoretical modeling approach followed by designing an LPV controller. There are few publications about the identification of LPV models of subsystems in the ICE using experimental data, which is considered in this thesis. One of the few publications is the research paper [Salcedo & Martínez 2008], in which local LTI models are first identified and then an LPV model is approximated to them.

1.2 Contributions

The main contributions in this thesis concern the identification of discrete-time LPV models, the state order reduction of discrete-time LPV state-space models, and the application to an industrial system, more precisely subsystems of the air path of a turbocharged combustion engine.

- The relation between the identification of LTI and LPV models is shown in Section 3.3.3. Consequently, off-the-shelf input-output LTI identification methods can be used to identify LPV ARX and ARMAX models. This results in Algorithm 3 as a contribution of this work, cf. [Schulz et al. 2018].

  Furthermore, a new LPV input-output identification algorithm is presented in Section 3.3.4, capable of identifying single-input single-output (SISO) and MIMO LPV models in Box-Jenkins or output error structure, see Algorithm 5 and [Schulz et al. 2018]. It is shown in

practical examples that Algorithm 5 provides low-order models of higher accuracy than those identified with a simpler input-output model structure, see Fig. 5.14 in Section 5.2.2.

A comparison and assessment of LPV input-output and state-of-the-art LPV subspace identification methods are given, cf. [Schulz et al. 2016]. Differences in the two approaches are worked out and limits are shown that arise either from assumptions or from necessary computations.

- A computationally efficient state order reduction scheme for discrete-time LPV state-space models with affine scheduling parameter dependence is presented in Algorithm 7 in Section 4.3.2, cf. [Schulz et al. 2017]. As key feature, the problem of exponentially growing matrices is solved, which allows a smaller approximation error in order reduction.

- The combination of LPV input-output identification methods with the computationally efficient model order reduction scheme enables the efficient identification of LPV state-space models of high order and/or with many scheduling parameters. In comparison, the computational load of such models increases for state-of-the-art LPV subspace identification methods and the memory required to store the data matrices can exceed the limits of an average desktop computer.

- LPV models are identified and evaluated in Chapter 5 for a real-world example system, the air path of a turbocharged ICE, using the introduced and state-of-the-art methods.

1.3 Thesis Outline

The thesis consists of six chapters including introduction in this chapter and the summary and future work in Chapter 6.

In Chapter 2, preliminary material on the LPV framework is given. As an entry, the historical development of the LPV framework is briefly presented and afterwards definitions for LPV systems in input-output and state-space representations are provided. Various dependencies on the scheduling parameters are discussed, where the dynamic and static dependence are addressed throughout this thesis. Essential LPV system properties such as stability, performance, observability, and controllability are presented, which are addressed in the remainder of this thesis. The properties are given for discrete-time systems, on which the present thesis is focused. Finally, transformations between input-output and state-space representations are presented as well as the state transformation in the state-space domain. These results are used later in identification and model order reduction.

The identification of discrete-time LPV models is investigated in Chapter 3. First, preliminary material to system identification is given, where some fundamental ideas and approaches are reviewed. A brief overview of LPV system identification methods is provided before LPV input-output identification methods for the common model structures are presented. A collection of existing and new algorithms for different model structure is given and the relation to LTI identification methods is investigated. The well-known subspace identification approaches for LPV state-space models are briefly presented and subsequently the two methods LPV input-output and subspace identification are compared in detail for specific model structures. Finally, it is referred to nonlinear optimization techniques that can be used to refine identified LPV models.

In Chapter 4, model order reduction of discrete-time LPV systems in state-space representation is studied. Model order reduction is motivated by the fact that in the MIMO case state-space representations of models identified with input-output methods are commonly non minimal. To reduce the model order, an (approximate) transformation into a balanced realization is sought such that a reduction can be carried out relatively simply while maintaining a high accuracy of the reduced-order model. In particular, the Ho-Kalman reduction technique is considered, in which the Hankel matrix is employed that appears in a similar form in subspace identification.

The presented methods are tested on an industrial application in Chapter 5. The air path of a turbocharged gasoline internal combustion engine is investigated, which is first described physically and phenomenologically to gain an understanding of the system. Two subsystems are studied in more detail. Firstly, a MIMO system is considered, where the throttle and inlet valve are used to actuate the intake manifold pressure and the amount of fresh air in the cylinder. Secondly, a SISO model is identified, where the input is the wastegate position and the output is the boost manifold pressure. The effectiveness of the proposed LPV model identification and order reduction methods are demonstrated. The results are based on measurements taken on an engine test bench.

2 LPV Systems

In this chapter, a review of LPV systems is given and the system theoretical basics needed for this thesis are presented. First, however, an outline of the history and development of LPV theory is given.

The class of LPV systems has been investigated since the late 1980s, early 1990s and the LPV framework emerged from the idea of extending known LTI approaches and applying them to operating point dependent systems, as used in classical gain-scheduling, e. g., in flight control [Hyde & Glover 1993; Nichols et al. 1993; Shamma 1988; Shamma & Athans 1991]. In this gain-scheduling approach a family of interpolated LTI controllers is used to control the system of interest, cf. [Hyde & Glover 1991]. For this purpose, the system is approximated at certain equilibrium operating points by LTI models for which individually LTI controllers are designed to ensure the performance requirements locally at the corresponding equilibrium. Consequently, interpolation of the linear controllers is required to control the nonlinear system on its entire operating range. Unfortunately, stability and performance guarantees are lost as a result and rapid changes of operating points can cause problems in this regard. Even an undesired dynamic behavior can occur due to the interpolation or approximation errors, cf. [Leith & Leithead 2000a; Shamma & Athans 1992].

In contrast, stability and performance can be guaranteed with the LPV approach. For the LPV system approach, it is assumed that the system changes with a time-varying scheduling parameter vector that is unknown in advance but can be measured online. In the first publications [Shamma 1988; Shamma & Athans 1991], external scheduling parameters were assumed. However, the scheduling parameter vector can also contain internal system variables, for example states, which results in a so-called quasi LPV system. This allows, on the one hand, the representation of nonlinear systems as LPV systems. On the other hand, the design of optimal controllers w. r. t. the $\mathcal{H}_\infty$ or $\mathcal{H}_2$ system norm in LTI theory has been extended to LPV systems utilizing optimization techniques with linear matrix inequalities (LMIs) constraints [Apkarian et al. 1996]. Thus, in terms of system representation and control design, the LPV theory is an important link between the world of nonlinear and linear system theory. The relationship between LPV, LTI, and linear time-varying (LTV) systems is depicted in [Apkarian et al. 1996, Section 8.4].

In the first LPV control design approaches, the set of admissible parameter values is assumed bounded and the scheduling parameters are allowed to vary arbitrarily fast [Becker et al. 1993; Packard et al. 1993]. Next, in the mid-1990s, the less conservative case of bounded parameter variations is considered [Apkarian et al. 1996; Feron et al. 1995; Gahinet et al. 1994; Wu 1995] and synthesis conditions for more complicated scheduling parameter dependencies than affine ones have been derived, e. g., rational or general dependencies [Apkarian et al. 1996; Apkarian & Gahinet 1995; Wu & Dong 2006]. This was also the time when research on discrete-time LPV controllers began, cf. [Apkarian et al. 2000; De Caigny et al. 2010; Gahinet et al. 1994; Wu & Dong 2006], which requires a considerable different treatment compared to the continuous-time domain as it is recently shown in [Cox et al. 2018]. Further improvements in LPV control design have been made over the last years, e. g., to reduce conservatism in the approaches, cf. [Hoffmann

2016; Scherer 2001, 2013]. Recent LPV control trends are, for example, design of structured LPV controllers, cf. [Adegas & Stoustrup 2011; Emedi & Karimi 2014, 2016; Kwiatkowski et al. 2009; Wollnack et al. 2013], combination of LPV systems with model predictive control, cf. [Abbas et al. 2015; Besselmann et al. 2012; Cisneros & Werner 2017], and design or direct learning of LPV controllers in input-output form, cf. [Formentin et al. 2016; Wollnack 2018; Wollnack & Werner 2016].

Usually, the LPV controller designs are based on a model, which means that an LPV model of the underlying system is required. This can either be obtained from a nonlinear system description, which in turn requires a nonlinear model, cf. [Giarré et al. 2006; Jung & Glover 2006]. General approaches to represent nonlinear models in the LPV framework are given in [Tóth 2010, Section 7.4] and [Kwiatkowski 2007, Section 3.3]. A different method to obtain an LPV model is data-driven LPV system identification, cf. [Verdult 2002, Chapter 6]. The first publications on the identification of LPV models were published in the mid to late 1990s [Lee & Poolla 1999; Nemani et al. 1995]. The identification of input-output and state-space models is investigated, e. g., in [Bamieh & Giarré 2001; Verdult 2002]. Again, there are numerous publications from 2010 to the present, where approaches to problem reduction [Cox & Tóth 2016; Wingerden & Verhaegen 2009], (model structure) extensions [Cerone et al. 2012; Laurain et al. 2010; Tóth, Heuberger, et al. 2012], and combination with other techniques [Boonto 2011; Schulz et al. 2017; Tóth et al. 2011] are presented. An overview of LPV system identification methods is given in [Bachnas et al. 2014] and [Cox 2018, Section 1.2 and Fig. 1.2].

Due to the attractive property of controlling nonlinear systems with advanced LTI methods, LPV theory is used in industrial research applications with an increase in publications in the early 2010s [Hoffmann & Werner 2015; Mohammadpour & Scherer 2012]. For example, LPV controllers are successfully used in robotics [Saupe & Pfifer 2011; Yu et al. 2002], aerospace [Bodenheimer et al. 1995; Marcos & Balas 2004] and automotive applications [Bartels et al. 2013; Jung & Glover 2006; Wei & Re 2007]. With regard to the automotive industry, however, this is mainly for research purposes and has hardly been used in series production to date. A detailed survey of applications including grouping into application types and synthesis techniques as well as a timeline is given in [Hoffmann & Werner 2015]. Compared to the literature on the application of LPV controllers, there is less literature on the practical application of LPV system identification. Commonly, numerical examples are considered to demonstrate effectiveness, cf. [Giarré et al. 2006; Laurain et al. 2010; Verdult & Verhaegen 2002]. The identification based on data of a nonlinear simulation model is considered in [Bachnas et al. 2014; Buchholz & Larimore 2013; Verdult et al. 2004], while real measurement data is considered in [Kominek et al. 2012; Schulz et al. 2016].

The outline of this chapter is as follows. The input-output and state-space representations of LPV systems are presented in Section 2.1. These LPV system representations are described with a general scheduling parameter dependency, which is discussed in more detail in Section 2.2. Particular dependencies are presented and in the remainder of this thesis it is focused on static affine scheduling parameter dependencies. In Section 2.3 basic properties of LPV systems are discussed, such as stability, observability, and reachability. These concepts form the basis for some of the methods in the following chapters. Since the LPV system properties are predominantly formulated for state-space representations, the transformation between LPV systems in input-output and state-space representation is discussed in Section 2.4. Particular LPV system representations are given and the state transformation of LPV state-space models is presented.

2.1 LPV System Representations

In this section, two different LPV system representations are presented: the input-output (IO) and the state-space (SS) representation. These two representations are in this thesis of great interest since the input-output representation has some computational advantages in system identification, whereas the state-space representation is commonly used for control design. In the following only discrete-time (DT) system representations are presented, because the focus of this thesis is on the identification of LPV models based on sampled signals and the implementation of controllers on a real-time hardware.

2.1.1 Input-Output Representations

The signals of a dynamical LPV system are separated in input signals $u(k) : \mathbb{N} \to \mathbb{R}^{n_u}$, output signals $y(k) : \mathbb{N} \to \mathbb{R}^{n_y}$, and scheduling signals $\rho(k) : \mathbb{N} \to \mathbb{R}^{n_\rho}$, where the sample instant is denoted by k. The input signals are free and the user or controller can actuate the LPV system by them. The output signals are not free and the measurements or information, which are obtained of the actuated system. The scheduling signals influence the system behavior. As discussed later, they can be external signals (free), internal signals (non-free), e. g., inputs and/or outputs, or a combination of both. All three signals can be confined to sets, however in this thesis only the definition of the confined set of scheduling signals is required.

Definition 2.1 (Set of admissible scheduling parameter trajectories). The set of admissible scheduling parameter trajectories $\mathcal{F}_\rho$ is the set of all unknown functions of time $\rho(k) : \mathbb{N} \to \mathbb{P}_\rho \subset \mathbb{R}^{n_\rho}$, where the set of admissible scheduling parameter values $\mathbb{P}_\rho$ is compact.

Note that rate constraints exist in DT due to compactness of $\mathbb{P}_\rho$ and they can be added to Definition 2.1. These rate constraints can be exploited to formulate less conservative conditions for evaluating LPV system properties. However, in this thesis rate constraints are not exploited.

Next, the definition for a DT LPV system in input-output (IO) representation is given, which is based on [Tóth 2010, Definition 3.18].

Definition 2.2 (DT LPV IO Representation). The discrete-time input-output representation of an LPV system G with scheduling parameter vector ρ, input vector u, and output vector y is denoted by $\mathcal{R}_{\mathrm{IO}}$ and is defined as a parameter-varying difference equation system of order n_a:

$$\mathcal{R}_{\mathrm{IO}}(G(\rho)) \;:\; \sum_{i=0}^{n_a} a_i(\rho(k))\mathrm{q}^i y(k) = \sum_{j=0}^{n_b} b_j(\rho(k))\mathrm{q}^j u(k) \;, \tag{2.1}$$

where q is the forward time shift operator, i. e. $\mathrm{q}^i y(k) = y(k+i)$, and $a_i(\rho) : \mathbb{P}_\rho \to \mathbb{R}^{n_y \times n_y}$ and $b_j(\rho) : \mathbb{P}_\rho \to \mathbb{R}^{n_y \times n_u}$ are the static coefficient matrix functions continuous on ρ. Moreover, the system coefficient matrix functions a_{n_a} and b_{n_b} must be non-singular for all values of $\rho \in \mathbb{P}_\rho$ and the orders are constrained to $n_a \geq n_b \geq 0$ and $n_a > 0$.

In order to express the sums in (2.1) of Definition 2.2, the scheduling parameter dependent matrix polynomials

$$A(\mathrm{q}, \rho) = \sum_{i=0}^{n_a} a_i(\rho(k))\mathrm{q}^i \;, \qquad B(\mathrm{q}, \rho) = \sum_{j=0}^{n_b} b_j(\rho(k))\mathrm{q}^j \tag{2.2}$$

can be introduced. The order of each polynomial is given by the highest exponent of the operator, e. g., the order of $A(\mathrm{q}, \rho)$ is n_a. Commonly, the polynomial $A(\mathrm{q}, \rho)$ is monic, which means that the zeroth coefficient function a_0 is identity, i. e. $a_0(\rho) = I$ with I the identity matrix.

2.1.2 State-Space Representations

In addition to the input, output, and scheduling signals of an LPV system in IO representation, latent variables can be introduced. These latent variables can be virtual and they correspond to inner variables or states $x : \mathbb{N} \to \mathbb{R}^{n_x}$ of the system. The order of a system in the state-space domain is given by the number of the states $n_x \in \mathbb{N}^+$. The following definition for discrete-time LPV systems in state-space (SS) representation is based on [Tóth 2010, Definition 3.19].

Definition 2.3 (DT LPV SS Representation). The discrete-time state-space representation of an LPV system G with scheduling parameter vector ρ, input vector u, output vector y, and state vector x is denoted by $\mathcal{R}_{\mathrm{SS}}$ and is defined as a first-order parameter-varying difference equation system:

$$\mathcal{R}_{\mathrm{SS}}(G(\rho)) \;:\; \begin{cases} \mathrm{q}x(k) = A(\rho(k))x(k) + B(\rho(k))u(k) \\ y(k) = C(\rho(k))x(k) + D(\rho(k))u(k) \end{cases}, \tag{2.3}$$

where $A(\rho) : \mathbb{P}_\rho \to \mathbb{R}^{n_x \times n_x}$, $B(\rho) : \mathbb{P}_\rho \to \mathbb{R}^{n_x \times n_u}$, $C(\rho) : \mathbb{P}_\rho \to \mathbb{R}^{n_y \times n_x}$, and $D(\rho) : \mathbb{P}_\rho \to \mathbb{R}^{n_y \times n_u}$ are the static state, input, output and feedthrough matrix functions, respectively, continuous on ρ.

A shorthand notation for an LPV system in SS representation is

$$\mathcal{R}_{\mathrm{SS}}(G(\rho)) \;:\; \left[\begin{array}{c|c} A(\rho) & B(\rho) \\ \hline C(\rho) & D(\rho) \end{array}\right]. \tag{2.4}$$

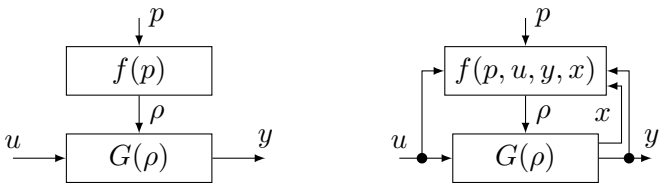
Note that the notation in the IO domain is similar to the SS domain, however $A(\mathrm{q}, \rho)$ denotes a polynomial in the input-output domain while $A(\rho)$ denotes the state matrix function in the state-space domain.

2.2 Dependency on the Scheduling Parameter

Up to here, a general dependency of the LPV system on the scheduling parameter has been considered. It is only assumed, that the matrices or coefficients of the system are static and continuous functions of the time-dependent and online measurable scheduling parameter vector $\rho^{\mathrm{T}} = [\rho_1 \quad \rho_2 \quad \dots \quad \rho_{n_\rho}]$, which is element of an a priori known compact set $\mathbb{P}_\rho$. To simplify the analysis of LPV systems, the synthesis of LPV controllers, or the identification of LPV models, particular scheduling parameter dependencies are presented in the following.

2.2.1 External and Internal Dependency

The scheduling parameter vector can depend on internal and external scheduling signals w. r. t. the system. Internal scheduling signals are assumed to depend on inputs u, outputs y, or states x. In contrast, external scheduling signals are assumed to depend on system independent signals, which are denoted by $p \in \mathbb{P}_p \subset \mathbb{R}^{n_p}$ with $\mathbb{P}_p$ being compact.



(a) LPV system **(b)** Quasi LPV system

Figure 2.1: Different scheduling parameter vector dependencies of an LPV system with dependency on (a) solely external scheduling signals and (b) internal scheduling signals as well

If the scheduling signal depends on internal signals, the LPV system is referred to as quasi LPV system, cf. Fig. 2.1 illustrating the difference between LPV and quasi LPV systems. Quasi LPV systems "often suffer from inherent couplings of the scheduling signals and thus from less tight parameter sets" [Kwiatkowski 2007, Chapter 2]. Moreover, the compact scheduling parameter set $\mathbb{P}_\rho$, which is assumed to be given a priori, depends on how and in which operating range the (controlled) system is driven.

With the extension of the scheduling parameter vector to depend on inputs, outputs, or states, it becomes possible to represent some nonlinear systems as quasi LPV systems. Thus, LPV methods, e. g., for control design can be applied to these nonlinear systems.

For the sake of simplicity, the term LPV system is used from here on to refer to systems with both types of scheduling parameter dependencies.

2.2.2 Static and Dynamic Dependency

In Definitions 2.2 and 2.3, LPV systems with static dependence on the scheduling parameter are assumed, i. e. they depend only on the scheduling parameter ρ at time instant k.

Definition 2.4 (Static Scheduling Parameter Dependency). A discrete-time LPV IO system (2.1) or SS model (2.3) has a static scheduling parameter dependency, if it depends only on the instantaneous value of the scheduling signal.

The time dependence of an LPV system on its scheduling parameters can be dynamic as well, where in this case the DT LPV system depends on at least one scheduling parameter shifted in time. In case of a dynamic dependency on the scheduling parameter, the operator $\diamond$ is used, which is introduced in [Tóth 2010, Section 3.1]. At the example of the matrix function a_i from Definition 2.2 for LPV IO system representations, the operator $\diamond$ is a shorthand notation for

$$a_i \diamond \rho(k) = a_i(\rho(k - k_-), \rho(k - k_- + 1), \ldots, \rho(k + k_+)) \ , \tag{2.5}$$

where $k_-, k_+ \in \mathbb{N}$ denote the dependency on past and future scheduling parameters with $k_- > 0$ and/or $k_+ > 0$. At this point, it is important to note that commutativity of the operator $\diamond$ with the operator q is not given, i. e. $\mathrm{q}^j(a_i \diamond \rho) \neq (a_i \diamond \rho)\mathrm{q}^j$. The time dependency must be taken into account accordingly, i. e. $\mathrm{q}^j(a_i \diamond \rho) = (a_i \diamond \mathrm{q}^j\rho)\mathrm{q}^j$ for dynamic dependency and $\mathrm{q}^j a_i(\rho(k)) = a_i(\rho(k + j))\mathrm{q}^j$ for static dependency, respectively.

The operator $\diamond$ covers both cases: static and dynamic scheduling parameter dependencies. In order to illustrate this, the following simple example is considered.

Example 2.1. Assume a first-order SISO DT LPV SS model G, see Definition 2.3, with $n_x = n_y = n_u = n_\rho = 1$ and state-space matrix functions

$$A \diamond \rho(k) = A(\rho(k), \mathrm{q}^{-1}\rho(k)) = 1 - \rho(k) + \frac{1}{\rho(k-1)} \ , \qquad B \diamond \rho(k) = 1 \tag{2.6a}$$

$$C \diamond \rho(k) = C(\rho(k)) = \cos(\rho(k)) \ , \qquad D \diamond \rho(k) = 1 \ . \tag{2.6b}$$

The state matrix function $A \diamond \rho$ has a dynamic scheduling parameter dependency while the output matrix function $C \diamond \rho = C(\rho)$ has a static parameter dependency. The input and feedthrough matrix functions are constant. Consequently, the LPV system $G \diamond \rho$ has a dynamic scheduling parameter dependency.

Methods for the analysis of an LPV system or for control design commonly require LPV systems in SS or IO representation with static scheduling parameter dependency. Therefore, in this thesis mainly LPV systems are considered that meet this requirement.

2.2.3 General Functional Dependency

In this and the following subsection, different types of functional dependencies of the LPV system on the scheduling parameters are discussed.

For a general functional scheduling parameter dependency, only the parameter set $\mathbb{P}_\rho$ is required to be compact and known a priori. Thus, w. r. t. Fig. 2.1, the image of each function $f_i(p, u, y, x)$ for each scheduling parameter ρ_i must be bounded.

The general type of scheduling parameter dependency is the less restrictive one. However, for analysis or synthesis an ad hoc gridding-based approach is required, where the parameter set is covered by a set of points (grid). The number of grid points increases exponentially with the number of scheduling parameters. Thus, the general dependency is limited to only a few scheduling parameters due to complexity of the system analysis and synthesis conditions that must be satisfied for each grid point. Furthermore, the gridding-based approach provides only approximate stability or performance guarantees. [Hoffmann 2016, Section 2.3], [Leith & Leithead 2000b; Packard et al. 1993; Scorletti & Ghaoui 1998]

To avoid the gridding problem, in addition to the already made assumption of a compact parameter set, it is also assumed that the scheduling parameter dependencies are rational or affine. In this thesis, only the affine dependency is considered and to obtain such a dependency a transformation function

$$f_{\theta\rho} \ : \ \mathbb{P}_\rho \to \mathbb{P}_\theta \ , \quad \rho \mapsto f_{\theta\rho}(\rho) = \theta \tag{2.7}$$

can be introduced in order to rewrite the LPV system with general parameter dependency to have affine dependency on the scheduling parameter vector θ. In many cases, this function is not unique and additional scheduling parameters must be introduced to obtain the affine dependency, hence $n_\rho \leq n_\theta$. The following notation for the scheduling parameters systems is used in this thesis:

- $p \in \mathbb{P}_p \subset \mathbb{R}^{n_p}$ for external scheduling parameters,
- $\rho \in \mathbb{P}_\rho \subset \mathbb{R}^{n_\rho}$ for a general dependency,
- $\theta \in \mathbb{P}_\theta \subset \mathbb{R}^{n_\theta}$ for an affine dependency, and

- $\mu \in \mathbb{P}_\mu \subset \mathbb{R}^{n_\mu}$ for a linear dependency with $\mu^{\mathrm{T}} = [1 \quad \theta^{\mathrm{T}}]$

of the LPV system on the scheduling parameter vector. The linear scheduling parameter μ is introduced to simplify the notation for LPV systems with affine scheduling parameter dependencies. It is preferably used in system identification to distinguish the scheduling parameters μ from the model coefficients that are denoted by θ.

2.2.4 Affine Dependency

Affine scheduling parameter dependency is more restrictive than the general dependence. However, LPV systems with affine parameter dependency can be interpreted as interpolation of several LTI systems[1)], which is appealing for novices, and simplifies the identification of LPV models.

LPV SS models $G(\theta)$ with affine parameter dependency are given by

$$G(\theta) \;:\; \begin{bmatrix} \mathrm{q}x(k) \\ y(k) \end{bmatrix} = \left(\begin{bmatrix} A_0 & B_0 \\ C_0 & D_0 \end{bmatrix} + \sum_{i=1}^{n_\theta} \theta_i(k) \begin{bmatrix} A_i & B_i \\ C_i & D_i \end{bmatrix} \right) \begin{bmatrix} x(k) \\ u(k) \end{bmatrix} \tag{2.8}$$

or equivalently

$$G(\mu) \;:\; \begin{bmatrix} \mathrm{q}x(k) \\ y(k) \end{bmatrix} = \sum_{i=0}^{n_\theta} \mu_{i+1}(k) \begin{bmatrix} A_i & B_i \\ C_i & D_i \end{bmatrix} \begin{bmatrix} x(k) \\ u(k) \end{bmatrix} \quad \text{with} \quad \mu(k) = \begin{bmatrix} 1 \\ \theta(k) \end{bmatrix} \tag{2.9}$$

with constant matrices $A_i \in \mathbb{R}^{n_x \times n_x}$, $B_i \in \mathbb{R}^{n_x \times n_u}$, $C_i \in \mathbb{R}^{n_y \times n_x}$, and $D_i \in \mathbb{R}^{n_y \times n_u}$ for $i \in \{0, 1, 2, \ldots, n_\theta\}$. Similarly, the coefficient matrix functions $a_i(\theta)$ and $b_j(\theta)$ of LPV IO systems in (2.1) can be expressed by a sum of products of constant matrices with entries of the scheduling parameter μ.

When it comes to system analysis or control design, a convex parameter set with small overbounding of the true set of admissible scheduling parameters is preferred, because constraints on (the vertices of) the convex parameter set must be met. Assuming that the affine scheduling parameters and thus their bounds are independent of each other, the admissible convex parameter set is a hyperrectangle. However, if the scheduling parameters θ_i or their bounds depend on each other, a convex polytope can be defined, which covers the admissible parameter values more closely, i. e. with less overbounding. Then, the scheduling parameters can be expressed by the convex combination

$$\theta(k) = \sum_{j=1}^{n_\alpha} \alpha_j(k) \theta_{\mathrm{v},j} \quad \text{with} \quad \sum_{j=1}^{n_\alpha} \alpha_j = 1 \;, \quad \alpha_j \geq 0 \quad \text{and} \quad \alpha \in \mathbb{P}_\alpha \subset \mathbb{R}^{n_\alpha} \;, \tag{2.10}$$

where $\theta_{\mathrm{v},j}$ and $\alpha_j(k)$ are the j-th vertex of the polytope and the corresponding time-varying convex coordinate, respectively. Hence, the LPV system in (2.8) can be written as

$$\begin{bmatrix} \mathrm{q}x(k) \\ y(k) \end{bmatrix} = \sum_{j=1}^{n_\alpha} \alpha_j(k) \begin{bmatrix} A_{\mathrm{v},j} & B_{\mathrm{v},j} \\ C_{\mathrm{v},j} & D_{\mathrm{v},j} \end{bmatrix} \begin{bmatrix} x(k) \\ u(k) \end{bmatrix} \tag{2.11a}$$

$$\begin{bmatrix} A_{\mathrm{v},j} & B_{\mathrm{v},j} \\ C_{\mathrm{v},j} & D_{\mathrm{v},j} \end{bmatrix} = \begin{bmatrix} A_0 & B_0 \\ C_0 & D_0 \end{bmatrix} + \sum_{i=1}^{n_\theta} \theta_{\mathrm{v},j}^{(i)} \begin{bmatrix} A_i & B_i \\ C_i & D_i \end{bmatrix} \;, \tag{2.11b}$$

[1)] However, especially in case of quasi LPV systems, these LTI systems do not represent the local dynamic behavior.

where $\theta_{\mathrm{v},j}^{(i)}$ is the i-th entry of the j-th vertex vector of the parameter polytope. Thus, an LPV system with affine parameter dependency is an interpolation of LTI systems at the vertices of the polytope. The vertices $\theta_{\mathrm{v},j}$ define a convex scheduling parameter set and the convex coordinates α_j are the weighting factors for each vertex.

In the following, an example is given to illustrate the different scheduling parameter dependencies and the conversion using a mapping $f_{\theta\rho}(\rho)$. In addition, the problem of conservatism caused by additional scheduling parameters is illustrated.

Example 2.2. Consider an LPV system in SS representation with one scheduling parameter $\rho = \rho_1 \in [0, 1]$ and state matrix function

$$A(\rho) = \begin{bmatrix} \rho_1 & f(\rho_1) \\ 0 & \rho_1^2 \end{bmatrix} , \tag{2.12}$$

where $f(\rho_1)$ represents, e. g., a lookup table. In order to obtain an LPV system with affine scheduling parameter dependency, the mapping

$$\theta^{\mathrm{T}} = \begin{bmatrix} \theta_1 & \theta_2 & \theta_3 \end{bmatrix} = f_{\theta\rho}(\rho) = \begin{bmatrix} \rho_1 & \rho_1^2 & f(\rho_1) \end{bmatrix} , \quad \theta \in \mathbb{P}_\theta \subset \mathbb{R}^3 , \tag{2.13}$$

hence conservatism is increased due to the additional independent scheduling parameters θ_2 and θ_3. From a numerical point of view, it makes sense to scale the affine scheduling parameters such that they have a similar magnitude. The conservatism due to additional scheduling parameters can be reduced by considering the parameter set as polytope instead of a hyperrectangle, where the polytope usually has more vertices than the hyperrectangle to cover the set of admissible parameter values more closely. Note that computing the convex coordinates is more complicated for a polytope in comparison to a hyperrectangle.

2.2.5 Determination of Scheduling Parameters and the Parameter Set

The choice of scheduling parameters and the functional dependence are typically not unique for quasi LPV systems. For example, this is dealt with in [Kwiatkowski 2007, Sections 2.3 and 3.2]. The author provides definitions for the equivalence of LPV systems with different scheduling parameter vectors. In addition, an example is given to demonstrate that a nonlinear system can have different LPV system representations with a different number of scheduling parameters.

Besides the choice of scheduling parameters, the parameter set also plays an important role. The set must be specified a priori, which is strictly speaking not possible for quasi LPV systems. Furthermore, the restriction to rational or affine parameter dependencies may require the introduction of additional (independent) scheduling parameters. This often leads to an overbounding of the initial parameter set and can have a considerable influence on the achievable control quality. Overbounding is discussed in, e. g., [Kwiatkowski 2007, Chapters 3 and 4]. Measures of overbounding are presented as well as the parameter set mapping method based on a principal component analysis with which the overbounding and the number of scheduling parameters can be reduced via coordinate transformation.

Note that a graphical visualization of the parameter set is possible for up to three (affine) scheduling parameters, making the overbounding and approximation of parameter sets with polytopes more intuitive.

2.3 LPV System Properties

So far, LPV system representations and their dependency on the scheduling parameter are discussed. In the following, particular properties of LPV systems are defined and (conservative) conditions are given with which these properties can be verified. The properties are fundamental for system analysis, model order reduction, and control design, among others.

Although LPV systems have a similar representation to LTI systems, some definitions and analysis concepts from LTI theory are not directly transferable to LPV systems. On the other hand, if an LPV system is (approximately) locally represented by an LTI system, the properties of the LTI system are denoted as local properties of the LPV system. However, they do not allow to draw conclusions about the properties of the LPV system.

2.3.1 Stability

Although LTI systems can be obtained from an LPV system by freezing the scheduling parameter, stability of LPV systems cannot be shown by analyzing the pole locations of the resulting LTI systems. This is demonstrated by examples in, e. g., [Shamma 2012] and [Apkarian et al. 1996, Section 6.2]. Even if no statement can be made about the stability of an LPV system, it is denoted as frozen or locally stable, if all LTI systems are stable that are obtained from the LPV system by freezing the scheduling parameter.

In order to derive a stability test for LPV systems, Lyapunov theory is employed. Then, asymptotic stability of the equilibrium at the origin of a DT LPV system in SS representation can be ensured. An equilibrium of a DT LPV SS model is given by $x_{\mathrm{eq}}(k+1) = x_{\mathrm{eq}}(k)$ and it is asymptotically stable, if it is both attractive and stable. This means that starting from an initial point in the neighborhood of the equilibrium x_{eq}, the state x converges to this x_{eq} and while doing so x remains within a certain distance from x_{eq} [Briat 2015, Section 2.2], [Hoffmann 2016, Section 2.2]. Note that formal definitions for stability, attractiveness, and asymptotic stability are given in, e. g., [Briat 2015, Section 2.2] and, since they correspond to the general understanding, they are omitted here. In the following definition, conditions are given with which asymptotic stability of an LPV system can be ensured. The conditions are based on [Wu 1995, Section 3.2] and adapted to the discrete-time case.

Definition 2.5 (Quadratic Stability, [Abbas 2010; Daafouz & Bernussou 2001]). The autonomous LPV system in SS representation

$$x_{k+1} = A(\rho_k)x_k \tag{2.14}$$

is called quadratically stable around the origin $x = 0$ for all admissible trajectories $\rho \in \mathcal{F}_\rho$, if there exists a quadratic Lyapunov function $V = x^{\mathrm{T}}Px$ with a constant and symmetric Lyapunov matrix $P = P^{\mathrm{T}}$, which satisfies

$$V(x,\rho) > 0\ , \quad \mathrm{q}V(x,\rho) - V(x,\rho) < 0\ , \quad \forall x \neq 0\ , \quad \text{and} \quad V(0,\rho) = 0\ . \tag{2.15}$$

Above inequalities are satisfied, if for all $\rho \in \mathbb{P}_\rho$ the inequalities

$$P \succ 0\ , \quad A^{\mathrm{T}}(\rho)PA(\rho) - P \prec 0 \tag{2.16}$$

are satisfied.

Note that the existence conditions of a Lyapunov matrix function P in (2.16) is only sufficient for stability, i. e. a system is not necessarily unstable at the origin, if no Lyapunov matrix P can be found.

Instead of a constant Lyapunov matrix P, a scheduling parameter dependent Lyapunov matrix function $P(\rho)$ can be assumed. Then, on the one hand, the conditions (2.16) become less conservative. On the other hand, this increases complexity of analysis and controller synthesis, since the number of unknowns increases and the inequalities must be satisfied for all admissible parameters and their rates. Nevertheless, parameter dependent Lyapunov matrix functions are outside the scope of this thesis and the interested reader is referred to, e. g., [De Caigny et al. 2010; Oliveira & Peres 2008].

Note that for nonlinear systems that can be represented by quasi LPV systems with state dependent scheduling parameters, asymptotic stability for equilibria other than the origin cannot be guaranteed with the conditions in Definition 2.5 [Koelewijn et al. 2020]. This means that other stability concepts, for example incremental stability, are required to ensure stability (and performance) when these nonlinear systems are operated with an LPV controller outside the origin [Koelewijn et al. 2019].

2.3.2 Performance

From a control engineering point of view, not only stability but also performance is important. Many performance measures are available in the literature, where usually performance of a system is measured by the ratio of the size of certain signals [Zhou et al. 1996, Chapter 4]. They are used to analyze systems and to find optimal controllers w. r. t. the performance criterion. For LTI systems, the most common performance measures are the $\mathcal{H}_2$ and the $\mathcal{H}_\infty$ system norm, see [Skogestad & Postlethwaite 2007, Section 4.10] for definitions of these two system norms. Here only the induced ℓ_2 system norm is considered, which can be seen as the energy gain from a system input u to a system output y and is equivalent to the $\mathcal{H}_\infty$ system norm in the LTI case. Other performance and control synthesis conditions can be found, e. g., in [Apkarian et al. 2000] and [White et al. 2013, Chapters 2 and 3] for the DT domain and [Apkarian et al. 1996, Chapter 11] for the continuous-time (CT) domain.

First, the ℓ_2 signal space is defined based on [Desoer & Vidyasagar 1975, Section 2.2].

Definition 2.6 (Signal Space ℓ_2, [Desoer & Vidyasagar 1975]). The space ℓ_2 is defined as the set of all square-summable signals, i. e. DT signals for which the quantity

$$\|x\|_{\ell_2} = \sqrt{\sum_{k=-\infty}^{\infty} x_k^{\mathrm{T}} x_k} \tag{2.17}$$

is finite. In this case $\|x\|_{\ell_2}$ is said to be the ℓ_2 signal norm of the DT signal $x \in \ell_2$.

Note that with a slight abuse of notation, $\|x\|_{\ell_2}$ is also used for local square-summable signals, i. e. signals with limited time interval. For example, the space $\ell_2[0, \infty)$ is the set of all DT signals with finite $\|x\|_{\ell_2}^2 = \sum_{k=0}^{\infty} x_k^{\mathrm{T}} x_k$.

In the following, a definition of the induced ℓ_2 system norm for LPV systems is given that is based on [Wu 1995, Definition 3.3.1], which is given for CT LPV systems.

Definition 2.7 (Induced ℓ_2 System Norm for LPV Systems, [Wu 1995]). The induced ℓ_2 system norm of a quadratically stable DT LPV SS model $G(\rho)$ in (2.3) with static scheduling

parameter dependence and zero initial conditions is given by

$$\|G(\rho)\|_{\mathrm{i},\ell_2} = \sup_{\substack{u\in\ell_2\setminus\{0\}\\ \rho\in\mathcal{F}_\rho}} \frac{\|y\|_{\ell_2}}{\|u\|_{\ell_2}} = \sup_{\substack{u\in\ell_2\setminus\{0\}\\ \rho\in\mathcal{F}_\rho}} \frac{\sqrt{\sum_{k=0}^{\infty} y^{\mathrm{T}}(k)y(k)}}{\sqrt{\sum_{k=0}^{\infty} u^{\mathrm{T}}(k)u(k)}} \,. \tag{2.18}$$

In above definition, $u \in \ell_2\setminus\{0\}$ means that $0 < \|u\|_{\ell_2} < \infty$, i. e. the energy in the input signal is non-zero and finite. Thus, the induced ℓ_2 system norm can be interpreted as the maximal amplification of finite energy input signals by the LPV system over all admissible parameter trajectories. In the LTI case, the induced ℓ_2 system norm is defined equivalently (without the scheduling parameters). Moreover, the induced ℓ_2 system norm is for LTI systems equal to the $\mathcal{H}_\infty$ norm and is determined by the peak of the maximum singular value of G over all frequencies [Skogestad & Postlethwaite 2007, Section 4.10].

An upper bound on the induced ℓ_2 system norm can be obtained from the bounded real lemma for LPV systems, which also implies a sufficient condition for quadratic stability of the considered LPV system and is given in the following.

Theorem 2.1 (Induced ℓ_2 System Norm Bound for LPV Systems, [Apkarian et al. 1995]). If there exists a Lyapunov matrix $P = P^{\mathrm{T}} \succ 0$ and a performance index $\gamma > 0$ such that for all $\rho_k \in \mathbb{P}_\rho$ the inequality

$$\begin{bmatrix} A^{\mathrm{T}}(\rho_k)PA(\rho_k) - P & A^{\mathrm{T}}(\rho_k)PB(\rho_k) & C^{\mathrm{T}}(\rho_k) \\ B^{\mathrm{T}}(\rho_k)PA(\rho_k) & B^{\mathrm{T}}(\rho_k)PB(\rho_k) - \gamma I & D^{\mathrm{T}}(\rho_k) \\ C(\rho_k) & D(\rho_k) & -\gamma I \end{bmatrix} \prec 0 \tag{2.19}$$

is satisfied, then the LPV system in state-space realization (2.3) with static scheduling parameter dependency is quadratically stable and its induced ℓ_2 system norm from input u to output y is smaller than γ.

Proof. The proof is given in [Apkarian et al. 1995]. □

The restriction to a constant Lyapunov matrix P simplifies the conditions in Theorem 2.1, however, such a matrix may not exist. Less conservative conditions can be obtained by allowing a scheduling parameter dependent Lyapunov matrix or by introducing a slack matrix. Again, these extensions are outside the scope of this thesis and the interested reader is referred to [De Caigny et al. 2010; Oliveira et al. 2002].

Theorem 2.1 represents conditions to analyze a DT LPV system but the conditions can also be used for controller synthesis. Then, however, the inequalities in Theorem 2.1 are no longer LMIs, since the closed-loop state-space matrices contain the unknown controller matrices and they are multiplied by the unknown Lyapunov matrix for the closed-loop system. The bilinear matrix inequalities can be converted back to LMIs by multiplying from the left and right by transformation matrices and using a change of variables. This is shown, e. g., in [Apkarian et al. 1996, Section 11.3] for the CT domain and is similar in the DT domain.

2.3.3 Observability and Reachability

Observability and reachability are basic properties of linear dynamic systems in SS representation. They are important for, e. g., state estimation, control, subspace identification, and model order reduction. Recall that a DT LTI system is observable, if the initial states of the system can be

uniquely constructed from the output signal (and input signal, if it is actuated) in a sufficiently long finite time horizon. Accordingly, a DT LTI system is reachable, if the states of the system can be driven by an unconstrained input signal from initial states to any other final states in the state-space in a sufficiently long finite time horizon. [Gu 2012, Section 3.1]

The concepts and definitions of the two system properties observability and reachability are similar for LPV and LTI SS models. However, the main difference between LTI and LPV systems is that in the LPV case the scheduling signal "acts as an extra 'time-axis' of the system" [Tóth 2010, p. 81]. Consequently, there are differently strict definitions for observability and reachability: LPV SS models can be complete observable/reachable, i. e. for all (admissible) scheduling parameter trajectories, or structural observable/reachable, which is a weaker, almost everywhere property, where the LPV system is not observable/reachable for specific scheduling parameter trajectories [Tóth 2010, Section 3.3].

Based on the ideas for LTI and LTV systems, definitions for matrix functions are provided in the following. The definitions are based on [Tóth 2010, Definitions 3.39 and 3.40] and the matrix functions can be used to determine observability or reachability of DT LPV SS models.

Definition 2.8 (State-Observability Matrix Function, [Tóth 2010]). The i-step state-observability matrix function of a DT LPV SS model (2.3) with dynamic scheduling parameter dependency is given by

$$\mathsf{O}_i^{\mathrm{T}} \diamond \rho = \begin{bmatrix} \mathsf{o}_1^{\mathrm{T}} \diamond \rho & \mathsf{o}_2^{\mathrm{T}} \diamond \rho & \dots & \mathsf{o}_i^{\mathrm{T}} \diamond \rho \end{bmatrix} \in \mathbb{R}^{n_x \times (in_y)} \tag{2.20}$$

with $\mathsf{o}_1 \diamond \rho = C \diamond \rho$ and for $l \in \{1, 2, \dots, i-1\}$

$$\mathsf{o}_{l+1} \diamond \rho = (\mathsf{o}_l \diamond \mathrm{q}\rho)(A \diamond \rho) \ . \tag{2.21}$$

Remark 2.1. *As in the LTI case, see [Zhou et al. 1996, Section 8.7], the observability operator* Ψ_{o} *with*

$$(\Psi_{\mathrm{o}} \diamond \rho(0))x(0) = \begin{bmatrix} y(0) \\ y(1) \\ y(2) \\ \vdots \end{bmatrix} = \begin{bmatrix} \mathsf{o}_1 \diamond \rho(0) \\ \mathsf{o}_2 \diamond \rho(0) \\ \mathsf{o}_3 \diamond \rho(0) \\ \vdots \end{bmatrix} x(0) = (\mathsf{O}_\infty \diamond \rho(0))x(0) \in \ell_2[0, \infty) \tag{2.22}$$

is given by the state-observability matrix function O_∞*. The observability operator* Ψ_{o} *maps the initial state* $x(0)$ *of a quadratically stable LPV system* (2.3) *that is non-actuated, i. e.* $u(k) = 0$ *for* $k \geq 0$*, on the systems future output trajectory for a given scheduling parameter trajectory* $\rho \in \mathcal{F}_\rho$*.*

Definition 2.9 (State-Reachability Matrix Function, [Tóth 2010]). The j-step state-reachability matrix function of a DT LPV SS model (2.3) with dynamic scheduling parameter dependency is given by

$$\mathsf{R}_j \diamond \rho = \begin{bmatrix} \mathsf{r}_1 \diamond \rho & \mathsf{r}_2 \diamond \rho & \dots & \mathsf{r}_j \diamond \rho \end{bmatrix} \in \mathbb{R}^{n_x \times (jn_u)} \tag{2.23}$$

with $\mathsf{r}_1 \diamond \rho = B \diamond \rho$ and for $l \in \{1, 2, \dots, j-1\}$

$$\mathsf{r}_{l+1} \diamond \rho = (A \diamond \rho)(\mathsf{r}_l \diamond \mathrm{q}^{-1}\rho) \ . \tag{2.24}$$

Remark 2.2. *As in the LTI case, see [Zhou et al. 1996, Section 8.7], the controllability operator* Ψ_c *with*

$$x(0) = (\Psi_\mathrm{c} \diamond \rho(-1)) \begin{bmatrix} u(-1) \\ u(-2) \\ u(-3) \\ \vdots \end{bmatrix} = \underbrace{\begin{bmatrix} \mathsf{r}_1 \diamond \rho(-1) & \mathsf{r}_2 \diamond \rho(-1) & \mathsf{r}_3 \diamond \rho(-1) & \dots \end{bmatrix}}_{\mathsf{R}_\infty \diamond \rho(-1)} \begin{bmatrix} u(-1) \\ u(-2) \\ u(-3) \\ \vdots \end{bmatrix} \in \mathbb{R}^{n_x} \tag{2.25}$$

is given by the state-reachability matrix function R_∞*. The controllability operator* Ψ_c *maps an input signal* u *in* $\ell_2(-\infty, 0)$ *on the initial state* $x(0)$ *for a given scheduling parameter trajectory* $\rho \in \mathcal{F}_\rho$*.*

Note that for LPV SS models with static scheduling parameter dependency the functions $\mathsf{o}_1 \diamond \rho$ and $\mathsf{r}_1 \diamond \rho$ have no dynamic dependency on the scheduling parameter, thus they are $\mathsf{o}_1(\rho)$ and $\mathsf{r}_1(\rho)$. However, the remaining o_l and r_l with $l \geq 2$ have a dynamic scheduling parameter dependence.

Structural state-observability is given, if $\mathsf{O}_{n_x} \diamond \rho$ has full rank in the functional sense [Tóth 2010, Section 3.3]. Note that this does not guarantee complete state-observability. Hence, for specific scheduling parameter trajectories, $\mathsf{O}_{n_x} \diamond \rho$ may not be invertible and reconstruction of the state by the n_x-step observability matrix is not possible.

Complete state-observability is given, if $\operatorname{rank}(\mathsf{O}_i \diamond \rho(k)) = n_x$ for all $k \in \mathbb{N}$ and all admissible scheduling parameter trajectories, where i is finite and possibly greater than n_x [Tóth 2010, Section 3.3]. If, moreover, the LPV SS representation has constant observability rank, i. e. the rank of the i-step observability matrix function remains the same from a certain $i \leq n_x$ on, see [Silverman & Meadows 1969], complete state-observability is given by $\operatorname{rank}(\mathsf{O}_{n_x} \diamond \rho(k)) = n_x$ [Tóth 2010, Section 3.3]. Thus, the condition for complete state-observability of LPV SS models is only equivalent to that of LTI SS models, if the LPV system has a constant observability rank representation.

The considerations and conditions for reachability are similar to those for observability. Thus, even if the rank of $\mathsf{R}_j \diamond \rho$ for $j = n_x$ is not full along each scheduling parameter trajectory (it may be full for $j > n_x$), it does not automatically mean that the LPV SS model is not completely state-reachable. Only if the LPV system has a constant reachability rank, then $\operatorname{rank}(\mathsf{R}_{n_x} \diamond \rho(k)) = n_x$ must be satisfied for reachability [Tóth 2010, Section 3.3].

Instead of the above given state-observability and -reachability matrix functions, observability and controllability Gramian matrix functions can be used to test for observability and reachability. In [Tóth 2010, Section 3.3], a definition for CT and DT parameter-varying Gramians is given on a finite time interval. Then, an LPV system is completely state-observable/reachable, if these parameter-varying Gramians have full rank for any finite time interval and each scheduling trajectory. Instead of parameter-varying Gramians on a finite time interval, upper bounds of them are considered on an infinite interval for CT LPV systems in [Wood 1995, Section 7.3]. These upper bounds are a conservative approximation and they are denoted as generalized Gramians, see [Beck et al. 1996] for their introduction for DT LTI systems. Existence conditions for these matrix functions for DT LPV SS models are given in [Abbas 2010, Section 3.3], cf. [Lall & Beck 2003] for DT LTV systems. In order to derive the conditions, it is assumed that the LPV system is quadratically stable. The following theorems are based on [Abbas 2010, Theorems 3.1 and 3.2] but simplified to scheduling parameter independent generalized Gramians.

Theorem 2.2 (Generalized Observability Gramian, [Abbas 2010]). Given a state-space realization of a DT LPV system with static parameter dependency (2.3), an initial state $x(0) = x_0$, and a real matrix $X_\mathrm{o} = X_\mathrm{o}^\mathrm{T} \succ 0$, denoted as generalized observability Gramian, satisfying

$$A^\mathrm{T}(\rho_k) X_\mathrm{o} A(\rho_k) - X_\mathrm{o} + C^\mathrm{T}(\rho_k) C(\rho_k) \prec 0 \quad \forall \rho_k \in \mathbb{P}_\rho \ , \tag{2.26}$$

then the LPV system is quadratically stable. Moreover, if $u(k) = 0$ for all $k \geq 0$, the energy in the output signal is bounded from above by

$$\|y\|_{\ell_2}^2 < \max x_0^\mathrm{T} X_\mathrm{o} x_0 \tag{2.27}$$

for all admissible scheduling parameter trajectories $\rho \in \mathcal{F}_\rho$.

Proof. The proof is given in [Abbas 2010, Section 3.3] for a scheduling parameter dependent generalized observability Gramian, i. e. $X_\mathrm{o}(\rho)$. However, it can be used for a constant generalized Gramian X_o as well. □

Theorem 2.3 (Generalized Controllability Gramian, [Abbas 2010]). Given a state-space realization of a DT LPV system with static parameter dependency (2.3), an initial state $x(0) = x_0$, and a real matrix $X_\mathrm{c} = X_\mathrm{c}^\mathrm{T} \succ 0$, denoted as generalized reachability or controllability Gramian, satisfying

$$A(\rho_k) X_\mathrm{c} A^\mathrm{T}(\rho_k) - X_\mathrm{c} + B(\rho_k) B^\mathrm{T}(\rho_k) \prec 0 \quad \forall \rho_k \in \mathbb{P}_\rho \ , \tag{2.28}$$

then the LPV system is quadratically stable and the energy required to drive the system from $x(-\infty) = 0$ to $x(0) = x_0$ with the input $u(k) \in \ell_2(-\infty, 0)$ is bounded from below by

$$\|u\|_{\ell_2}^2 > \max x_0^\mathrm{T} X_\mathrm{c}^{-1} x_0 \tag{2.29}$$

for all admissible scheduling parameter trajectories $\rho \in \mathcal{F}_\rho$.

Proof. The proof is given in [Abbas 2010, Section 3.3] for a scheduling parameter dependent generalized controllability Gramian, i. e. $X_\mathrm{c}(\rho)$. However, it can be used for a constant generalized Gramian X_c as well. □

Remark 2.3. *The observability and controllability Gramians in the LTI case are given by the observability, respectively controllability, operator as*

$$X_\mathrm{o} = \Psi_\mathrm{o}^\mathrm{T} \Psi_\mathrm{o} \ , \qquad\qquad X_\mathrm{c} = \Psi_\mathrm{c} \Psi_\mathrm{c}^\mathrm{T} \ , \tag{2.30}$$

see [Zhou et al. 1996, Section 8.1]. The generalized observability and controllability Gramians are an upper bound, i. e. for the LPV case in Theorems 2.2 and 2.3

$$X_\mathrm{o} \succ (\Psi_\mathrm{o}^\mathrm{T} \diamond \rho)(\Psi_\mathrm{o} \diamond \rho) \ , \qquad\qquad X_\mathrm{c} \succ (\Psi_\mathrm{c} \diamond \mathrm{q}^{-1}\rho)(\Psi_\mathrm{c}^\mathrm{T} \diamond \mathrm{q}^{-1}\rho) \tag{2.31}$$

for all $\rho \in \mathcal{F}_\rho$. In the observability case, this can be concluded from (2.22) *and* (2.27) *or the proof of [Abbas 2010, Theorem 3.1] and in the controllability case from* (2.25) *and* (2.29) *or the proof of [Abbas 2010, Theorem 3.2].*

Remark 2.4. *For DT LPV SS models with static and affine scheduling parameter dependence it is possible to compute the generalized Gramians without gridding the parameter set. For this purpose, the Schur complement can be applied to the LMIs in* (2.26) *and* (2.28)*, which become*

$$\begin{bmatrix} -X_\mathrm{o} & 0 & X_\mathrm{o}A(\theta) \\ 0 & -I & C(\theta) \\ A^\mathrm{T}(\theta)X_\mathrm{o} & C^\mathrm{T}(\theta) & -X_\mathrm{o} \end{bmatrix} \prec 0 \ , \qquad \begin{bmatrix} -X_\mathrm{c} & 0 & X_\mathrm{c}A^\mathrm{T}(\theta) \\ 0 & -I & B^\mathrm{T}(\theta) \\ A(\theta)X_\mathrm{c} & B(\theta) & -X_\mathrm{c} \end{bmatrix} \prec 0 \ . \tag{2.32}$$

As a result, the LMIs become linear in the affine scheduling parameters θ so that only the vertices of the scheduling parameter set must be checked.

The generalized Gramians are required to be positive definite, i. e. they have full rank on the entire parameter set, and thus if they exist, the system is completely state-observable/reachable, [Theis et al. 2018].

Note that the generalized Gramians in Theorems 2.2 and 2.3 are not unique. Hence, it is common to introduce further constraints or a cost function in order to derive unique generalized Gramians. Besides, the generalized Gramians are a conservative quadratic stability test, since each generalized Gramian is also a Lyapunov matrix, compare (2.26) with (2.16). The geometric interpretation of constant generalized Gramians is as follows: X_o and X_c^{-1} define the maximal inner ellipsoids with center at origin and which are inside all other parameter dependent ellipsoids. Hence, depending on the variation of the shapes of the parameter dependent ellipsoids, it is conceivable that these constant generalized Gramians can be quite conservative estimates of (2.27) and (2.29). In the following an example is considered in order to illustrate the conservatism of a scheduling parameter independent generalized Gramian for an LPV system.

Example 2.3. The geometric interpretation of the parameter independent generalized observability Gramian is the maximal inner ellipsoid with center at the origin and which bound from below the energy in the output signal for a given initial state. As an example, the DT LPV SS model

$$G(\theta) \ : \ \left[\begin{array}{c} \mathrm{q}x_1 \\ \mathrm{q}x_2 \\ \hline y \end{array}\right] = \left[\begin{array}{cc|c} 0.9 & 0 & 0.1 \\ 0 & 0.99 & 0.1 \\ \hline \theta & 1 & 0 \end{array}\right] \left[\begin{array}{c} x_1 \\ x_2 \\ \hline u \end{array}\right] \ , \quad \theta \in [0.5, 2] \tag{2.33}$$

is considered. First, the observability Gramians for frozen scheduling parameters are computed using LTI methods, e. g., the `gram` command in MATLAB. The corresponding ellipsoids are plotted in Fig. 2.2 (dark blue, thin) for unit energy in the output signal, where the largest ellipsoid belongs to $\theta = 0.5$ and the smallest to $\theta = 2$. Note that these ellipsoids are the maximal achievable ones in the LPV case. Next, the scheduling parameter independent generalized observability Gramian X_o is computed satisfying the LMI (2.26). The ellipsoid for this generalized Gramian is also plotted in Fig. 2.2 (light blue, thick) and, as already mentioned, it is the result of an optimization problem finding the maximal inner ellipsoid. Then, the energy in the output signal is less than one for any initial state on/in this inner ellipsoid for any scheduling parameter.

2.4 Transformation of LPV Systems

In the preceding section, properties of LPV systems in SS representation are given. In order to verify these properties also for LPV systems in IO representation, it is useful to find LPV systems

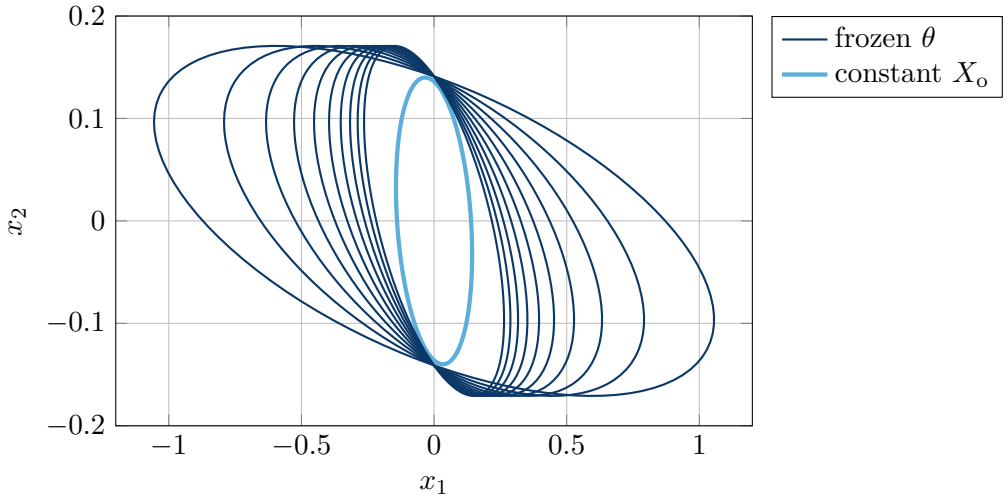


Figure 2.2: Initial states for which the energy in the output signal is one (LTI case) or less (LPV case) for gridded (frozen) scheduling parameters using LTI methods (dark blue) and for a scheduling parameter independent generalized observability Gramian using LPV methods (light blue)

in IO representation that have an equivalent SS representation with static and affine scheduling parameter dependence. These LPV system representations are also of great interest for the combination of identification and control design because some LPV identification methods assume IO model structures whereas LPV SS models are typically assumed for control design.

Besides the transformation between IO and SS representations, also state transformations are of interest. This concerns especially model order reduction methods, which are dealt with in Chapter 4. Therefore, the basics for state transformations of LPV SS models are presented in Section 2.4.2.

2.4.1 Transformation between IO and SS Representations

For general LPV IO system representations, transformations to LPV SS representation are available, see [Tóth 2010, Sections 4.2 and 4.3]. However, only particular LPV IO system representations have the desired static scheduling parameter dependence in the SS representation. Note that the transformations can be applied in both directions such that certain LPV SS representations have a simple IO representation.

As depicted in [Tóth 2010, Section 4.3], the objective is to construct a state-map for a given LPV IO representation that defines an LPV SS representation with identical input-output behavior. Then, the LPV IO and the LPV SS representations are denoted as IO equivalent, i. e. they possess the identical IO behavior, see [Tóth et al. 2007, definition 3]. In order to obtain such a SS representation from an IO system, latent variables (states) must be introduced. However, in the LPV case this requires more attention compared to the LTI case, since the system coefficient functions are scheduling parameter (and thus time) dependent and do not commute with forward time shift operator q. Therefore, a transformation from one domain into the other usually causes dynamic scheduling parameter dependency, even if the original system has static dependence [Tóth et al. 2007]. Furthermore, in [Tóth et al. 2007] equations for equivalent transformations of DT LPV systems with affine scheduling parameter dependency

from SS to IO domain and vice versa are given. The results are presented for SISO systems, however according to the authors they can be easily extended to the MIMO case. In order to illustrate the influence of dynamic scheduling parameter dependency, the following example from [Tóth et al. 2007] is considered.

Example 2.4. The DT LPV system in IO representation

$$G_{\mathrm{IO}}(p) \;:\; y(k) = -p(k)y(k-1) - p(k)y(k-2) + p(k)u(k-1) \tag{2.34}$$

has a static dependency on the external scheduling parameter $p \in \mathbb{P}_p = [-0.5, 1]$. The canonical LPV SS representations of (2.34) are

$$G_{\mathrm{SS}}^{\mathrm{o}} \diamond p \;:\; \left[\begin{array}{cc|c} 0 & 1 & p(k+1) \\ -p(k+2) & -p(k+2) & -p(k+2)p(k+1) \\ \hline 1 & 0 & 0 \end{array}\right] \tag{2.35}$$

$$G_{\mathrm{SS}}^{\mathrm{r}} \diamond p \;:\; \left[\begin{array}{cc|c} 0 & -p(k-1) & 1 \\ 1 & -p(k-1) & 0 \\ \hline p(k) & -p(k)p(k-1) & 0 \end{array}\right] \tag{2.36}$$

for the observability and reachability form, respectively, where the time dependence of the scheduling parameter is considered during the transformation. Neglecting the time dependence of p and simply applying the transformation procedure from LTI systems results in these LPV SS models

$$G_{\mathrm{SS}}^{\mathrm{o,LTI}}(p) \;:\; \left[\begin{array}{cc|c} 0 & 1 & p(k) \\ -p(k) & -p(k) & -p^2(k) \\ \hline 1 & 0 & 0 \end{array}\right], \quad G_{\mathrm{SS}}^{\mathrm{r,LTI}}(p) \;:\; \left[\begin{array}{cc|c} 0 & -p(k) & 1 \\ 1 & -p(k) & 0 \\ \hline p(k) & -p^2(k) & 0 \end{array}\right]. \tag{2.37}$$

The systems $G_{\mathrm{SS}}^{\mathrm{o,LTI}}$ and $G_{\mathrm{SS}}^{\mathrm{r,LTI}}$ have a static scheduling parameter dependency, whereas the dependency is dynamic for $G_{\mathrm{SS}}^{\mathrm{o}}$ and $G_{\mathrm{SS}}^{\mathrm{r}}$. Moreover, the observability and reachability forms $G_{\mathrm{SS}}^{\mathrm{o}}$ and $G_{\mathrm{SS}}^{\mathrm{r}}$ have different dynamic dependencies on $p(k)$, where causality of the system w. r. t. the scheduling parameter is violated in the observability form.

In Fig. 2.3 simulation results are shown for

$$u(k) = \sin(0.5k) \quad \text{and} \quad p(k) = \min\left(0.4 - 0.9\cos(0.3k), 1\right). \tag{2.38}$$

The input and scheduling parameter signals are plotted in the top. In the middle, the outputs of all five systems are shown, where y_{IO} denotes the output of the LPV IO system G_{IO} in (2.34). The outputs $y_{\mathrm{SS}}^{\mathrm{o}}$, $y_{\mathrm{SS}}^{\mathrm{r}}$, $y_{\mathrm{SS}}^{\mathrm{o,LTI}}$, and $y_{\mathrm{SS}}^{\mathrm{r,LTI}}$ denote the outputs of $G_{\mathrm{SS}}^{\mathrm{o}}$, $G_{\mathrm{SS}}^{\mathrm{r}}$, $G_{\mathrm{SS}}^{\mathrm{o,LTI}}$, and $G_{\mathrm{SS}}^{\mathrm{r,LTI}}$, respectively, see (2.35)–(2.37). The output errors (differences) w. r. t. the original output of the LPV IO system are shown at the bottom. It can be seen, that the SS models $G_{\mathrm{SS}}^{\mathrm{o}}$ and $G_{\mathrm{SS}}^{\mathrm{r}}$ do not have an output error w. r. t. G_{IO}, which confirms the identical IO behavior. In contrast, $G_{\mathrm{SS}}^{\mathrm{o,LTI}}$ and $G_{\mathrm{SS}}^{\mathrm{r,LTI}}$ have an output error that results from neglecting time dependence in p. Note that all the systems approach each other for time intervals with constant scheduling parameters, because then the effect of dynamic dependency on p vanishes.

For the design of an LPV controller based on an LPV SS model, static scheduling parameter dependency is assumed for the LPV SS model. Consequently, in order to avoid a dynamic

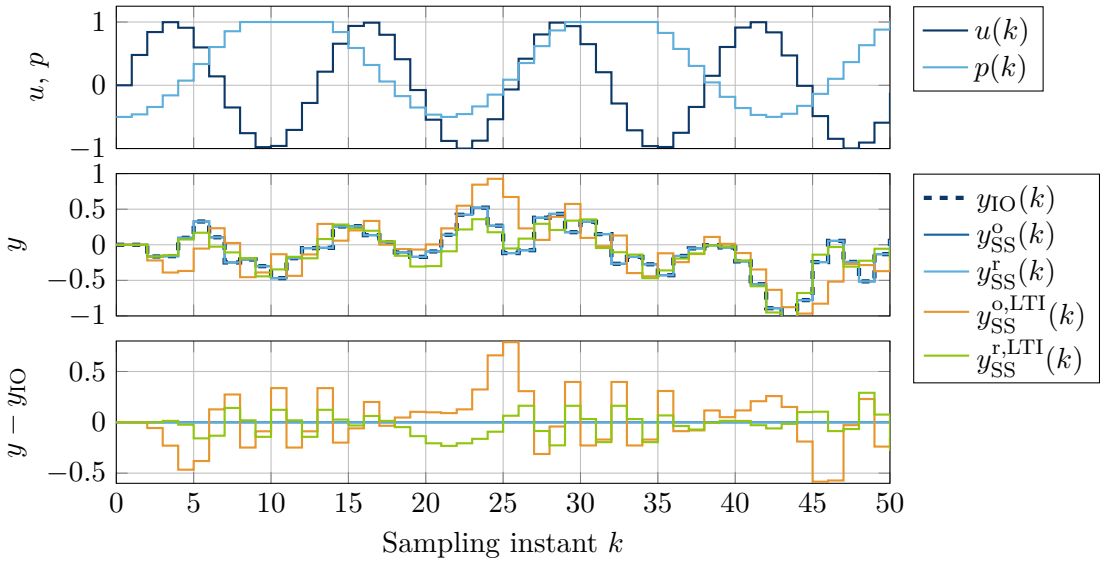


Figure 2.3: Comparison of an LPV IO system and various derived canonical LPV SS models, where the time dependence of the scheduling parameter is considered or neglected

dependency in the state-space representation, dedicated dynamic scheduling parameter dependencies need to be assumed for the LPV IO representation. In the following four different transformations are presented. Note that transformation rules for the continuous-time domain are also available, see [Tóth 2010, Section 4.3], however they are more complicated due to the product rule.

Three forms of dynamically parameter dependent DT LPV IO representations are well known, which are introduced in [Abbas et al. 2010] and allow a direct transformation into LPV SS representations with static scheduling parameter dependence.

1. The IO representation in shifted form (SF)

$$\mathcal{R}_{\mathrm{IO}}^{\mathrm{DT,SF}}(G \diamond \rho) \;:\; y_k = -\sum_{i=1}^{n_a} a_i\left(\rho_{k-i}\right) y_{k-i} + \sum_{j=0}^{n_b} b_j\left(\rho_{k-j}\right) u_{k-j} \;, \tag{2.39}$$

whose state-space representation $\mathcal{R}_{\mathrm{SS}}^{\mathrm{DT,SF}}(G \diamond \rho)$ is

$$\left[\begin{array}{c|c} A(\rho) & B(\rho) \\ \hline C(\rho) & D(\rho) \end{array}\right] = \left[\begin{array}{ccccc|c} -a_1(\rho) & I & 0 & \dots & 0 & b_1(\rho) - a_1(\rho)b_0(\rho) \\ \vdots & 0 & \ddots & \ddots & \vdots & \vdots \\ \vdots & \vdots & \ddots & \ddots & \vdots & \vdots \\ -a_{n_a-1}(\rho) & 0 & \dots & 0 & I & b_{n_a-1}(\rho) - a_{n_a-1}(\rho)b_0(\rho) \\ -a_{n_a}(\rho) & 0 & \dots & \dots & 0 & b_{n_a}(\rho) - a_{n_a}(\rho)b_0(\rho) \\ \hline I & 0 & \dots & \dots & 0 & b_0(\rho) \end{array}\right] \tag{2.40}$$

and is denoted by $\mathcal{R}_{\mathrm{SS}}^{\mathrm{DT,SF}}(G \diamond \rho)$. It is clear from the input matrix function that the state-space representation has the same scheduling parameter dependence as the input-output representation, only if b_0 is parameter independent.

2. The IO representation in augmented form (AF)

$$\mathcal{R}_{\mathrm{IO}}^{\mathrm{DT,AF}}(G \diamond \rho) \ : \ y_k = -\sum_{i=1}^{n_a} a_i\left(\rho_{k-1}\right) y_{k-i} + b_0(\rho_k) u_k + \sum_{j=1}^{n_b} b_j\left(\rho_{k-1}\right) u_{k-j} \ , \tag{2.41}$$

whose state-space representation $\mathcal{R}_{\mathrm{SS}}^{\mathrm{DT,AF}}(G \diamond \rho)$ is

$$\left[\begin{array}{c|c} A(\rho) & B(\rho) \\ \hline C(\rho) & D(\rho) \end{array}\right] = \left[\begin{array}{cccccccc|c} -a_1(\rho) & \dots & \dots & -a_{n_a}(\rho) & b_2(\rho) & \dots & \dots & b_{n_b}(\rho) & b_1(\rho) - a_1(\rho)b_0(\rho) \\ I & 0 & \dots & \dots & \dots & \dots & \dots & 0 & b_0(\rho) \\ \vdots & \ddots & \ddots & \ddots & \ddots & \ddots & \ddots & \vdots & \vdots \\ 0 & \dots & I & 0 & 0 & \dots & \dots & 0 & 0 \\ 0 & \dots & 0 & 0 & 0 & \dots & \dots & 0 & I \\ 0 & \dots & 0 & 0 & I & 0 & \dots & 0 & 0 \\ \vdots & \ddots & \ddots & \ddots & \ddots & \ddots & \ddots & \vdots & \vdots \\ 0 & \dots & \dots & \dots & \dots & \dots & I & 0 & 0 \\ \hline I & 0 & \dots & \dots & \dots & \dots & \dots & 0 & b_0(\rho) \end{array}\right] . \tag{2.42}$$

Note that the AF as presented in [Tóth, Abbas, et al. 2012] is regained, if $b_0(\rho_k)$ is zero. Similar to the SF, the state-space representation has the same scheduling parameter dependence as the input-output representation, only if b_0 is parameter independent.

3. The IO representation in observability form (OF)

$$\mathcal{R}_{\mathrm{IO}}^{\mathrm{DT,OF}}(G \diamond \rho) \ : \ y_k = -\sum_{i=1}^{n_a} a_i\left(\rho_{k-n_a}\right) y_{k-i} + b_{n_a}\left(\rho_{k-n_a}\right) u_{k-n_a} \ , \tag{2.43}$$

whose state-space representation $\mathcal{R}_{\mathrm{SS}}^{\mathrm{DT,OF}}(G \diamond \rho)$ is

$$\left[\begin{array}{c|c} A(\rho) & B(\rho) \\ \hline C(\rho) & D(\rho) \end{array}\right] = \left[\begin{array}{ccccc|c} 0 & I & 0 & \dots & 0 & 0 \\ \vdots & 0 & \ddots & \ddots & \vdots & \vdots \\ \vdots & \vdots & \ddots & \ddots & \vdots & \vdots \\ 0 & 0 & \dots & 0 & I & 0 \\ -a_{n_a}(\rho) & -a_{n_a-1}(\rho) & \dots & \dots & -a_1(\rho) & b_{n_a}(\rho) \\ \hline I & 0 & \dots & \dots & 0 & 0 \end{array}\right] . \tag{2.44}$$

In addition to these three forms, the augmented form can be extended, resulting in the fourth form.

4. The IO representation in extended AF

$$\mathcal{R}_{\mathrm{IO}}^{\mathrm{DT,eAF}}(G(\rho)) \ : \ y_k = -\sum_{i=1}^{n_a} a_i\left(\rho_k\right) y_{k-i} + \sum_{j=0}^{n_b} b_j\left(\rho_k\right) u_{k-j} \ , \tag{2.45}$$

whose state-space representation $\mathcal{R}_{\mathrm{SS}}^{\mathrm{DT,eAF}}(G(\rho))$ is

$$\left[\begin{array}{c|c} A(\rho) & B(\rho) \\ \hline C(\rho) & D(\rho) \end{array}\right] = \left[\begin{array}{cccccccc|c} -a_1(\rho) & \dots & \dots & -a_{n_a}(\rho) & b_1(\rho) & \dots & \dots & b_{n_b}(\rho) & b_0(\rho) \\ I & 0 & \dots & \dots & \dots & \dots & \dots & 0 & 0 \\ \vdots & \ddots & \ddots & \ddots & \ddots & \ddots & \ddots & \vdots & \vdots \\ 0 & \dots & I & 0 & 0 & \dots & \dots & 0 & 0 \\ 0 & \dots & 0 & 0 & 0 & \dots & \dots & 0 & I \\ 0 & \dots & 0 & 0 & I & 0 & \dots & 0 & 0 \\ \vdots & \ddots & \ddots & \ddots & \ddots & \ddots & \ddots & \vdots & \vdots \\ 0 & \dots & \dots & \dots & \dots & \dots & I & 0 & 0 \\ \hline -a_1(\rho) & \dots & \dots & -a_{n_a}(\rho) & b_1(\rho) & \dots & \dots & b_{n_b}(\rho) & b_0(\rho) \end{array}\right]. \tag{2.46}$$

The extended augmented form has n_u more states compared to the AF, however the extended AF has the special property that its transformation between the IO and the SS domain allows static scheduling parameter dependence in both domains.

Compared to the LPV IO system given in Definition 2.2, for all four IO systems the coefficient function a_0 is identity, i. e. $a_0 \diamond \rho = I$. The transformation of all four IO system representations into the SS domain is given in detail in Appendix A.2.

Some of the obvious differences of the different forms are as follows. The four input-output forms differ in the dynamic dependency of the coefficient functions a_i, b_j on the scheduling parameter, where the SF has the property that the scheduling parameter on which the system coefficient functions depend is of the same time instant as the in- and outputs. This is in contrast to the other forms, where it is assumed that the whole IO system behavior varies according to the (delayed) scheduling parameter from a single time step. The observability form is restricted to just one coefficient for the input signal, since $b_j \diamond \rho = 0$ for $j \in \{0, 1, \dots, n_{a-1}\}$, s. t. only the input $u(k - n_a)$ is directly influencing the output. Thus, the OF only represents systems with an input time delay of the same size as the order of polynomial $A(\mathrm{q}, \rho)$. Another difference is that the order in the state-space domain, i. e. the number of states, of the augmented form (AF) and its extended version is inherently higher than those of the SF and OF, if the order in the input-output domain, i. e. the highest exponent of the time shift operator, is for all four systems the same. In the SISO case, the state-space representations of the SF and OF are in a canonical form[2)], thus they are structurally state-observable [Tóth 2010, Section 4.1].

The first three IO representations, i. e. the shifted, the augmented, and the observability form, depend in the case of no feedthrough, i. e. $b_0(\rho_k) = 0$, on a delayed scheduling parameter. In contrast, the SS representations depend on the current scheduling parameter. This property is adverse for quasi LPV systems, where the scheduling parameter depends on the output of the LPV system. Then, there exists an algebraic loop in the SS representation while this is not the case for the IO representation, which makes simulation and real-time control more demanding. Thus, it is favorable to find an IO equivalent SS representation with a delayed scheduling parameter for quasi LPV systems. Thus, starting with a DT LPV SS representation with static parameter dependence

$$\mathrm{q}x_k = A(\rho_k)x_k + B(\rho_k)u_k \tag{2.47a}$$

$$y_k = C(\rho_k)x_k + D(\rho_k)u_k \tag{2.47b}$$

[2)] A canonical state-space representation possesses the least possible number of meaningful parameters, see [Tóth et al. 2007, Section 2] and references therein.

and shifting the state equation by one time step backwards, i. e. multiplying with q^{-1}, yields

$$x_k = A(\rho_{k-1})x_{k-1} + B(\rho_{k-1})u_{k-1} \tag{2.48}$$

with the new state vector $[x_{k-1}^{\mathrm{T}} \quad u_{k-1}^{\mathrm{T}}]^{\mathrm{T}}$. The state x_k in the output equation can now be substituted so that

$$\begin{aligned} y_k &= C(\rho_k)\mathrm{q}x_{k-1} + D(\rho_k)u_k \\ &= C(\rho_k)(A(\rho_{k-1})x_{k-1} + B(\rho_{k-1})u_{k-1}) + D(\rho_k)u_k \ , \end{aligned} \tag{2.49}$$

which has in general a dynamic scheduling parameter dependency due to ρ_k and ρ_{k-1}. Hence, the state-space representation with additionally the one-step delayed scheduling parameter becomes

$$\left[\begin{array}{c} \mathrm{q}x_{k-1} \\ \mathrm{q}u_{k-1} \\ \hline y_k \end{array}\right] = \left[\begin{array}{cc|c} A(\rho_{k-1}) & B(\rho_{k-1}) & 0 \\ 0 & 0 & I \\ \hline C(\rho_k)A(\rho_{k-1}) & C(\rho_k)B(\rho_{k-1}) & D(\rho_k) \end{array}\right] \left[\begin{array}{c} x_{k-1} \\ u_{k-1} \\ \hline u_k \end{array}\right] . \tag{2.50}$$

This transformation is achieved at the expense of n_u additional state(s). Assuming that the system has no feedthrough, i. e. $b_0(\rho) = 0$, the output equations of the state-space representations of the SF, AF, and OF are parameter independent. Then, the LPV SS model (2.50) depends only on the one-step delayed scheduling parameter, thus a quasi-static, delayed dependence is obtained. For example, the state-space representation $\mathcal{R}_{\mathrm{SS}}^{\mathrm{DT,SF}}(G \diamond \rho)$ of an LPV system without feedthrough using the one-step delayed scheduling parameter $\mathrm{q}^{-1}\rho$ is

$$\left[\begin{array}{c|c} A(\mathrm{q}^{-1}\rho) & B(\mathrm{q}^{-1}\rho) \\ \hline C(\mathrm{q}^{-1}\rho) & D(\mathrm{q}^{-1}\rho) \end{array}\right] = \left[\begin{array}{ccccc|c} -a_1(\mathrm{q}^{-1}\rho) & I & 0 & \dots & 0 & b_1(\mathrm{q}^{-1}\rho) & 0 \\ \vdots & 0 & \ddots & \ddots & \vdots & \vdots & \vdots \\ \vdots & \vdots & \ddots & \ddots & \vdots & \vdots & \vdots \\ -a_{n_a-1}(\mathrm{q}^{-1}\rho) & 0 & \dots & 0 & I & b_{n_a-1}(\mathrm{q}^{-1}\rho) & 0 \\ -a_{n_a}(\mathrm{q}^{-1}\rho) & 0 & \dots & \dots & 0 & b_{n_a}(\mathrm{q}^{-1}\rho) & 0 \\ 0 & 0 & \dots & \dots & 0 & 0 & I \\ \hline -a_1(\mathrm{q}^{-1}\rho) & I & 0 & \dots & 0 & b_1(\mathrm{q}^{-1}\rho) & 0 \end{array}\right] . \tag{2.51}$$

Another approach to avoid the algebraic loop in the SS domain is to use a delayed output in the scheduling parameter vector already in the IO domain. Note that the controller output is the system input, hence the LPV SS model used for control design should have a dependency, if any, on the delayed system input to avoid the algebraic loop in the LPV SS controller. Further, note that this ad hoc approach induces a modeling error, if the true system has a dependency on the current output (and/or input).

2.4.2 Transformation of State-Space Representations

Different IO equivalent LPV SS representations can differ in the static/dynamic scheduling parameter dependence, among others. This is already demonstrated in Example 2.4. Hence, state transformations that maintain IO equivalence can influence the time dependence on the scheduling parameter. This is discussed in more detail in this section.

Similar to the LTI case, state transformations can be performed for LPV SS models, cf. [Wood 1995, Section 2.4]. However, in the LPV case the state transformation matrix T can be

scheduling parameter dependent, thus the similarity transformation from the original states x to the new, transformed states z is

$$z = (T \diamond \rho)x \ . \tag{2.52}$$

Since the scheduling parameter is time-dependent, the transformation matrix function T is also time-dependent. This must be considered if it is applied to an LPV system in SS representation. For a DT LPV SS model

$$G_x \diamond \rho \ : \ \left[\begin{array}{c|c} A \diamond \rho & B \diamond \rho \\ \hline C \diamond \rho & D \diamond \rho \end{array}\right] \tag{2.53}$$

the state equations with state x and the transformed z are

$$x_{k+1} = (A \diamond \rho_k)x_k + (B \diamond \rho_k)u_k \tag{2.54}$$

$$(T \diamond \rho_{k+1})^{-1} z_{k+1} = (A \diamond \rho_k)(T \diamond \rho_k)^{-1} z_k + (B \diamond \rho_k)u_k \tag{2.55}$$

$$z_{k+1} = (T \diamond \rho_{k+1})(A \diamond \rho_k)(T \diamond \rho_k)^{-1} z_k + (T \diamond \rho_{k+1})(B \diamond \rho_k)u_k \tag{2.56}$$

using (2.52) for a state transformation. Note that on the left hand side in (2.55) the forward shift operator must be applied to the transformed state and the scheduling parameter in the state transformation matrix. Thus, the transformed DT LPV SS model becomes

$$G_z \diamond \rho \ : \ \left[\begin{array}{c|c} (T \diamond \mathrm{q}\rho)(A \diamond \rho)(T \diamond \rho)^{-1} & (T \diamond \mathrm{q}\rho)(B \diamond \rho) \\ \hline (C \diamond \rho)(T \diamond \rho)^{-1} & (D \diamond \rho) \end{array}\right] \ . \tag{2.57}$$

The transformed LPV SS model $G_z \diamond \rho$ is non-causal w. r. t. the scheduling parameter. Moreover, applying parameter dependent state transformations can increase considerably the functional complexity on the scheduling parameter.

Now it also becomes clear why in Section 2.3.3 the Theorems 2.2 and 2.3 are restricted to constant generalized Gramians. From the generalized Gramians, state transformation matrices can be calculated for a balanced realization. Thus, LPV SS models for a control design or even the LPV SS controller itself can be reduced in its state order. In both cases a non-causal, dynamic scheduling parameter dependency leads to problems and as demonstrated in Example 2.4, the dynamic dependency should not be neglected.

Next, based on the gained insights on state transformations, a minimal state representation is defined as follows.

Definition 2.10 (Minimal LPV SS Representation, [Tóth 2010]). A DT LPV SS model $\bar{x}_{k+1} = (\bar{A} \diamond \rho_k)\bar{x}_k + (\bar{B} \diamond \rho_k)u_k$, $y_k = (\bar{C} \diamond \rho_k)\bar{x}_k + (\bar{D} \diamond \rho_k)u_k$, and state dimension $n_{\bar{x}}$ is called minimal, if there exist no equivalent SS representation $x_{k+1} = (A \diamond \rho_k)x_k + (B \diamond \rho_k)u_k$, $y_k = (C \diamond \rho_k)x_k + (D \diamond \rho_k)u_k$, and state dimension n_x of lower order, i. e. with $n_x < n_{\bar{x}}$. This means that there exists no invertible square state transformation matrix function $T \diamond \rho$ of dimension $n_{\bar{x}}$ such that

$$(T \diamond \mathrm{q}\rho)(\bar{A} \diamond \rho)(T \diamond \rho)^{-1} = \begin{bmatrix} A \diamond \rho & 0 \\ \star & \star \end{bmatrix} \ , \quad (T \diamond \mathrm{q}\rho)(\bar{B} \diamond \rho) = \begin{bmatrix} B \diamond \rho \\ \star \end{bmatrix} \tag{2.58a}$$

$$(\bar{C} \diamond \rho)(T \diamond \rho)^{-1} = \begin{bmatrix} C \diamond \rho & 0 \end{bmatrix} \ , \qquad \bar{D} \diamond \rho = D \diamond \rho \tag{2.58b}$$

holds for all $\rho \in \mathcal{F}_\rho$, where terms of no interest are denoted by $\star$.

Remark 2.5. *A definition for equivalent LPV SS representations is given in [Tóth 2010, definition 3.29] and determines two LPV SS models as equivalent, if they possess the same IO behavior.*

A minimal LPV SS representation is completely state-observable and state-reachable since only the observable and reachable states contribute to the IO behavior. More details on minimal state representations of static, affine scheduling parameter dependent LPV SS models are given in [Petreczky et al. 2016].

Note that in the SISO case the LPV SS realizations of the shifted form and the observability form are minimal, while the realizations of the AF and its extended version are non-minimal [Tóth, Abbas, et al. 2012, Section 4]. For LPV IO systems, analog definitions can be given for equivalent and minimal representations. However, they are not required in this thesis and can be found in [Tóth 2010, Section 3.2].

For illustration, the state transformation matrices for the state-space representations $G^{\mathrm{r}}_{\mathrm{SS}} \diamond p$ and $G^{\mathrm{o}}_{\mathrm{SS}} \diamond p$ in Example 2.4 are given in the following.

Example 2.4 (Continued). The scheduling parameter dependent state transformation matrix $T_{\mathrm{r,o}} \diamond p$ that transfers the canonical observability form $G^{\mathrm{o}}_{\mathrm{SS}} \diamond p$ in (2.35) into the canonical reachability form $G^{\mathrm{r}}_{\mathrm{SS}} \diamond p$ in (2.36) is given by

$$T_{\mathrm{r,o}} \diamond p(k) = \begin{bmatrix} \frac{1}{p(k)} - 1 & -\frac{1}{p(k+1)} \\ -\frac{1}{p(k-1)} & -\frac{1}{p(k-1)p(k+1)} \end{bmatrix} \tag{2.59}$$

and, accordingly, the shifted as well as the inverse matrices are

$$T_{\mathrm{r,o}} \diamond p(k+1) = \begin{bmatrix} \frac{1}{p(k+1)} - 1 & -\frac{1}{p(k+2)} \\ -\frac{1}{p(k)} & -\frac{1}{p(k)p(k+2)} \end{bmatrix} \tag{2.60}$$

$$(T_{\mathrm{r,o}} \diamond p(k))^{-1} = \begin{bmatrix} p(k) & -p(k-1)p(k) \\ -p(k)p(k+1) & p(k-1)(p(k)-1)p(k+1) \end{bmatrix} . \tag{2.61}$$

3 Identification of LPV Models

In the preceding chapter, representations and characteristics of LPV systems have been presented. In this chapter, the problem is addressed how to obtain such an LPV system. More precisely, the chapter is about obtaining an LPV model that (approximately) represents an unknown LPV system. In the following, as a review, different approaches for modeling a system are discussed. They can be divided into theoretical and experimental modeling as well as intermediate stages, which are also known as white, gray, and black box approaches. Irrespective of the theoretical or experimental modeling approach, measurements of the system are always required if model coefficients are unknown and therefore need to be estimated from data.

The theoretical modeling (white box) has the advantage that through first principle modeling many insights into and thus knowledge about the system can be gained, for example (influences on) the stationary and dynamic behavior, properties, and limits. This knowledge can be exploited when designing the controller or determining unknown system coefficients. The disadvantage of the theoretical approach is that this knowledge about the system is only as trustworthy as the modeling so that errors or simplifications in the modeling can have a significant impact [Leith & Leithead 2000a; Plastino & Muzzio 1992]. In addition, at the end of the modeling process, commonly a nonlinear model is obtained, which usually cannot be used directly for model-based LPV control design. Thus, the nonlinear model must first be (approximately) represented by an LPV model, for which there are various techniques, see [Packard & Kantner 1996; Rugh & Shamma 2000; Shamma & Cloutier 1993], [Kwiatkowski 2007, Section 1.2], [Tóth 2010, Section 7.3], [Theis 2018, Section 2.1] and references therein. However, several methods for converting a nonlinear model into an LPV model either are valid only in the neighborhood of linearization points or require a specific class of nonlinear systems [Leith & Leithead 1999]. In addition, interpolation of linearized models at several linearization points can lead to a complete misfit, e. g., due to the interpolation of local state-spaces.

In contrast to theoretical modeling, in the black box approach or experimental modeling, also known as system identification, a model is estimated based on a recorded data set. In this approach, the identified model depends on the structural decisions made in advance, such as the choice of system class and model structure. Then, for an appropriate choice, the obtained model is directly suitable for control design. However, for a successful control design it is important to keep the modeling error small, i. e. the model structure must be suitable for the system and not for the control design method, respectively it must be suitable for the problem and not the method. Further, the accuracy of the identified model depends on whether the relevant information about the system is contained in the recorded data set and if the information can be extracted. This usually requires on one hand a good understanding of the system and on the other hand knowledge of the purpose or requirements of the model in order to generate an appropriate data set.

In this chapter, the identification of discrete-time LPV models is presented, where the models have a state-space representation with static and affine dependence on the scheduling parameter. The motivation for this is that model-based LPV control design methods exist for such models. In Fig. 3.1, a schematic workflow is depicted for model-based LPV control design. On the right

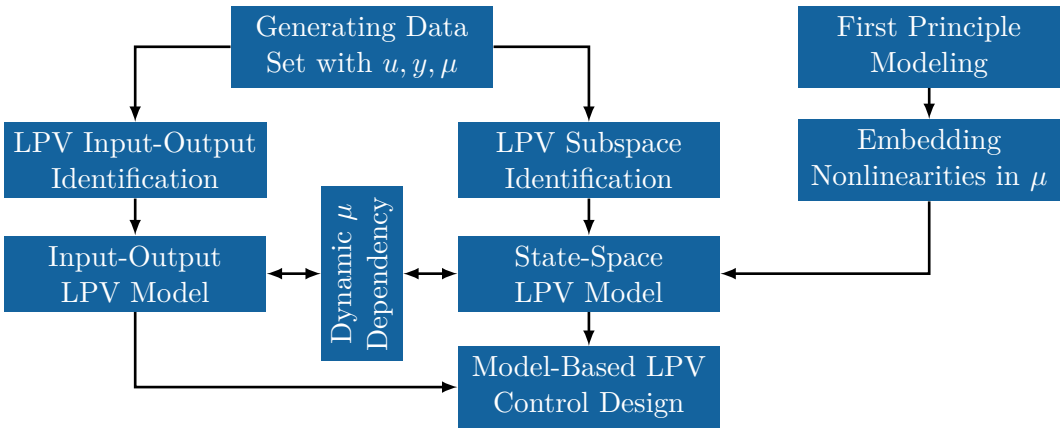


Figure 3.1: Schematic workflow to obtain an LPV model for model-based LPV control design with input u, output y, and linear scheduling parameter μ

side the theoretical modeling is depicted and on the left side and in the middle two identification methods are shown, which belong to the experimental approach: input-output and subspace LPV system identification. The advantage of subspace identification is that an LPV SS model with static scheduling parameter dependence is obtained directly from the identification method. As opposed to this, the identified IO models need to be converted into SS models, if well-known LPV SS model-based control design methods are to be used, cf. Fig. 3.1. In order to ensure the requirement of a static dependence on the scheduling parameter in the LPV SS model, only the forms presented in Section 2.4.1 with specific (dynamic) dependencies of the model coefficients on the scheduling parameter are considered in the IO identification. Note that LPV IO models with static scheduling parameter dependence can be used directly for LPV IO control design, see [Wollnack & Werner 2016] and Fig. 3.1. Overall, the identification of an LPV model is more demanding than that of an LTI model, because on the one hand the number of unknown model coefficients increases linearly[1)] with the number of linear scheduling parameters μ and on the other hand the data set must contain sufficient information not only regarding time (dynamic behavior) but also regarding the scheduling parameter set, cf. [Schoukens & Ljung 2019].

In Section 3.1, the four key steps of system identification are recapitulated: excite the system, choose a model structure, find (optimal) model coefficients, and validate the model. An overview of identification methods for LPV models is given in Section 3.2. The methods are classified into the identification of LPV models in IO and SS representation. In Section 3.3, the identification of LPV IO models is discussed. Model structures and a collection of existing and new algorithms are presented. The concept of the subspace identification method of LPV SS models introduced in [Verdult 2002; Wingerden & Verhaegen 2009] is briefly reviewed in Section 3.4. Then, the subspace identification is compared with IO methods in Section 3.5 and the remainder of this thesis. In Section 3.6, a reference to existing nonlinear optimization methods of LPV SS models is provided, since some of them are applied in Chapter 5.

[1)] Here the number of unknown coefficients in the LPV model structure is meant. In subspace identification, however, an intermediate problem must be solved that grows exponentially with the number of scheduling parameters.

3.1 Preliminaries for LPV System Identification

In the following, four key ingredients are presented that are typically required for system identification, cf. [Schoukens & Ljung 2019].

System Excitation One of the key ingredients in system identification is the data set obtained from measurements or simulations of high-fidelity models using an excitation signal as input. Central issues are the sampling rate, signal processing, information content in the data by excitation and coverage of the operating range, etc.

Model Structure It is assumed that the system can be (approximately) represented by a model of a certain class (e. g., LTI or LPV) and structure with adjustable coefficients. Since a true system cannot usually be represented exactly by the (mathematically expressed) model, a modeling error occurs here. The choice of the model structure depends on the requirements (accuracy, intended use) and the system behavior.

Model Coefficients Once the model structure is given and a data set is available, suitable model coefficients have to be found. For simplicity's sake, the search for the coefficients is formulated as an optimization problem. There are many different approaches and a focus of system identification is on the formulation and solution of the optimization problem.

Validation It is useful to validate the estimated model via accuracy criteria on a fresh data set, which is not used for estimation and should be representative for the intended use of the model.

Remark 3.1. *Typically, the term model parameters is used in the system identification community, while in this thesis they are referred to as model coefficients θ. In order to distinguish them from the affine scheduling parameters, which are also denoted by θ, the scheduling parameter μ is preferably used, which denotes a linear dependency. Thus, the linear and affine scheduling parameter are related by $\mu^{\mathrm{T}} = [1 \quad \theta^{\mathrm{T}}]$.*

For identifying, respectively optimizing, the model coefficients θ of an LPV model structure, a data set

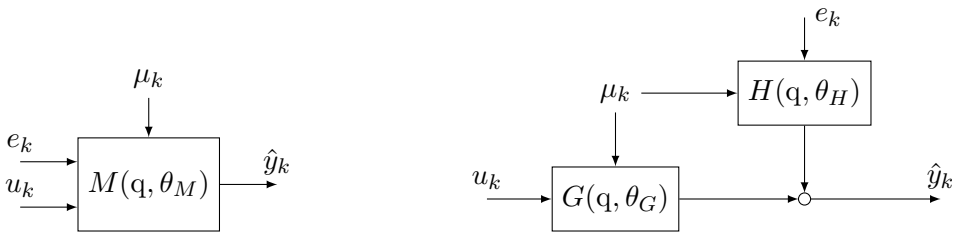
$$\mathbb{D} = \{u(k), y(k), p(k) \ , \ k \in \{1, 2, \ldots, N\}\} \tag{3.1}$$

is required, which contains N[2)] consecutive samples of the inputs u, outputs y, and external scheduling parameters p. It is assumed that the signals are sampled synchronously[3)] and equidistantly, i. e. with a constant sampling time T_{s}. Moreover, it is assumed that proper signal processing has been performed such that e. g., aliasing is avoided. Finally, the estimation of the model coefficients and validation of the estimated model should be performed on different data sets to achieve a higher reliability for the results.

The choice of input signal is of large importance for the identification result. For the identification of LTI models, an independent and identically or normally distributed random signal (over a sufficiently long time interval) is commonly used in the literature. The assumption

[2)] Note that DT models depend on past values and therefore only a reduced number of samples N^{red} of the model output can be reconstructed from the data set. Despite this difference, the number of samples N and the reduced number N^{red} are used as synonyms herein.

[3)] The assumption of sampling in a perfectly synchronized way is common, but it cannot always be satisfied in practical applications, cf. [Sehnke et al. 2018] where different temporal properties and their effects are discussed.



(a) Combined model M **(b)** Separation of system model G and noise model H

Figure 3.2: Block diagram of general LPV model structures

is that such a signal is persistently exciting the system, see [Isermann & Münchhof 2011, Theorem 9.3] for the condition on the input signal satisfying persistence of excitation. This type of excitation is also frequently used for the identification of LPV systems in academic examples, cf. [Verdult 2002, Chapter 6], [Laurain et al. 2010; Wingerden & Verhaegen 2009]. However, an independent and identically or normally distributed random signal may be insufficient, especially in the quasi LPV case, because homogeneity[4)] of the system can be lost and the randomly excited system may remain close to its initial- or steady-state. Consequently, in case of identifying LPV models, the system should be (persistently) excited in the time domain and the scheduling parameter set, respectively in its dynamics and operating range. Therefore, the excitation signals must be carefully designed such that the operating range of inputs, outputs, and scheduling parameters is well covered. This is especially true, if the scheduling variables are functions of states, inputs, or outputs of the system. Only few literature about persistent excitation for nonlinear systems is available. In [Gorinevsky 1995], persistence of excitation is investigated for a radial basis function network, i. e. a linear interpolation of radial basis functions. In [Nelles & Isermann 1995], a pseudo-random binary signal (PRBS) is modified by assigning different amplitudes to the signal, hence an amplitude modulated pseudo-random binary signal is derived, cf. [Nelles 2001], which is claimed by the authors to perform best in comparison with other excitation signals in the identification of radial basis function networks for different nonlinear systems. However, the authors suggest to choose the minimum hold time of the amplitude modulated pseudo-random binary signal long enough such that the system can (nearly) settle, which results in a quasi-static step signal with poor dynamic excitation. Thus, here a different approach is followed for the excitation signal, using a combination of at least two signals for identification: a low-frequency, high-amplitude signal superimposed by a high-frequency, low-amplitude signal to cover both, operating range and dynamics. Note that some systems can be excited in open-loop, whereas, e. g., unstable systems should or need to be operated in closed-loop. However, in this thesis it is assumed that the system is excited in open-loop.

The second key ingredient of system identification is the model structure, where in the following the focus is on IO model structures. A general LPV model structure is shown in Fig. 3.2a with input u_k and noise e_k as inputs, the modeled output $\hat{y}_k$, and the dynamic DT LPV model structure M parametrized by the model coefficients θ_M and scheduled by the linear scheduling parameter μ_k. For the identification of IO models, it is generally assumed that the

[4)] Homogeneity is one of the two principles that linear systems obey and states that the output scales with the same factor as the input scales [Ogunfunmi 2007, Section 1.1].

model structure can be divided into an LPV system model G and an LPV noise model H, i. e.

$$\hat{y}(k) = G(\mathrm{q}, \mu(k), \theta_G)u(k) + H(\mathrm{q}, \mu(k), \theta_H)e(k) \tag{3.2}$$

with model coefficients θ_G and θ_H for G and H, respectively. The model structure is shown in Fig. 3.2b. The noise $e(k)$ is assumed to be white, stationary, and normally distributed sequence with zero mean and finite variance, thus

$$\mathrm{E}\left\{e_k\right\} = 0 \ , \qquad \mathrm{E}\left\{e_k e_k^{\mathrm{T}}\right\} = \Sigma_e \succ 0 \ , \qquad \mathrm{E}\left\{e_k e_{k-l}^{\mathrm{T}}\right\} = 0 \ , \ l \neq 0 \ . \tag{3.3}$$

For simplicity, a noise signal that satisfies all these properties will be referred to as white in the following. The output of the system model G is supposed to represent the deterministic, noise free part of the output y. The output of the noise model H, which is colored noise and sometimes denoted as disturbance, is intended to represent the stochastic part of y. Note that there are several structural error sources in the approximation, of the true system by a model. Examples of these error sources are the model class (e. g., LTI or LPV), the structure of the models, or the scheduling parameters and their dependencies.

The objective in this thesis is to identify LPV models that are suitable for LPV SS control design. Thus, the model structure w. r. t. the scheduling parameters is chosen such that the identified DT LPV models have a SS representation with static and affine scheduling parameter dependency. Although there are also approaches available for the design of LPV IO controllers based on IO models or even on a data set, e. g., [Ali et al. 2010; Formentin et al. 2016; Wollnack & Werner 2016], the majority of the design methods are based on LPV SS models with a static parameter dependency, e. g., [Apkarian & Adams 1998; De Caigny et al. 2012; Gahinet & Apkarian 1994; Hoffmann & Werner 2015; Scherer 2001; Wu & Dong 2006]. Moreover, it is assumed that the affine scheduling parameters are known a priori, i. e. the general scheduling parameters ρ and their functional dependence are assumed as known. Note that a general functional scheduling parameter dependency can be resolved with an affine one, where additional affine parameters commonly need to be introduced. However, assuming the affine scheduling parameter as independent of each other causes conservatism in polytopic LPV control design, cf. [Kwiatkowski 2007, Chapter 2]. In LPV identification, more scheduling parameters due to an affine dependency only means an increase of unknown model coefficients.

Once the model structure is specified, an error between the measured and modeled output can be determined. In system identification, two different types of error are commonly used: the output error and the prediction error. The output error is obtained by subtracting from the measured output y the noise free simulated system model output $\hat{y} = G(\mathrm{q}, \mu, \theta_G)u$. This error is commonly used for model validation and also denoted as simulation error. The prediction error ε, on the other hand, is

$$\varepsilon_k = y_k - \hat{y}_{k|k-1} \tag{3.4}$$

$$\hat{y}_{k|k-1} = H^{\dagger}(\mathrm{q}, \mu_k, \theta_H)G(\mathrm{q}, \mu_k, \theta_G)u_k - (I - H^{\dagger}(\mathrm{q}, \mu_k, \theta_H))y_k \ , \tag{3.5}$$

where $\hat{y}_{k|k-1}$ is the predicted output for the time instant k assuming the model structure (3.2) and using measured input and output samples from the past up to time instant $k-1$. The derivation of (3.5) is, e. g., given in [Tóth, Heuberger, et al. 2012], [Ljung 1999, Section 3.2]. In order to derive the equation for the predicted output $\hat{y}_{k|k-1}$, it is assumed that the inverse[5)] of

[5)] The inverse is given as $H^{\dagger}(\mathrm{q}, \mu_k) = \sum_{i=0}^{\infty}(I - H(\mathrm{q}, \mu_k, \theta_H))^i$, which can be shown based on telescopic sums, see [Cox 2018, Lemma 4.3].

H exists, i. e. $H^\dagger(\mathrm{q}, \mu_k, \theta_H)H(\mathrm{q}, \mu_k, \theta_H) = I$ in the functional sense. Further, it is assumed that $H^\dagger G$ and $H^\dagger$ are asymptotically stable and that H is bi-proper and normalized (monic), i. e. $H^\dagger(0, \mu_k, \theta_H) = I$. Note that the normalization of H is not a restriction, since the normalization can be compensated by scaling the variance of e. The prediction error ε is commonly used in IO system identification methods, which are thus denoted prediction error methods.

For the determination of the model coefficients, an informative data set and identifiable model structure is desired. A data set is informative, if two models with different coefficients do not produce the same modeled output for this data set, see [Gevers et al. 2009] and [Tóth, Heuberger, et al. 2012, Definition 2.2], and a model structure is identifiable, if two models with non-identical model coefficients differ, see [Boonto 2011, Definition 5.10]. Note that a data set can only be informative, if the model structure is identifiable (necessary condition). Consequently, if both properties are met, the function from model coefficients to modeled output is bijective, which simplifies the determination of the model coefficients. For example, if the identification problem is a linear regression, the regression matrix has full rank, if the data set is informative and the model structure is identifiable.

The key idea in system identification is to map a data set $\mathbb{D}$ to the model coefficients θ or in other words, to shrink the information of a large number of samples to a small number of coefficients. There are different methods for estimating the model coefficients, see [Isermann & Münchhof 2011, Chapter 11] and [Nelles 2001, Chapter 2]. In this thesis, the method of least squares (LS) is used, which is compared to the other methods the computationally most efficient one. Using the LS method, the cost function is

$$\sum_{k=1}^{N} \varepsilon^{\mathrm{T}}(k, \theta)\varepsilon(k, \theta) \ , \tag{3.6}$$

hence the squared signal 2-norm of the prediction error is minimized. Since the prediction error depends for most model structures nonlinearly on the model coefficients θ, minimizing the cost function (3.6) for θ requires numerical optimization tools. Consequently, the optimization methods are faced with, e. g., local minima, convergence problems and the initial guess. Moreover, constraints such as stability of $H^\dagger G$ and $H^\dagger$ must be included in the optimization problem. However, these constraints are not considered in the literature for the identification of LPV models, cf. [Laurain et al. 2010; Tóth, Heuberger, et al. 2012], and also neglected here for reasons of simplicity. Note that the prediction error depends for particular model structures linearly on the unknowns and thus minimizing (3.6) can be solved in closed form.

Note that the objective of this thesis is to develop computational efficient estimation methods, since a large number of variables are unknown (for example the order and scheduling parameter dependency) and thus many different models need to be estimated. Moreover, models with a good estimation result can be used as initial guess in a subsequent model refinement step using computationally elaborated methods. However, even if the coefficients are obtained as a solution of an optimization problem, the model can have a low accuracy due to a poor choice in the previous steps, e. g., the model structure, the system excitation, and local minima or the loss function in the optimization problem.

In order to evaluate an identification method, statistical properties of the estimated coefficients are often used. For this purpose, it is assumed that the true, data-generating system corresponds to the selected model structure completely or at least to the system model structure G and that the corresponding coefficients are known. Then, an estimator is called consistent, if the expected value of the estimated coefficients approximates the true coefficients with a probability of one as

the number of samples N goes toward infinity, see [Tóth, Heuberger, et al. 2012, Definition 2.4]. Further, the estimator is unbiased, if this is also true for any (finite) number of samples N, see [Nelles 2001, p. 755]. However, these properties do not yet say anything about the variance of the estimated coefficients and a small variance is desirable. The unbiased estimator with the smallest variance is called efficient, see [Nelles 2001, p. 756]. Indeed, biased estimators can have a lower variance than the efficient estimator and, in some cases, the biased estimator is the preferable one, because it can provide more reliable estimates, see the illustrative example in [Nelles 2001, Fig. B.10]. However, all these estimator properties only apply if the true system corresponds to the selected model structure. In real-world applications, this is usually not or only approximately the case. Thus, these (important) properties are rather of theoretical nature. Nevertheless, even if the system does not correspond to the model structure, an estimation with prediction error method represents a meaningful approximation, because with this method the signal 2-norm of the prediction error is minimized.

Finally, the accuracy of the identified model must be validated and a detailed discussion on model validation can be found in [Ljung 1999, Chapter 16] and [Nelles 2001, Section 7.3]. However, "due to the lack of a general LPV validation theory, mis-modeled dynamics only show up in the performance loss of the designed controllers" [Tóth 2010, Section 7.1], which can be evaluated, e. g., by comparing the closed-loop behavior with the model and the true system in the loop. In this thesis, the best fit rate (BFR) and variance accounted for (VAF) with

$$\mathrm{BFR} = \max\left(1 - \frac{\|y_i - \hat{y}_i\|_{\ell_2}}{\|y_i - \bar{y}_i\|_{\ell_2}}, 0\right) \tag{3.7}$$

$$\mathrm{VAF} = \max\left(1 - \frac{\mathrm{var}(y_i - \hat{y}_i)}{\mathrm{var}(y_i)}, 0\right) \tag{3.8}$$

are used to determine the model accuracy for the i-th output, see [Tóth 2010, Definitions 2.3 and 2.4]. In above equations, $\hat{y}$ denotes the modeled and y the measured output signal vector, $\bar{y}$ is the vector of mean values for each signal in y, and var is the variance operator. The index i denotes the i-th signal in the vector of (output) signals. If the system has multiple outputs, the BFR and VAF need to be determined for each output, which allows evaluating the accuracy for each output individually and is in contrast to [Tóth 2010, Definitions 2.3 and 2.4]. A low BFR or VAF value indicates invalidity of the model. If the BFR is equal to zero, then the modeled output is no better than approximating the measured output with its mean value over all time instants. The variance accounted for indicates how much of the output variation is explained by the model, disregarding possible bias of the modeled output [Tóth 2010]. Hence, if the identified model is used for control design and the open control loop has integral action (a high gain at low frequencies), which is commonly demanded for stationary accuracy, a high VAF value for the identified model can be a sufficient measure for indicating the model accuracy. The modeled output $\hat{y}$ can be the predicted output $\hat{y}_{k|k-1}$ or the simulated output $\hat{y} = G(\mathrm{q}, \mu_k, \theta_G)u$, see (3.5), (3.2). In this thesis, the simulated output is used because it highlights model inaccuracies more. Hence, different error types are used in estimation and validation. In addition to the two performance criteria in the time domain, the local model behavior at different operating points obtained by Jacobian linearization can be analyzed in the frequency domain and compared with data sets containing a local system excitation at the corresponding operating points. Furthermore, validation can be performed using residual correlation analysis [Ljung 1999, Section 16.6]. Here, the cross-correlation of the prediction error ε with the input u and the auto-correlation of the prediction error ε (with past values)

are determined. If the prediction error ε can hardly be predicted by the past input nor by past prediction errors (correspondingly small values in the cross- and auto-correlation), the model has a high accuracy. However, residual correlation analysis is not used for validation in this thesis.

In conclusion, system identification is usually not a straightforward process but rather iterative, where each step should be subject to an analysis. It may be even necessary to have multiple iteration loops over single or multiple of the above four steps until a satisfactory result is achieved.

3.2 Review of LPV Identification Methods

In the last two decades, identification of LPV models has developed considerably. With its origin in LTI system identification, two major methods evolved: input-output and subspace identification. These methods provide LPV models in IO or SS representation that are suitable for model-based LPV SS control design, assuming that their state-space representation has static scheduling parameter dependence. In the same period, there are also several publications on the approach of identifying LPV models based on local LTI models. An extensive overview of different methods in LPV system identification is given in [Cox 2018, Fig. 1.2].

The identification of LPV IO models has its origin in the late 1990s and is based, analogous to the LTI case, on the minimization of the prediction error. In the first publications, the identification of LPV SISO systems is presented for the ARX model structure [Bamieh & Giarré 2002]. Polynomial dependence on a single scheduling parameter was assumed and it was established that the prediction error is linear in the model coefficients. Consequently, the estimation can be formulated as a linear least squares problem for which a global solution can be found. The extension to multiple scheduling parameters with an affine dependency was done in, e. g., [Bamieh & Giarré 2001]. In [Butcher et al. 2008] consistency in the identification of LPV ARX models is investigated, which can be achieved under certain conditions with the instrumental variables method. Around 2010 the identification of LPV SISO models is extended to the Box-Jenkins model structure in [Laurain et al. 2010] and further modified for spatially interconnected systems in [Ali et al. 2011]. In [Tóth, Heuberger, et al. 2012] a collection is given for several algorithms based on the prediction error method for the identification of IO models with different model structures. Finally, the transformation of IO models into SS models is discussed in [Tóth, Abbas, et al. 2012] that allows the connection between the identification in IO and SS domain, see Section 2.4.1. Note that most of the literature is about the identification in the discrete-time domain and one of the few publications investigating continuous-time LPV system identification is [Laurain et al. 2011], where the SISO case is considered in the IO domain.

One of the first publications on the identification of LPV SS models is from the mid 1990s [Nemani et al. 1995]. Here, the affine dependence on a single scheduling parameter and full-state measurement is assumed. At the end of the 1990s, the first research paper on nonlinear optimization of LPV SS models was published [Lee & Poolla 1999]. Both affine and rational scheduling parameter dependence are considered in nonlinear optimization, see [Lee & Poolla 1999; Wills & Ninness 2011]. However, the nonlinear, non-convex optimization problem depends on the initial estimate, is prone to local minima, and the number of optimization variables increases quadratically with the number of states and linearly with the number of scheduling parameters (assuming an affine scheduling parameter dependence). Parallel to the nonlinear

optimization approaches, the first research paper on subspace identification of LPV models were published, which do not require state measurements, e. g., [Verdult & Verhaegen 2000]. However, the presented approach contains matrices that grow exponentially, where the base is the number of scheduling parameters μ and the exponent is related to the model order. Around 2010 a kernel method is introduced to reduce the problem dimensions [Kulcsár et al. 2009; Wingerden & Verhaegen 2009] and is modified in [Proimadis et al. 2015]. Other approaches, in which an IO model serves as an intermediate step for the identification of a SS model, have recently been published [Cox & Tóth 2016; Schulz et al. 2017]. They also have a lower computational effort than the reduced subspace identification problem.

In the majority of publications on LPV system identification, as in this thesis, offline identification methods are addressed. Thus, the measured data set is first stored and then processed on a computer. In contrast, recursive identifications methods can run online or in real-time. Some methods are presented in [Isermann & Münchhof 2011, Section 9.4] and a recursive algorithm for the identification of LPV IO models with one scheduling parameter is given in [Bamieh & Giarré 2002]. However, similar to the identification in the continuous-time domain, only few literature exists on recursive methods for LPV system identification.

3.3 LPV Input-Output Model Identification

In Section 3.1 a general introduction to the field of system identification is given. In the following, the identification of DT LPV IO models is considered and dedicated methods are provided to solve the system identification problem, respectively the optimization problem. All methods are based on the prediction error framework. Note that only LPV IO model structures with one of the four forms in Section 2.4.1, i. e. with a specific dynamic scheduling parameter dependency, are considered in order to guarantee LPV SS models with static scheduling parameter dependence.

3.3.1 Model Structures

In the following, different model structures are presented and ordered by their complexity, which can all be represented with the general model structure in Fig. 3.2b and (3.2).

LPV ARX model structure For the LPV IO model in autoregressive with exogenous input (ARX) structure, the model equation is

$$A(\mathrm{q}, \mu_k)\hat{y}_k = B(\mathrm{q}, \mu_k)u_k + e_k \ . \tag{3.9}$$

LPV ARMAX model structure Extending the ARX model structure by an LPV moving average model structure for the noise model, i. e. incorporating the polynomial C, leads to the LPV IO model in autoregressive moving average with exogenous input (ARMAX) structure

$$A(\mathrm{q}, \mu_k)\hat{y}_k = B(\mathrm{q}, \mu_k)u_k + C(\mathrm{q}, \mu_k)e_k \ . \tag{3.10}$$

LPV OE model structure The LPV IO output error (OE) model structure is

$$\hat{y}_k = F^{\dagger}(\mathrm{q}, \mu_k)B(\mathrm{q}, \mu_k)u_k + e_k \ , \tag{3.11}$$

Table 3.1: Overview of system model G and noise model H for different model structures

	$G(\mathrm{q}, \mu_k)$	$H(\mathrm{q}, \mu_k)$
ARX	$A^\dagger(\mathrm{q}, \mu_k)B(\mathrm{q}, \mu_k)$	$A^\dagger(\mathrm{q}, \mu_k)$
ARMAX	$A^\dagger(\mathrm{q}, \mu_k)B(\mathrm{q}, \mu_k)$	$A^\dagger(\mathrm{q}, \mu_k)C(\mathrm{q}, \mu_k)$
OE	$F^\dagger(\mathrm{q}, \mu_k)B(\mathrm{q}, \mu_k)$	I
BJ	$F^\dagger(\mathrm{q}, \mu_k)B(\mathrm{q}, \mu_k)$	$D^\dagger(\mathrm{q})C(\mathrm{q})$

i. e. the noise is directly added to the output. Note that for this model structure the prediction error is equal to the output error since in both cases the error is the difference between y_k and $F^\dagger(\mathrm{q}, \mu_k)B(\mathrm{q}, \mu_k)u_k$.

LPV BJ model structure Extending the OE model structure by a dynamic LTI noise model structure, the LPV IO model in Box-Jenkins (BJ) structure is obtained with

$$\hat{y}_k = F^\dagger(\mathrm{q}, \mu_k)B(\mathrm{q}, \mu_k)u_k + D^\dagger(\mathrm{q})C(\mathrm{q})e_k \ . \tag{3.12}$$

Note that the noise model is assumed to be scheduling parameter independent.

The four model structures are summarized in Table 3.1, where the system and noise model are given for each model structure, cf. Fig. 3.2b. Note that a constant input delay can easily be incorporated into the model structures by multiplying the system model G, respectively the polynomial B, with a delay at the input. However, it is assumed here that the constant delay, if any, is automatically identified by the B polynomial with the first coefficient functions then becoming zero.

The terminology of some model structures can be confusing, since it originates from modeling in economics, where stochastic models are considered with input e. Hence, the deterministic input u is regarded as exogenous input (X). The first two model structures contain an autoregressive (AR) part, since the current output $y(k)$ depends due to the polynomial $A(\mathrm{q}, \mu_k)$ on weighted past values of the output. Moving average denotes the $C(\mathrm{q}, \mu_k)$ polynomial on the noise such that the output depends on a weighted sum of (delayed) noise samples.

The ARX model structure is the simplest but also most commonly used one, since the identification problem becomes linear in the unknowns and thus it can be solved without iterations. All other model structures require an iterative procedure to determine the model coefficients, usually using the coefficients of an ARX model as starting values.

For the two model structures ARX and ARMAX, the input u and the noise e have common dynamics in the output $\hat{y}$. Thus, the system model G and the noise model H share dynamics and so θ_G and θ_H have common coefficients. In contrast, it is assumed that the system model G and the noise model H in the two model structures OE and BJ have entirely different dynamics.

The presented LPV polynomials A, B, C, D, and F have as indeterminate the forward time shift operator q and they are of orders n_a, n_b, etc. They are, by the example of the shifted form, given as

$$A(\mathrm{q}, \mu_k) = I + \sum_{i=1}^{n_a} a_i(\mu_{k-i})\mathrm{q}^{-i} \ , \qquad B(\mathrm{q}, \mu_k) = \sum_{j=0}^{n_b} b_j(\mu_{k-j})\mathrm{q}^{-j} \tag{3.13}$$

with coefficient matrix functions $a_i(\mu_{k-i}) : \mathbb{P}_\mu \to \mathbb{R}^{n_y \times n_y}$ and $b_j(\mu_{k-j}) : \mathbb{P}_\mu \to \mathbb{R}^{n_y \times n_u}$. The polynomials $F(\mathrm{q}, \mu_k)$, $C(\mathrm{q}, \mu_k)$[6)], and $D(\mathrm{q})$ are given analogously as the monic polynomial $A(\mathrm{q}, \mu_k)$. Note that the polynomials in (3.13) have a dynamic dependency on the scheduling parameter. Despite this, the notation with the operator $\diamond$ is omitted because the polynomials already contain the forward time shift operator q as an argument, which allows and indicates a dynamic dependence. Because all the coefficient functions in polynomial B are free, i. e. B is not normalized, the polynomials A and F need to be monic, otherwise the model structure is overparameterized and not identifiable. Since $H(0, \mu_k) = I$ as already demanded in the model structure, see (3.5), the polynomials C and D are also assumed as monic.

It is assumed that the coefficient matrix functions of the LPV polynomials have an affine scheduling parameter dependency such that they can be parametrized by constant coefficient matrices. For example, the scheduling parameter dependent coefficient matrix functions in $A(\mathrm{q}, \mu_k)$ are

$$a_i(\mu_{k-i}) = \sum_{l=1}^{n_\mu} a_i^{(l-1)} \mu_{k-i}^{(l)} = a_i^{(0)} + \sum_{l=2}^{n_\mu} a_i^{(l-1)} \mu_{k-i}^{(l)} \tag{3.14}$$

for $i \in \{1, 2, \ldots, n_a\}$, where the constant coefficient matrices $a_i^{(l)} \in \mathbb{R}^{n_y \times n_y}$ are unknown and the superscript (l) denotes the l-th entry of the scheduling parameter vector and the corresponding coefficient matrix, where $\mu^{(1)} = 1$.

In (3.13) and (3.14) the shifted form is used, however other dynamic dependencies on the scheduling parameter as these in the OF, AF, or extended AF are also possible. Nevertheless, in the following, the problem formulation will be shown by the example of the SF. For the other forms, the time shifts of the scheduling parameter must be replaced by the appropriate ones of the corresponding form and in the case of the OF the structure of polynomial B must be respected.

Finally, a remark on identifiability is given for the presented IO model structures assuming an affine scheduling parameter dependence.

Remark 3.2. *Identifiability, see [Boonto 2011, Definition 5.10], implies for LPV model structures with linear scheduling parameter dependence that each set of basis functions is linearly independent on $\mathbb{P}_\mu$ [Tóth, Heuberger, et al. 2012, Theorem 2.1]. The basis functions are used to compute the linear scheduling parameter vector based on the (quasi) scheduling parameter candidates p, u, and y, i. e. $\mu^{\mathrm{T}} = [1 \quad f_2(p,u,y) \quad \ldots \quad f_{n_\mu}(p,u,y)]$ with n_μ scalar basis functions f_i, $i \in \{1, 2, \ldots, n_\mu\}$, and $f_1(p,u,y) = 1$.*

For quasi LPV models, it is no longer sufficient to consider only the basic functions, which are used to calculate the affine scheduling parameters, to assess identifiability. Instead, the entire regressor vector, respectively matrix, must be examined for linear dependencies. For example, a quasi LPV SISO model with linear scheduling parameter vector $\mu^{\mathrm{T}} = [1 \quad u \quad y]$ is not identifiable, since the regressor then contains $[u_{k-j} \quad u_{k-j}u_{k-l} \quad u_{k-j}y_{k-l} \quad y_{k-i} \quad y_{k-i}u_{k-l} \quad y_{k-i}y_{k-l}]$. The regressor becomes linearly dependent, whenever $i = j = l$, i. e. the third entry is equal to the fifth in above vector with $u_{k-i}y_{k-i}$. This occurs several times in the SF and one time in the AF and OF.

[6)] The polynomial C may depend on the scheduling parameter, as it is assumed in the ARMAX model structure, or not, as in the BJ model structure. If both are possible, the more general scheduling parameter dependent notation is used. The same applies to the noise model H.

Considering that and removing the redundant entries in the regressor vector, it is argued in [Tóth, Heuberger, et al. 2012] and [Boonto 2011, Section 5.5], that the LPV ARX model structure is globally identifiable.

Next, the identification problems for the different model structures are formulated and algorithms for solving them are presented. In addition, properties of the estimators for the different algorithms are discussed. The following problem formulations and algorithms are based on [Schulz et al. 2018; Tóth, Heuberger, et al. 2012].

3.3.2 LPV ARX Model

First, the identification of LPV ARX models is reviewed, see [Bamieh & Giarré 2002; Tóth, Heuberger, et al. 2012]. Then, it is shown how the LPV ARX identification problem can be reformulated such that LTI identification tools can be used, which is one of the main contributions in this thesis, cf. [Schulz et al. 2018].

The predicted output $\hat{y}_{k|k-1}$ of the LPV model in SF ARX structure with the affine scheduling parameter dependent polynomials in (3.13) becomes

$$\begin{aligned}\hat{y}_{k|k-1} &= (I - A(\mathrm{q}, \mu_k))y_k + B(\mathrm{q}, \mu_k)u_k \\ &= -\sum_{i=1}^{n_a}\sum_{l=1}^{n_\mu} a_i^{(l-1)}\mu_{k-i}^{(l)} y_{k-i} + \sum_{j=0}^{n_b}\sum_{l=1}^{n_\mu} b_j^{(l-1)}\mu_{k-j}^{(l)} u_{k-j} = \theta^{\mathrm{T}} z_k \end{aligned} \tag{3.15}$$

with $z_k \in \mathbb{R}^{n_\mu(n_y n_a + n_u(n_b+1))}$ the regressor vector and $\theta \in \mathbb{R}^{n_\mu(n_y n_a + n_u(n_b+1))\times n_y}$ the coefficient matrix that are for the ARX structure

$$z_k^{\mathrm{T}} = \begin{bmatrix} -\mu_{k-1}^{\mathrm{T}} \otimes y_{k-1}^{\mathrm{T}} & \dots & -\mu_{k-n_a}^{\mathrm{T}} \otimes y_{k-n_a}^{\mathrm{T}} & \mu_k^{\mathrm{T}} \otimes u_k^{\mathrm{T}} & \dots & \mu_{k-n_b}^{\mathrm{T}} \otimes u_{k-n_b}^{\mathrm{T}} \end{bmatrix} \tag{3.16}$$

$$\theta^{\mathrm{T}} = \begin{bmatrix} a_1^{(0)} & \dots & a_1^{(n_\mu-1)} & \dots & a_{n_a}^{(0)} & \dots & a_{n_a}^{(n_\mu-1)} \\ & & & & & & b_0^{(0)} & \dots & b_0^{(n_\mu-1)} & \dots & b_{n_b}^{(0)} & \dots & b_{n_b}^{(n_\mu-1)} \end{bmatrix}, \tag{3.17}$$

where $\otimes$ denotes the Kronecker product, see Appendix A.1. Note that usually the predicted output $\hat{y}_{k|k-1}$ depends, in prediction error method, only on past inputs u_{k-i} and outputs y_{k-i} with $i \in \mathbb{N}^+$ [Ljung 1999, Chapters 3 and 4]. However, if the true system contains a feedthrough, this must also be taken into account when predicting the output, as is done in (3.16). This means that $\hat{y}_{k|k-1}$ is no longer dependent only on past inputs and outputs, but also on the current input u_k.

The solution for the LS in the prediction error, see (3.4) and minimizing (3.6), is obtained directly since the prediction error depends linearly on the model coefficients, hence

$$\theta_{\mathrm{LS}} = \left(Z^{\mathrm{T}} Z\right)^{-1} Z^{\mathrm{T}} Y \tag{3.18}$$

$$Y^{\mathrm{T}} = \begin{bmatrix} y_{n_a+1} & y_{n_a+2} & \dots & y_N \end{bmatrix} \in \mathbb{R}^{n_y \times N - n_a} \tag{3.19}$$

$$Z^{\mathrm{T}} = \begin{bmatrix} z_{n_a+1} & z_{n_a+2} & \dots & z_N \end{bmatrix} \in \mathbb{R}^{n_\mu(n_y n_a + n_u(n_b+1))\times N - n_a} \tag{3.20}$$

with N the number of data samples and assuming that $n_a \geq n_b$. In the following, the regressor vectors and coefficient matrices are given for the other three IO forms with (dynamic) dependence

Algorithm 1 Identification of an LPV ARX model using the (regularized) LS method, [Tóth, Heuberger, et al. 2012]

Require: Measured u, y, p, sampling time T_s, orders n_a, n_b, regularization $K \succeq 0$
Ensure: M_LPV a DT LPV IO ARX model
1: $\mu \leftarrow f_\mu(p, u, y)$ ▷ compute linear scheduling parameter vector, cf. Fig. 2.1b
2: $Y \leftarrow [y_{n_a+1}, y_{n_a+2}, \ldots, y_N]^\mathrm{T}$ ▷ see (3.19)
3: $Z \leftarrow \mathrm{regressor}(y, u, \mu, [n_a, n_b], \text{'form'})$ ▷ cf. (3.16), (3.20); 'form' is, e. g., 'SF'
4: $\theta \leftarrow (Z^\mathrm{T}Z + K)^{-1}Z^\mathrm{T}Y$
5: $[a_i^{(l)}, b_j^{(l)}] \leftarrow \mathrm{coefficients}(\theta, n_y, n_u, n_\mu, [n_a, n_b], \text{'form'})$ ▷ get model coefficients, cf. (3.17)
6: $M_\mathrm{LPV} \leftarrow \{a_i^{(l)}, b_j^{(l)}, T_\mathrm{s}\}$

on the scheduling parameter. For the AF, they are

$$z_{k,\mathrm{AF}}^\mathrm{T} = \begin{bmatrix} -\mu_{k-1}^\mathrm{T} \otimes y_{k-1}^\mathrm{T} & \ldots & -\mu_{k-1}^\mathrm{T} \otimes y_{k-n_a}^\mathrm{T} \\ & & \mu_k^\mathrm{T} \otimes u_k^\mathrm{T} & \mu_{k-1}^\mathrm{T} \otimes u_{k-1}^\mathrm{T} & \ldots & \mu_{k-1}^\mathrm{T} \otimes u_{k-n_b}^\mathrm{T} \end{bmatrix} \tag{3.21a}$$

$$\theta_\mathrm{AF}^\mathrm{T} = \begin{bmatrix} a_1^{(0)} & \ldots & a_1^{(n_\mu-1)} & \ldots & a_{n_a}^{(0)} & \ldots & a_{n_a}^{(n_\mu-1)} \\ & & & & b_0^{(0)} & \ldots & b_0^{(n_\mu-1)} & \ldots & b_{n_b}^{(0)} & \ldots & b_{n_b}^{(n_\mu-1)} \end{bmatrix} \tag{3.21b}$$

and for the OF, where additionally the constrain on the structure of the B polynomial is taken into account, they are

$$z_{k,\mathrm{OF}}^\mathrm{T} = \begin{bmatrix} -\mu_{k-1}^\mathrm{T} \otimes y_{k-1}^\mathrm{T} & \ldots & -\mu_{k-n_a}^\mathrm{T} \otimes y_{k-n_a}^\mathrm{T} & \mu_{k-n_a}^\mathrm{T} \otimes u_{k-n_a}^\mathrm{T} \end{bmatrix} \tag{3.22a}$$

$$\theta_\mathrm{OF}^\mathrm{T} = \begin{bmatrix} a_1^{(0)} & \ldots & a_1^{(n_\mu-1)} & \ldots & a_{n_a}^{(0)} & \ldots & a_{n_a}^{(n_\mu-1)} & b_{n_a}^{(0)} & \ldots & b_{n_a}^{(n_\mu-1)} \end{bmatrix} . \tag{3.22b}$$

Concluding, the regressor vector and coefficient matrix for the extended AF are

$$z_{k,\mathrm{eAF}}^\mathrm{T} = \begin{bmatrix} -\mu_k^\mathrm{T} \otimes y_{k-1}^\mathrm{T} & \ldots & -\mu_k^\mathrm{T} \otimes y_{k-n_a}^\mathrm{T} & \mu_k^\mathrm{T} \otimes u_k^\mathrm{T} & \ldots & \mu_k^\mathrm{T} \otimes u_{k-n_b}^\mathrm{T} \end{bmatrix} \tag{3.23a}$$

$$\theta_\mathrm{eAF}^\mathrm{T} = \begin{bmatrix} a_1^{(0)} & \ldots & a_1^{(n_\mu-1)} & \ldots & a_{n_a}^{(0)} & \ldots & a_{n_a}^{(n_\mu-1)} \\ & & & & b_0^{(0)} & \ldots & b_0^{(n_\mu-1)} & \ldots & b_{n_b}^{(0)} & \ldots & b_{n_b}^{(n_\mu-1)} \end{bmatrix} . \tag{3.23b}$$

As pointed out in [Nelles 2001, Section 3.1], the covariance matrix of θ_LS depends linearly on the inverse of $Z^\mathrm{T}Z$, hence the eigenvalues of $Z^\mathrm{T}Z$ should be large or at least increase with the number of samples N in order to achieve a small variance. The identification procedure is given in Algorithm 1. Note that instead of the LS solution in (3.18), a regularized solution is suggested in Line 4 of the algorithm, more precisely the Tikhonov regularization method. With regularization overfitting can be avoided, i. e. representing the noise e by the system model is prevented. Usually, the intention of regularization is to search for entries in the coefficient matrix θ that only marginally affect the loss function, i. e. the signal 2-norm of the prediction error. Then, these coefficients are treated as redundant and pushed toward zero instead of reaching almost arbitrary values due to their low sensitivity (high variance) in the optimization such that only the significant coefficients remain [Sima 2006, Section 1.2]. This is particularly relevant for model structures with high flexibility, i. e. many entries in θ.

Note that coefficients can have a high variance not only if the model structure has too many degrees of freedom to approximate the true system, but also if the system excitation is too poor or the noise is too large to collect enough information in the data set for all coefficients [Nelles 2001, Section 3.1]. Thus, a high covariance of the coefficients in an estimate means redundancy for the data set but not necessarily redundancy for the system. Regardless of this, the redundant coefficients or an estimate of the covariance matrix of the coefficients must be known for a sophisticated regularization in order to penalize primarily redundant coefficients w. r. t. the data set that cannot be estimated reliably, i. e. they possess a high variance. In simpler approaches, all coefficients are penalized and pushed toward zero, even the significant ones. With Tikhonov regularization, the loss function is modified and the squared values of the model coefficients weighted by the entries in matrix K are added to the signal 2-norm of the prediction error. Consequently, the condition number of $Z^{\mathrm{T}}Z + K$ can be reduced, however the resulting reduction of the variance in the coefficients is accompanied by a bias (bias-variance trade-off).

Due to the favorable choice of the noise model in the ARX model structure, the model coefficients can be determined directly. However, the model coefficients are biased if the assumption of the noise model does not apply for the true system. To prevent the bias, the instrumental variables (IV) method can be applied. By simulating the estimated model, a regressor matrix Z_{IV} can be constructed in a similar way as Z but with the simulated, noise free outputs $\hat{y} = G(\mathrm{q}, \mu, \theta)u$ instead of the noisy measured outputs y. Assuming a noise-free scheduling signal, the matrix Z_{IV} is then uncorrelated with the prediction error, in contrast to Z, and thus a bias-free or consistent estimate is achieved. However, the variance of the estimated model coefficients is increased. If the model accuracy is high, then Z_{IV} highly correlates with Z, hence the variance becomes small, where the minimal variance is achieved for $Z_{\mathrm{IV}} = Z$, which gives a bias. The IV method is explained in more detail in, e. g., [Nelles 2001, Section 16.5]. By applying the IV method, the model coefficients

$$\theta_{\mathrm{IV}} = \left(Z_{\mathrm{IV}}^{\mathrm{T}} Z\right)^{-1} Z_{\mathrm{IV}}^{\mathrm{T}} Y \tag{3.24}$$

are estimated iteratively, where the simulated output (entering Z_{IV}) is updated in each iteration step by simulating the model with the estimated model coefficients in θ_{IV} of the preceding step. The iterative identification procedure is given in Algorithm 2, where regularization can be similarly included as in Algorithm 1. The while loop usually requires just a few iterations and (a combination of) many criteria can be tested to examine convergence, e. g., maximal number of iterations or the variation of θ or of the prediction error between two iterations[7)].

The identification problem for the LPV ARX model structure can be reformulated such that LTI identification methods can be used. The idea is to extract the so-called LTI part in the scheduling parameter dependent AR model structure $A(\mathrm{q}, \mu_k)y_k$ and treat the remaining signals as new extended input signals. Thus, using (3.15) and (3.14) the predicted model output

[7)] Similar criteria to examine convergence are used in the iterative identification of models with ARMAX and BJ structure, see Algorithms 4 and 5.

Algorithm 2 Identification of an LPV ARX model using the IV method, [Tóth, Heuberger, et al. 2012]

Require: Measured u, y, p, sampling time T_s, orders n_a, n_b
Ensure: M_LPV a DT LPV IO ARX model

$\mu \leftarrow f_\mu(p, u, y)$ ▷ compute linear scheduling parameter vector, cf. Fig. 2.1b
$Y \leftarrow [y_{n_a+1}, y_{n_a+2}, \ldots, y_N]^\mathrm{T}$ ▷ see (3.19)
$Z \leftarrow \mathrm{regressor}(y, u, \mu, [n_a, n_b], \text{'form'})$ ▷ cf. (3.16), (3.20); 'form' is, e. g., 'SF'
$\theta \leftarrow (Z^\mathrm{T} Z)^{-1} Z^\mathrm{T} Y$
while loop is not converged **do**
    $\hat{y} \leftarrow \mathrm{simulate}(\theta, u, \mu, [n_a, n_b], \text{'form'})$
    $Z_\mathrm{IV} \leftarrow \mathrm{regressor}(\hat{y}, u, \mu, [n_a, n_b], \text{'form'})$
    $\theta \leftarrow (Z_\mathrm{IV} Z^\mathrm{T})^{-1} Z_\mathrm{IV} Y$
end while
$[a_i^{(l)}, b_j^{(l)}] \leftarrow \mathrm{coefficients}(\theta, n_y, n_u, n_\mu, [n_a, n_b], \text{'form'})$ ▷ get model coefficients, see (3.17)
$M_\mathrm{LPV} \leftarrow \{a_i^{(l)}, b_j^{(l)}, T_\mathrm{s}\}$

becomes

$$\begin{aligned}
\hat{y}_{k|k-1} &= -\sum_{i=1}^{n_a}\sum_{l=1}^{n_\mu} a_i^{(l-1)} \mu_{k-i}^{(l)} y_{k-i} + \sum_{j=0}^{n_b}\sum_{l=1}^{n_\mu} b_j^{(l-1)} \mu_{k-j}^{(l)} u_{k-j} \\
&= \underbrace{-\sum_{i=1}^{n_a} a_i^{(0)} y_{k-i}}_{(I-\tilde{A}(\mathrm{q}))y_k} - \sum_{i=1}^{n_a}\sum_{l=2}^{n_\mu} a_i^{(l-1)} \mu_{k-i}^{(l)} y_{k-i} + \sum_{j=0}^{n_b}\sum_{l=1}^{n_\mu} b_j^{(l-1)} \mu_{k-j}^{(l)} u_{k-j} \\
&= (I - \tilde{A}(\mathrm{q}))y_k + \tilde{b}^\mathrm{T} \tilde{u}_{k|k-p}
\end{aligned} \tag{3.25}$$

with the extended coefficient matrix $\tilde{b}^\mathrm{T} \in \mathbb{R}^{n_y \times n_{\tilde{b}}}$ and the extended input signal vector $\tilde{u}_{k|k-p} \in \mathbb{R}^{n_{\tilde{b}}}$, where $n_{\tilde{b}} = (n_\mu - 1)n_y n_a + n_\mu n_u (n_b + 1)$. The index $k|k-p$ denotes that $\tilde{u}_{k|k-p}$ depends on outputs, inputs, and scheduling parameters from the past time instant $k-p$ with $p = \max(n_a, n_b)$ up to k. Both $\tilde{u}_{k|k-p}$ and $\tilde{b}^\mathrm{T}$ are given similarly to (3.16) and (3.17) as

$$\begin{aligned}
\tilde{u}_{k|k-p}^\mathrm{T} = \Big[&-\theta_{k-1}^\mathrm{T} \otimes y_{k-1}^\mathrm{T} \quad -\theta_{k-2}^\mathrm{T} \otimes y_{k-2}^\mathrm{T} \quad \ldots \quad -\theta_{k-n_a}^\mathrm{T} \otimes y_{k-n_a}^\mathrm{T} \\
&\mu_k^\mathrm{T} \otimes u_k^\mathrm{T} \quad \mu_{k-1}^\mathrm{T} \otimes u_{k-1}^\mathrm{T} \quad \ldots \quad \mu_{k-n_b}^\mathrm{T} \otimes u_{k-n_b}^\mathrm{T} \Big]
\end{aligned} \tag{3.26}$$

$$\begin{aligned}
\tilde{b}^\mathrm{T} = \Big[&a_1^{(1)} \quad \ldots \quad a_1^{(n_\mu-1)} \quad \ldots \quad a_{n_a}^{(1)} \quad \ldots \quad a_{n_a}^{(n_\mu-1)} \\
&b_0^{(0)} \quad \ldots \quad b_0^{(n_\mu-1)} \quad \ldots \quad b_{n_b}^{(0)} \quad \ldots \quad b_{n_b}^{(n_\mu-1)} \Big] .
\end{aligned} \tag{3.27}$$

Hence, in the SISO case, the identification of an LPV ARX model is the same as the identification of an LTI ARX multi-input single-output model[8]. This is illustrated in Fig. 3.3, neglecting the right block for the C polynomial. Now, software packages for LTI system identification can be applied to estimate the parameters of the makeshift ARX model structure (like the `arx` command in the MATLAB system identification toolbox). The output y and the order

[8] In the MIMO case, both the LPV and the LTI models are MIMO, where the LTI model structure has significantly more inputs.

n_a remain the same, while the extended input needs to be calculated as in (3.26) and the order of the polynomial for the extended input is one. The identification procedure is given in Algorithm 3, where in Line 6 the `armax` command must be replaced by the `arx` command. Then, only the coefficients of the estimated LTI polynomials $\tilde{A}(\mathrm{q})$ and $\tilde{b}$ have to be assigned to the LPV polynomials $A(\mathrm{q}, \mu_k)$ and $B(\mathrm{q}, \mu_k)$ using (3.13), (3.25), and (3.27).

Next, some remarks on the estimator properties of the LPV ARX model structure are in order. If the inverse of the square matrix $Z^{\mathrm{T}}Z$ in (3.18) exists, i. e. $Z^{\mathrm{T}}Z \succ 0$, the system is referred to as persistently excited, cf. [Wei & Re 2006]. Sometimes the term persistence of excitation is connected to a requirement on the input signal u, however this only applies if the regressor matrix Z consists exclusively of input signals. In dynamic systems, the output y usually also appears in the regressor, which is also the case for the ARX model structure. Therefore, strictly speaking, persistence of excitation is only met if u and y satisfy certain conditions to guarantee full rank of $Z^{\mathrm{T}}Z$. Conditions for persistence of excitation are also given in [Bamieh & Giarré 2002], which are similarly based on the requirement that the inverse of $Z^{\mathrm{T}}Z$ exists. According to [Bamieh & Giarré 2002], two main conditions must be satisfied from an LTI point of view. For a finite set of scheduling parameter samples, the LPV system is a family of LTI systems and thus the input signals must be rich enough to excite persistently each LTI system of the family (first condition). Since this must be satisfied for each LTI system, the scheduling parameter trajectory must visit the dedicated scheduling parameter values sufficiently many times (second condition). In addition to these two conditions of persistence of excitation, the conditions for identifiability must be satisfied to guarantee structurally a full rank of $Z^{\mathrm{T}}Z$.

Consistency of the estimator for SISO LPV ARX systems with static dependence on the scheduling parameters is investigated in [Butcher et al. 2008]. Therein it is shown that the estimation of the coefficients using the linear least squares method is – similar to the LTI case – typically not consistent. Only if the regressor vector z_k is uncorrelated with the noise contained in the current data set (which is not the case if the model structure contains an AR part), the linear LS method provides consistent estimates. Using the ARX model structure, consistent coefficient estimates are obtained with the instrumental variables method, if instruments are chosen that are uncorrelated with the noise in the current data set. Although consistency is achieved with the IV method, the variance in the coefficient estimate increases[9)] so that the (problem dependent) question arises whether the bias-variance trade-off provides any advantages [Nelles 2001, Section 16.5 and Fig. B.10]. Moreover, the consistency benefit using the IV method is unclear in case of modeling errors, where no consistency exists. Consistency analysis for MIMO LPV systems or LPV IO systems with dynamic scheduling parameter dependency has not yet been investigated.

3.3.3 LPV ARMAX Model

The identification of LPV ARMAX models is rarely considered in the literature. In [Lu et al. 2013] the coefficients of locally identified LTI ARMAX models are interpolated to obtain an LPV ARMAX model. An identification algorithm for LPV ARMAX models with LPV C polynomial is given in [Tóth, Heuberger, et al. 2012]

In the following, two contributions of this thesis are presented. First, the identification of LPV ARMAX models with LTI C polynomial is considered. Due to the restriction to LTI C

[9)] A high correlation between the instrumental variables in Z_{IV} and the regressors in Z is desired for a low variance in the coefficient estimate (the highest correlation is achieved if Z_{IV} would be equal to Z) [Nelles 2001, Section 16.5].

Algorithm 3 Identification of an LPV ARMAX model using LTI methods, [Schulz et al. 2018]

Require: Measured u, y, and p, sampling time T_s, and orders n_a, n_b, n_c
Ensure: M_LPV a DT LPV IO ARMAX model with LTI C polynomial

1: $\mu \leftarrow f_\mu(p, u, y)$ ▷ compute linear scheduling parameter vector, cf. Fig. 2.1b
2: $Y \leftarrow [y_{n_a+1}, y_{n_a+2}, \ldots, y_N]^\mathrm{T}$ ▷ see (3.19)
3: $\tilde{u}_{k|k-p} \leftarrow [-\theta_{k-1}^\mathrm{T} \otimes y_{k-1}^\mathrm{T}, \ldots, -\theta_{k-n_a}^\mathrm{T} \otimes y_{k-n_a}^\mathrm{T}, \mu_k^\mathrm{T} \otimes u_k^\mathrm{T}, \ldots, \mu_{k-n_b}^\mathrm{T} \otimes u_{k-n_b}^\mathrm{T}]$ ▷ see (3.26) for SF
4: $\tilde{U} \leftarrow [\tilde{u}_{n_a+1|1}, \tilde{u}_{n_a+2|2}, \ldots, \tilde{u}_{N|N-n_a}]^\mathrm{T}$
5: $\mathbb{D} \leftarrow \{Y, \tilde{U}, T_\mathrm{s}\}$
6: $M_\mathrm{LTI} \leftarrow \mathrm{armax}(\mathbb{D}, [n_a, 1, n_c])$
7: $[a_i^{(l)}, b_j^{(l)}, c_m^{(0)}] \leftarrow \mathrm{coefficients}(M_\mathrm{LTI})$ ▷ get model coefficients, see (3.27), (3.28)
8: $M_\mathrm{LPV} \leftarrow \{a_i^{(l)}, b_j^{(l)}, c_m^{(0)}, T_\mathrm{s}\}$

polynomials, well-known LTI algorithms can be used for the identification, cf. [Schulz et al. 2018]. This is one of the main contributions of this thesis. Second, the algorithm given in [Tóth, Heuberger, et al. 2012] is modified by introducing a gradient related update mechanism in the iteration procedure, cf. Algorithm 4. This contributes to the convergence of the coefficients and the identification of models with high accuracy.

From a noise-handling point of view, the ARMAX model structure seems to be appealing, since its state-space representation is structurally similar to the one assumed in the subspace identification approach, see [Schulz et al. 2017] and Section 3.5.1, and is thus suitable for comparison of both identification approaches.

First, the LPV ARMAX model structure is considered with an LTI C polynomial. The restriction to an LTI C polynomial makes sense, since the resulting algorithm is considerably less complicated than the one presented in [Tóth, Heuberger, et al. 2012] for LPV C polynomials, it also supports MIMO systems, and it is a straightforward extension of identifying DT LPV ARX models using LTI methods in the preceding subsection.

The identification problem with an LTI C polynomial is formulated as an extension of (3.25) using now additionally a monic noise polynomial $C(\mathrm{q})$, hence the predicted output is

$$\hat{y}_{k|k-1} = -\sum_{i=1}^{n_a} a_i^{(0)} y_{k-i} + \tilde{b}^\mathrm{T} \tilde{u}_{k|k-p} + \sum_{m=1}^{n_c} c_m^{(0)} e_{k-m} \ . \tag{3.28}$$

The C polynomial only has the zeroth coefficients $c_m^{(0)}$ w. r. t. the scheduling parameter so that it becomes LTI, see (3.14). The first term on the right hand side in (3.28) represents the scheduling parameter independent or LTI part of $A(\mathrm{q}, \mu_k)$. The remaining coefficients of polynomial A, i. e. $a_i^{(l)}$ with $l \in \{1, 2, \ldots, n_\theta\}$, are contained in $\tilde{b}$, see (3.27). This rewriting of the LPV ARMAX model structure in (3.28) is illustrated in Fig. 3.3. This is again an identification problem of an LTI system and well-known approaches can be deployed, where the identification procedure is presented in Algorithm 3. In comparison with the ARX model structure, the ARMAX identification problem is no longer linear in the unknowns, since the unknown noise signal e is filtered by the unknown polynomial $C(\mathrm{q}) - I$. The unknown noise e that enters the regressor is usually replaced by the prediction error ε. Hence, the regressor depends on the predicted output $\hat{y}_{k|k-1}$ and the resulting nonlinear identification problem needs to be solved iteratively.

Next, some remarks on the estimator properties of Algorithm 3 are in order. To this end, it

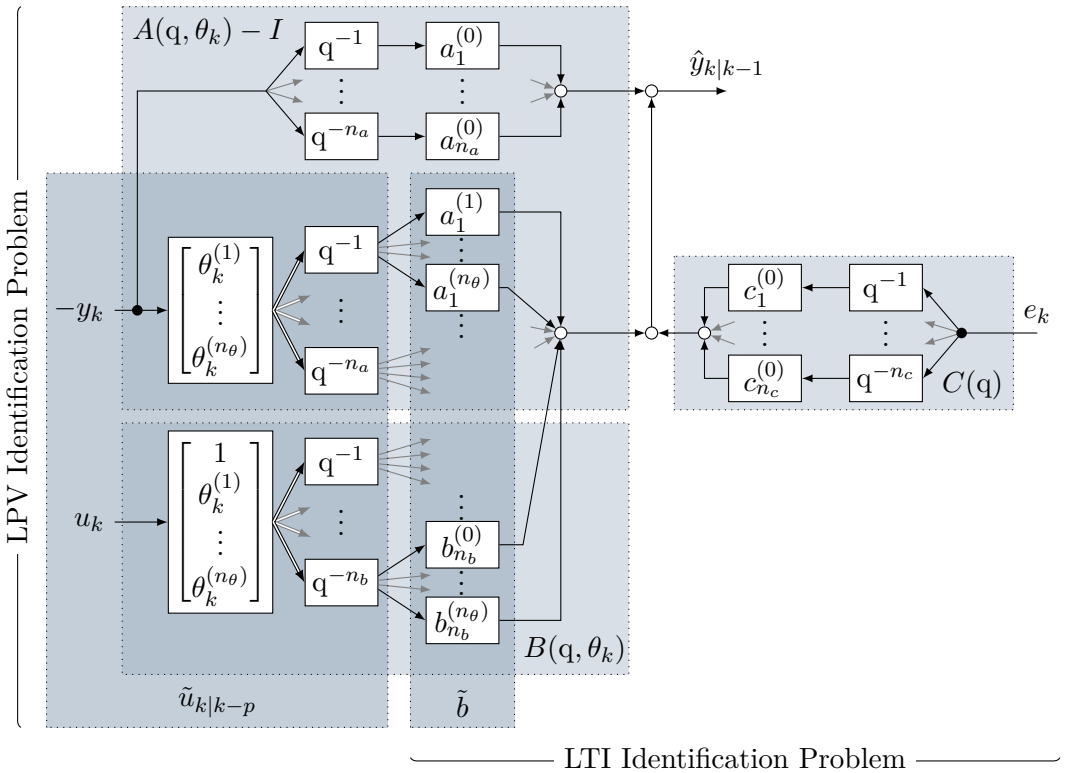


Figure 3.3: Block diagram of an LPV ARMAX model in SF with the affine scheduling parameter vector θ and an LTI C polynomial for illustrating the relation between LPV and LTI system identification problem, where the double arrows indicate that the entire output (vector) is fed to the next block

is instructive to look at the auxiliary LTI problem, which is solved in Line 6, in more detail. The input vector $\tilde{u}_{k|k-p}$ to the LTI ARMAX model is made up of two components: the delayed inputs u_{k-j} and outputs y_{k-i} with $j \in \{0, 1, \ldots, n_b\}$ and $i \in \{1, 2, \ldots, n_a\}$, see (3.26). Hence, the construction of $\tilde{u}_{k|k-p}$ can be interpreted as having a given LTV block diagonal controller $K(\mathrm{q}, \mu_k)$ with two-degrees-of-freedom, where

$$\tilde{u}_{k|k-p} = K(\mathrm{q}, \mu_k) \begin{bmatrix} y_k \\ u_k \end{bmatrix} = \begin{bmatrix} K_1(\mathrm{q}, \mu_k) & 0 \\ 0 & K_2(\mathrm{q}, \mu_k) \end{bmatrix} \begin{bmatrix} y_k \\ u_k \end{bmatrix} . \tag{3.29}$$

Note that the LTV controller has no freely adjustable coefficients, but the scheduling parameters can be seen as time varying coefficients. Consequently, the estimation problem can be interpreted as the identification of an LTI model operated in closed-loop with a given LTV block diagonal controller $K(\mathrm{q}, \mu_k)$ with two-degrees-of-freedom, see Fig. 3.4. It is straightforward to show that $K_1(\mathrm{q}, \mu_k)$ and $K_2(\mathrm{q}, \mu_k)$ can be written as a concatenation of $n_a n_\theta$, respectively $(n_b + 1)n_\mu$, matrix functions, i. e.

$$K_1(\mathrm{q}, \mu_k) = -\begin{bmatrix} \theta_{k-1}\mathrm{q}^{-1} \\ \theta_{k-2}\mathrm{q}^{-2} \\ \vdots \\ \theta_{k-n_a}\mathrm{q}^{-n_a} \end{bmatrix} \otimes I_{n_y} , \qquad K_2(\mathrm{q}, \mu_k) = \begin{bmatrix} \mu_k \mathrm{q}^0 \\ \mu_{k-1}\mathrm{q}^{-1} \\ \vdots \\ \mu_{k-n_b}\mathrm{q}^{-n_b} \end{bmatrix} \otimes I_{n_u} , \tag{3.30}$$

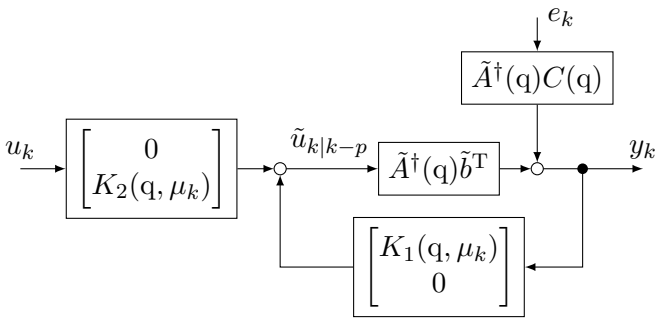


Figure 3.4: Closed-loop identification setup for the ARMAX LPV-LTI algorithm

where K_1 depends on the affine scheduling parameter θ and K_2 depends on the linear scheduling parameter μ with $\mu^\mathrm{T} = [1 \quad \theta^\mathrm{T}]$. Analysis of consistency of the original LPV estimation problem is now transferred to that of an LTI problem in closed-loop. Identification in closed-loop with an LTV, two-degree-of-freedom controller is thoroughly analyzed in [Söderström & Stoica 1989, Chapter 10]. There, it is assumed that the controller coefficients change at certain time instants, which are collected in the set $\mathbb{I}_K = \{k_1, k_2, \ldots, k_{n_K}\}$ with

$$\mathbb{I}_K = \{k \ : \ K_1(\mathrm{q},\mu_k) \neq K_1(\mathrm{q},\mu_{k-1}) \vee K_2(\mathrm{q},\mu_k) \neq K_2(\mathrm{q},\mu_{k-1})\} \ . \tag{3.31}$$

Furthermore, it is assumed that $k_1 > n_a$, since the first few data samples of a measurement record usually have to be discarded when setting up the system of equations, see (3.19). Then, it is shown in [Söderström & Stoica 1989, Section 10.3] that a sufficient condition for a consistent parameter estimate by any prediction error method is that the matrix function

$$\begin{bmatrix} I_{n_y} & \ldots & I_{n_y} & 0 & \ldots & 0 \\ -K_1(\mathrm{q},\mu(k_1)) & \ldots & -K_1(\mathrm{q},\mu(k_{n_K})) & 0 & \ldots & 0 \\ 0 & \ldots & 0 & K_2(\mathrm{q},\mu(k_1)) & \ldots & K_2(\mathrm{q},\mu(k_{n_K})) \end{bmatrix} \tag{3.32}$$

is of full row rank almost everywhere. In this context, full rank almost everywhere means that the matrix function (3.32) has full rank for almost every q^{-1} with q^{-1} regarded as complex variable [Söderström & Stoica 1989, Section 10.3]. Note that this holds assuming that the prediction error method in Line 6 of Algorithm 3 converges to the global minimum. Thus, assuming there is no modeling error, the estimator is consistent.

Since the noise e_k is unknown, it is not possible to combine a scheduling parameter dependent noise polynomial C with the LTI identification methods, because the multiplication of e_k with the scheduling parameter μ_k needs to be done a priori. Only if the ARMAX model is identified iteratively and an update of the noise sequence e_k in each iteration is available, then the noise sequence multiplied by the scheduling parameters can be included in the extended input $\tilde{u}_{k|k-p}$.

Next, the restriction to an LTI C polynomial is removed and the identification of LPV ARMAX models with a scheduling parameter dependent C polynomial is reviewed. The procedure results in the identification algorithm in [Tóth, Heuberger, et al. 2012, Algorithm 3] that is also known as the extended LS method, cf. [Ljung 1999, Section 11.5] for the LTI case.

Algorithm 4 Identification of an LPV ARMAX model, based on [Tóth, Heuberger, et al. 2012]

Require: Measured u, y, p, sampling time T_s, orders n_a, n_b, n_c, regularization $K \succeq 0$
Ensure: M_LPV a DT LPV IO ARMAX model with LPV C polynomial

1: $\mu \leftarrow f_\mu(p, u, y)$ ▷ compute linear scheduling parameter vector, cf. Fig. 2.1b
2: $Y \leftarrow [y_{n_a+1}, y_{n_a+2}, \ldots, y_N]^\mathrm{T}$ ▷ see (3.19)
3: $Z_{yu} \leftarrow \mathrm{regressor}(y, u, \mu, [n_a, n_b], \text{'form'})$ ▷ cf. (3.16), (3.20); 'form' is, e. g., 'SF'
4: $Z_{e,0} \leftarrow 0$
5: $\theta_0 \leftarrow \begin{bmatrix} (Z_{yu}^\mathrm{T} Z_{yu})^{-1} Z_{yu}^\mathrm{T} Y \\ 0 \end{bmatrix}$
6: $\Delta\theta_0 \leftarrow 0$
7: $i \leftarrow 0$
8: **while** loop is not converged **do**
9: $\quad i \leftarrow i + 1$
10: $\quad \hat{Y} \leftarrow [Z_{yu}, Z_{e,i-1}]\theta_{i-1}$
11: $\quad E \leftarrow Y - \hat{Y}$
12: $\quad Z_{e,i} \leftarrow \mathrm{regressor}(e, \mu, n_c, \text{'form'})$ ▷ cf. (3.16), (3.20) with $y \leftarrow e$ and u discarded
13: $\quad Z^\mathrm{T} \leftarrow [Z_{yu}, Z_{e,i}]$
14: $\quad \Delta\theta_i \leftarrow (Z^\mathrm{T} Z + K)^{-1} Z^\mathrm{T} Y - \theta_{i-1}$
15: $\quad \alpha \leftarrow \mathrm{vec}(\theta_{i-1} - \theta_{i-2})^\mathrm{T} \frac{\mathrm{vec}(\Delta\theta_i - \Delta\theta_{i-1})}{\|\mathrm{vec}(\Delta\theta_i - \Delta\theta_{i-1})\|_2^2}$ ▷ alternative: fix α
16: $\quad \theta_i \leftarrow \theta_{i-1} + \alpha\Delta\theta_i$
17: **end while**
18: $[a_i^{(l)}, b_j^{(l)}, c_m^{(l)}] \leftarrow \mathrm{coefficients}(\theta_i, n_y, n_u, n_\mu, [n_a, n_b, n_c], \text{'form'})$ ▷ cf. (3.17), (3.35)
19: $M_\mathrm{LPV} \leftarrow \{a_i^{(l)}, b_j^{(l)}, c_m^{(l)}, T_\mathrm{s}\}$

The predicted output is given by

$$\begin{aligned} \hat{y}_{k|k-1} &= (I - A(\mathrm{q}, \mu_k))y_k + B(\mathrm{q}, \mu_k)u_k - (I - C(\mathrm{q}, \mu_k))e_k \\ &= \begin{bmatrix} \theta_{AB}^\mathrm{T} & \theta_C^\mathrm{T} \end{bmatrix} \begin{bmatrix} z_k \\ z_{e,k} \end{bmatrix} \qquad (3.33) \\ z_{e,k}^\mathrm{T} &= \begin{bmatrix} \mu_{k-1}^\mathrm{T} \otimes e_{k-1}^\mathrm{T} & \cdots & \mu_{k-n_c}^\mathrm{T} \otimes e_{k-n_c}^\mathrm{T} \end{bmatrix} \qquad (3.34) \\ \theta_C^\mathrm{T} &= \begin{bmatrix} c_1^{(0)} & \cdots & c_1^{(n_\mu-1)} & \cdots & c_{n_c}^{(0)} & \cdots & c_{n_c}^{(n_\mu-1)} \end{bmatrix} \qquad (3.35) \end{aligned}$$

with regressor vector z_k as in (3.16) and the system model coefficient matrix θ_{AB} as θ in (3.17) for the ARX case. Note that the C polynomial is assumed in shifted form, i. e. with a specific dynamic scheduling parameter dependence, although this is not necessary and other dynamic or static dependencies are possible as well.

Since e is not known and hence z_e is unknown, too, an iterative procedure is suggested in Algorithm 4[10)] for the identification of LPV ARMAX models, where the noise signal e is estimated in each iteration. Note that Algorithm 4 is a modification of the algorithm in [Tóth, Heuberger, et al. 2012] by employing a gradient descent method. In Lines 14 to 16 of Algorithm 4, the gradient is approximated and a coefficient update is performed based on the

[10)] Note that the notation $\mathrm{vec}(\Delta\theta)$, e. g., in Line 15, means that a vector is constructed by stacking the columns of the matrix $\Delta\theta$ underneath each other.

scalar step length $\alpha \in \mathbb{R}$ that is commonly in the interval $(0, 1]$. The step length can be fixed, e. g., $\alpha = 0.2$ or it can be updated in each iteration. In Line 15, the Barzilai-Borwein method is used to iteratively update the step length [Barzilai & Borwein 1988]. Other computational more demanding line search methods can also be deployed that satisfy, e. g., the Wolfe conditions or the Goldstein conditions [Nocedal & Wright 2006, Chapter 3]. Note that Algorithm 4 and the algorithm in [Tóth, Heuberger, et al. 2012] are the same, if the step length is set to $\alpha = 1$.

3.3.4 LPV OE and BJ Model

Due to the common dynamics in the system model G and the noise model H for the ARX and ARMAX model structures, any structural modeling error of the noise has an effect on the coefficients of both G and H. In order to avoid this disadvantage, the OE and BJ model structures can be chosen that allow completely independent dynamics in G and H.

Only few literature is available on the identification of LPV OE and BJ models. One approach is the LPV refined instrumental variables (LPV-RIV) method that is published in [Laurain et al. 2010] and revisited in [Tóth, Heuberger, et al. 2012]. This method is only applicable to SISO systems. Thus, a new algorithm is introduced, denoted as LPV prediction error (LPV-PE) method, which allows handling MIMO LPV BJ and OE model structures and is one of the main contributions of this thesis, cf. [Schulz et al. 2018]. Since the OE model structure is a simplification of the BJ model structure, the more general BJ structure is considered in the following and at the end, it is pointed out in the algorithm where a modification, respectively simplification, takes place in order to estimate an OE model.

For the identification of an LPV model in the BJ structure the following model structure is assumed

$$F(\mathrm{q}, \mu_k)\breve{y}_k = B(\mathrm{q}, \mu_k)u_k \tag{3.36a}$$
$$D(\mathrm{q})v_k = C(\mathrm{q})e_k \tag{3.36b}$$
$$\hat{y}_k = \breve{y}_k + v_k \ , \tag{3.36c}$$

where $\breve{y}_k$ is the unknown noise free output[11)] of the LPV system model G and v_k the colored noise output of the LTI noise model H.

First, the LPV system part (3.36a) is again reformulated as an LTI model with extended inputs, i. e.

$$\tilde{F}(\mathrm{q})\breve{y}_k = \tilde{b}^{\mathrm{T}}\tilde{u}_{k|k-p} \ , \tag{3.37}$$

where $\tilde{u}_{k|k-p}$ contains this time the unknown noise free outputs $\breve{y}_k$, cf. (3.26). Then, the noise free output and the colored noise are given by

$$\breve{y}_k = (I - \tilde{F}(\mathrm{q}))\breve{y}_k + \tilde{b}^{\mathrm{T}}\tilde{u}_{k|k-p} \tag{3.38a}$$
$$v_k = (I - D(\mathrm{q}))v_k + C(\mathrm{q})e_k \ . \tag{3.38b}$$

Replacing (3.38a) and (3.38b) in (3.36c), the output $\hat{y}_k$ becomes

$$\hat{y}_k = (I - \tilde{F}(\mathrm{q}))\breve{y}_k + \tilde{b}^{\mathrm{T}}\tilde{u}_{k|k-p} + (I - D(\mathrm{q}))v_k + C(\mathrm{q})e_k \ . \tag{3.39}$$

[11)] Due to solving the identification problem iteratively, $\breve{y}_k$ needs to be updated (simulated) in each iteration step, thus the `bj` algorithm from MATLAB's system identification toolbox for LTI systems cannot be applied; the same applies to the identification of LPV OE models.

Using $\breve{y}_k = \hat{y}_k - v_k$, a predicted output

$$\hat{y}_{k|k-1} = (I - \tilde{F}(\mathrm{q}))y_k + \tilde{b}^{\mathrm{T}}\tilde{u}_{k|k-p} + (\tilde{F}(\mathrm{q}) - D(\mathrm{q}))v_k + (C(\mathrm{q}) - I)e_k \ . \tag{3.40}$$

is obtained, which depends on an estimated colored and a white noise signal v and e, respectively. The model coefficients w. r. t. the regressors in (3.40) are collected in the coefficient matrix θ in following order

$$\begin{aligned}\theta^{\mathrm{T}} = \Big[& f_1^{(0)} \ \dots \ f_{n_f}^{(0)} \ \ f_1^{(1)} \ \dots \ f_1^{(n_\mu-1)} \ \dots \ f_{n_f}^{(1)} \ \dots \ f_{n_f}^{(n_\mu-1)} \\ & b_0^{(0)} \ \dots \ b_0^{(n_\mu-1)} \ \dots \ b_{n_b}^{(0)} \ \dots \ b_{n_b}^{(n_\mu-1)} \\ & f_1^{(0)} - d_1^{(0)} \ \dots \ f_{n_d}^{(0)} - d_{n_d}^{(0)} \ \ c_1^{(0)} \ \dots \ c_{n_c}^{(0)} \Big]\end{aligned} \tag{3.41}$$

with $\theta \in \mathbb{R}^{(n_\mu(n_y n_f + n_u(n_b+1)) + n_y(n_c+n_d)) \times n_y}$. Since the predicted model output $\hat{y}_{k|k-1}$ in (3.40) depends on $(\tilde{F}(\mathrm{q}) - D(\mathrm{q}))v_k$, the difference of the coefficients in the polynomials $\tilde{F}$ and D is estimated, see (3.41). To obtain the coefficients of D this difference must be subtracted from the coefficients of $\tilde{F}$, where the coefficients of $\tilde{F}$ are estimated in the first $n_y n_f$ columns of θ^{T}, see (3.41). Note that if the order $n_f < n_d$, the missing coefficients $f_{n_{f+1}}^{(0)}, \dots, f_{n_d}^{(0)}$ in $\tilde{F} - D$ can be set to zero. On the other hand, if $n_f > n_d$, then the coefficients in $\tilde{F} - D$ and $\tilde{F}$ must match from order $n_d + 1$. Since this constraint is not considered in the LS problem, it will generally not be met. Instead, in this case, only the LTI part of F up to order n_d is extracted in $\tilde{F}$ and the remaining part of F enters $\tilde{b}$.

The regressor of the BJ (and also OE) structure in (3.40) are based more on modeled than measured (output) signals compared to the ARX and ARMAX structures. This becomes clear when the extended input $\tilde{u}_{k|k-p}$ is considered. In the ARX and ARMAX model structures, the input $\tilde{u}_{k|k-p}$ depends on (delayed) measured outputs y_{k-i} with $i \in \mathbb{N}^+$. In the OE and BJ structure the delayed noise-free modeled output $\breve{y}_{k-i}$ enters $\tilde{u}_{k|k-p}$. Thus, the information in the output signal enters the estimation more indirectly via the iteratively updated $\breve{y}_k$.

In Algorithm 5 the resulting LPV-PE method for the identification of LPV BJ models is given. The noise free output $\breve{y}$ is simulated in Line 11 of this algorithm, hence for simulation only the coefficients of the system model, i. e. the polynomials B and F, are taken, see (3.36a). Moreover, the orders of all polynomials must be taken into account when creating the regressor matrices so that there is no time shift in the respective lines of the matrices. Analogous to the other algorithms, the LS problem is regularized with a positive semi-definite matrix K.

In comparison, the Lines 15 to 21 of Algorithm 5 differ from the LPV-RIV algorithm in [Laurain et al. 2010]. In Lines 15 to 19 of Algorithm 5, the system and noise model are estimated together, while with the LPV-RIV algorithm both are estimated separately. The separation is possible because filtered signals are used for the system model. Again, a gradient descent method is used in Lines 15 to 19 of Algorithm 5, which is not used in the LPV-RIV algorithm that, respectively, corresponds to a step length of $\alpha = 1$.

The identification of LPV OE models is obtained by neglecting the noise model estimation steps in Algorithm 5. Omitting the dynamic noise model means that Lines 13 to 17 can be skipped in Algorithm 5 and the regressors $Z_{e,i}$ and Z_v vanish, i. e. Line 18 becomes $Z \leftarrow [Z_y, Z_{\breve{y},u}]$.

Next, consistency for the LPV-BJ estimators is discussed. It is shown in [Laurain et al. 2010] that the LPV-RIV method provides under the common conditions consistent estimates, i. e. $Z_{\mathrm{IV}}^{\mathrm{T}} Z$ is invertible, the instrumental variables are uncorrelated with the noise, and the

Algorithm 5 Identification of an LPV BJ model using the LPV-PE method, [Schulz et al. 2018]

Require: Measured u, y, p, sampling time T_s, orders n_f, n_b, n_c, n_d, regularization $K \succeq 0$
Ensure: M_LPV a DT LPV IO BJ model with LTI noise model

1: $\mu \leftarrow f_\mu(p, u, y)$ ▷ compute linear scheduling parameter vector, cf. Fig. 2.1b
2: $Y \leftarrow [y_{n_f+1}, y_{n_f+2}, \ldots, y_N]^\mathrm{T}$ ▷ see (3.19)
3: $Z_y \leftarrow \mathrm{regressor}(y, n_f)$ ▷ i. e. $[\mathrm{q}^{-1}Y, \ldots, \mathrm{q}^{-n_f}Y]$
4: $Z_{y,u} \leftarrow \mathrm{regressor}(y, u, \mu, [n_f, n_b], \text{'form'})$ ▷ cf. (3.16), (3.20); 'form' is, e. g., 'SF'
5: $Z_{e,0} \leftarrow 0$
6: $\theta_0 \leftarrow \begin{bmatrix} (Z_{y,u}^\mathrm{T} Z_{y,u})^{-1} Z_{y,u}^\mathrm{T} Y \\ 0 \end{bmatrix}$
7: $\Delta\theta_0 \leftarrow 0$
8: $i \leftarrow 0$
9: **while** loop is not converged **do**
10: $i \leftarrow i + 1$
11: $\breve{y} \leftarrow \mathrm{simulate}(\theta_{i-1}, u, \mu, [n_f, n_b], \text{'form'})$ ▷ noise free output $\breve{y}_k = G(\mathrm{q}, \mu_k)u_k$
12: $Z_{\breve{y},u} \leftarrow \mathrm{regressor}(\breve{y}, u, \mu, [n_f, n_b], \text{'form'})$ ▷ w/o "LTI-part" of F, i. e. $F - \tilde{F}$
13: $v \leftarrow y - \breve{y}$
14: $Z_v \leftarrow \mathrm{regressor}(v, n_d)$ ▷ i. e. $[\mathrm{q}^{-1}V, \ldots, \mathrm{q}^{-n_d}V]$
15: $\hat{Y} \leftarrow [Z_y, Z_{\breve{y},u}, Z_v, Z_{e,i-1}]\theta_{i-1}$ ▷ see (3.40)
16: $E \leftarrow Y - \hat{Y}$
17: $Z_{e,i} \leftarrow \mathrm{regressor}(e, n_c)$ ▷ i. e. $[\mathrm{q}^{-1}E, \ldots, \mathrm{q}^{-n_c}E]$
18: $Z \leftarrow [Z_y, Z_{\breve{y},u}, Z_v, Z_{e,i}]$
19: $\Delta\theta_i \leftarrow (Z^\mathrm{T}Z + K)^{-1}Z^\mathrm{T}Y - \theta_{i-1}$
20: $\alpha \leftarrow \mathrm{vec}(\theta_{i-1} - \theta_{i-2})^\mathrm{T} \frac{\mathrm{vec}(\Delta\theta_i - \Delta\theta_{i-1})}{\|\mathrm{vec}(\Delta\theta_i - \Delta\theta_{i-1})\|_2^2}$ ▷ alternative: fix α
21: $\theta_i \leftarrow \theta_{i-1} + \alpha\Delta\theta_i$
22: **end while**
23: $[a_i^{(l)}, b_j^{(l)}, c_m^{(0)}, d_n^{(0)}] \leftarrow \mathrm{coefficients}(\theta_i, n_y, n_u, n_\mu, [n_a, n_b, n_c, n_d], \text{'form'})$ ▷ see (3.41)
24: $M_\mathrm{LPV} \leftarrow \{a_i^{(l)}, b_j^{(l)}, c_m^{(0)}, d_n^{(0)}, T_\mathrm{s}\}$

algorithm converges to the global minimum. In contrast, the estimator in Algorithm 5 is not consistent. Assume that the true system with G_0 and H_0 can be represented by the LPV BJ model structure (3.12) and is driven by a true white noise signal e_0. Due to the estimated noises v and e in Line 13 and Line 16, which enter the regressor Z in Line 18, the regressor is correlated with the true noise e_0 as the estimated G and H approaches the true G_0 and H_0, since then $v \to H_0 e_0$ and $e \to e_0$. In addition, the regressor Z_y contains noisy outputs and correlates with the noise e_0. To avoid the correlation, instrumental variables must be introduced. A possible choice for the IV matrix is $Z_\mathrm{IV} = [Z_{\breve{y}}, Z_{\breve{y},u}, Z_{\bar{v}}, Z_{\bar{e}}]$, where, e. g., $\bar{v} = D^\dagger(\mathrm{q})C(\mathrm{q})\bar{e}$ using the estimated coefficients for C and D and $\bar{e}$ is white noise that is uncorrelated with e_0. Then, Z_IV is uncorrelated with the noise (consistency) and similar to Z in Line 18 (high correlation between Z_IV and Z, i. e. low covariance of the estimated coefficients). Of course, consistency only applies if Algorithm 5 converges to the global minimum. However, case studies have shown that the quality of the identification result, both in terms of the model accuracy and the estimated coefficients, is frequently worse with the IV matrix Z_IV compared to the

Table 3.2: Overview of identification algorithms for different LPV model structures and different noise models $H = D^{\dagger}C$

Structure	H	Algorithm
ARX	1	[Bamieh & Giarré 2002] (only for SISO) [Tóth, Heuberger, et al. 2012, Algorithm 1] (extensible for MIMO)
ARMAX	LTI	Algorithm 3
ARMAX	LPV	[Tóth, Heuberger, et al. 2012, Algorithm 3] (extensible for MIMO) Algorithm 4
OE	1	[Laurain et al. 2010] (only for SISO) Algorithm 5
BJ	LTI	[Laurain et al. 2010] (only for SISO) Algorithm 5

biased estimator in Algorithm 5.

In [Schulz et al. 2018], the identification method LPV-RIV from the literature, see [Laurain et al. 2010], is compared with the proposed LPV-PE method, see Algorithm 5. In a numerical SISO example, see [Laurain et al. 2010], the bias with the LPV-RIV method is nearly four times lower than with the LPV-PE method. On noise free validation data, the BFR is however similar for the models identified using the two methods and is slightly better than using the IV method for an LPV ARX structure. If the order of the C and D polynomials is increased, the estimated models get unstable with the LPV-RIV method while this does not emerge with Algorithm 5 [Schulz et al. 2018]. Moreover, in a second numerical MIMO example, see [Verdult & Verhaegen 2002], where the LPV-RIV method cannot be used, the LPV-PE method gives similar good results in the BFR as the LPV ARMAX identification with LTI C polynomial, see Algorithm 3. Note that the data generating system is an LPV SS model that cannot be represented exactly by the IO model structures. In this thesis, instead of numerical examples, the LPV-RIV and the LPV-PE methods are applied to identify models for a real-world example system, see Section 5.2.2.

In conclusion of the identification of DT LPV IO models, a summary of the considered model structures and identification algorithms is given in Table 3.2. Algorithm 1 is the same as in the literature and Algorithm 4 is based on [Tóth, Heuberger, et al. 2012, Algorithm 3] with slight modifications on how to update the estimated coefficients. A new algorithm is introduced in Algorithm 3, where models of simpler structure (assuming an LTI instead of an LPV C polynomial) are identified, however this algorithm allows the use of established existing LTI identification methods [Schulz et al. 2018]. The algorithm in [Laurain et al. 2010] for OE and BJ models is introduced for SISO systems and cannot be extended to the MIMO case. To fill this gap, some steps in this algorithm have been changed, resulting in Algorithm 5, which consequently differs from the algorithm in [Laurain et al. 2010] in Lines 15 to 21. Thus, it is now possible to identify MIMO LPV models in OE and BJ structure with this new algorithm.

3.4 LPV Subspace Identification

After presenting the identification of LPV IO models, the direct identification of DT LPV SS models using a subspace identification (SID) method is briefly reviewed in this section. With the objective of designing controllers based on LPV models in SS representation, one advantage of SID over IO identification is that the identified model is directly obtained in the state-space representation, hence no transformation from the IO to the SS domain is required. Another advantage is that the state-space matrix functions of the identified LPV model can be fully populated allowing a wider set of LPV models in SS representation. In contrast, the SS matrix functions in SF, (extended) AF, and OF are constrained by a certain structure, see Sections 2.4.1 and 2.4.2. The downside of the SID method is a significantly higher computational effort and an unfavorable scaling of the problem with the number of samples and scheduling parameters. Advantageous is again that no iterations are required (as for the ARMAX or BJ structure) and that the model order can be determined based on singular values (automatically or manually).

Subspace identification of LPV SS models is introduced in [Verdult 2002; Wingerden & Verhaegen 2009]. In the following, the model structures are presented and the approach for the problem formulation is briefly reviewed.

Again, an affine scheduling parameter dependency is assumed that can be measured or calculated from measurements. Moreover, the realistic assumption is made that the states cannot be measured and are therefore not explicitly available for identification.

3.4.1 Model Structures

In LPV subspace identification an LPV model in SS representation (2.3) is assumed with affine and static scheduling parameter dependency. In addition, the model structure is extended to have noise inputs. In general, it is assumed that both the state and the output equation are affected by noise, hence the DT SS representation becomes

$$x_{k+1} = A(\mu_k)x_k + B(\mu_k)u_k + w_k \tag{3.42a}$$

$$y_k = C(\mu_k)x_k + D(\mu_k)u_k + e_k \ , \tag{3.42b}$$

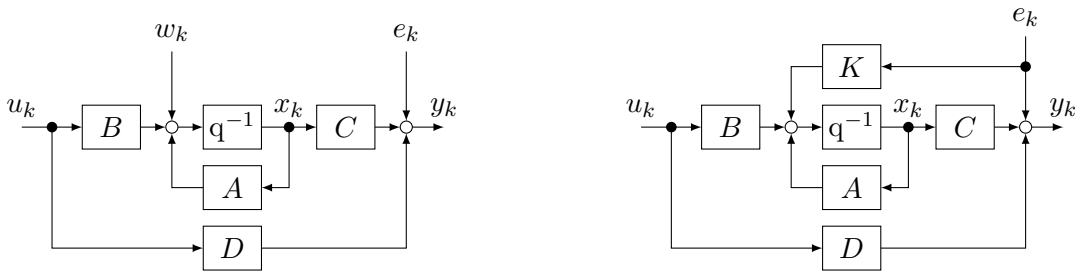
where w_k is denoted as process noise and e_k as measurement noise[12)]. Both noise signals are assumed to be white noise, i. e. uncorrelated random variables with zero mean and time independent finite variance, thus for $l \neq 0$

$$\mathrm{E}\left\{\begin{bmatrix} w_k \\ e_k \end{bmatrix}\begin{bmatrix} w_{k-l} \\ e_{k-l} \end{bmatrix}^{\mathrm{T}}\right\} = 0 \ , \qquad \mathrm{E}\left\{\begin{bmatrix} w_k \\ e_k \end{bmatrix}\right\} = 0 \ , \qquad \mathrm{cov}\left\{\begin{bmatrix} w_k \\ e_k \end{bmatrix}\right\} = \begin{bmatrix} \Sigma_w & 0 \\ 0 & \Sigma_e \end{bmatrix} \succ 0 \ . \tag{3.43}$$

The model structure is depicted in Fig. 3.5a with omitted scheduling parameter dependency for brevity. It can be represented by the general model structure in Fig. 3.2a, hence a separation into (finite) models G and H as in the IO case, see Fig. 3.2b, is in general no longer possible.

An equivalent state-space representation of (3.42) can be obtained that has the structure as shown in Fig. 3.5b, where K is the Kalman gain and e is zero mean white noise that has a different covariance matrix as the noise e in (3.42) [Cox 2018, Section 4.3]. The state of this SS representation is then the minimum variance estimate of the state in (3.42). However, in

[12)] Usually, the measurement noise e is denoted by v. However, to achieve a consistent notation with the IO methods, e is used here.



(a) General noise structure **(b)** Innovation type noise structure

Figure 3.5: Model structures for subspace identification with omitted scheduling parameter dependence for brevity

the LPV case, the Kalman gain has a dynamic and rational dependence on the scheduling parameter [Cox 2018, Lemma 4.1]. For simplification, an observer matrix function with affine scheduling parameter dependence is assumed, which results in a special noise model structure that is no longer equivalent to the general noise structure. The LPV DT SS representation of this innovation type noise model structure is

$$x_{k+1} = A(\mu_k)x_k + B(\mu_k)u_k + K(\mu_k)e_k \tag{3.44a}$$
$$y_k = C(\mu_k)x_k + D(\mu_k)u_k + e_k \ , \tag{3.44b}$$

where K is the observer matrix function and e is zero mean white noise. The structure is depicted in Fig. 3.5b. It is possible to eliminate e_k in the state equation by solving for e_k in the output equation, i. e.

$$e_k = y_k - C(\mu_k)x_k - D(\mu_k)u_k \ , \tag{3.45}$$

and inserting this in the state equation that results in the predictor form

$$x_{k+1} = \bar{A}(\mu_k)x_k + \bar{B}(\mu_k)u_k + K(\mu_k)y_k = \bar{A}(\mu_k)x_k + \tilde{B}(\mu_k)z_k \tag{3.46a}$$
$$y_k = C(\mu_k)x_k + D(\mu_k)u_k + e_k \tag{3.46b}$$

with $\bar{A}(\mu_k) = A(\mu_k) - K(\mu_k)C(\mu_k)$ and $\bar{B}(\mu_k) = B(\mu_k) - K(\mu_k)D(\mu_k)$. In addition, combining the measured in- and outputs in $z_k^{\mathrm{T}} = [u_k^{\mathrm{T}} \quad y_k^{\mathrm{T}}]$ with the input matrix $\tilde{B}(\mu_k) = [\bar{B}(\mu_k) \quad K(\mu_k)]$ simplifies the notation.

Note that there is no restriction on the state-space matrices in above SS representations, hence they can be fully populated that is usually an overparameterization. Moreover, since the state trajectory is not known, identifiability of the SS matrices is not given and they can only be estimated up to similarity transformation. Thus, consistency cannot be evaluated except for particular, e. g., canonical SS representations.

3.4.2 Problem Formulation

Usually, subspace identification (SID) consists of the following four steps [Favoreel et al. 2000; Overschee & Moor 1996]:

1. Select one of the model structures (general or innovation type noise structure, cf. Fig. 3.5). The structure is accompanied by a matrix equation that has in general the form

$$Y = \Phi_X Z_X + \Phi_U Z_U + \Phi_E Z_E \ , \tag{3.47}$$

   i. e. a linear combination of an unknown state matrix Z_X, a matrix Z_U that contains measured inputs (and outputs), and a noise term Z_E. The matrices Φ_X, Φ_U, and Φ_E contain the (unknown) model matrices.

2. Solve the matrix equation for the unknowns. For this purpose, either $\Phi_U Z_U + \Phi_E Z_E$ is eliminated via projections and weighting matrices or the unknown state sequence in Z_X is replaced by known terms and (after neglecting some terms) solved in the sense of least squares. This step usually causes an approximation error.

3. Get an estimate either of the (extended) observability matrix in Φ_X or the state signal (up to a similarity transformation). This is achieved by the singular value decomposition (SVD) of a matrix obtained from the solution of the preceding step. Here, there is another approximation error because only the first n_x singular values from the SVD are used for the estimation.

4. Use this estimate to solve a least squares problem for the model matrices.

In the LPV case, similar matrix equations can be obtained as in the LTI case. However, since the model matrices are scheduling parameter and therefore time-dependent, the derivation of the matrix equation is more involved and the dimensions of the resulting matrices are much larger. In the second and third steps of the above procedure, the second approach is used in LPV SID, i. e. replacing the state sequence in Z_X by known terms and, consequently, estimating the state signal, see [Wingerden & Verhaegen 2009].

In LPV SID, the innovation type noise model structure is commonly assumed with affine scheduling parameter dependence in the state equation and a parameter independent output equation, i. e. the matrix functions $C(\mu)$ and $D(\mu)$ in (3.44b) and (3.46b) are simplified to constant matrices C and D. The assumption of scheduling parameter independent matrices in the output equation is typical and reduces the dimension of the identification problem. Note that the LPV SS representations in SF, OF, and AF also have a scheduling parameter independent output equation, see Section 2.4.1. The formulation of the LPV SID problem for the general noise model structure is given in [Verdult 2002, Sections 3.1–3.3], where also a scheduling parameter independent output equation is assumed.

The procedure of LPV SID can be summarized as follows. By recursively replacing the state equation (3.46a) in the output equation (3.46b), the output at the current time y_k can be determined from a past state x_{k-p} and past input, output, and scheduling signals u_{k-i}, y_{k-i}, and μ_{k-i}, respectively, with $i \in \{1, 2, \ldots, p\}$. Repeating this for several time instants results in a matrix equation in the form of (3.47) (first step). If the system is asymptotically stable and the past time window p is large enough, the influence of the state x_{k-p} and thus the first term in (3.47) can be neglected. Then, the second term $\Phi_U Z_U$ can be seen as predicted output and $Y - \Phi_U Z_U$ as prediction error, which gives the predictor-based subspace identification (PBSID) method its name. The solution of the simplified matrix equation (second step) can then be used to construct the product of the extended observability matrix and state trajectory. With an SVD a state trajectory can then be constructed (third step), which can then be used together with the input, output, and scheduling signal to estimate the model matrices (fourth step).

The detailed procedure and matrices used in LPV SID is given in, e. g., [Verdult 2002, Sections 3.1–3.3]. In this thesis, the method implemented in the PBSID toolbox is used, where the dimension of some matrices in the SID is reduced, see [Wingerden & Verhaegen 2009]. A review of these two approaches is given in Appendix A.3.1, where the main steps and simplifications are summarized.

3.5 Comparison of Input-Output and Subspace Identification

In this section, the two identification approaches LPV IO and LPV SID are compared. Different criteria are considered to work out differences, advantages, disadvantages, and limitations of both methods. The comparison is based on [Schulz et al. 2016] and [Bussa 2016, Section 3.4]. For the IO identification, the algorithms introduced in Section 3.3 are used, while for SID the predictor-based subspace identification (PBSID) toolbox[13)] is used, which is based on the reduced SID problem formulation, see Appendix A.3.2. Note that the comparison is focused on LPV SF ARX and ARMAX structures in the IO domain and on the innovation type noise structure in the SS domain, since these model structures are frequently used and allow some direct comparisons.

3.5.1 Relation of Model Structures

In order to compare the subspace with the IO method, first the model structures are compared. This requires to transfer the model structures used in IO and subspace identification to the respective other system representation.

SID Model Structures in IO Representation

First, the innovation type noise model structure is transformed into the IO domain. The recursion of the state equation for the innovation type noise model structure (3.44a) results in

$$\begin{aligned} x_k &= A(\mu_{k-1})x_{k-1} + B(\mu_{k-1})u_{k-1} + K(\mu_{k-1})e_{k-1} \\ &= A(\mu_{k-1})(A(\mu_{k-2})x_{k-2} + B(\mu_{k-2})u_{k-2} + K(\mu_{k-2})e_{k-2}) \\ &\quad + B(\mu_{k-1})u_{k-1} + K(\mu_{k-1})e_{k-1} \\ &\vdots \\ &= \phi_{k-1|0}x_0 + \sum_{i=1}^{k} \phi_{k-1|k-i+1}B(\mu_{k-i})u_{k-i} + \sum_{i=1}^{k} \phi_{k-1|k-i+1}K(\mu_{k-i})e_{k-i} \end{aligned} \tag{3.48}$$

with the state transition matrix function $\phi_{k-1|k-i+1} = A(\mu_{k-1})A(\mu_{k-2})\dots A(\mu_{k-i+1})$ for $i > 1$ and with $\phi_{k-1|k} = I$. Note that the second term on the right side is the state-reachability matrix function $\mathsf{R}_k \diamond \mu_{k-1}$ multiplied by the past input signal, i. e.

$$\sum_{i=1}^{k} \phi_{k-1|k-i+1}B(\mu_{k-i})u_{k-i} = (\mathsf{R}_k \diamond \mu_{k-1}) \begin{bmatrix} u_{k-1} \\ u_{k-2} \\ \vdots \end{bmatrix} . \tag{3.49}$$

[13)] This free toolbox is available for MATLAB and is provided by the Delft Center of Systems and Control, Delft University of Technology [Wingerden & Verhaegen 2009].

Inserting (3.48) in the scheduling parameter dependent output equation (3.46b) results in

$$y_k = C(\mu_k)\phi_{k-1|0}x_0 + D(\mu_k)u_k + C(\mu_k)\sum_{i=1}^{k}\phi_{k-1|k-i+1}B(\mu_{k-i})u_{k-i} + e_k + C(\mu_k)\sum_{i=1}^{k}\phi_{k-1|k-i+1}K(\mu_{k-i})e_{k-i}\ . \quad (3.50)$$

If the model is asymptotically stable, the transition matrix function $\phi_{k+i|k}$ tends for increasing i to zero. Hence, for a sufficiently large past window p and $i > p$, the matrix $\phi_{k+i|k}$ can be neglected at the cost of an approximation error [Verdult 2002, Lemma 3.5]. Thus, the high-order input-output representation can be approximately truncated to the finite one

$$y_k \approx D(\mu_k)u_k + C(\mu_k)\sum_{i=1}^{p}\phi_{k-1|k-i+1}B(\mu_{k-i})u_{k-i} + e_k + C(\mu_k)\sum_{i=1}^{p}\phi_{k-1|k-i+1}K(\mu_{k-i})e_{k-i}\ . \quad (3.51)$$

It can be seen, that the DT LPV SS model structure used in subspace identification represents for asymptotically stable models a high-order DT LPV IO moving average with exogenous input (MAX) model structure.

Instead of a high-order MAX structure, a high-order ARX model structure can be derived, too. If the predictor form of the innovation type noise structure (3.46) is used, the truncated output equation becomes

$$y_k \approx C(\mu_k)\sum_{i=1}^{p}\bar{\phi}_{k-1|k-i+1}K(\mu_{k-i})y_{k-i} + D(\mu_k)u_k + C(\mu_k)\sum_{i=1}^{p}\bar{\phi}_{k-1|k-i+1}\bar{B}(\mu_{k-i})u_{k-i} + e_k\ . \quad (3.52)$$

Converting the SS model structure taken for SID into an IO model structure and assuming that $x_0 = 0$ results in high-order MAX or ARX model structures, where the model coefficients depend on each other. Asymptotically stable models can be approximated by truncated high-order IO model structures. For the identification, either inputs only (MAX) or inputs and outputs (ARX) can be used. Note that in the MAX case outputs will be used indirectly via the noise estimation. In the PBSID toolbox, the ARX approach is pursued to obtain an intermediate result, see [Wingerden & Verhaegen 2009, (13)] or (A.59).

IO Model Structures in SS Representation

If an IO LPV ARMAX model structure in shifted form is chosen, i. e.

$$\begin{aligned} y_k &= (I - A(\mathrm{q},\mu_k))y_k + B(\mathrm{q},\mu_k)u_k + C(\mathrm{q},\mu_k)e_k \\ &= -\sum_{i=1}^{n_a} a_i\left(\mu_{k-i}\right)y_{k-i} + \sum_{j=0}^{n_b} b_j\left(\mu_{k-j}\right)u_{k-j} + \sum_{m=0}^{n_c} c_m\left(\mu_{k-m}\right)e_k\ , \end{aligned} \quad (3.53)$$

and the polynomial $C(\mathrm{q}, \mu_k)$ is monic (common assumption), the state-space representation has the structure

$$x_{k+1} = A(\mu_k)x_k + B(\mu_k)u_k + K(\mu_k)e_k \tag{3.54a}$$
$$y_k = C(\mu_k)x_k + D(\mu_k)u_k + e_k \ . \tag{3.54b}$$

The matrix functions A, B, C, D, and K, cf. (2.40), have the following static dependency structure (in the case of $n_a = n_b = n_c$)

$$A(\mu) = \begin{bmatrix} -a_1(\mu) & I & 0 & \cdots & 0 \\ -a_2(\mu) & 0 & \ddots & \ddots & \vdots \\ \vdots & \vdots & \ddots & \ddots & \vdots \\ -a_{n_a-1}(\mu) & 0 & \cdots & 0 & I \\ -a_{n_a}(\mu) & 0 & \cdots & \cdots & 0 \end{bmatrix} \tag{3.55a}$$

$$\begin{bmatrix} B(\mu) & K(\mu) \end{bmatrix} = \begin{bmatrix} b_1(\mu) - a_1(\mu)b_0(\mu) & c_1(\mu) - a_1(\mu) \\ \vdots & \vdots \\ b_{n_a}(\mu) - a_{n_a}(\mu)b_0(\mu) & c_{n_a}(\mu) - a_{n_a}(\mu) \end{bmatrix} \tag{3.55b}$$

$$C(\mu) = \begin{bmatrix} I & 0 & \cdots & 0 \end{bmatrix} \ , \qquad D(\mu) = b_0(\mu) \ . \tag{3.55c}$$

Affine dependency is regained, if the feedthrough is scheduling parameter independent, i. e. $b_0(\mu) = b_0^{(0)}$. Consequently, the SS representation of the innovation type noise is obtained with a scheduling parameter independent output equation, thus similar to the model structure in the LPV SID approach. If the ARX structure is considered instead of the ARMAX, the $c_m(\mu_k)$ entries vanish in $K(\mu_k)$ in (3.55b) for $m \in \{1, 2, \ldots, n_a\}$. As a result, the entries in $K(\mu_k)$ depend on and coincide with those in the first n_y columns of $A(\mu_k)$.

3.5.2 Noise Handling

For the IO methods a variety of (dynamic) noise model structures are available in order to represent the influence of the noise on the output as well as possible and thus to separate the noise from the system model. It is assumed that the noise can only affect the system at one point either in the middle (ARX and ARMAX) or at the output (OE) with system-independent dynamics (BJ).

In contrast, it is assumed in SID that there are two points where noise interferes: as an additional input to the system model and at the system output. The two noise sources are assumed as either independent (general noise structure) or dependent (innovation type noise structure) on each other. In both cases, the process noise (multiplied by a matrix) has an additive effect on the states or outputs. An additional dynamic is not provided to allow possible correlation in the noise signal acting on the system.

In Section 3.5.1, it is shown that on the one hand the IO ARMAX model structure is related to the SS innovation type noise structure and that on the other hand the innovation type noise structure can be represented by the ARX or MAX structure of high order. However, there are limitations in both cases. Either the state-space matrices obtained from IO models have a special structure or the coefficients in the polynomials of the ARX, respectively MAX, structure are not independent of each other. Thus, in terms of handling white noise, the identification of high-order IO ARX or MAX models is superior to the SID framework and the SID is superior to the identification of ARX, respectively ARMAX, models of the same order.

3.5.3 Structure of Model Coefficient or Matrix Functions

First, the IO representations of the SS innovation type noise model structure are discussed. The coefficients of the polynomials in the truncated, high-order ARX and MAX model structures are related to each other. Both the coefficients of the individual polynomials and those of both polynomials depend on each other. For clarification, the coefficient functions of the AR polynomial for q^i and q^{i+1} in (3.52) are $C(\mu_k)\bar{\phi}_{k-1|k-i+1}K(\mu_{k-i})$ and $C(\mu_k)\bar{\phi}_{k-1|k-i+2}K(\mu_{k-i-1})$, thus $C(\mu_k)\bar{\phi}_{k-1|k-i+1}[K(\mu_{k-i}) \quad \bar{A}(\mu_{k-i})K(\mu_{k-i-1})]$. They only differ in the additional multiplication with the state matrix function $\bar{A}$ and the different time instants of the scheduling parameters in the matrix function K. Similar applies to the polynomials in the MAX model structure. In addition, (3.51) and (3.52) can be simplified to

$$y_k \approx \begin{bmatrix} D(\mu_k) & I \end{bmatrix} \begin{bmatrix} u_k \\ e_k \end{bmatrix} + C(\mu_k) \sum_{i=1}^{p} \phi_{k-1|k-i+1} \begin{bmatrix} B(\mu_{k-i}) & K(\mu_{k-i}) \end{bmatrix} \begin{bmatrix} u_{k-i} \\ e_{k-i} \end{bmatrix} \tag{3.56}$$

$$y_k \approx \begin{bmatrix} D(\mu_k) & I \end{bmatrix} \begin{bmatrix} u_k \\ e_k \end{bmatrix} + C(\mu_k) \sum_{i=1}^{p} \bar{\phi}_{k-1|k-i+1} \begin{bmatrix} \bar{B}(\mu_{k-i}) & K(\mu_{k-i}) \end{bmatrix} \begin{bmatrix} u_{k-i} \\ y_{k-i} \end{bmatrix} , \tag{3.57}$$

respectively, which shows that the coefficients of both polynomials in the ARX and MAX structures cannot be selected completely independently of each other. Thus, the coefficient functions of the IO model structures exhibit strong and complicated dependencies. However, these dependencies allow the generation of low-order SS models from the high-order IO models and are furthermore useful from the point of view of the assumed model structure with the noise sources, where the transition of inputs and noise through the system is similar and given by the state matrix.

Next, the SS representation of the IO ARMAX model structure is discussed. The SS representation obtained for the SF with matrices $A(\mu_k)$, $B(\mu_k)$, $C(\mu_k)$, and $D(\mu_k)$ in (3.55) is in the companion observability canonical form [Tóth 2010, Section 4.1.3]. Thus, any LPV SS model that has this companion observability form with static scheduling parameter dependence and scheduling parameter independent output matrix C can be exactly represented by the IO model structure in SF. Of course, these systems can also have fully populated state-space matrices by applying scheduling parameter independent state transformations.

It is shown in [Tóth 2010, Section 4.1], that structural state observable SISO LPV systems can be transformed in the companion observability form. Thus, structural state observability is a necessary condition for equivalence of both identification methods. Unfortunately, even if the true system has a static scheduling parameter dependence, the transformed canonical state-space representation does not necessarily have to preserve the static dependency due to scheduling parameter dependent transformation matrices. However, LPV SS models in companion observability form with dynamic scheduling parameter dependence can be also identified with IO methods, where only the dynamic scheduling parameter dependence in the IO domain must be modified. Note that in the MIMO case the restriction to the specific structures in the LPV state-space matrices is even stricter, because even in the LTI MIMO case not all systems can be represented by a canonical representation.

Therefore, for LPV MIMO and SISO systems the model structure used in the subspace method provides a more general LPV SS representation, which allows lower modeling errors in the SS domain.

3.5.4 Order of the Identified Model

It has been already mentioned that the IO models obtained from the subspace approach are of high order. Due to the assumption that the transition matrix $\bar{\phi}_{k+i|k} \approx 0$ for $i > p$, the order of the polynomials in the IO domain are given by the value for the appropriately chosen past window p.

In the following, the orders of SS representations of models identified in the IO domain are analyzed in more detail. The coefficient functions of the polynomials in the IO domain are, for example, $a_i(\mu) : \mathbb{P}_\mu \to \mathbb{R}^{n_y \times n_y}$ and $b_j(\mu) : \mathbb{P}_\mu \to \mathbb{R}^{n_y \times n_u}$. Therefore, the order of the SS models, determined by the number of states, depends on the chosen form (SF, (e)AF, OF), the orders of the polynomials (n_a, n_b), and the number of inputs and outputs (n_u, n_y). Hence, the orders n_x of the SS representations of the identified system models G using IO methods are for the

- shifted form (SF): $\max(n_a, n_b)n_y$, see (2.40),
- augmented form (AF) and extended AF: $n_a n_y + (n_b - 1)n_u$, see (2.42), and $n_a n_y + n_b n_u$, see (2.46),
- observability form (OF): $n_a n_y$, see (2.44).

By varying the orders n_a and n_b, the order of the SS representation of the IO model varies only as multiples of the number of outputs n_y or inputs n_u. Consequently, in the MIMO case, the LPV SS models transformed from the IO domain are often of high order and thus usually non-minimal. For example, the SS representation of the shifted form for a system with a single input and four outputs has orders four, eight, and twelve with increasing values of n_a from one, two, and three, respectively, and $n_b \leq n_a$. Then, there is no possibility of getting models of any order in between, e. g., one, two, three, or five, six, seven without performing model order reduction, where some of the model accuracy can be lost in the reduction step. Therefore, IO methods do not provide the flexibility to identify SS models of any order $n_x \in \mathbb{N}^+$ for LPV MIMO systems without performing a model reduction step. By contrast, any model order can be chosen in subspace identification, irrespective of the number of inputs or outputs. The order n_x is only limited by $n_x \leq p n_y$. In the case of LPV MIMO systems, illustrative examples have shown that for the same model accuracy LPV SID methods provide models with fewer states than LPV IO identification methods, cf. [Schulz et al. 2016].

3.5.5 Algorithms

The basic steps of the algorithms for the identification of LPV models using IO methods and the SID method are given in Table 3.3 [Schulz et al. 2016; Tóth, Heuberger, et al. 2012; Wingerden & Verhaegen 2009]. However, modifications and extensions are possible in both algorithms, which turn both into more complicated algorithms.

In the IO case, the identification of more complicated model structures than the ARX requires the repetition of steps two and three. Due to the iterative procedure using gradient like methods of finding the optimum, the estimator depends on the initial guess and is prone to local minima. Moreover, the regressors are more involved, since simulated or estimated signals also enter the regressors.

In contrast to the IO methods, where one LS problem must be solved (several times in case of iteration), the solution of two LS problems and an SVD are required for the SID approach. In the first LS problem in step three, the high-order ARX model in (3.52) is estimated (using

Table 3.3: Comparison of the basic steps for the estimation of LPV IO models (at the example of the ARX structure) and LPV SS models

	LPV IO identification method		LPV SID method
1.	A priori settings e. g., n_a, n_b, and μ independent polynomials	1.	A priori settings e. g., p, f, and μ independent matrices
2.	Regressor matrices using u, y and μ, see (3.16)	2.	Regressor matrices using u, y and μ, see (A.49)
		3.	Solve an LS problem for Φ, see (A.53)
		4.	Construct matrix observability times reachability, see (A.61)
		5.	Estimate a state sequence $\hat{X}_k$ using SVD, see (A.56)
3.	Solve an LS problem for the model coefficients, see (3.18)	6.	Solve an LS problem for the model matrices, see (A.57)

the dual problem formulation) with scheduling parameter independent C and D matrices, see [Wingerden & Verhaegen 2009, (13) and (19)]. The remaining steps then consist of the approximation of a lower order SS model based on this ARX model of order p. In addition, there are different ways how to perform the steps two to four. The two-block method, see [Verdult 2002, Section 3.3] or Appendix A.3.1, or even more sophisticated methods can be used [Verdult 2002, Chapter 3]. Instead, there are also reduced methods and two different algorithms for the subspace identification are introduced in [Wingerden & Verhaegen 2009], cf. Appendix A.3.2. One is using an LS approach and the other one is using the kernel method in combination with the LS approach.

Obviously, the subspace identification algorithm contains three more steps as compared to IO methods. However, all IO model structures except the ARX structure require iterations in the third step while SID is iteration free. The regressors with input, output and scheduling data created in the second step of the algorithms are of different size. For the comparison of both identification methods, mainly the third step is considered in more detail in the remainder of this section.

3.5.6 Scheduling Parameter Handling

In the following, the scheduling parameter handling in the third step of both algorithms in Table 3.3 is discussed. As mentioned, in both algorithms an IO model is identified of structure ARX (or even ARMAX for IO methods). The ARX polynomials are of order n_a and n_b (and possibly n_c) in the IO and of order p in the subspace approach. Hence, the regressor matrices contain inputs and outputs from a different number of time shifts. However, the regressors contain in a structural different way the scheduling parameters and the matrices with the corresponding model coefficients differ considerably in both approaches. The reason for this is the more complicated dependency on the scheduling parameters and SS matrices in the IO model structures obtained from the SID approach, which do not occur in the IO approach.

Thus, if both in the IO and in the SS domain the scheduling parameter dependency is supposed to be simple, it is recommended to use IO identification methods, because with these the scheduling parameter dependency in both domains can be influenced easily.

It is obvious that the coefficients of the high-order ARX model structure in (3.52) contains products of scheduling parameter dependent matrices. To obtain a separation between the scheduling parameters and model coefficients requires much larger matrices, see [Wingerden & Verhaegen 2009], [Verdult 2002, Section 3.3], or Appendix A.3.1. For the clarification of the scheduling parameter handling in the two approaches, the following example is considered.

Example 3.1. The identification of a SISO LPV model with only one affine scheduling parameter is considered, i. e. $\mu_k^{\mathrm{T}} = [1 \quad \theta_k]$, and without feedthrough, i. e. $D = 0$. Step three in the subspace identification algorithm for a past window of $p = 2$, i. e. two past time instants to estimate the output, leads to solving

$$\hat{y}_{k|k-1} = C\acute{\bar{\mathcal{R}}}_2 \grave{\mathcal{Z}}_{k-2|k-1} = \Phi_{\mathrm{SID}} \begin{bmatrix} \mu_{k-1} \otimes I_2 & 0 \\ 0 & \mu_{k-1} \otimes \mu_{k-2} \otimes I_2 \end{bmatrix} \begin{bmatrix} u_{k-1} \\ y_{k-1} \\ u_{k-2} \\ y_{k-2} \end{bmatrix} , \tag{3.58}$$

where $\hat{y}_{k|k-1}$ is the predicted output. This is equivalent to the reduced problem formulation, see [Wingerden & Verhaegen 2009, (13)] or (A.59). The matrix of unknowns is

$$\Phi_{\mathrm{SID}} = \begin{bmatrix} C\tilde{B}_0 & C\tilde{B}_1 & C\bar{A}_0\tilde{B}_0 & C\bar{A}_0\tilde{B}_1 & C\bar{A}_1\tilde{B}_0 & C\bar{A}_1\tilde{B}_1 \end{bmatrix} \in \mathbb{R}^{1\times 12} \tag{3.59}$$

with $\bar{A}_l = A_l - K_l C$ and $\tilde{B}_l = [B_l \quad K_l]$. Hence, Φ_{SID} is equivalent to Φ_U in Appendix A.3.1, cf. (A.51d). Note that as usual the C matrix is assumed as scheduling parameter independent.

In contrast, LPV IO identification of an ARX model in shifted form with $n_a = n_b = 2$ leads to solving the equation

$$\hat{y}_{k|k-1} = \Phi_{\mathrm{IO}}^{\mathrm{SF}} \begin{bmatrix} \mu_{k-1} \otimes I_2 & 0 \\ 0 & \mu_{k-2} \otimes I_2 \end{bmatrix} \begin{bmatrix} u_{k-1} \\ y_{k-1} \\ u_{k-2} \\ y_{k-2} \end{bmatrix} . \tag{3.60}$$

The matrix of unknowns is

$$\Phi_{\mathrm{IO}}^{\mathrm{SF}} = \begin{bmatrix} b_1^{(0)} & b_1^{(1)} & a_1^{(0)} & a_1^{(1)} & b_2^{(0)} & b_2^{(1)} & a_2^{(0)} & a_2^{(1)} \end{bmatrix} \in \mathbb{R}^{1\times 8} , \tag{3.61}$$

which is a permutation of the coefficient matrix θ in (3.17).

In (3.58) and (3.60), the data matrices on the right hand side with the past input-output data are exactly the same due to the special choice $p = n_a = n_b = 2$. The only differences in both formulations are, firstly, the contents of the unknown matrices $\Phi_{\mathrm{IO}}^{\mathrm{SF}}$ and Φ_{SID} and, secondly, the matrices with scheduling parameters represented by the middle matrices on the right hand side. As already discussed, the difference occurs due to the multiplications of scheduling parameter dependent matrices in the subspace approach that do not occur in the IO approach, compare (3.59) and (3.61).

As seen from the blocks along the diagonal of the matrix with scheduling parameters in (3.58), the subspace algorithm uses the Kronecker product of the past scheduling parameters

from the time instant of the corresponding input-output data up to $k-1$. In contrast, the IO algorithms depend on the scheduling parameters of only one time instant for each pair of input-output data that is valid even for the other forms, i. e. (extended) AF and OF, where in these cases the time instant remains the same if there is no feedthrough. It is obvious that increasing the past window size, hence allowing more multiplications of scheduling parameter dependent matrices, increases the matrix dimensions in (3.58) significantly. Extending (3.58) and (3.60) for the estimation of the output using three instants of past input-output data, i. e. $p = n_a = n_b = 3$, the last block along the diagonal in the matrix containing the scheduling parameters contains $\mu_{k-1} \otimes \mu_{k-2} \otimes \mu_{k-3} \otimes I_2$ in the subspace and only $\mu_{k-3} \otimes I_2$ in the IO SF approach. This illustrates how both methods use different ways of handling scheduling parameters in their algorithm.

An interpretation of the difference is as following. In the subspace approach, the information in the input-output data is propagated from their time instant through the system up to the current time instant. This requires repeated multiplications with the scheduling parameter dependent state matrix A, respectively $\bar{A}$. On the other hand, in the IO approach it is assumed that all information from past data is captured in the output data for the corresponding time instant. Thus, a recursive propagation is not necessary and the computation of the output at the current time instant requires the forward propagation by only one time instant of past input-output data. The two different propagation approaches make both sense, however the IO approach provides a simpler computation of the estimated output.

3.5.7 Number of Unknowns

In the preceding section and in the derivation of both identification methods, it emerged clearly that the number of unknowns differs significantly in both approaches. This is now examined in more detail.

Due to the multiplication of scheduling parameter dependent matrices, the number of unknowns inside Φ_{SID} increases exponentially with increasing the past window size p[14)]. On the contrary, the number of unknowns involved in IO identification $n^{\text{IO}}_{\text{unknowns}}$ increases linearly with n_a and n_b (and n_c). Thus, the number of unknowns in step three of both algorithms in Table 3.3 can be computed as following

$$n^{\text{SID}}_{\text{unknowns}} = n_y(n_y + n_u) \sum_{i=1}^{p} n_\mu^i \tag{3.62}$$

$$n^{\text{IO}}_{\text{unknowns}} = n_y n_\mu \left(n_y(n_a + n_c) + n_u(n_b + 1)\right) \ . \tag{3.63}$$

Here, the reduced LPV SID method that is implemented in the PBSID toolbox with future window length $f = 1$ is considered. The computation of $n^{\text{SID}}_{\text{unknowns}}$ for Φ_{SID} follows from the output dimension and the number of rows in the regressor matrix, cf. (A.60). In the IO case, the identification of an LPV model in SF[15)] and ARMAX structure is assumed and $n^{\text{IO}}_{\text{unknowns}}$ for $\Phi^{\text{SF}}_{\text{IO}}$ follows from (3.17) and (3.35). In the case of the simpler ARX structure, n_c is zero. For illustrative purposes, the number of unknowns for a different number of inputs, outputs and orders of the polynomials is listed in Table 3.4 for the subspace and in Table 3.5 for the SF IO identification method. It is obvious that the past window has an immense effect on the

[14)] If a future window is used, the number of unknowns also increases exponentially with the future window size f, cf. (A.58).

[15)] This also holds for the (extended) AF but not for the OF, where n_b is zero.

Table 3.4: Number of unknowns in SID for $n_\mu = 4$ and $n_u = n_y$

	$n_y = 1$	$n_y = 2$	$n_y = 3$	$n_y = 4$	$n_y = 5$
$p = 2$	40	160	360	640	1000
$p = 3$	168	672	1512	2688	4200
$p = 4$	680	2720	6120	10880	17000
$p = 5$	2728	10912	24552	43648	68200

Table 3.5: Number of unknowns in IO SF for $n_\mu = 4$, $n_a = n_b = n_c$ and $n_u = n_y$

	$n_y = 1$	$n_y = 2$	$n_y = 3$	$n_y = 4$	$n_y = 5$
$n_a = 2$	28	112	252	448	700
$n_a = 3$	40	160	360	640	1000
$n_a = 4$	52	208	468	832	1300
$n_a = 5$	64	256	576	1024	1600

number of unknowns. Even for the simple SISO case (first column of Tables 3.4 and 3.5), the number of unknowns required for estimation of high-order models using the SID method is higher as compared to the IO identification method. Note that the order n_x in SID is limited by the past window size p by the inequality $n_x \leq pn_y$ and for the SF state-space representation it is given by $n_x = n_y \max(n_a, n_b)$. In the MIMO case, the dimensionality problem is more severe with the subspace algorithm running into the "out of memory" problem because of the need to store huge data matrices. In addition, as inversion of huge matrices requires more time, subspace is much slower as compared to IO identification. The same is the case for low-order LPV models with many affine scheduling parameters. Therefore, the subspace method will be impractical for identification of high-complexity models, i. e. a high model order and/or number of scheduling parameters. The IO approach is more resource efficient in the sense of both memory and computation.

The curse of dimensionality in Φ_{SID} can be circumvented by using the kernel method, see [Wingerden & Verhaegen 2009] or Appendix A.3.2. Then, the dimensionality of the LS problem is limited to the number of data samples N. Nevertheless, the number of data points for the identification of LPV models can increase rapidly as the system needs to be sufficiently excited, including sufficient coverage of the scheduling parameter set.

So far, in the SID case, a scheduling parameter independent C matrix is assumed so that the dimension only increases with n_μ^i and not with n_μ^{2i}, see Remark A.1. However, the number of unknowns is significantly reduced when the MAX representation in (3.51) is used instead of the ARX representation in (3.52) [Cox & Tóth 2016]. This is shown by the following equations for the number of unknowns for the ARX and MAX structures

$$n_{\text{unknowns}}^{\text{SID,ARX}} = n_y \left(n_y \sum_{i=1}^{p} n_\mu^{2i} + n_u \sum_{j=0}^{p} n_\mu^{2j+1} \right) \tag{3.64}$$

$$n_{\text{unknowns}}^{\text{SID,MAX}} = n_y \left(n_y \sum_{i=1}^{p} n_\mu^{i+1} + n_u \sum_{j=0}^{p} n_\mu^{j+1} \right) . \tag{3.65}$$

Reason for this is the state matrix, which enters the transition matrix. In the MAX case, the state matrix is $A(\mu_k)$, while in the ARX case it is $\bar{A}(\mu_k) = A(\mu_k) - K(\mu_k)C(\mu_k)$, hence a

product of scheduling parameter dependent matrices. Thus, using the MAX representation for the identification of SS models allows scheduling parameter dependent output matrices without a significant additional increase of the identification problem complexity, since both $n_{\text{unknowns}}^{\text{SID}}$ and $n_{\text{unknowns}}^{\text{SID,MAX}}$ increase with n_μ^i.

Besides, it should be noted that in this section the term number of unknowns is used. In the case of IO methods, this number corresponds to the model coefficients. In the SID approach, however, this is not the case. As already mentioned, the model coefficients of the ARX or MAX model structure are interdependent. However, this dependency is ignored when solving the LS problem in step three of the SID algorithm. This is on the one hand a simplification of the problem and on the other hand allows considerably more degrees of freedom than the SS model structure actually provides. In order to compensate these degrees of freedom, further approximations are made in the subsequent steps of the SID algorithm, which are reviewed in the following. According to Table 3.3, in step four a matrix is constructed based on the LS solution that represents the observability times reachability matrix (this time considering the matrix structure), where only the LTI part of the observability matrix is considered, see [Wingerden & Verhaegen 2009, (6) and (14)] or (A.42) and (A.61). This constructed matrix also represents the LTI part of the observability matrix multiplied by the state sequence and it has $n_y p$ rows and N columns, with N the number of data samples. The SVD of this matrix must be computed, from which an estimate of the states results, however this time the structure of the observability matrix is ignored by the SVD. In the last step another LS problem is solved, where this time the number of unknowns correspond to the number of model coefficients, i. e. $n_\mu n_x(n_x + n_u) + n_y(n_x + n_u)$. Note that the number of unknowns in the third step of the SID algorithm is in (3.59) $n_{\text{unknowns}}^{\text{SID}} = 12$, while the number of SS model coefficients is 6 for $n_x = 1$ or 15 for $n_x = 2 \leq n_y p$. However, the number of SS model coefficients is usually smaller than $n_{\text{unknowns}}^{\text{SID}}$ due to the exponential dependence on the past window p in $n_{\text{unknowns}}^{\text{SID}}$.

In summary, there are significantly more unknowns in the LPV SID approach that need to be determined than in the LPV IO identification approach – and up to n_μ times more model coefficients than in LTI case. Due to their canonical SS representation, the model structures in the IO methods even represent in the SISO case a minimal parameterization. However, not all LPV models possess a static scheduling parameter dependence in canonical SS representations, which is why fully populated SS matrices are advantageous, as they are assumed in the SID approach, even if fully populated SS matrices are usually an overparameterization.

3.5.8 Regularization

As already shown, the number of unknowns in the third step of the algorithms is significantly higher in the subspace than in the IO approach. Thus, it is likely that the LS problem in the SID case is ill-conditioned, see also the discussion on Page 43. Therefore, in the subspace case regularization is necessary to obtain a meaningful estimate. Note that regularization makes the estimated coefficients biased. In the case of IO identification, regularization is not mandatory due to the smaller number of unknowns, but it can also be included to the LS problem for increasing the similarity of both algorithms and for achieving the same regularization effect. However, in the IO approach the model coefficients are obtained directly from the LS problem in the third step of the algorithm, see Table 3.3. On the other hand, a high-order model is obtained as an intermediate result from the LS problem in the third step of SID and model reduction is performed afterwards in step five. Hence, even if choosing the same regularization method for the LS problem in step three, the SID has additional regularization in step five

(SVD) and possibly also in step six (another LS problem).

3.5.9 Summary

In summary, it can be stated that both identification methods, IO and subspace, can be compared well under certain criteria. It is discussed that the IO methods scale much better with the order and number of affine scheduling parameters of LPV models, whereas less structural restrictions on the system model G are made by the SID method.

The structure of noise handling is similar in the IO ARMAX and SS innovation type noise model structures. The innovation type noise structure in SID can be represented by high-order IO model structures, but with interdependent model coefficients. LPV IO models, on the other hand, have a canonical LPV SS representation. These represent a minimal parameterization, but do not always exist with a static scheduling parameter dependence. Furthermore, SS representations of MIMO IO models are often non-minimal in the model order, which is why a reduction is useful. The SS model structure assumed in the SID method is more flexible at the same state order. However, this entails the formulation of huge matrices. Only with modern approaches the calculation becomes manageable, whereby however approximation errors arise that are caused by, e. g., regularization and neglecting matrix structures. The computational effort is still considerably higher than with the IO methods, which however often require iterative computations.

It appears that LPV SID is the better of the two methods for the identification of models with low order and low number of affine scheduling parameters. IO identification is able to perform better for high-order and more complicated systems, at the same time being resource efficient from the perspective of memory and computation.

3.6 Nonlinear Optimization

In the preceding sections, the identification of LPV models in the IO and SS domain is discussed. They are obtained based on the solution of LS problems, i. e. the identification problem is formulated such that it is linear in the unknowns. A different approach is to minimize the output error, i. e. the difference between the simulated and the measured output. Consequently, linearity in the parameters is lost and a nonlinear, non-convex optimization problem is obtained that is prone to local minima, hence its solution depends strongly on the initial value. With input-output and subspace identification, such an initial value is provided and the model accuracy of these models can be improved by nonlinear optimization. However, note that the computational load increases using nonlinear optimization.

In [Gunes et al. 2018] tensor networks are used to refine LPV SS models. The proposed algorithm circumvents the dimensionality problem of the convex method in subspace identification, cf. [Verdult & Verhaegen 2002, Chapter 3], [Wingerden & Verhaegen 2009], and Appendix A.3.1. However, the non-convex optimization problem is solved iteratively using alternating LS, where each core of the tensor network is updated by solving an LS problem while fixing the remaining (unknown) cores of the tensor network.

Other nonlinear optimization methods for LPV models are given in, e. g., [Verdult & Verhaegen 2002, Chapter 5] and [Cox 2018, Chapter 7]. The approaches are based on [Lee & Poolla 1999] and [Wills & Ninness 2011], respectively. An LPV model in SS representation is assumed and the coefficients of this SS model are optimized using a gradient-based or an expectation maximization approach, i. e. minimizing the output error based on its gradient or minimizing

the variance of the coefficients while preserving the minimum prediction error property. For the gradient-based approach the innovation type noise model structure is assumed, while for the expectation maximization approach the general noise model structure is taken, see Fig. 3.5 for the two model structures. Since the SS model is overparameterized and state transformation does not affect the input-output behavior of the model, the output error does not change on a manifold that is determined by state transformations. Thus, the gradient is based on a rank deficient Jacobian matrix, which is why a projected gradient is employed, where the directions are projected out in which the output error does not change. In contrast, the expectation maximization approach requires an estimate of the state trajectory that can be obtained via, e. g., Kalman filtering, cf. [Cox 2018, Section 7.3]. As pointed out in [Cox 2018, Section 7.4], the gradient-based method requires an asymptotically stable model in predictor form while this is not the case for the expectation maximization method where the state reconstruction via Kalman filter is stable even for unstable models.

The methods will not be presented further, since they are applied in this work exactly as presented in the literature, see [Lee & Poolla 1999; Wills & Ninness 2011], [Verdult & Verhaegen 2002, Chapter 5], and [Cox 2018, Chapter 7]. This serves to put the methods presented here in relation, e. g., to which accuracies can be achieved with a nonlinear optimization at which calculation time.

4 Model Order Reduction of LPV Systems

In this chapter, the model order reduction of LPV SS models is treated, where the model order is given by the number of states. This is motivated by the high model order of MIMO LPV SS models obtained with IO identification methods. Consequently, the purpose of model order reduction in this thesis is to reduce the complexity of the model in terms of the number of states while ensuring a high model accuracy. Then, the problem is to find for a full-order model G a low-order model $\hat{G}$ with a small approximation error in the input-output behavior. There are many different measures for the approximation error and a common one is

$$\|G - \hat{G}\|_{\mathrm{i},\ell_2} \ , \tag{4.1}$$

i. e. the induced ℓ_2 system norm of the difference. Model order reduction is motivated by, for instance, system identification, control design and implementation complexity. For example, classical designed model-based optimal controllers in the induced ℓ_2 system norm are at least of the generalized plant order, which results from the sum of the model order and the orders of the weights. Hence, it can be useful to perform order reduction to both, identified models and designed controllers. In [Antoulas 2008], model order reduction of LTI models is extensively discussed.

The complexity of LPV systems is not only given by the number of states, but also by the dependence on the scheduling parameters and their number. The reduction in relation to the number of scheduling parameters is treated in [Kwiatkowski & Werner 2008; Rizvi et al. 2018, 2016; Siraj et al. 2012], where a (kernel-based) principal component analysis or an autoencoder neural network is used. However, the reduction of scheduling parameters is not the subject of this chapter. In [Tóth, Abbas, et al. 2012] an approach is presented with which the dynamic dependency can be reduced to a static one, hence the complexity of the scheduling parameter dependence is reduced. On the other hand, this may require the introduction of further states so that the reduction of states is of additional interest for LPV systems. The combined reduction of states and scheduling parameter dependence is the topic in [Farhood & Dullerud 2007; Siraj et al. 2012].

In the remainder of this chapter, the general model order reduction procedure is discussed in Section 4.1 as an entry and the literature on LPV model order reduction is reviewed. Next, the balanced realization and truncation is recapitulated in Section 4.2, where the generalized Gramians are employed for order reduction, hence an optimization problem with LMI constraints must be solved. In Section 4.3, a different approach is presented based on the Hankel matrix, also known as the Ho-Kalman reduction method. It is first presented for the LTI case and then extended to the LPV case, where similar matrices occur as in LPV SID. Due to the dimensionality problem in the LPV case, computational simplifications are introduced for the Ho-Kalman method from which an algorithm is derived in Section 4.3.2 that can be combined with, e. g., LPV IO identification methods. The resulting two-step procedure is compared with state-of-the-art in LPV subspace identification using an academic benchmark example and it yields promising results.

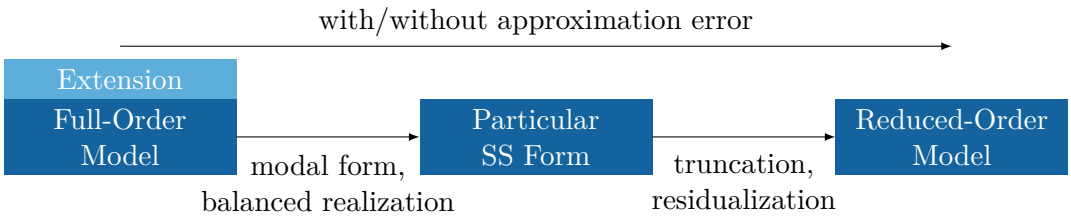


Figure 4.1: Overview of particular model order reduction approaches

4.1 General Procedure in Model Order Reduction

An overview of typical steps in model order reduction regarding the number of states is shown in Fig. 4.1. First, the full-order model is converted into a particular state-space form and then this form is used for the reduction step. There are different methods for both steps, the model transformation step and how to proceed with the transformed model. Some methods combine both steps in order to avoid the explicit computation of the (full-order) particular SS form. Furthermore, other methods cannot be divided into the two model order reduction steps. For example, in [Wu 1996; Zhang & Shi 2008], an optimization-based approach is investigated for LPV systems, where the approximation error in (4.1) is minimized, i. e. the induced ℓ_2 system norm, see Definition 2.7, of the difference between the full-order and the reduced-order model.

In the first step of model order reduction of SS models, a state transformation is usually performed. The resulting SS form must be such that unimportant parts of the full-order model, e. g., for the IO behavior are easily recognizable and can be removed. Typically, the modal form or balanced realization is chosen for this. In the modal form, the state matrix A of the SS model is in block diagonal form and non-relevant modes can be removed. However, as discussed in [Theis 2018, Section 3.1], such a decoupled structure is generally not available for LPV systems. Therefore, the balanced realization is chosen, which allows in a simple way to choose the states that can be removed without getting a large approximation error. Another advantage of model order reduction based on a balanced realization is that stability is maintained and limits for the approximation error can be specified a priori [Skogestad & Postlethwaite 2007, Chapter 11], [Wood et al. 1996]. The techniques to obtain the transformation matrices for a balanced realization can be exact or approximate. An overview of different techniques (mainly for LTI systems) is given in [Antoulas & Sorensen 2001; Gugercin & Antoulas 2000]. Exact methods for calculating the transformation matrices are based on the (generalized) observability and controllability Gramians[1)], both resulting from Lyapunov equations, followed by an SVD. The same approaches are used for LTV systems [Sandberg & Rantzer 2004]. Approximate methods are based on Krylov subspaces, which are (in a special case) the subspaces spanned by the observability and reachability matrices [Antoulas & Sorensen 2001]. Approximate methods are numerically more efficient and can be applied to systems of high order with $n_x \gg 10^3$, since projections onto the subspaces are calculated iteratively and both steps of model order reduction are combined into one using fat/tall transformation matrices. Unfortunately, guarantees for preserving stability and for error bounds are lost. The (approximate) transformation matrices can also be seen as oblique projections, which are discussed for LTI systems intensively in

[1)] Note that uncontrollable and unobservable states provide singular transformation matrices for a balanced realization, where it is of great interest to remove these states. For this reason, there are algorithms that avoid an explicit calculation of the transformation matrices [Safonov et al. 1990].

[Villemagne & Skelton 1987]. Recently, the idea of oblique projection is transferred to the CT LPV case in [Theis et al. 2018] and the full block S-procedure is used in [Heeren & Werner 2020] for the grid-free calculation of generalized Gramians of CT LPV systems in linear fractional representation.

In the second step, the order reduction itself is performed, where model order reduction can be done with or without an approximation error. As already mentioned, SS models can have identical IO behavior even though they may have different orders, cf. [Tóth 2010, Definition 3.29]. Thus, model order reduction without approximation error implies the reduction to a minimal realization, see Definition 2.10, which has the minimal number of states while preserving IO equivalence. In contrast, approximate model order reduction techniques induce an error between the full- and the reduced-order model so that it is even possible to achieve a lower order than the minimal order. In this thesis, the approximation error is not evaluated as in (4.1), but by (the decrease in) the performance criteria from model validation, i. e. the (change in) BFR and VAF, see (3.7) and (3.8). The reason for this is that in this thesis the full-order model results from system identification and its accuracy is evaluated by these two measures, too.

There are different techniques to remove states or dynamics from the full-order model. These techniques cause different properties of the reduced-order model and the most common are truncation, residualization (also known as singular perturbation), and optimal Hankel norm approximation. They all have in common that they assume a partitioning of the SS model as follows

$$G(\rho) \;:\; \left[\begin{array}{c} \mathrm{q}x_1 \\ \hline \mathrm{q}x_2 \\ \hline y \end{array}\right] = \left[\begin{array}{cc|c} A_{11}(\rho) & A_{12}(\rho) & B_1(\rho) \\ A_{21}(\rho) & A_{22}(\rho) & B_2(\rho) \\ \hline C_1(\rho) & C_2(\rho) & D(\rho) \end{array}\right] \left[\begin{array}{c} x_1 \\ \hline x_2 \\ \hline u \end{array}\right], \tag{4.2}$$

here at the example of an LPV SS model. In truncation, the states x_2 and the associated system dynamics with these states are discarded, hence the reduced-order model becomes

$$\hat{G}(\rho) \;:\; \left[\begin{array}{c} \mathrm{q}x_1 \\ \hline y \end{array}\right] = \left[\begin{array}{c|c} A_{11}(\rho) & B_1(\rho) \\ \hline C_1(\rho) & D(\rho) \end{array}\right] \left[\begin{array}{c} x_1 \\ \hline u \end{array}\right] \tag{4.3}$$

whereas in residualization, the states x_2 are assumed constant, i. e. in DT $x_2(k+1) = x_2(k)$, and subsequently x_2 is eliminated from the state-space equations. This, however, increases complexity in the scheduling parameter dependence and requires that $I - A_{22}(\rho)$ is invertible, which follows directly from the LTI case, see [Skogestad & Postlethwaite 2007, Section 11.2]. The steady-state gain of the full-order model is preserved with residualization, whereas truncation maintains the feedthrough accuracy [Skogestad & Postlethwaite 2007, Section 11.4]. As a result, a reduced-order model obtained with the truncation method can achieve a low BFR on a validation data set but still a high VAF, because steady-state deviations are not taken into account in the VAF. Since residualization preserves accuracy at low frequencies and truncation at high frequencies, there exist frequency weighted reduction approaches in order to preserve accuracy at a user specified frequency range [Abbas & Werner 2011]. Consequently, it makes sense to choose the method in the reduction step depending on the intended use of the reduced-order model.

Procedures for the optimal Hankel approximation, which is also a truncation method, are provided for the LTI case in [Glover 1984; Safonov et al. 1990] and for the LPV case in [Wood et al. 1996]. Unfortunately, as the authors in [Wood et al. 1996] emphasize, in the LPV case only the manipulations are similar to those in the LTI case, however the interpretation of the

procedure being optimal is not clear. In addition, the complexity of the scheduling parameter dependence introduced by this method "may exceed any gains obtained by reducing the order" [Wood et al. 1996]. Furthermore, in the LTI model order reduction comparison study in [Gugercin & Antoulas 2000], the optimal Hankel norm approximation results in the worst $\mathcal{H}_2$ norm compared to balanced truncation and residualization. Consequently, the truncation method, illustrated in (4.2) and (4.3), is used in this thesis for DT LPV SS model order reduction, because the functional dependence on the scheduling parameters is preserved. In addition, the truncation method is commonly used in the literature for LPV systems [Abbas & Werner 2011; Beck 2006; Jaimoukha et al. 2005; Schulz et al. 2017; Theis et al. 2018; Wood et al. 1996].

In addition to the two model order reduction steps, the full-order model can be extended. For example, the procedure in Fig. 4.1 is only applicable to asymptotically stable models. Thus, unstable models must first be converted into a coprime factorization (extension into two stable models) and subsequently the above given model order reduction procedure can be performed. In [Wood et al. 1996, Section 7.5] it is shown, how a coprime factorization and model order reduction of unstable, affine scheduling parameter dependent LPV models can be obtained. The same approach is used for unstable LPV models in linear fractional representation in [Beck 2006]. Another extension is to employ (dynamic, proper, and stable) weights W_u and W_y at the inputs and outputs, respectively. This allows to influence which frequency range is to be preserved in the reduction. Then, the objective is to find a lower order model such that $\|W_y(G - \hat{G})W_u\|$ is minimized instead of (4.1). In the LPV case, this is discussed in [Wood 1995, Section 7.4], where fully populated generalized Gramians for the augmented, weighted full-order model are allowed. However, in this case it cannot be guaranteed that asymptotic stability is maintained for the reduced-order model. Only if either the input or the output weight is equal to identity, asymptotic stability will remain. In [Abbas & Werner 2011] structured generalized Gramians are assumed, i. e. they are block diagonal and partitioned in the same way as the augmented, weighted full-order model. Then, asymptotic stability is maintained for the reduced-order model, even if input and output weights are used simultaneously.

4.2 Balanced Realization and Truncation of LPV Systems

In this section, the first step of model order reduction is considered. More precisely, the transformation into a balanced realization and the properties of this particular SS form are discussed. The balanced realization is favored for model order reduction using truncation or residualization, since then the approximation error between the full- and reduced-order model in the induced ℓ_2 system norm becomes small, i. e. a small value of (4.1).

A state-space model in balanced realization has (generalized) controllability and observability Gramians, see Theorems 2.2 and 2.3, which are diagonal, equal, and therefore simply denoted as (generalized) Gramian. Hence, due to the equality, each state is as controllable as observable and the entries on the diagonal of the (generalized) Gramian are denoted as Hankel singular values, which represent a measure of the joint controllability and observability. Since the diagonal entries of the (generalized) Gramian are in descending order, the states of such a SS model in balanced realization are accordingly ordered by their contribution to the input-output behavior of the model. Discarding the states x_2 implies, removing the states with the lowest effect on the input-output behavior of the model, which is desirable for model order reduction. [Skogestad & Postlethwaite 2007, Section 11.3]

Algorithm 6 Transformation Matrix for Balanced Realization, [Laub et al. 1987]

Require: Generalized Gramians X_o and X_c of an LPV system
Ensure: Transformation matrix T_bal for a balanced realization
1: $L_\mathrm{o} \leftarrow \mathrm{chol}(X_\mathrm{o}, \text{'lower'})$
2: $L_\mathrm{c} \leftarrow \mathrm{chol}(X_\mathrm{c}, \text{'lower'})$
3: $[U, \Sigma, V] \leftarrow \mathrm{svd}(L_\mathrm{o}^\mathrm{T} L_\mathrm{c})$
4: $T_\mathrm{bal} \leftarrow L_\mathrm{c} V \Sigma^{-1/2}$
5: $T_\mathrm{bal}^{-1} \leftarrow \Sigma^{-1/2} U^\mathrm{T} L_\mathrm{o}^\mathrm{T}$

Due to the beneficial properties of a balanced realization for model order reduction, a state transformation matrix is required for the LPV SS model. There are different ways to obtain the transformation matrix. They are usually based on the generalized observability and controllability Gramians or on the Hankel operator. In [Laub et al. 1987], the square root algorithm is presented for LTI systems to obtain the transformation matrix T_bal (and its inverse) such that $x = T_\mathrm{bal} x_\mathrm{bal}$ and it is given in Algorithm 6. This algorithm requires the generalized Gramians X_o and X_c. In the LPV case, they need to satisfy the conditions in Theorems 2.2 and 2.3 and they can be scheduling parameter dependent. Then, the above given square root algorithm for LTI systems is the same as in the LPV case. However, decompositions (Cholesky and singular value decomposition) of scheduling parameter dependent matrices are then necessary to obtain the scheduling parameter dependent transformation matrix. Moreover, as shown in Section 2.4.2, scheduling parameter dependent state transformations may cause the transformed model to have a dynamic dependency on the scheduling parameter. For practical reasons[2)] it is demanded in this thesis, that static scheduling parameter dependency is preserved by LPV model order reduction techniques. The dynamic dependency vanishes, if in the DT case the forward time shifted dependence on the scheduling parameter in the transformation matrix $T_\mathrm{bal}(\rho_{k+1})$ is eliminated due to the multiplication with the remaining matrices, i. e. if the two equations

$$T_\mathrm{bal}(\rho_{k+1}) A(\rho_k) T_\mathrm{bal}^{-1}(\rho_k) = A_\mathrm{bal}(\rho_k) \tag{4.4a}$$
$$T_\mathrm{bal}(\rho_{k+1}) B(\rho_k) = B_\mathrm{bal}(\rho_k) \tag{4.4b}$$

hold such that the transformed matrices $A_\mathrm{bal}(\rho_k)$ and $B_\mathrm{bal}(\rho_k)$ have a static dependence. Unfortunately, these conditions are only satisfied for specific combinations of transformation and state-space matrices. Thus, in the CT case, a second scheduling parameter state transformation can be used to eliminate the dynamic dependence. However, this results in a more involved functional dependence on the scheduling parameters [Wood 1995, Section 7.3]. At the cost of conservatism, the requirement to retain the static scheduling parameter dependency is automatically satisfied, if only scheduling parameter independent transformation matrices are allowed, as they are used in [Wood et al. 1996].

As mentioned in Section 2.3.3, the generalized controllability and observability Gramians are not unique in the LPV case due to the inequalities in Theorems 2.2 and 2.3. The square roots of the eigenvalues of $X_\mathrm{o} X_\mathrm{c}$ are comparable with the Hankel singular values, hence they

[2)] LPV models with dynamic scheduling parameter dependency require commonly more scheduling parameters in control design (static dependency is required, hence the dynamic dependencies are added to the scheduling parameter vector), thus there is more conservatism, and it can prevent real-time implementations, if unknown derivatives or predictions of the scheduling parameter are required.

characterize the input-output mapping of a given LPV system [Wood et al. 1996]. Thus, for model order reduction it is desired to find solutions to the convex inequalities in Theorems 2.2 and 2.3 where the objective is to reduce the smallest n_{x_2} eigenvalues in $X_\mathrm{o}X_\mathrm{c}$, belonging to the part of the system that will be discarded. Mathematically, this means

$$\min_{X_\mathrm{o},X_\mathrm{c}} \operatorname{trace}(X_\mathrm{o}X_\mathrm{c}) \ , \tag{4.5}$$

which is a non-convex cost function. In [Wood et al. 1996], an iterative procedure for solving this optimization problem is proposed, based on the alternating search for optimal generalized controllability and observability Gramians X_c and X_o, respectively. This approach is adapted in this thesis for the DT case, i. e. solving the LMIs (2.26) and (2.28) for scheduling parameter independent generalized Gramians X_o and X_c with objective (4.5) in an alternating way. The solution of the optimization problem for X_o and X_c are then used in Algorithm 6 to obtain the constant transformation matrix T_bal and then perform an order reduction via truncation. This approach is the current state-of-the-art for order reduction of LPV systems using scheduling parameter independent transformation matrices. In this thesis, it is called LMI or Gramian based approach and is used for comparison with the newly developed Ho-Kalman based order reduction method, since both approaches are structurally similar.

4.3 Ho-Kalman Reduction

In this section, a balanced model order truncation technique is presented that does not require the generalized observability and controllability Gramians. This approach belongs to the approximate methods in the first step of model order reduction, see Fig. 4.1. This is because, considering the LTI case, matrices are used with which the Gramians can be approximated. However, the dimensions of the matrices are usually larger than the model order so that the computational effort increases. On the other hand, there is no optimization problem to solve. In addition, the full order model does not have to be asymptotically stable, so that the technique is directly applicable to unstable models [Tóth, Abbas, et al. 2012].

The idea is to construct the matrix representation of the Hankel operator, also denoted as Hankel matrix, and obtain from this huge matrix the order reduction transformation matrix or directly the reduced-order state-space matrices. The technique was introduced for LTI systems in [Ho & Kalman 1966] using the sequence of Markov parameters. For LPV systems it was first presented in [Abbas et al. 2010; Tóth, Abbas, et al. 2012] based on the extended observability and reachability matrices proposed in [Wingerden & Verhaegen 2009], cf. Appendix A.3. First, the technique is explained in the LTI case, see [Zhou et al. 1996, Section 8.7] and [De Schutter 2000]. Then, the procedure is extended to the LPV case.

4.3.1 Reduction of DT LTI SS Models

Assume a discrete-time LTI system in SS representation

$$x(k+1) = Ax(k) + Bu(k) \tag{4.6a}$$
$$y(k) = Cx(k) + Du(k) \tag{4.6b}$$

with $x(-\infty) = 0$. Then, the controllability operator Ψ_c is equal to the state-reachability matrix R_∞ and maps the input signal $u \in \ell_2(-\infty, 0)$ onto the state $x(0) = x_0$, see Remark 2.2.

Analogously, the observability operator Ψ_o is equal to the state-observability matrix O_∞ and maps an initial state $x(0) = x_0$ onto the output signal $y \in \ell_2[0, \infty)$, if the input is set to zero, i. e. $u(k) = 0$ for $k \geq 0$, see Remark 2.1. Consequently, the Hankel operator $\Psi_\mathrm{o}\Psi_\mathrm{c}$ that maps the past input $u(k) \in \ell_2(-\infty, 0)$ on the system future output $y(k) \in \ell_2[0, \infty)$ has the matrix representation

$$\mathsf{H}_{\infty,\infty} = \mathsf{O}_\infty \mathsf{R}_\infty = \begin{bmatrix} C \\ CA \\ CA^2 \\ \vdots \end{bmatrix} \begin{bmatrix} B & AB & A^2B & \dots \end{bmatrix} = \begin{bmatrix} G_1 & G_2 & G_3 & \dots \\ G_2 & G_3 & G_4 & \dots \\ G_3 & G_4 & G_5 & \dots \\ \vdots & \vdots & \vdots & \ddots \end{bmatrix} , \tag{4.7}$$

see [Zhou et al. 1996, Chapter 8]. Accordingly, the future outputs can be calculated using the past inputs and the Hankel matrix, thus

$$\begin{bmatrix} y_k \\ y_{k+1} \\ \vdots \end{bmatrix} = \mathsf{H}_{\infty,\infty} \begin{bmatrix} u_{k-1} \\ u_{k-2} \\ \vdots \end{bmatrix} . \tag{4.8}$$

The entries of the Hankel matrix $\mathsf{H}_{\infty,\infty}$ are the Markov parameters

$$G_0 = D , \quad G_k = CA^{k-1}B \quad \text{for } k \in \mathbb{N}^+ , \tag{4.9}$$

where the sequence of Markov parameters $\{G_k\}_{k=0}^{\infty}$ corresponds to the impulse response. For asymptotically stable systems, the impulse response converges to zero and the Hankel matrix $\mathsf{H}_{\infty,\infty}$ has only a finite number of approximately nonzero elements, thus

$$\mathsf{H}_{\infty,\infty} \approx \begin{bmatrix} \mathsf{H}_{n,n} & 0 \\ 0 & 0 \end{bmatrix} \quad \text{with} \quad \mathsf{H}_{n,n} \approx \begin{bmatrix} G_1 & G_2 & \dots & G_{n-1} & G_n \\ G_2 & G_3 & \dots & G_n & 0 \\ G_3 & G_4 & \dots & 0 & 0 \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ G_n & 0 & \dots & 0 & 0 \end{bmatrix} . \tag{4.10}$$

Similar to this result, a condition for the minimal system order of an LTI SS model is given in [Silverman 1971, Theorem 11] by evaluating the rank of the Hankel matrix. The authors state that although the dimensions of the Hankel matrix increase, the rank is limited, cf. (4.10), and that this maximal rank of the Hankel matrix is the minimal system order of the LTI system. Thus, the minimal order can be determined by a finite sequence of the impulse response or finite state-observability and -reachability matrices.

For a given Hankel matrix $\mathsf{H}_{i,j}$ with sufficiently large i and j, the SVD can be computed and from this the state-space matrices of the reduced-order model can be obtained, hence

$$\mathsf{H}_{i,j} = \mathsf{O}_i \mathsf{R}_j = U\Sigma V^\mathrm{T} \approx \hat{U}\hat{\Sigma}\hat{V}^\mathrm{T} = \hat{\mathsf{O}}_i \hat{\mathsf{R}}_j = \hat{\mathsf{H}}_{i,j} , \qquad \bar{\mathsf{H}}_{i,j} = \mathsf{O}_i A \mathsf{R}_j , \tag{4.11}$$

where $\hat{U}$, $\hat{\Sigma}$, and $\hat{V}$ are reduced matrices with only the first $n_{\hat{x}} < n_x$ columns (and rows) of the SVD matrices U, Σ, and V. Then, the matrices $\hat{B}$ and $\hat{C}$ are the first column or row block in $\hat{\mathsf{R}}_j$ or $\hat{\mathsf{O}}_i$, respectively, see (4.7). The state-space matrices of reduced order are completed by $\hat{A} = \hat{\mathsf{O}}_i^\dagger \bar{\mathsf{H}}_{i,j} \hat{\mathsf{R}}_j^\dagger$ with the shifted Hankel matrix $\bar{\mathsf{H}}_{i,j}$ in (4.11). The shifted Hankel matrix $\bar{\mathsf{H}}_{i,j}$ can also be obtained from $\mathsf{H}_{i,j+1}$ by canceling the first n_u columns in $\mathsf{H}_{i,j+1}$, respectively R_{j+1}.

Hence, the remaining part is $\mathsf{O}_i A \mathsf{R}_j$, respectively $A\mathsf{R}_j$, cf. (4.7). This model order reduction procedure can be summarized in an algorithm, cf. [De Schutter 2000]. Note that also the impulse response (Markov parameters) of a DT LTI model can be used to construct the Hankel matrices $\mathsf{H}_{i,j}$ and $\bar{\mathsf{H}}_{i,j}$ and perform the model order reduction, cf. (4.10). If the full-order model has a feedthrough D, then $\hat{D} = D$ in case of a SS representation and $\hat{D} = G_0$ in case of an impulse response of the full-order model. The number of states in the reduced-order model $\hat{G}$ can be either provided a priori or the singular values of the Hankel matrix $\mathsf{H}_{i,j}$ can be analyzed for a (significant) decay in order to achieve a good trade-off between state reduction and approximation error.

Note that the Ho-Kalman reduction algorithm is related to balanced truncation and it can be shown that the reduced-order SS model using the Ho-Kalman algorithm with SVD is balanced, see [De Schutter 2000]. The transformation matrices for a balanced realization are obtained from the Cholesky decompositions of the observability and controllability Gramians, i. e. $X_\mathrm{o} = L_\mathrm{o} L_\mathrm{o}^\mathrm{T}$ and $X_\mathrm{c} = L_\mathrm{c} L_\mathrm{c}^\mathrm{T}$. By definition, the Gramians and the Hankel matrix are given by the observability and controllability operators, i. e. $X_\mathrm{o} = \Psi_\mathrm{o}^\mathrm{T} \Psi_\mathrm{o}$, $X_\mathrm{c} = \Psi_\mathrm{c} \Psi_\mathrm{c}^\mathrm{T}$, and $\mathsf{H}_{\infty,\infty} = \mathsf{O}_\infty \mathsf{R}_\infty = \Psi_\mathrm{o} \Psi_\mathrm{c}$, respectively. Thus, the singular values of the Hankel matrix $\mathsf{H}_{\infty,\infty}$ are the same as those of $L_\mathrm{o}^\mathrm{T} L_\mathrm{c}$. If only a finite Hankel matrix is considered, the Hankel singular values, respectively the Gramians, are approximated. For asymptotically stable systems, where a finite number of approximately nonzero elements in $\mathsf{H}_{\infty,\infty}$ exists, see (4.10), the approximation error of the Hankel singular values is negligible for sufficiently large values i and j in $\mathsf{H}_{i,j}$.

Unfortunately, it is unknown which finite values for i and j are required to keep the approximation error small. The assumption can be erroneous to choose i and j equal to the model order n_x. This assumption is based on the state observability (reachability) test that states: a system is state observable (reachable), if the matrix O_{n_x} (R_{n_x}) has full column (row) rank [Skogestad & Postlethwaite 2007, Section 4.2]. Indeed, it can be shown with the Cayley-Hamilton Theorem, see for example [Zhou et al. 1996, Theorem 2.5], that the matrices O_i and R_j do not change their rank for i or j greater than n_x. However, despite the rank remaining the same, an increase in i and j can still affect the singular values of the Hankel matrix (product of O_i and R_j) and, accordingly, the unitary matrices U and V from the SVD. In [Kung 1978] it is also pointed out that it is often preferable to choose higher i and j for a more accurate reduced-order model, if the Hankel matrix is constructed from an estimated model. The influence of i and j on the Ho-Kalman algorithm is shown in the following case study.

Example 4.1. A family of 420 stable, random DT LTI models is considered. They are of order $n_x = 6$ and partitioned in two subsystems G_1 and G_2, i. e. $G = G_1 G_2$. The first subsystem G_1 is of fourth order and has random poles and zeros with the maximum value of the pole absolute values being less than 0.8, i. e. all poles of G_1 are within the circle with center at the origin and radius of 0.8. The second subsystem G_2 has an order of two and a complex pole pair, where a grid on a set of radii and angles is utilized to determine the pole locations. The purpose of the partition is to investigate a relationship between the pole location and the required values for i and j.

The order of these LTI models is reduced to four, where truncation based on the square root algorithm (using Gramians) and on the Ho-Kalman algorithm (using Hankel matrix) is performed. For simplicity, i is chosen equal to j. The value of i is increased until the ratio of the approximation error in the induced ℓ_2 system norm of both methods is greater

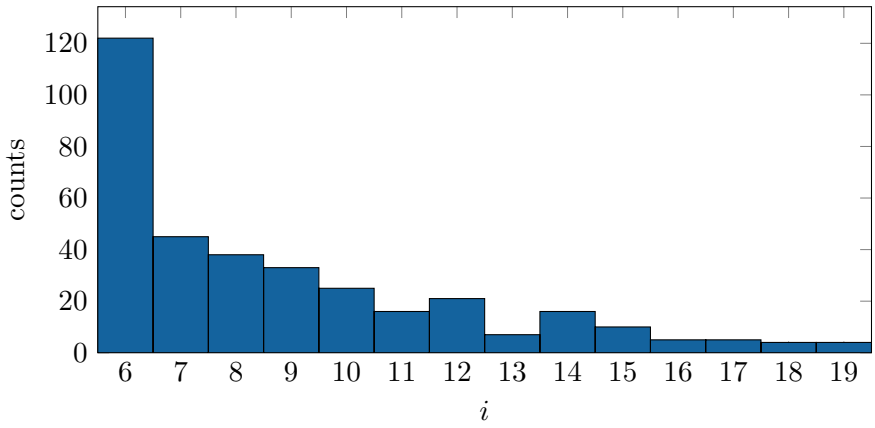


Figure 4.2: Histogram of i in Hankel matrix $\mathsf{H}_{i,i}$

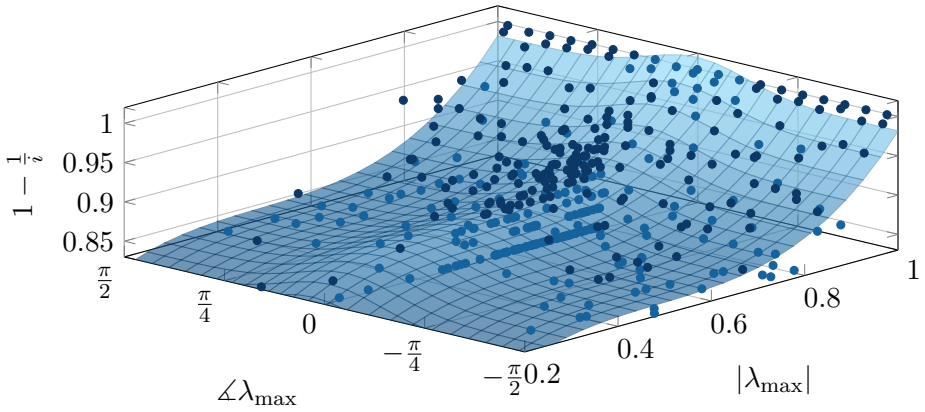


Figure 4.3: Dependence of i in Hankel matrix $\mathsf{H}_{i,i}$ on pole location with $\lambda_{\max}$ the pole of G with maximal distance to the origin

than 99 %, i. e.

$$\frac{\|G-\hat{G}_X\|_{\mathrm{i},\ell_2}}{\|G-\hat{G}_H\|_{\mathrm{i},\ell_2}} > 0.99\ , \tag{4.12}$$

where the reduced-order models $\hat{G}_X$ and $\hat{G}_H$ are based on the Gramians and the Hankel matrix, respectively.

First, the statistics of i are analyzed. In Fig. 4.2 the histogram of i is plotted. It can be seen, that for 29 % of the models, the choice $i = n_x$ is valid. However, the value of i can get very large and Fig. 4.2 is clipped at $i = 19$ for a better visualization. For example, 10 % of the models require $i > 45$ and 5 % require even $i > 250$. Consequently, it is analyzed next what may cause these huge values for i. In Fig. 4.3, the value of i is plotted over the absolute value of the maximal eigenvalue/pole of G (spectral radius) and its phase. Note that due to the significant increase in i, a nonlinear transformation is applied, s. t. a value around one represents a very large i and a value around 0.83 represents $i = n_x = 6$. Moreover, a function is fitted to the data points with the trade-off to find both an approximation and an upper bound. As a result, 48 % of the data points are above the

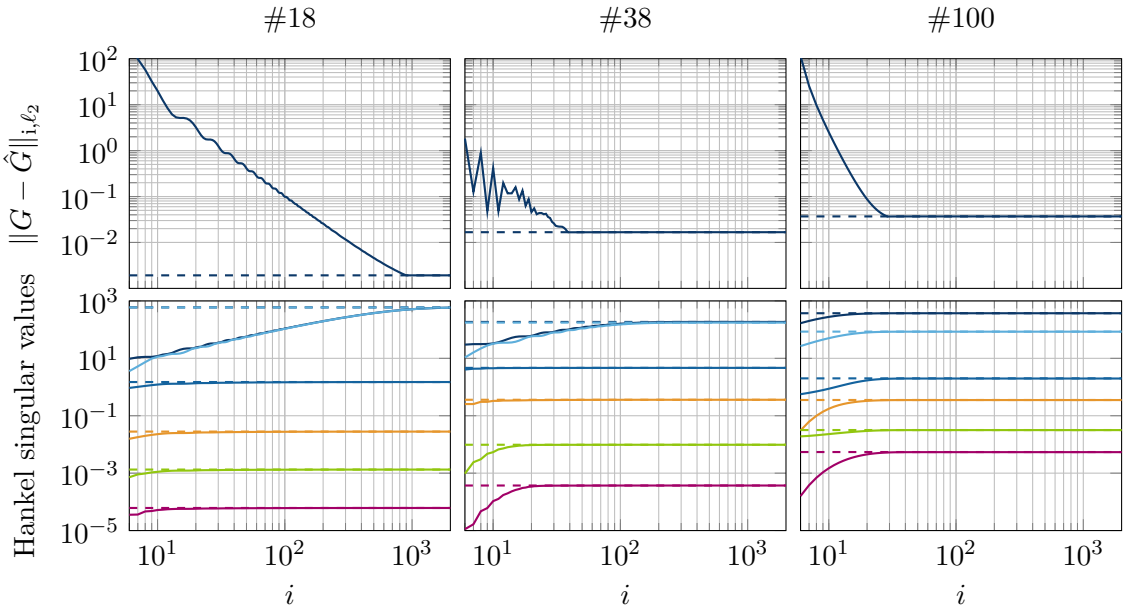


Figure 4.4: Approximation error and Hankel singular values for selected models using Gramians (dashed) or Hankel matrix (solid)

function (marked by the dark blue color). If the distances between data points and surface are considered and the sum of positive distances (data points above the function) is set in relation to the absolute value of the sum of negative distances (data points below the surface), a ratio of 2:3 is achieved. Thus, the function is a good approximate while it tends to give an upper bound. It can be concluded that poles near the boundary of the stability region (unit circle) effect a significant increase of i. From Fig. 4.3, it can be seen that this is slightly more severe for poles $\lambda_{\max}$ near the real axis (low tendency to oscillate). This also corresponds to the argumentation in [Wingerden & Verhaegen 2009]. The i-th power of the models state matrix A, i. e. A^i, converges more slowly to zero, if A has eigenvalues close to the unit circle. Therefore, less approximately nonzero entries occur in the Hankel matrix, see (4.10).

Nevertheless, the choice of i has a significant influence on the approximation accuracy and must be chosen carefully. This is shown in Fig. 4.4, where for different models the (exact) balanced truncation based on Gramians is compared with the (approximate) Ho-Kalman reduction based on the Hankel matrix. The approximation error based on Ho-Kalman reduction converges for all three models with increasing i to that one based on balanced truncation (independent of i). However, it can be seen from model #38 that the approximation error from Ho-Kalman reduction is not monotonically decreasing with increasing i. Note that the largest Hankel singular values have a significant effect on the approximation error. This is evident in models #18 and #38.

4.3.2 Reduction of DT LPV SS Models

Next, the idea of the Ho-Kalman algorithm is extended to DT LPV SS models. The procedure is similar to LPV SID, cf. [Verdult 2002, Section 3.3] or Appendix A.3.1. First, it is assumed that the LPV system has affine scheduling parameter dependence (or it is converted into

such a system), see Section 2.2.4. Then, the linear mapping of the past inputs on the future outputs with the Hankel matrix in (4.8) is extended to the LPV case, resulting in a dynamically scheduling parameter dependent Hankel matrix function $\mathsf{H}_{\infty,\infty} \diamond \theta_k$ with

$$\begin{bmatrix} y_k \\ y_{k+1} \\ \vdots \end{bmatrix} = (\mathsf{H}_{\infty,\infty} \diamond \theta_k) \begin{bmatrix} u_{k-1} \\ u_{k-2} \\ \vdots \end{bmatrix} = (\mathsf{O}_\infty \diamond \theta_k)(\mathsf{R}_\infty \diamond \theta_{k-1}) \begin{bmatrix} u_{k-1} \\ u_{k-2} \\ \vdots \end{bmatrix} . \tag{4.13}$$

Note that this equation is a simplified version of the matrix equation in (3.47) in LPV SID, where the initial state $x_{k-p} = x_{k-\infty}$, future input u_{k+i}, and noise e_{k+i} are assumed zero[3)] with $i \in \mathbb{N}$. Thus, the first and third term on the right hand side in (3.47) vanish. More details can be found in Appendix A.3.1, where the matrix equation for the LPV SID is given in (A.50). Only the third and fourth terms remain in (A.50), and the first and second block entries in Φ_U and Z_U, respectively, see (A.51d) and (A.51e).

Next, the state-space matrices and the scheduling parameters in $\mathsf{H}_{\infty,\infty}$ or O_∞ and R_∞, respectively, can be partitioned in separate matrices. For this purpose, the finite matrix functions are considered and it follows

$$\mathsf{O}_i \diamond \theta_k = \acute{\mathcal{P}}^{\text{diag}}_{k+i|k} \grave{\mathcal{O}}_i \tag{4.14a}$$

$$\mathsf{R}_j \diamond \theta_{k-1} = \acute{\mathcal{R}}_j \grave{\mathcal{P}}^{\text{diag}}_{k-1|k-j} \tag{4.14b}$$

$$\mathsf{H}_{i,j} \diamond \theta_k = \acute{\mathcal{P}}^{\text{diag}}_{k+i|k} \grave{\mathcal{O}}_i \acute{\mathcal{R}}_j \grave{\mathcal{P}}^{\text{diag}}_{k-1|k-j} . \tag{4.14c}$$

In contrast to LPV SID, where all parameter dependencies are extracted to the right side, here the parameter dependencies are extracted to the left and right side, cf. [Verdult 2002, Section 3.3] or Appendix A.3.1. This reduces the dimension of the scheduling parameter independent matrix. The different matrices in the above equations are presented next. The j-step extended reachability matrix

$$\acute{\mathcal{R}}_j = \begin{bmatrix} \acute{\mathcal{M}}_1 & \acute{\mathcal{M}}_2 & \dots & \acute{\mathcal{M}}_j \end{bmatrix} \in \mathbb{R}^{n_x \times \left(n_u \sum_{l=1}^{j} n_\mu^l\right)} \tag{4.15}$$

is constructed from the recursive auxiliary matrices

$$\acute{\mathcal{M}}_1 = \begin{bmatrix} B_0 & B_1 & \dots & B_{n_\theta} \end{bmatrix} , \qquad \acute{\mathcal{M}}_l = \begin{bmatrix} A_0 \acute{\mathcal{M}}_{l-1} & A_1 \acute{\mathcal{M}}_{l-1} & \dots & A_{n_\theta} \acute{\mathcal{M}}_{l-1} \end{bmatrix} , \tag{4.16}$$

which are based on the SS matrices of the full-order DT LPV system with affine scheduling parameter dependence. Note that the auxiliary matrices $\acute{\mathcal{M}}_l$ and the extended reachability matrix $\acute{\mathcal{R}}_j$ are also used in LPV SID, see [Wingerden & Verhaegen 2009, Definition 2] or (A.41) and (A.48). The i-step extended observability matrix $\grave{\mathcal{O}}_i$ is given in a similar way by

$$\grave{\mathcal{N}}_1 = \begin{bmatrix} C_0 \\ C_1 \\ \vdots \\ C_{n_\theta} \end{bmatrix} , \qquad \grave{\mathcal{N}}_l = \begin{bmatrix} \grave{\mathcal{N}}_{l-1} A_0 \\ \grave{\mathcal{N}}_{l-1} A_1 \\ \vdots \\ \grave{\mathcal{N}}_{l-1} A_{n_\theta} \end{bmatrix} , \qquad \grave{\mathcal{O}}_i = \begin{bmatrix} \grave{\mathcal{N}}_1 \\ \grave{\mathcal{N}}_2 \\ \vdots \\ \grave{\mathcal{N}}_i \end{bmatrix} \in \mathbb{R}^{\left(n_y \sum_{l=1}^{i} n_\mu^l\right) \times n_x} \tag{4.17}$$

with auxiliary matrices $\grave{\mathcal{N}}_l$ and the SS matrices of the full-order DT LPV SS model. The extended observability matrix $\grave{\mathcal{O}}_i$ differs from that one used in LPV SID, since here the

[3)] In LPV SID the same is assumed, except for the future input, which is non-zero.

scheduling parameter dependence is extracted to the left while in SID it is extracted to the right, see [Verdult 2002, Lemma 3.4] or $\acute{\mathcal{O}}_i$ in (A.40b).

The matrix $\grave{\mathcal{P}}^{\text{diag}}_{k-1|k-j}$ in (4.14c) contains the samples of the scheduling parameters, which are multiplied with the corresponding coefficients in the extended observability matrix $\grave{\mathcal{O}}_i$, and is based on $\grave{\mathcal{P}}_{k+l|k}$, see (A.34). However only the first column is required, since only one column of inputs (and outputs) is considered in (4.13). Therefore, it follows:

$$\grave{\mathcal{P}}_{k-1|k-j}(:,1) = \grave{\mathcal{P}}^{(1)}_{k-1|k-j} = \mu_{k-1} \otimes \mu_{k-2} \otimes \ldots \otimes \mu_{k-j} \tag{4.18a}$$

$$\grave{\mathcal{P}}^{\text{diag}}_{k-1|k-j} = \operatorname{diag}\left(\grave{\mathcal{P}}^{(1)}_{k-1|k-1} \otimes I_{n_u}, \grave{\mathcal{P}}^{(1)}_{k-1|k-2} \otimes I_{n_u}, \ldots, \grave{\mathcal{P}}^{(1)}_{k-1|k-j} \otimes I_{n_u}\right) . \tag{4.18b}$$

Each block on the diagonal of $\grave{\mathcal{P}}^{\text{diag}}_{k-1|k-j}$ is then multiplied from the left by the corresponding $\acute{\mathcal{M}}_l$ in the extended reachability matrix. Moreover, it follows that

$$\grave{\mathcal{U}}_{k-j|k-1} = \grave{\mathcal{P}}^{\text{diag}}_{k-1|k-j} \begin{bmatrix} u_{k-1} \\ \vdots \\ u_{k-j} \end{bmatrix} = \begin{bmatrix} \grave{\mathcal{P}}^{(1)}_{k-1|k-1} \otimes u_{k-1} \\ \vdots \\ \grave{\mathcal{P}}^{(1)}_{k-1|k-j} \otimes u_{k-j} \end{bmatrix} , \tag{4.19}$$

so $\grave{\mathcal{U}}_{k-j|k-1}$ is similar to the data matrix $\grave{\mathcal{Z}}_{k-p|k-1}$ in the SID, see (A.49). In a similar way, matrix $\acute{\mathcal{P}}^{\text{diag}}_{k+i|k}$ is given. However, it is the left hand side expansion, hence transformation is required such that

$$\acute{\mathcal{P}}^{\text{diag}}_{k+i|k} = \operatorname{diag}\left(\acute{\mathcal{P}}^{(1)}_{k|k} \otimes I_{n_y}, \acute{\mathcal{P}}^{(1)}_{k+1|k} \otimes I_{n_y}, \ldots, \acute{\mathcal{P}}^{(1)}_{k+i|k} \otimes I_{n_y}\right)^{\mathrm{T}} . \tag{4.20}$$

These matrices can be used to determine the state x_k (for a sufficiently large j) and the autonomous output sequence y_{k+l} for $l \in \{0, 1, \ldots, i\}$ (non-actuated system). Thus, x_k and y_{k+l} are given by

$$x_k = \underbrace{\acute{\mathcal{R}}_j \grave{\mathcal{P}}^{\text{diag}}_{k-1|k-j}}_{\mathsf{R}_j \diamond \theta_{k-1}} \begin{bmatrix} u_{k-1} \\ \vdots \\ u_{k-j} \end{bmatrix} = \acute{\mathcal{R}}_j \grave{\mathcal{U}}_{k-j|k-1} , \qquad \begin{bmatrix} y_k \\ \vdots \\ y_{k+i} \end{bmatrix} = \underbrace{\acute{\mathcal{P}}^{\text{diag}}_{k+i|k} \grave{\mathcal{O}}_i}_{\mathsf{O}_i \diamond \theta_k} x_k \tag{4.21}$$

for a given input signal u_{k-l} with $l \in \{1, 2, \ldots, j\}$ and a given state x_k, respectively, see [Tóth, Abbas, et al. 2012]. It is also stated in [Tóth, Abbas, et al. 2012] that full row rank of $\acute{\mathcal{R}}_{n_x}$ corresponds to reachability in a structural sense, see also Section 2.3.3. An analogous conclusion can be drawn for $\grave{\mathcal{O}}_{n_x}$: full column rank of $\grave{\mathcal{O}}_{n_x}$ corresponds to observability in a structural sense. Consequently, the extended Hankel matrix is given by

$$\mathcal{H}_{i,j} = \grave{\mathcal{O}}_i \acute{\mathcal{R}}_j \in \mathbb{R}^{\left(n_y \sum_{l=1}^{i} n_\mu^l\right) \times \left(n_u \sum_{l=1}^{j} n_\mu^l\right)} , \tag{4.22}$$

see also (4.14c), which contains time independent information about the full-order LPV system. Thus, the objective of model order reduction is to find – similar to the LTI case – an approximation of $\mathcal{H}_{i,j}$, which contains the most relevant information. Hence, the SVD of the extended Hankel matrix (4.22) reveals the underlying model order and allows to find a realization of the state-space. A state-minimal realization can be obtained by selecting all non-zero singular values, whose amount is $n_{\bar{x}}$, i. e. $\operatorname{rank}(\mathcal{H}_{n_x,n_x}) = n_{\bar{x}}$ [Petreczky et al. 2016]. If

the initial full-order LPV system with n_x states is already state-minimal, then $n_{\bar{x}} = n_x$, see also Definition 2.10. A lower order approximation of the full-order LPV SS representation can be found by selecting only the first $n_{\hat{x}} < n_{\bar{x}}$ singular values.

In subspace identification, the (extended) Hankel matrix is the key ingredient to find a realization of the state or the state-space matrices [Cox et al. 2015; Wingerden & Verhaegen 2009]. Accordingly, the extended Hankel matrix $\mathcal{H}_{f-1,p}$ can be found in LPV SID, see Appendix A.3, in the second line of (A.50), partitioned into two matrices.

As a downside of using (4.22) directly for model order reduction as initially suggested in the Ho-Kalman like scheme [Abbas 2010, Section 6.5] and [Abbas et al. 2010; Tóth, Abbas, et al. 2012], the size of the extended Hankel matrix increases exponentially with i and j leading easily to numerical limitations of the Ho-Kalman algorithm. Nevertheless, i and j need to be chosen large enough such that at least $\operatorname{rank}(\mathcal{H}_{i,j}) = n_{\bar{x}}$, see also Example 4.1. To overcome this numerical problem, a bases reduced algorithm is proposed in [Cox et al. 2015]. The idea is to select only the non-repetitive parts of the extended Hankel matrix, which will reduce the computational load. The reduced basis results in smaller sized sub-Hankel matrices on which matrix decompositions are performed. Unfortunately, due to the structure of $\grave{\mathcal{O}}_i$ and $\acute{\mathcal{R}}_j$ for DT LPV SS representations, there are no obvious bases that can always be avoided. Hence, there should be an automated method of selecting bases until certain condition holds. However, how to do this selection in a computationally efficient and automated way is still an open question.

A different approach is the following fast Ho-Kalman reduction scheme, which is based on the joint work [Schulz et al. 2017], see also [Cox 2018, Section 8.5]. The computational load to obtain the SVD of the full Hankel matrix $\mathcal{H}_{i,j}$ can be decreased significantly by applying QR decompositions on the extended observability and reachability matrices. More specifically, compute the following two QR decompositions $\grave{\mathcal{O}}_i = Q_{\mathcal{O}} R_{\mathcal{O}}$ and $\acute{\mathcal{R}}_j^{\mathrm{T}} = Q_{\mathcal{R}} R_{\mathcal{R}}$. Consequently, the SVD of only $R_{\mathcal{O}} R_{\mathcal{R}}^{\mathrm{T}} \in \mathbb{R}^{n_x \times n_x}$ is required, i. e.

$$\mathcal{H}_{i,j} = \grave{\mathcal{O}}_i \acute{\mathcal{R}}_j = Q_{\mathcal{O}} R_{\mathcal{O}} R_{\mathcal{R}}^{\mathrm{T}} Q_{\mathcal{R}}^{\mathrm{T}} = Q_{\mathcal{O}} U \Sigma V^{\mathrm{T}} Q_{\mathcal{R}}^{\mathrm{T}} \,. \tag{4.23}$$

Thus, a fast model order reduction scheme for DT LPV systems with affine scheduling parameter dependence is available.

The matrices $\grave{\mathcal{O}}_i$ and $\acute{\mathcal{R}}_j$, however, still grow exponentially. In fact, they grow only in one dimension instead of both as it is the case for $\mathcal{H}_{i,j}$, but they must be saved and processed in a QR decomposition. Consequently, only small values for i and j are viable. For this reason, the fast Ho-Kalman reduction scheme of [Schulz et al. 2017] is further modified. As already stated in [Schulz et al. 2017], only the upper triangular matrices $R_{\mathcal{O}}$ and $R_{\mathcal{R}}$ from the QR decomposition are required in order to compute the projection matrix T for model order reduction. Following the idea in [Schulz et al. 2017], both matrices are efficiently computed by the UDU decomposition of

$$\grave{\mathcal{O}}_i^{\mathrm{T}} \grave{\mathcal{O}}_i = R_{\mathcal{O}}^{\mathrm{T}} Q_{\mathcal{O}}^{\mathrm{T}} Q_{\mathcal{O}} R_{\mathcal{O}} = R_{\mathcal{O}}^{\mathrm{T}} R_{\mathcal{O}} = U_{\mathcal{O}}^{\mathrm{T}} D_{\mathcal{O}} U_{\mathcal{O}} \in \mathbb{R}^{n_x \times n_x} \,, \qquad R_{\mathcal{O}} = D_{\mathcal{O}}^{\frac{1}{2}} U_{\mathcal{O}} \tag{4.24a}$$

$$\acute{\mathcal{R}}_j \acute{\mathcal{R}}_j^{\mathrm{T}} = R_{\mathcal{R}}^{\mathrm{T}} Q_{\mathcal{R}}^{\mathrm{T}} Q_{\mathcal{R}} R_{\mathcal{R}} = R_{\mathcal{R}}^{\mathrm{T}} R_{\mathcal{R}} = U_{\mathcal{R}}^{\mathrm{T}} D_{\mathcal{R}} U_{\mathcal{R}} \in \mathbb{R}^{n_x \times n_x} \,, \qquad R_{\mathcal{R}} = D_{\mathcal{R}}^{\frac{1}{2}} U_{\mathcal{R}} \,. \tag{4.24b}$$

In addition, the matrices $\grave{\mathcal{O}}_i^{\mathrm{T}} \grave{\mathcal{O}}_i$ and $\acute{\mathcal{R}}_j \acute{\mathcal{R}}_j^{\mathrm{T}}$ can be computed recursively, i. e. using (4.15)–(4.17)

to obtain

$$\grave{\mathcal{O}}_i^{\mathrm{T}}\grave{\mathcal{O}}_i = \sum_{l=1}^{i} \grave{\mathcal{N}}_l^{\mathrm{T}}\grave{\mathcal{N}}_l \ , \qquad \grave{\mathcal{N}}_l^{\mathrm{T}}\grave{\mathcal{N}}_l = \sum_{m=0}^{n_\theta} A_m^{\mathrm{T}}\grave{\mathcal{N}}_{l-1}^{\mathrm{T}}\grave{\mathcal{N}}_{l-1}A_m \ , \quad l>1 \tag{4.25a}$$

$$\acute{\mathcal{R}}_j\acute{\mathcal{R}}_j^{\mathrm{T}} = \sum_{l=1}^{j} \acute{\mathcal{M}}_l\acute{\mathcal{M}}_l^{\mathrm{T}} \ , \qquad \acute{\mathcal{M}}_l\acute{\mathcal{M}}_l^{\mathrm{T}} = \sum_{m=0}^{n_\theta} A_m\acute{\mathcal{N}}_{l-1}\acute{\mathcal{N}}_{l-1}^{\mathrm{T}}A_m^{\mathrm{T}} \ , \quad l>1 \ . \tag{4.25b}$$

Consequently, no matrices with exponentially growing dimensions have to be stored and only a repeated multiplication of quadratic matrices of dimension n_x has to be performed. This means, that the computational load of the Ho-Kalman scheme in the LPV case becomes very similar to that one in the LTI case and the following Lemma can be introduced. It is a modification of the Ho-Kalman like scheme, see [Abbas et al. 2010; Tóth, Abbas, et al. 2012], and is proven to provide a state-minimal representation, if $n_{\hat{x}} = n_{\bar{x}}$ [Petreczky et al. 2016].

Lemma 4.1 (Quick Ho-Kalman Reduction Scheme). Choose i and j large enough such that at least $\operatorname{rank}(\grave{\mathcal{O}}_i^{\mathrm{T}}\grave{\mathcal{O}}_i) = n_{\bar{x}}$ and $\operatorname{rank}(\acute{\mathcal{R}}_j\acute{\mathcal{R}}_j^{\mathrm{T}}) = n_{\bar{x}}$, respectively. Compute the quadratic matrices $\grave{\mathcal{O}}_i^{\mathrm{T}}\grave{\mathcal{O}}_i$ and $\acute{\mathcal{R}}_j\acute{\mathcal{R}}_j^{\mathrm{T}}$ of dimension n_x recursively as in (4.25) with $\grave{\mathcal{N}}_1$ and $\acute{\mathcal{M}}_1$ in (4.17) and (4.16), respectively. Perform the UDU decompositions as in (4.24), compute the matrices $R_{\mathcal{O}}$ and $R_{\mathcal{R}}$ and the SVD of their product as follows

$$R_{\mathcal{O}} = D_{\mathcal{O}}^{\frac{1}{2}}U_{\mathcal{O}} \ , \qquad R_{\mathcal{R}} = D_{\mathcal{R}}^{\frac{1}{2}}U_{\mathcal{R}} \ , \qquad R_{\mathcal{O}}R_{\mathcal{R}}^{\mathrm{T}} = U\Sigma V^{\mathrm{T}} \ . \tag{4.26}$$

Then, a realization of the reduced-order LPV state-space matrices is found by

$$\hat{A}_l = TA_lT^{\dagger} \ , \qquad \hat{B}_l = TB_l \ , \qquad \hat{C}_l = C_lT^{\dagger} \ , \tag{4.27}$$

where $l \in \{0, 1, \ldots, n_\theta\}$ and the projection matrix is

$$T = \hat{\Sigma}^{-\frac{1}{2}}\hat{U}^{\mathrm{T}}R_{\mathcal{O}} \ , \qquad T^{\dagger} = R_{\mathcal{R}}^{\mathrm{T}}\hat{V}\hat{\Sigma}^{-\frac{1}{2}} \ , \tag{4.28}$$

where $\hat{U}$, $\hat{V}$ denotes the first $n_{\hat{x}}$ columns of the matrices U, V, respectively, $\hat{\Sigma}$ denotes the upper left $n_{\hat{x}}$ by $n_{\hat{x}}$ matrix of Σ, and $1 \leq n_{\hat{x}} \leq n_{\bar{x}}$. In (4.28), $T^{\dagger}$ is the right pseudo-inverse of the projection matrix T, i. e. $TT^{\dagger} = I_{n_{\hat{x}}}$.

Proof. In the Ho-Kalman like scheme [Tóth, Abbas, et al. 2012], the shifted Hankel matrix is chosen as:

$$\bar{\mathcal{H}}_{i,j} = \grave{\mathcal{O}}_i \begin{bmatrix} A_0 & A_1 & \ldots & A_{n_\theta} \end{bmatrix} (I_{n_\mu} \circledcirc \acute{\mathcal{R}}_j) \ , \tag{4.29}$$

where the block wise Kronecker product $\circledcirc$ is defined as

$$(I_{n_\mu} \circledcirc \acute{\mathcal{R}}_j) = \begin{bmatrix} I_{n_\mu} \otimes \acute{\mathcal{M}}_1 & I_{n_\mu} \otimes \acute{\mathcal{M}}_2 & \ldots & I_{n_\mu} \otimes \acute{\mathcal{M}}_j \end{bmatrix} \ . \tag{4.30}$$

In this scheme, the extended observability and reachability matrices and there approximates are

$$\grave{\mathcal{O}}_i = Q_{\mathcal{O}}R_{\mathcal{O}} \ , \qquad \hat{\mathcal{O}}_i = Q_{\mathcal{O}}\hat{R}_{\mathcal{O}} \ , \qquad \hat{R}_{\mathcal{O}} = \hat{U}\hat{\Sigma}^{\frac{1}{2}} \tag{4.31a}$$

$$\acute{\mathcal{R}}_j^{\mathrm{T}} = Q_{\mathcal{R}}R_{\mathcal{R}} \ , \qquad \hat{\mathcal{R}}_j^{\mathrm{T}} = Q_{\mathcal{R}}\hat{R}_{\mathcal{R}} \ , \qquad \hat{R}_{\mathcal{R}} = \hat{V}\hat{\Sigma}^{\frac{1}{2}} \tag{4.31b}$$

using QR decomposition and the reduced matrices from SVD decomposition in (4.26). Then,

$$\begin{aligned}\hat{\mathcal{O}}_i^{\dagger}\bar{\mathcal{H}}_{i,j}(I_{n_\mu} \circledcirc \hat{\mathcal{R}}_j^{\dagger}) &= \hat{\Sigma}^{-\frac{1}{2}}\hat{U}^{\mathrm{T}}Q_{\mathcal{O}}^{\mathrm{T}}Q_{\mathcal{O}}R_{\mathcal{O}}\begin{bmatrix}A_0 & A_1 & \dots & A_{n_\theta}\end{bmatrix}(I_{n_\mu} \circledcirc R_{\mathcal{R}}^{\mathrm{T}}Q_{\mathcal{R}}^{\mathrm{T}}Q_{\mathcal{R}}\hat{V}\hat{\Sigma}^{-\frac{1}{2}}) \\ &= T\begin{bmatrix}A_0 & A_1 & \dots & A_{n_\theta}\end{bmatrix}(I_{n_\mu} \circledcirc T^{\dagger}) \ . \end{aligned} \tag{4.32}$$

Hence, the construction of the $\hat{A}_l$ matrices is proven. Recall that C_l with $l \in \{0, 1, \dots, n_\theta\}$ is constructed from the first $n_y n_\mu$ rows of the Hankel matrix, i. e. $\mathcal{H}_{1,j} = \grave{\mathcal{N}}_1\acute{\mathcal{R}}_j$ with $\grave{\mathcal{N}}_1^{\mathrm{T}} = [C_0^{\mathrm{T}} \quad C_1^{\mathrm{T}} \quad \dots \quad C_{n_\theta}^{\mathrm{T}}]$, see (4.17). Therefore, $\hat{C}_l$ is

$$\mathcal{H}_{1,j}\hat{\mathcal{R}}_j^{\dagger} = \grave{\mathcal{N}}_1\acute{\mathcal{R}}_j\hat{\mathcal{R}}_j^{\dagger} = \grave{\mathcal{N}}_1 T^{\dagger} = \begin{bmatrix}\hat{C}_0^{\mathrm{T}} & C_1^{\mathrm{T}} & \dots & \hat{C}_{n_\theta}^{\mathrm{T}}\end{bmatrix}^{\mathrm{T}} \ . \tag{4.33}$$

Hence, the construction of $\hat{C}_l$ is provided. Similar argument can be made for the construction of $\hat{B}_l$ and is therefore omitted. Finalizing, it is shown that $TT^{\dagger} = I_{n_{\hat{x}}}$. Note that $\hat{U}^{\mathrm{T}}U = [I_{n_{\hat{x}}} \quad 0]$, $\hat{V}^{\mathrm{T}}V = [I_{n_{\hat{x}}} \quad 0]$ for any $n_{\hat{x}}$ (orthogonality property). Hence,

$$TT^{\dagger} = \hat{\Sigma}^{-\frac{1}{2}}\hat{U}^{\mathrm{T}}R_{\mathcal{O}}R_{\mathcal{R}}^{\mathrm{T}}\hat{V}\hat{\Sigma}^{-\frac{1}{2}} = \hat{\Sigma}^{-\frac{1}{2}}\begin{bmatrix}I_{n_{\hat{x}}} & 0\end{bmatrix}\Sigma\begin{bmatrix}I_{n_{\hat{x}}} & 0\end{bmatrix}^{\mathrm{T}}\hat{\Sigma}^{-\frac{1}{2}} = I_{n_{\hat{x}}} \ , \tag{4.34}$$

which completes the proof. □

In contrast to the LPV Ho-Kalman like scheme in [Tóth, Abbas, et al. 2012] or its fast version in [Schulz et al. 2017], it is shown here how a reduced-order realization can be obtained by finding the static projection matrix T that does not require the computation of any exponential growing matrix, such as $\grave{\mathcal{O}}_i$, $\acute{\mathcal{R}}_j$, $\mathcal{H}_{i,j}$ or the shifted Hankel matrix $\bar{\mathcal{H}}_{i,j}$ in (4.29). In addition, in the original Ho-Kalman like scheme the shifted Hankel matrix $\bar{\mathcal{H}}_{i,j}$ is constructed from $\acute{\mathcal{R}}_{j-1}$, whereas in (4.29) the matrix $\acute{\mathcal{R}}_j$ is employed for $\bar{\mathcal{H}}_{i,j}$. However, this modification does not change the realization in any way. Note that the singular values of $R_{\mathcal{O}}R_{\mathcal{R}}^{\mathrm{T}}$ are the first n_x Hankel singular values, which is in case of a minimal realization also the amount of (numerically approximately) nonzero singular values, respectively the rank of $\mathcal{H}_{i,j}$.

If the full-order LPV SS model is quadratically stable, it can be shown that the reduced LPV SS model obtained with the (quick) Ho-Kalman scheme with SVD is balanced [De Schutter 2000; Tóth, Abbas, et al. 2012]. Theoretical global error bounds between the original and reduced model have not been developed yet (as known in the LTI case [Antoulas 2008]) and remain objectives for future research. However, approximate error bounds can be computed using the μ-test in the enhanced linear fractional representation toolbox [Magni 2006].

Finally, the quick Ho-Kalman reduction scheme in Lemma 4.1 is summarized in Algorithm 7.

4.3.3 Analysis of the Reduction Scheme

It is worth mentioning that the canonical form, which IO models have in SS representation, is lost during an order reduction by the transformation or projection matrix, regardless of which order reduction method is used. However, due to a constant transformation matrix, state-space matrices that are scheduling parameter independent remain constant.

Unfortunately, there is no interpretation from a system theoretic point of view for the matrices $\grave{\mathcal{O}}_i^{\mathrm{T}}\grave{\mathcal{O}}_i$ and $\acute{\mathcal{R}}_j\acute{\mathcal{R}}_j^{\mathrm{T}}$ in (4.25). In the LTI case, the Gramians and the Hankel matrix are defined by the observability and reachability matrix, thus

$$\mathsf{O}_\infty^{\mathrm{T}}\mathsf{H}_{\infty,\infty}\mathsf{R}_\infty^{\mathrm{T}} = \mathsf{O}_\infty^{\mathrm{T}}\mathsf{O}_\infty\mathsf{R}_\infty\mathsf{R}_\infty^{\mathrm{T}} = X_{\mathrm{o}}X_{\mathrm{c}} \ , \tag{4.35}$$

Algorithm 7 Quick Ho-Kalman Reduction for DT LPV models with affine scheduling parameter dependence

Require: DT LPV SS model $G(\theta)$, i, j, reduced order $n_{\hat{x}}$
Ensure: Approximate DT LPV SS model $\hat{G}(\theta)$ with reduced order $n_{\hat{x}} < n_x$

$\{A(\theta), B(\theta), C(\theta), D(\theta), T_\mathrm{s}\} \leftarrow G(\theta)$
$N_1 \leftarrow [C_0; C_1; \ldots; C_{n_\theta}]$
$N_{ii-1} \leftarrow N_1^\mathrm{T} N_1$
$O_{ii} \leftarrow N_{ii-1}$
for $l = 2$ **to** i **do**
    $N_{ii} \leftarrow \sum_{m=0}^{n_\theta} A_m^\mathrm{T} N_{ii-1} A_m$
    $N_{ii-1} \leftarrow N_{ii}$
    $O_{ii} \leftarrow O_{ii} + N_{ii}$
end for
$M_1 \leftarrow [B_0, B_1, \ldots, B_{n_\theta}]$
$M_{jj-1} \leftarrow M_1 M_1^\mathrm{T}$
$R_{jj} \leftarrow M_{jj-1}$
for $l = 2$ **to** j **do**
    $M_{jj} \leftarrow \sum_{m=0}^{n_\theta} A_m M_{jj-1} A_m^\mathrm{T}$
    $M_{jj-1} \leftarrow M_{jj}$
    $R_{jj} \leftarrow R_{jj} + M_{jj}$
end for
$[U_O, D_O] \leftarrow \mathrm{udu}(O_{ii})$ ▷ MATLAB: ldl(O_{ii}, 'upper')
$R_O \leftarrow D_O^{1/2} U_O$
$[U_R, D_R] \leftarrow \mathrm{udu}(R_{jj})$
$R_R \leftarrow D_R^{1/2} U_R$
$[U, \Sigma, V] \leftarrow \mathrm{svd}(R_O R_R^\mathrm{T})$
$T \leftarrow \Sigma^{-1/2}(1{:}n_{\hat{x}}, 1{:}n_{\hat{x}}) U^\mathrm{T}(:, 1{:}n_{\hat{x}}) R_O$
$T^\dagger \leftarrow R_R^\mathrm{T} V(:, 1{:}n_{\hat{x}}) \Sigma^{-1/2}(1{:}n_{\hat{x}}, 1{:}n_{\hat{x}})$
for $l = 0$ **to** n_θ **do**
    $\hat{A}_l \leftarrow T A_l T^\dagger$
    $\hat{B}_l \leftarrow T B_l$
    $\hat{C}_l \leftarrow C_l T^\dagger$
end for
$\hat{G}(\theta) \leftarrow \{\hat{A}(\theta), \hat{B}(\theta), \hat{C}(\theta), D(\theta), T_\mathrm{s}\}$

see also Section 2.3.3 and the introductory LTI case of this section. Consequently, the Gramians X_o and X_c can be computed by O_∞ and R_∞, respectively, and an approximate of the Gramians is obtained for finite observability and reachability matrices. However, an equivalent computation in the LPV case with the matrices in (4.24) is not possible, since for a sufficiently large i

$$X_\mathrm{o} \succ \mathsf{O}_i^\mathrm{T} \mathsf{O}_i = \grave{\mathcal{P}}_i^\mathrm{T} \acute{\mathcal{O}}_i^\mathrm{T} \acute{\mathcal{O}}_i \grave{\mathcal{P}}_i = \grave{\mathcal{O}}_i^\mathrm{T} \acute{\mathcal{P}}_i^\mathrm{T} \acute{\mathcal{P}}_i \grave{\mathcal{O}}_i \ , \tag{4.36}$$

where $\acute{\mathcal{P}}_i = \acute{\mathcal{P}}_{k+i|k}^{\mathrm{diag}}$, and $\grave{\mathcal{P}}_i = \grave{\mathcal{P}}_{k+i|k}(:, 1) \otimes I_{n_x}$ as well as $\acute{\mathcal{O}}_i$ can be derived as in (A.32), (A.33), and (A.40b). The approximation for the generalized controllability Gramian is similar and thus not considered further here. The product $\acute{\mathcal{O}}_i^\mathrm{T} \acute{\mathcal{O}}_i$ is a huge square matrix of dimension $n_x n_\mu^i$, where $\acute{\mathcal{O}}_i^\mathrm{T} \acute{\mathcal{O}}_i$ contains scheduling parameter independent information about the system. In

contrast to this, the product of $\grave{\mathcal{O}}_i^{\mathrm{T}}$ and $\grave{\mathcal{O}}_i$ is used in the UDU decomposition in (4.24). If the products $\grave{\mathcal{O}}_i^{\mathrm{T}}\grave{\mathcal{O}}_i$ and $\mathsf{O}_i^{\mathrm{T}}\mathsf{O}_i$ are compared with each other, it is clear that the first one contains only a subset of the matrix products of the second one. This is illustrated by the following calculation: assume $i = 1$ and dependency on one affine scheduling parameter, then it follows

$$\grave{\mathcal{O}}_1^{\mathrm{T}}\grave{\mathcal{O}}_1 = \begin{bmatrix} C_0^{\mathrm{T}} & C_1^{\mathrm{T}} \end{bmatrix} \begin{bmatrix} C_0 \\ C_1 \end{bmatrix} = C_0^{\mathrm{T}}C_0 + C_1^{\mathrm{T}}C_1 \tag{4.37}$$

$$\begin{aligned} \mathsf{O}_1^{\mathrm{T}}\mathsf{O}_1 &= C^{\mathrm{T}}(\theta)C(\theta) = (C_0 + C_1\theta_1)^{\mathrm{T}}(C_0 + C_1\theta_1) \\ &= C_0^{\mathrm{T}}C_0 + C_1^{\mathrm{T}}C_1\theta_1^2 + (C_0^{\mathrm{T}}C_1 + C_1^{\mathrm{T}}C_0)\theta_1 \ . \end{aligned} \tag{4.38}$$

Hence, the cross multiplications $C_0^{\mathrm{T}}C_1$ and its transposed do not appear in $\grave{\mathcal{O}}_1^{\mathrm{T}}\grave{\mathcal{O}}_1$. Consequently, the matrix products in the UDU decomposition in (4.24) cannot be used to approximate the generalized Gramians X_{o} and X_{c}.

Nevertheless, the QR decomposition of $\acute{\mathcal{O}}_i^{\mathrm{T}} = \acute{Q}_{\mathcal{O}}\acute{R}_{\mathcal{O}}$ and subsequently the SVD of $\acute{R}_{\mathcal{O}}\acute{R}_{\mathcal{O}}^{\mathrm{T}}$ can be used to extract the characteristics of $\acute{\mathcal{O}}_i^{\mathrm{T}}\acute{\mathcal{O}}_i$, respectively, and thus obtain an approximation of X_{o}. However, from a computational point of view, this is not as efficient as the UDU decomposition in (4.24), thus this approach is no longer pursued.

The pre-decomposition of the extended observability and reachability matrix is useful from three points of view. Firstly, it avoids construction of exponentially growing matrices as the extended Hankel matrix and shifted Hankel matrix and instead computes the matrix products in (4.25) recursively. Secondly, the UDU decomposition is computationally cheaper than the QR decomposition and applied to a much smaller matrix. Thirdly, in Lemma 4.1 as well as in [Schulz et al. 2017, Lemma 1] the SVD is performed on a much smaller sized matrix, i. e. the SVD of $R_{\mathcal{O}}R_{\mathcal{R}}^{\mathrm{T}} \in \mathbb{R}^{n_x \times n_x}$ and not of $\mathcal{H}_{i,j} \in \mathbb{R}^{\left(n_y \sum_{l=1}^{i} n_\mu^l\right) \times \left(n_u \sum_{l=1}^{j} n_\mu^l\right)}$ is required. To be more specific, the UDU decomposition of a square matrix requires $n^3/3$ flops [Golub & Van Loan 2013, Algorithm 4.2.2], while QR decomposition of a matrix $A \in \mathbb{R}^{m \times n}$ with $\operatorname{rank}(A) = n$ needs $2mn^2$ flops using a modified Gram-Schmidt algorithm [Golub & Van Loan 2013, Algorithm 5.2.6]. On the other hand, SVD consists usually of two steps and needs $4m^2n + 22n^3$ flops (or $4m^2n + 8mn^2 + 9n^3$, if $m \approx n$) [Golub & Van Loan 2013, Fig. 8.6.1]. Nevertheless, pre-decomposition is only possible, if the state-space matrices are known a priori. Therefore, the concept cannot be used for LPV SID, since the state-space matrices are still unknown there, but it can be used in a two-step identification procedure: in the first step a (high-order) LPV IO model is identified and in the second step the SS representation of this model is reduced in its state order via Lemma 4.1. This results in LPV SS models, which can be obtained numerically more efficient than LPV SID.

As mentioned at the beginning of this section, the Ho-Kalman reduction method is an approximation method. Thus, the projection matrices T and $T^{\dagger}$ in (4.28) are only approximate transformation matrices for a balanced realization. Consequently, the error limits known for LTI and LPV systems, see [Zhou et al. 1996, Section 8.5] and [Wood et al. 1996], are not transferable to the Ho-Kalman reduction method. Only if i and j are sufficiently large such that the Hankel singular values do no longer change, it can be assumed that the error bounds apply. Theoretical global error bounds between the full- and reduced-order model have not been developed yet for $n_{\hat{x}} \leq n_x$ (as known in the LTI case [Antoulas 2008]) and remain objectives for future research. However, approximate error bounds can be computed using the μ-test in the enhanced linear fractional transformation toolbox [Magni 2006]. The error bound can also be analyzed a posteriori via the induced ℓ_2 system norm for (4.1).

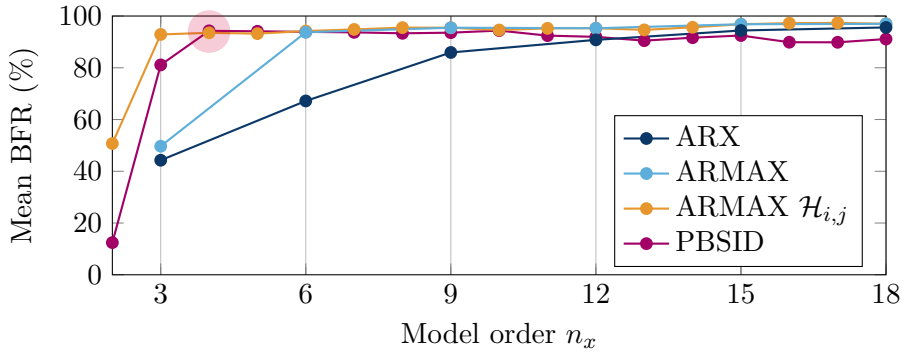


Figure 4.5: Identification results on fourth order MIMO example with averaged fit using IO identification with ARX and ARMAX model structures as well as PBSID, where the model order reduction scheme is applied to the identified ARMAX models

To conclude this chapter, the performance of the proposed model order reduction scheme is presented on a numerical example.

Example 4.2. A fourth order DT LPV MIMO system with innovation type noise model structure, see Fig. 3.5 and (3.44) with scheduling parameter independent output equation, introduced in [Verdult & Verhaegen 2002] is tested with two inputs, three outputs and three scheduling parameters. For the estimation and validation data with 1000 samples, the input, scheduling parameter, and noise signals are designed as in [Verdult & Verhaegen 2002], with an averaged signal-to-noise ratio of about 20 dB. The validation outputs are simulated without noise, in order to evaluate the simulated output of the identified model. The identification results are shown in Fig. 4.5 in terms of BFR versus model order.

To find the best LPV model for each order, in SID, both parameter dependent and independent input and observer gain matrices are considered, i. e. $B(\theta) = B$ and $K(\theta) = K$, respectively. In addition, for each past window $p \leq 6$ different models of order up to pn_y are identified. Further settings (e. g., regularization) are chosen as suggested in the PBSID toolbox documentation. Then, in Fig. 4.5 only the best results are plotted, i. e. the highest mean fit for a given order, and the identified models are denoted as PBSID models in the following. In the IO identification case, first LPV ARX models in SF are obtained using Algorithm 1 with all combinations of n_a and n_b with $n_b \leq n_a \leq 6$ and without regularization, i. e. $K = 0$, cf. [Schulz et al. 2016]. Since the model order depends only on $n_a n_y$, the same model order is obtained for different $n_b \leq n_a$ and again the best results are plotted in Fig. 4.5. Furthermore, LPV ARMAX models are identified using the iterative procedure in Algorithm 4. Now, also the order n_c is varied with the limitation $n_c \leq n_a$. Hence, various IO models are identified with equivalent model order and only the model with the highest BFR are displayed. In Fig. 4.5, it is shown that the models of this iterative method slightly outperform the (iteration-free) PBSID models for high orders. However, the PBSID models are significantly better for model orders less than six, i. e. models of low order that are preferred herein.

If IO models in SF are identified, it is not possible to obtain an LPV model that has a SS representation of the same order as the true system. However, if order reduction is applied to the identified models, any order can be obtained (up to order 18 in this example). Here, the Ho-Kalman approach in Lemma 4.1 with $i = j = 50$ is applied. Then,

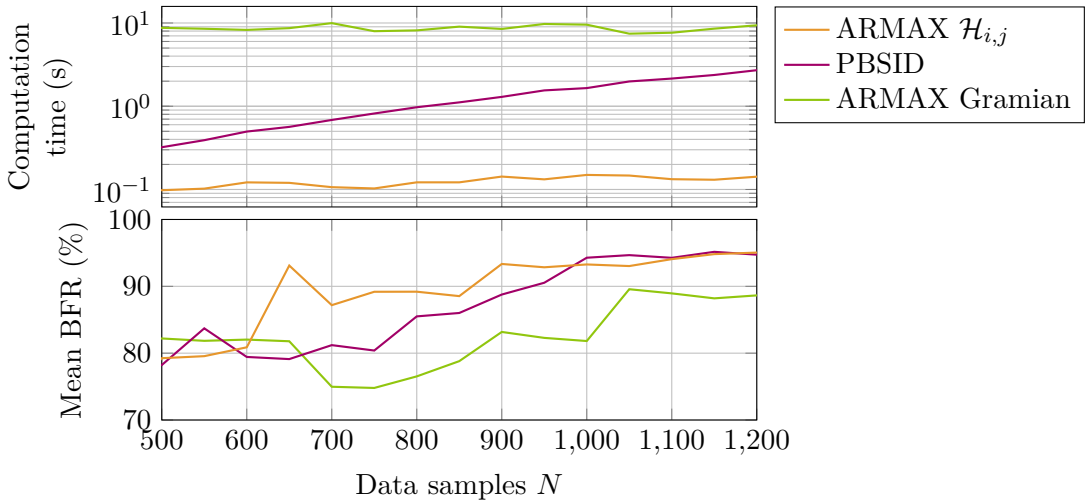


Figure 4.6: Computation time to obtain LPV SS models of fourth order for the LPV MIMO example and corresponding averaged fit

by choosing for a given model order the highest BFR over all (reduced-order) models of this order, the orange line is achieved and the models are denoted as ARMAX $\mathcal{H}_{i,j}$ models in the following. The approximation error due to model order reduction is small, based on the comparison of the BFR for the ARMAX and the ARMAX $\mathcal{H}_{i,j}$ models. In fact, it can be seen that the true LPV system can be approximated very well by a third order ARMAX $\mathcal{H}_{i,j}$ model, based on the BFR. The ARMAX $\mathcal{H}_{i,j}$ models of second and third order also have a significantly higher BFR than the corresponding PBSID models. In addition, it is clear that the proposed ARMAX identification method in combination with the quick Ho-Kalman model order reduction provides LPV models with similar accuracy as the PBSID method on this particular example while keeping computational complexity low. The computation time for the two models of fourth order (highlighted by the light red circle in Fig. 4.5) is less than 0.2 s using ARMAX $\mathcal{H}_{i,j}$ and 1.6 s using PBSID.

In Fig. 4.6, the computation time and the corresponding averaged BFR are shown for the two models highlighted by the light red circle in Fig. 4.5. The calculations are performed on a computer with 2.6 GHz CPU and 16 GB of RAM. In the comparison of the ARMAX $\mathcal{H}_{i,j}$ models with the PBSID models, the elapsed time is for the two-step procedure used in ARMAX $\mathcal{H}_{i,j}$ considerably shorter, scales better with the number of samples, and the fit is comparable or even slightly better. The result also indicates that the PBSID approach can lead to long computation times or even problems, if a large number of samples is required, for example to capture the system dynamics and the parameter set of LPV systems with many scheduling parameters. In the IO case, only the identification step affects the computation time w. r. t. the number of samples N, but not the order reduction step that is approximately constant at 5 ms. Of course, the computation time for the ARMAX $\mathcal{H}_{i,j}$ models scales with i and j, however the scaling is approximately linear[4)] and

[4)] With the fast Ho-Kalman scheme of [Schulz et al. 2017], the computation time increases strongly with i and j since the extended observability and reachability matrices $\grave{\mathcal{O}}_i$ and $\acute{\mathcal{R}}_j$ grow exponentially with i and j. Additionally, using the fast Ho-Kalman scheme, i and j are in this example limited to values less than 13, otherwise storage problems occur for the matrices $\grave{\mathcal{O}}_i$ and $\acute{\mathcal{R}}_j$.

therefore the identification step takes most of the time. In addition, the fit and computation time of reduced-order models is plotted using generalized Gramians, where a non-convex optimization is solved on the scheduling parameter set using the LMIs in Theorems 2.2 and 2.3 and the cost function in (4.5). The calculation time is the longest with about 8.5 s and the BFR is mostly not as good as using the quick Ho-Kalman scheme. Again, the computational load of this model order reduction technique is independent of the number of data samples as with the Ho-Kalman scheme. Instead, the computation time is mainly determined by the number of iterations required to solve the non-convex optimization problem, the number of scheduling parameters and the functional dependency[5)] on them. Nevertheless, the computational efficiency may be increased by a different choice of solver settings or by avoiding the gridding of the parameter set using, e. g., the S-procedure or enforcing multi-convexity [Apkarian et al. 1996, Section 11.9].

[5)] Only the (LTI) systems at the vertices of the convex polytope that spans the parameter set need to be considered in case of an affine scheduling parameter dependency. In contrast, a grid of the parameter set is required for a general functional dependency.

5 Application Examples on the Air Path

The algorithms developed in the preceding chapters are tested on an industrial application in the present chapter. For this purpose, the air path of a turbocharged gasoline engine is considered, which shows a large number of nonlinearities in theoretical modeling. To demonstrate the applicability and effectiveness of the system identification approach, the different LPV IO identification methods are compared with each other and with the LPV subspace identification method. If necessary, a two-step procedure is used: the identification of LPV IO models with subsequent order reduction of the corresponding model in state-space representation. First, however, the application and the approach are motivated.

Today, the vast majority of vehicles are powered by internal combustion engines (ICEs). Such an engine is subject to strict legal requirements regarding emissions and, thus indirectly, fuel consumption. In addition, the customer demands a dynamic driving behavior, i. e. a fast response without oscillations. To meet these competing requirements, ICEs are constantly being further developed and equipped with additional actuators, such as variable valve actuation, variable compression ratio or electrically adjusted turbochargers [Pischinger & Seiffert 2016, Chapter 5]. As a result, engine control must both achieve a high level of quality so that, e. g., emission requirements are met and deal with increasingly elaborated systems, since the additional actuators must be coordinated and controlled on a (commonly) coupled system.

In general, ICEs contains many subsystems that must be controlled. The most important of which are for spark-ignition (SI) engines the fuel system, the ignition system, and the air system [Isermann 2014, Section 1.3]. In the following, the air path responsible for providing fresh air in the cylinder for combustion is examined in more detail for a turbocharged gasoline engine. Several actuators are available for this purpose, the throttle valve, the inlet and exhaust valves of the cylinders and – in the case of exhaust turbocharged engines – an actuator at the turbine. With these actuators, pressures in the intake tract can be controlled or the air charge in the cylinder, i. e. the proportion of fresh air in the cylinder, can be directly influenced. Due to nonlinearities in the system, including the nonlinearities in the actuators, linear control approaches reach their limits. A nonlinear control approach is presented, e. g., in [Beckmann 2015], however linearized gain scheduled controllers are commonly used in the automotive industry, where the controller parameters are stored in lookup maps to control the local (linearized) behavior at the operating point of the ICE. Due to the dependence of the system behavior on the operating point, it is obvious to use LPV methods for control designs, which also provide guarantees for stability and performance. Application examples for the control (of subsystems) of the air path with LPV methods can be found in, e. g., [Jung & Glover 2006; Kominek et al. 2012; Kwiatkowski 2007; Wei & Re 2007]. Since the design of LPV controllers, like other modern linear and nonlinear control designs, is model-based, a model of the air path is required. A gray box model approach is often used, in which the nonlinear system is modeled by physical-phenomenological relations, whose parameters must be determined by measurements, cf. [Isermann 2014, Chapter 3] and [Beckmann 2015, Chapter 4]. In this thesis, a different approach is followed, in which a black box LPV model is identified assisted by the previous knowledge about the modeling of the system. This approach proves

to be efficient because both approaches, gray and black box modeling, require measurements, whereas the latter can be more time efficient as it avoids accurate physical modeling and many steps in identification can be automated. In addition, the gray box model is nonlinear and must be converted into an LPV model to use LPV control design techniques, cf. [Jung & Glover 2003]. Furthermore, (structural) modeling errors may be present in the gray box model. Instead, a DT LPV model is directly obtained with the black box approach. Unfortunately, this depends substantially on the amount and type of information about the system, its dynamics, and the operating region that is contained in the estimation data set. A direct comparison of the nonlinear modeling approach with the identification of LPV models is outside the scope of this work due to the measurement effort required for the nonlinear modeling approach and is therefore not discussed here.

In the remainder of this chapter, the theoretical modeling of the air path is presented in Section 5.1. Then, two subsystems are considered in Section 5.2 for which LPV models are identified based on measurement data. The identified LPV models, which are obtained with different identification algorithms, are compared with each other. In this process, it is elaborated which approaches and changes bring which improvement, where computational effort, problem scaling, and model accuracy improvement are evaluated.

5.1 Modeling of the Turbocharged SI Engine

For a better understanding of the SI engine a semi-physical modeling approach, consisting of physical and phenomenological equations, is recapitulated, cf. [Daasch et al. 2016; Wahlström & Eriksson 2011], [Beckmann 2015, Chapter 4], [Guzzella & Onder 2010, Chapter 2]. In this modeling approach, the level of abstraction is chosen to cover the relevant effects that influence the dynamic behavior. This leads to a mean value model, i. e. the strokes of the engine are not considered but a mean process with averaged states and process variables. Such a mean value modeling approach for the air path is very common and state-of-the-art, cf. [Isermann 2014, Chapter 4], [Guzzella & Onder 2010, Chapter 2], [Eriksson & Nielsen 2014, Chapters 7 and 8]. The modeled components are the air intake, the air outlet, the boost, intake, and exhaust manifolds, the cylinder (including the inlet and exhaust valves), the throttle valve, and the turbocharger. The corresponding schematic model of the turbocharged SI engine is shown in Fig. 5.1. Other components of the air path are not considered, such as the air filter upstream and the intercooler downstream of the compressor as well as catalysts, filters, and traps in the exhaust path. Their influence on supplying the cylinder with fresh air is rather small, which is why they are neglected here.

Note that for the modeling of the air path in this section a common notation is used in order to be easily comprehensible, although it is ambiguous with the notation in the previous sections. However, the ambiguity can always be resolved from the context.

On the layer of physical modeling the model is divided into receivers (also denoted as manifolds) and valves, where the receivers are modeled by differential equations and the valves by static functions. Each valve connects two receivers. Most valves serve as actuators and can be controlled, where the opening angle or timing of the valves is denoted by α and the reference is referred to as u in this thesis, because the references are the inputs of the system to be identified.

A receiver is considered as a well-mixed fixed volume where the thermodynamic states, temperature T, pressure p and mass m, are lumped variables. In Fig. 5.1, the modeled receivers

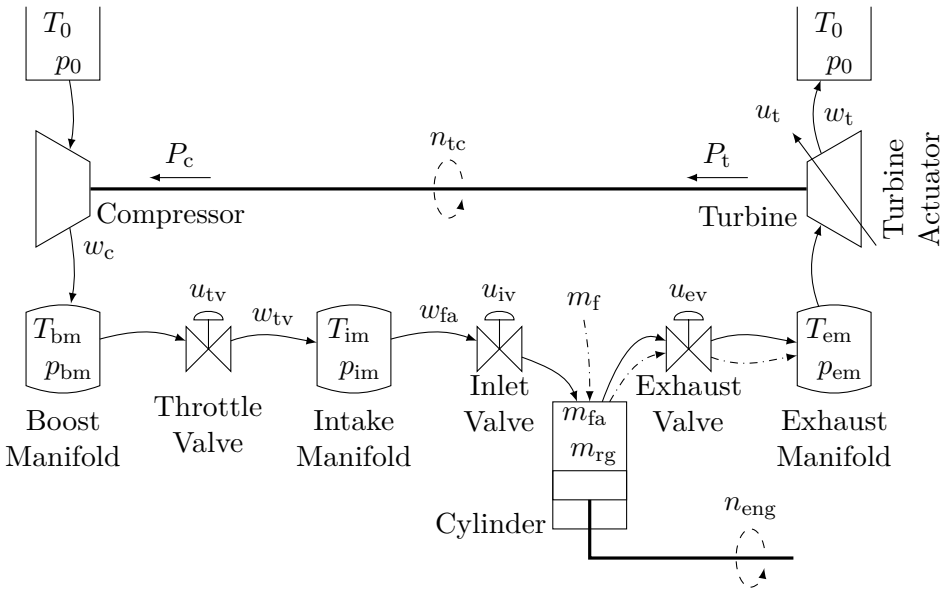


Figure 5.1: Simplified air path model of the turbocharged gasoline engine

are the boost, intake, and exhaust manifold. Additionally, the ambiance is modeled as a receiver with constant temperature T_0 and pressure p_0. Since the effects of the temperature states on the dynamic behavior are negligible as simulation investigations have shown and that is also established in [Wahlström & Eriksson 2011], a simplified reduced-order model is obtained by assuming constant temperatures.

The throttle valve is modeled by the valve equation, where the mass flow w_{tv} results from the states in the adjacent receivers and the effective flow area. The effective flow area is the area the fluid flows through corrected by a discharge coefficient and it depends in a nonlinear way on the opening angle of the throttle valve α_{tv} [Eriksson & Nielsen 2014, Section 7.3]. This opening angle is controlled by an electric machine and influenced by return springs of different stiffness, friction, and aging processes. A possible approach for controlling α_{tv} is, e. g., adaptive control, cf. [Thiel et al. 2016]. Note that the signal u_{tv} is the reference position for the throttle valve opening angle α_{tv} and in this thesis it is normalized so that $u_{tv} = 0$ represents a closed (minimal opened) and $u_{tv} = 1$ an entirely opened throttle valve.

The cylinder including the inlet and exhaust valve can be modeled by a volumetric pump [Eriksson & Nielsen 2014, Section 7.4]. In general, the volumetric pump can be approximated by a static nonlinear function f_{cyl}. The outputs of this nonlinear function are the masses of fresh air (air charge) m_{fa}[1)] and of residual gas m_{rg} in the cylinder, which can be converted into mass flows using the engine speed n_{eng}. In addition, a variable valve train with variable valve phasing is being considered with which the closing/opening of the exhaust/inlet valve can be adjusted to early or late, cf. [Eriksson & Nielsen 2014, Fig. 12.2]. With the inlet valve timing, the amount of fresh air in the cylinder is influenced. The valve overlapping influences the residual gases in the cylinder. Similar to the throttle valve, the positions of the inlet and

[1)] Note that a relative air charge is often used, which in this thesis is also referred to as m_{fa} and is given in percent as the throttle valve position u_{tv}. The relative air charge is obtained by relating the fresh air mass to a nominal mass that results from the ideal gas law using the piston displacement volume, pressure $p_0 = 1013\,\text{mbar}$, and temperature $T_0 = 273\,\text{K}$.

exhaust valve must also be controlled, where u_{iv} and u_{ev} in Fig. 5.1 are reference angle positions for α_{iv} and α_{ev}, respectively. Additionally to the fresh air and the residual gas in the cylinder, fuel is injected directly into the cylinder, assuming that complete (stoichiometric) combustion is achieved. Thus, in the cylinder is as much oxygen as required for a complete reaction with all of the fuel. Consequently, modeling the amount of fresh air mass is essential and errors cause increased raw emissions and reduced engine efficiency.

The model of the turbocharger can be divided into three parts: the (radial) turbine, whose output power P_{t} drives the turbocharger, the (radial) compressor, which consumes the power P_{c}, and the mechanics of the turbocharger, where the speed n_{tc} is determined by the power balance including losses. The turbine is either equipped with a wastegate or with a variable-geometry turbocharger (VGT) to control the turbocharger speed and thus the boost manifold pressure. The wastegate is a valve that can be opened such that (a part of the mass flow) bypasses the turbine, hence the output power of the turbine and thus the turbocharger speed can be reduced. A VGT is equipped with guiding vanes to change the flow characteristics of the turbine, with a similar effect as the wastegate, see [Eriksson & Nielsen 2014, Section 8.7]. Similar to the throttle valve, the normalized position of the VGT guide vanes, respectively the opening of the wastegate valve, α_{t} is controlled with an electric machine as actuator, where u_{t} is the normalized reference position. In contrast to the throttle valve, the reference $u_{\mathrm{t}} = 0$ means open and $u_{\mathrm{t}} = 1$ closed guide vanes, respectively open and closed wastegate valve. With this definition, the system gain becomes positive because increasing u_{t}, respectively closing the guide vanes or the wastegate, results in an increase of the turbocharger speed n_{tc} and the boost manifold pressure p_{bm}.

To summarize the modeling approach of the air path, the system states boost manifold pressure p_{bm}, intake manifold pressure p_{im}, exhaust manifold pressure p_{em}, and the turbocharger speed n_{tc}, can be written in a compact form as a continuous-time nonlinear system with

$$\dot{p}_{\mathrm{bm}} = f_{\mathrm{bm}}(p_{\mathrm{bm}}, p_{\mathrm{im}}, n_{\mathrm{tc}}, \alpha_{\mathrm{tv}}) \tag{5.1a}$$
$$\dot{p}_{\mathrm{im}} = f_{\mathrm{im}}(p_{\mathrm{bm}}, p_{\mathrm{im}}, p_{\mathrm{em}}, n_{\mathrm{eng}}, \alpha_{\mathrm{tv}}, \alpha_{\mathrm{iv}}, \alpha_{\mathrm{ev}}) \tag{5.1b}$$
$$\dot{p}_{\mathrm{em}} = f_{\mathrm{em}}(p_{\mathrm{im}}, p_{\mathrm{em}}, n_{\mathrm{eng}}, n_{\mathrm{tc}}, \alpha_{\mathrm{iv}}, \alpha_{\mathrm{ev}}, \alpha_{\mathrm{t}}) \tag{5.1c}$$
$$\dot{n}_{\mathrm{tc}} = f_{\mathrm{tc}}(p_{\mathrm{bm}}, p_{\mathrm{em}}, n_{\mathrm{tc}}, \alpha_{\mathrm{t}}) \ , \tag{5.1d}$$

where, at the example of the boost manifold, $\dot{p}_{\mathrm{bm}}$ is the time derivative of p_{bm} and f_{bm} is a nonlinear, static function. Note that the system complexity is reduced by assuming constant temperatures in the receivers. Actuator dynamics must be added, if they are not sufficiently fast, i. e. if the dynamics of the closed-loop reference tracking is in the same range as the dynamics in (5.1) or if they are even slower. The orders of the actuator dynamics from u_{tv}, u_{iv}, u_{ev}, u_{t} to α_{tv}, α_{iv}, α_{ev}, α_{t}, respectively, depend on the corresponding controllers. A simple approach is to approximate them by LTI models of second order.

The model coefficients of the functions in (5.1) are adjusted based on stationary measurements. To obtain this model the following common simplifications are made: adiabatic and isotherm processes, ideal gas, and no friction, cf. [Guzzella & Onder 2010; Isermann 2014; Kolmanovsky et al. 1999; Wahlström & Eriksson 2011]. In order to design and implement an LPV controller, this continuous-time nonlinear model must be converted into an LPV model and discretization must be performed (before or after the control design). This can cause many approximation errors, which is why in this thesis the identification approach is chosen and the model is only used to gain insights about the system.

5.2 Identification of the Turbocharged SI Engine

Based on the derived air path model extensive simulations are performed to investigate the system behavior and DT LPV model identification techniques. One of the main topics is a sophisticated system excitation with which the measurement data contain a lot of information about the system behavior and at the same time damage of components is avoided, for example through too high intake pressures, exhaust gas temperatures or turbocharger speeds. In addition, an unfavorable combination of actuator positions leads to misfiring, knocking, or even stalls the engine.

In the next sections, the identification of two subsystems of the air path is investigated, which is based on the identification results in [Bussa 2016; Kettlitz 2018]. Firstly, a MIMO system is considered, where the efficiency of LPV model identification and order reduction techniques is demonstrated. Secondly, a SISO system is considered with input u_t and output p_bm. The separation in these two subsystems makes sense since the turbocharged engine operates in suction operation at partial load or in charged operation at high or full load. In suction operation, actuating the turbine is not required and, in addition, has only minor effects. In charged operation, on the other side, the throttle valve is completely opened and the pressure in the intake path is controlled by the turbocharger. Note that in the following the signals are scaled for the identification such that they vary in a similar range and they are labeled by "sc" in the index. The scaling is not only useful for numerical reasons, but also from control design perspective, e. g., to compare gains in the different input-output paths easily and also from identification perspective, e. g., to weight errors in each output similarly.

In both cases, a sampling time of $T_\mathrm{s} = 10\,\mathrm{ms}$ is used that represents a good trade-off. On the one hand, frequency analysis of the input and output signals have shown, that this sampling time is sufficient to capture the characteristic system behavior. On the other hand, the discretization errors are assumed small and this sampling time results in a number of data samples that is still manageable for the LPV SID method. Moreover, in the engine control unit (ECU) only dedicated time synchronous sampling times are available, e. g., 1 ms, 10 ms, and 100 ms. Further, from the control point of view, a faster sampling time may not be manageable since this requires efficient and low load computations to achieve real-time requirements on the ECU.

5.2.1 LPV MIMO Identification in Suction Operation

In this section, the identification procedure for DT LPV MIMO models with affine scheduling parameter dependency for the air path in suction operation is presented. Inputs are the reference positions for the throttle and inlet valve u_tv and u_iv, respectively, and outputs are the intake manifold pressure p_im and mass of fresh air in the cylinder m_fa. Note that the output p_im can be measured, while the output m_fa is the output of the cylinder model. Thus, the nonlinear function f_cyl is required. The engine speed n_eng varies during the identification experiment due to the excitation and the other actuators of the air path, i. e. the exhaust valve u_ev and the VGT actuator u_t, are kept constant.

Identification Data Set

First, the design of the input signals is considered. For this purpose, the steady-state behavior is investigated at an operating point at the beginning. As simulation experiments indicate, the outputs are affected only by small throttle valve angles, which is shown in Fig. 5.2 because a

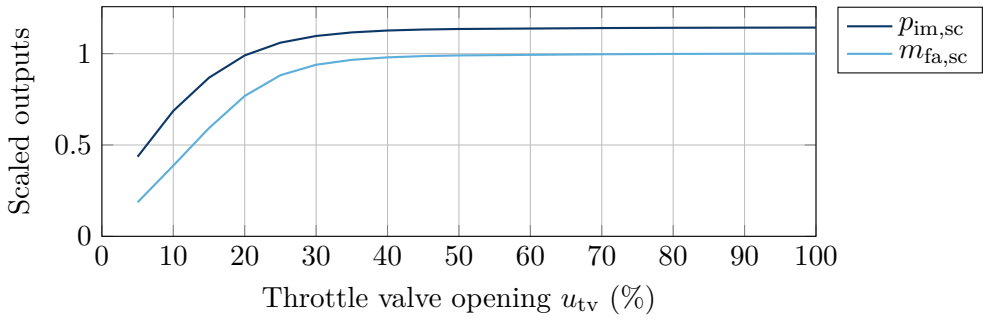


Figure 5.2: Steady-state behavior of both outputs over throttle valve position for fixed engine speed and inlet valve position

small angle is sufficient to provide the maximal intake manifold pressure that is limited from above by the ambiance pressure (the engine is in suction operation). Both outputs show a similar steady-state behavior since they are coupled by the mass flow through the air path. Moreover, this steady-state mass flow in the air path that is influenced by, e. g., the engine speed affects the input-output steady-state behavior, since higher mass flows are more sensitive to a reduced throttling area. Consequently, exciting the system with a uniform distributed random signal[2)] on the throttle valve results in a poor data set, because the outputs are mainly constant. A simple approach to circumvent this problem is to fit a (surjective) function to the steady-state measurement data and to feed its (right) inverse with the (random) excitation signal to obtain a proper input signal. This is also illustrated in Fig. 5.4, where the randomly distributed excitation signals $u_{\mathrm{r,tv}}$ and $u_{\mathrm{r,iv}}$ are scaled with a static input nonlinearity that approximates the inverse steady-state behavior to eventually obtain the input signals u_{tv} and u_{iv} used for identification. Note that then the random signal can be seen as steady-state reference values. Although the behavior in Fig. 5.2 depends on the engine speed, it is possible to simplify the problem and neglect engine speed when the steady-state function is fitted such that only the (averaged) characteristic behavior is represented by the steady-state function.

Additionally, the local system behavior is analyzed by using small excitation signals around different mean values for u_{tv}. From these experiments, a nonlinear behavior is concluded because the local linearized models vary significantly with the mean value.

Similar experiments are conducted for the inlet valve opening angle and it can be concluded that the throttle valve essentially allows the setting of the operating point in the outputs around which small changes are possible with the inlet valve opening angle. [Bussa 2016, Section 4]

To protect the components further limitations in the input signal must be considered. For example, the rate of the throttle valve angle is limited, if the angle is small. To construct the inlet valve excitation, a low frequency signal of high amplitude and a high frequency random signal of low amplitude with a sampling rate of 0.5 s are added. Thus, the inlet valves are (slowly) driven through their entire operating range and (dynamically) exciting the system. [Bussa 2016, Section 5]

The resulting input, output and scheduling signals are shown in Fig. 5.3. As can be seen, both outputs vary similarly and they are dominated by the throttle valve. However, the air charge m_{fa} is affected by the inlet valve that can be seen in the time range 112 s to 116 s, where

[2)] A random signal is commonly denoted as persistently exciting because it contains the entire frequency spectrum.

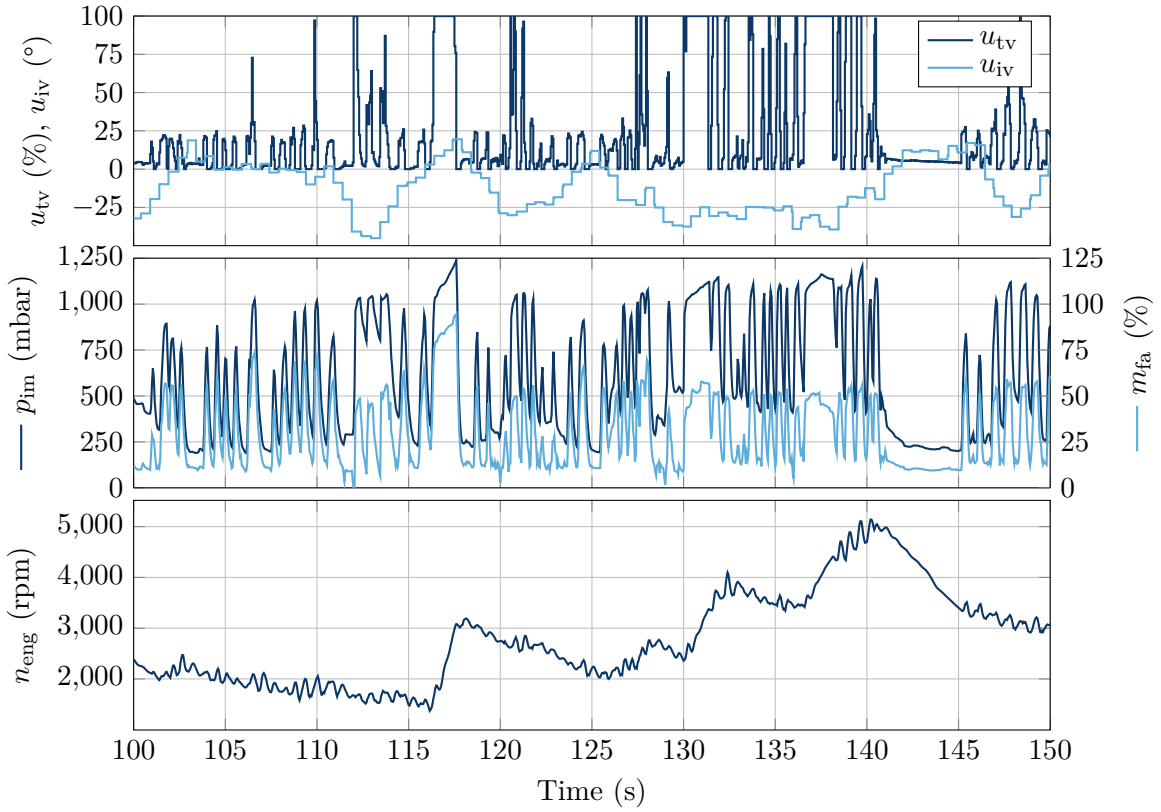


Figure 5.3: Segment of input, output, and engine speed signals for the identification of the engine in suction operation

the inlet valve increases and the maximal values of m_{fa} also increase, while the maximal values of p_{im} remain constant (depending on the throttle valve opening). It can be seen from Fig. 5.3 that the outputs are not very noisy, which is why noise has less influence than a modeling error.

The measurement is partitioned in two data sets of equal length: one for estimation and one for validation. Moreover, a second measurement, using similar input signals, is also divided in two data sets, where both are used for validation. A total of four data sets are thus available.

Scheduling Parameters

Next, the scheduling parameters are discussed since they are not known a priori but must be provided to the LPV identification algorithm. One approach is to derive the scheduling parameters and their functional dependence from the nonlinear model. As can be deduced from (5.1), all the states of the air path, angles of the actuators, and the engine speed n_{eng} are candidates for scheduling parameters. Since some actuators are kept constant and some states have a negligible effect in suction operation, the number of candidates for the scheduling parameters can be reduced. Consequently, the following candidates remain: p_{im}, m_{fa}, α_{tv}, α_{iv}, and n_{eng}. Extensive studies have shown that instead of the actual actuator positions, the reference positions can be used as scheduling parameters, i. e. u_{tv} and u_{iv}. Based on this scheduling parameter vector $\rho^{\mathrm{T}} = [n_{\mathrm{eng}} \quad p_{\mathrm{im}} \quad u_{\mathrm{tv}} \quad u_{\mathrm{iv}}]$, further investigations are carried out to find the affine scheduling parameter vector, i. e. also the functional dependence on the

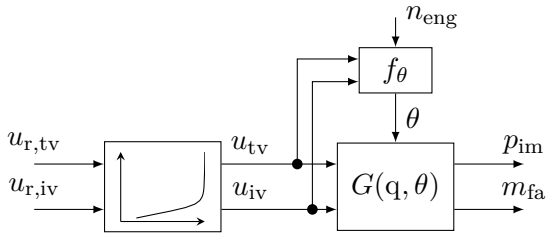


Figure 5.4: Inputs, outputs, and affine scheduling parameter dependence for the identification of the air path MIMO subsystem in suction operation with static input nonlinearity to obtain the inputs u_{tv} and u_{iv} from excitation signals $u_{\text{r,tv}}$ and $u_{\text{r,iv}}$ with randomly distributed values

Table 5.1: Scaling coefficients for the signals used in the identification of the engine in suction operation with $x_{\text{sc}} = (x - x_{\text{off}})x_{\text{rng}}^{-1}$ for the signal x (sc: scaled, off: offset, rng: range)

Signal	Offset	Range
u_{tv}	0 %	50 %
u_{iv}	0°	50°
p_{im}	0 mbar	1000 mbar
m_{fa}	0 %	100 %
n_{eng}	0 rpm	1000 rpm

scheduling parameters ρ. For this purpose, two LPV model structures (IO ARX in SF and SS in predictor form) are chosen where only the affine scheduling parameter vector can vary. Then, affine scheduling parameter vectors θ were formed based on different combinations of scheduling parameter candidates (entries in ρ) along with different basis functions, see [Bussa 2016]. Based on the identification results, the affine scheduling vector $\theta = [n_{\text{eng}} \quad u_{\text{tv}} \quad u_{\text{iv}}]$ provides good results and thus is used from here on in this work. Based on the measurement data, the parameter set is given as $\mathbb{P}_\theta = [1000\,\text{rpm}, 5000\,\text{rpm}] \times [0\,\%, 100\,\%] \times [-45°, 20°]$.

Note that the affine scheduling parameter vector θ contains both inputs, hence quasi LPV models are identified, see Fig. 5.4. In addition, the LPV IO models are not identifiable, see Remark 3.2. The reason for this is that the regressor, which consists of multiplications of the affine scheduling parameters with the inputs and outputs, contains multiple times the product $u_{\text{tv}}u_{\text{iv}}$, in the case of the SF there are $2\min(n_a, n_b)$ redundant entries due to the time shifts. However, identifiability can be regained for this affine scheduling parameter vector, if the repetitions of the (possibly time shifted) product $u_{\text{tv}}u_{\text{iv}}$ are removed from the regressor and accordingly the entries in the coefficients of the $B(\text{q},\theta)$ polynomial.

Identification Results

In the following, the identification results are shown. It is assumed that the data sets contain a sufficient system excitation for the identification. Subsequently, the effectiveness of the order reduction and the nonlinear optimization are shown. Before the LPV identification techniques are used, the signals are scaled such that they are dimensionless and vary in a comparable range. The corresponding scaling coefficients are given in Table 5.1.

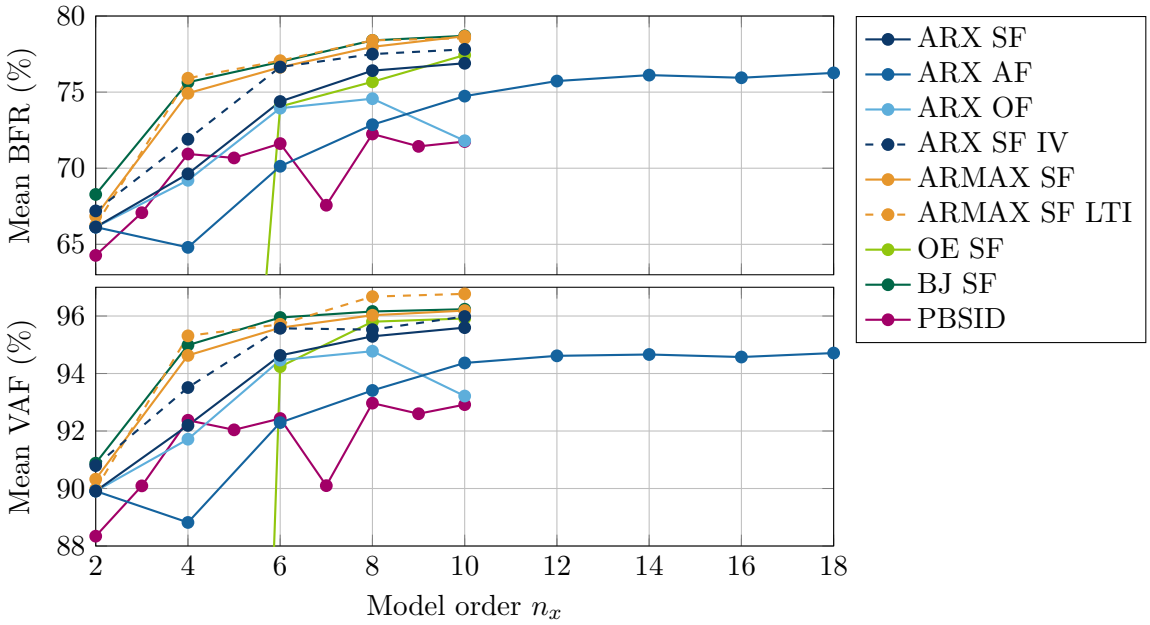


Figure 5.5: Identification results on the air path MIMO system with averaged fit for both outputs and all data sets

Due to the unknown model order, LPV models[3)] of different order are identified. From the modeling point of view, the system order of the air path is two, see (5.1) with neglected p_{bm} and n_{tc}, plus the order of the controlled actuators. Hence, assuming that the actuator dynamics are relevant and of second order, the overall system is of sixth order. With this in mind, in the following, the model order is determined in a data-based approach by identifying models of different order and analyzing their accuracy with respect to the data sets. This has the advantage that the influence of the order on the model accuracy is directly visible instead of interpreting intermediate values as for example in [He & Asada 1993] and [Verdult 2002, Section 3.2]. The averaged model accuracy over the model order is shown in Fig. 5.5. On the ordinates, the mean BFR and VAF are plotted, where again mean indicates that the mean value of the two outputs for the four data sets (including the estimation data set) is calculated. All models, with a few exceptions, have an accuracy in the BFR of 65 % to 80 % and in the VAF of 88 % to 97 %. This shows that they can reproduce the dynamic behavior in the data sets very well while there is a slight lack of stationary accuracy (high VAF and lower BFR value). However, the qualitative variation of both accuracy criteria is similar. Based on this, only the BFR is considered in the following evaluations.

In Fig. 5.5, the DT LPV IO identification method for the ARX model structure is employed for the three dynamic scheduling parameter dependencies, i. e. the SF, the AF, and the OF. The SF provides the best results regarding the fit and the model order, where the OF performs very similar to the SF despite the model restrictions in the OF, which affect the accuracy only at higher orders, see the deviations to the SF starting at order $n_x = 8$. The AF has

[3)] Note that in the following identified DT LPV SS models are denoted as ARX models or PBSID models, for example. In both cases, the LPV models are labeled by the identification approach to distinguish the models. Thus, an ARX model is obtained via IO identification of a DT LPV IO model in ARX structure while a PBSID model is obtained via subspace identification of a DT LPV SS model with innovation type noise.

inherently a higher order in the SS representation, hence comparable fits to the SF and OF are achieved at higher state order. However, in the IO domain the orders of the polynomials are comparable for all three forms, where $(n_a, n_b) \leq 5$. Using the IV method for the LPV ARX model in SF improves the model accuracy nearly to the level of the ARMAX model structure. This more advanced structure is considered only for the SF and the additional model freedom allows a better fit, especially for low-order models. Moreover, despite the more restrictive model structure, the identification of LPV models in ARMAX structure with an LTI noise polynomial $C(\mathrm{q})$ using off-the-shelf LTI algorithms (Algorithm 3) provides slightly better fits in comparison to Algorithm 4 that allows an LPV C polynomial. The introduced Algorithm 5 for the identification of LPV MIMO models in BJ structure provides models with slightly better accuracy than Algorithm 4 for ARMAX models, although Algorithm 5 is biased. OE models can be also identified with Algorithm 5, where the noise model estimation steps are skipped. However, the OE models are of good accuracy only from model order $n_x = 6$. The DT LPV SS models identified with the PBSID toolbox, which are denoted by PBSID in Fig. 5.5, achieve fits in a similar range as the IO methods. Note that these fits are obtained by taking the best fit of the following three cases: firstly, all matrices A, B and K are assumed as scheduling parameter dependent, secondly, B is assumed as constant, and, thirdly, K is assumed as constant, where in the latter two cases, the other two matrices are allowed to be scheduling parameter dependent. However, the resulting fit with PBSID is worse than for the IO ARX model in SF, except for model order $n_x = 4$. Note that the identification of LTI models[4)] result in models with a BFR and a VAF of approximately 15 % and 65 %, respectively. Accordingly, modeling the system with LTI approaches is insufficient and an extension to LPV systems enables a significantly better model quality based on the identification results in Fig. 5.5.

In conclusion, a model order of four or six seems to be appropriate. Especially for more elaborated model structures than the ARX structure, the accuracy increases only slightly from the fourth order.

The model orders of the SS representations of the identified LPV IO models are a multiple of the number of outputs, see Section 2.4.1. Thus, the orders of the IO models for this MIMO system are a multiple of two, cf. Fig. 5.5. In contrast, PBSID provides models for any state number $n_x \leq 10$ (due to the choice of $(p, f) \leq 5$) and not only for even orders, however usually with a better fit for an even number of states, see Fig. 5.5. To obtain also IO models for any state number less than ten, model order reduction is performed on the IO models. For the reduction, Algorithm 7 is used for the quick Ho-Kalman scheme and an LMI based approach is used to calculate the generalized Gramians, see Theorems 2.2 and 2.3. In both cases, scheduling parameter independent transformation matrices are calculated as (approximate) solution for a transformation in the balanced realization.

From the model order reduction results, it can be stated for this application example that the Ho-Kalman approach often provides models with better accuracy than using the LMI-Gramian based approach. For example, when reducing from the identified ARX models (dark blue line in Fig. 5.5) with order greater than seven to an LPV model with order $n_x = 7$, a BFR of 76.3 % is achieved with the Ho-Kalman approach, whereas only 74.8 % BFR can be achieved with the LMI-Gramian based approach. When reducing to model order $n_x = 5$, however, both approaches yield models with similar BFR: 74.5 % with the Ho-Kalman and 74.3 % with the LMI-Gramian based approach. In addition to the tendency for the Ho-Kalman approach to achieve a better BFR, the calculation is also faster compared to the LMI-Gramian based

[4)] The `tfest` command in the MATLAB system identification toolbox is used for the LTI model identification.

order reduction. The average computation time for the order reduction is 0.01 s for the quick Ho-Kalman scheme and 6.0 s for the LMI-Gramian approach[5)]. However, it should be pointed out that due to, among other things, the restrictive approach of a constant transformation matrix, order reduction is not always successful and models with poor accuracy can be obtained as well (with both reduction approaches). Comparing to LPV subspace identification, the IO methods with order reduction lead to better fits than the LPV PBSID method, if order reduction is successful, i. e. the reduced-order model has a high BFR.

Next, possible reasons why model reduction is not successful are discussed. For the LMI-Gramian based approach, the explanation is a bit simpler, therefore it is started with this method. Often no or only a conservative solution for the generalized Gramian matrices can be found with the chosen approach. Since the transformation matrix should be scheduling parameter independent, the generalized observability and controllability Gramians X_o and X_c, respectively, must also be constant. This in turn means that there exists at least a constant Lyapunov matrix P with which quadratic stability of the LPV system is shown. For example, the identified LPV ARX models in SF are quadratically stable for orders $n_x \leq 8$, where also reduced-order models with the LMI-Gramian approach show good accuracy, while for the model of order $n_x = 10$ no constant Lyapunov matrix can be found. Consequently, the order reduction with the LMI-Gramian approach fails, too. However, the LPV models show non-divergent behavior in the simulations, even if no constant Lyapunov matrix can be found[6)]. On the other hand, even if a Lyapunov matrix exists, the generalized Gramians are subject to stricter conditions. Therefore, either no scheduling parameter independent generalized Gramians exist or the determined matrices X_o and X_c are conservative, see also Example 2.3 for an illustrative example for the conservatism. These problems in the search for X_o and X_c can be reduced by scheduling parameter dependent generalized Gramians, but then the scheduling parameter dependence on the reduced LPV system becomes more complicated, see Section 2.4.2.

In the following, the reduction with the Ho-Kalman approach for this practical example is examined in more detail. For this purpose, the ARX SF reduced-order model of order $n_x = 5$ is considered, which has a BFR of 74.5 %. The singular values of the Hankel matrix $\mathcal{H}_{i,j}$[7)], computed for different i and j are shown in Fig. 5.6, where for illustrative purposes $i = j$. Note that i and j are chosen as 200 to obtain the reduced LPV fifth-order model from the identified LPV ARX SF sixth-order model. In addition, for comparison purposes, the Hankel singular values obtained from the LMI-Gramian based approach are also plotted (dashed lines), which are independent of i and j, thus constant. As can be seen in Fig. 5.6, an i and j of at least 40 is needed so that the Hankel singular values are approximately converged (note that the largest Hankel singular value still increases from $i = j = 40$ to $i = j = 100$ by more than 22 % while the increase from $i = j = 100$ to $i = j = 200$ is less than 3 %). It should be pointed out that for commercially available computers such large values for i and j lead to computational problems (memory requirements and computational load) for the three existing

[5)] The computation time for the LMI-Gramian approach is considerably longer, if the time for the reduced-order models is considered that have a poor accuracy, since here only a conservative solution can be found (if at all) and the algorithm terminates after several iterations. Usually, only few iterations are necessary for those reduced-order models with a high BFR.

[6)] That no constant Lyapunov matrix can be found does not necessarily mean that the model is unstable. The model can still be stable, but the approach with a constant Lyapunov matrix can be too conservative for a stability proof.

[7)] The singular values are calculated via an SVD of the product $R_\mathcal{O} R_\mathcal{R}^\mathrm{T}$ instead of the Hankel matrix $\mathcal{H}_{i,j}$, see Section 4.3.2.

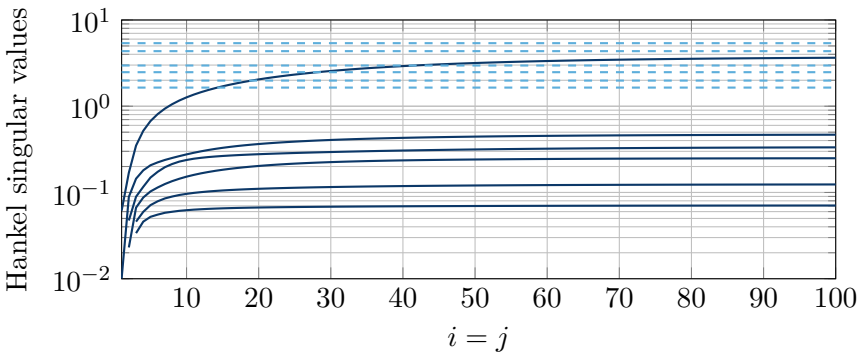


Figure 5.6: Hankel singular values of both model order reduction techniques (solid: Ho-Kalman, dashed: LMI-Gramian) over $i = j$ for the identified LPV ARX SF model of order six for the air path MIMO system

approaches: the classical LPV Ho-Kalman order reduction scheme presented in [Tóth, Abbas, et al. 2012], the fast Ho-Kalman scheme in [Schulz et al. 2017], and the bases reduced scheme in [Cox et al. 2015]. In this practical example with two inputs, two outputs, and three affine scheduling parameters, the Ho-Kalman reduction scheme in [Tóth, Abbas, et al. 2012] and its fast version in [Schulz et al. 2017] are limited to i and j less than 15. Using such small values for i and j results in reduced-order models with a poor accuracy (BFR), which is why the computational simplifications for the Ho-Kalman approach presented in this thesis are essential for the practical application of LPV model order reduction using the Ho-Kalman approach. Although the calculated Hankel singular values of both reduction approaches differ, the order reduction is successful with both techniques and models of similar accuracy are obtained (a BFR of 74.5 % with the Ho-Kalman approach and a BFR of 74.3 % with the LMI-Gramian based approach).

In a final step of system identification, the models can be refined with nonlinear optimization, cf. Section 3.6. For this purpose, the SS representations of the (reduced-order) LPV IO SF models based on the ARX and ARMAX structure are selected as well as the LPV SS models from SID. The two optimization algorithms (gradient-based and expectation maximization) presented in [Cox 2018, Chapter 7] are used, which are briefly reviewed in Section 3.6. The results with the expectation maximization algorithm and the settings made there are not as good in the BFR as with the gradient-based method. Therefore, only the results of the gradient-based method are shown in Fig. 5.7 for two selected models, see the diamonds in Fig. 5.7. For the IO model, the reduced-order model (square in Fig. 5.7) served as the starting value for the optimization.

First, it can be stated that the BFR (and the VAF that is not shown) is significantly improved with the optimization. However, the best BFR is achieved with the optimized PBSID models with nearly 85 % BFR. In conclusion, LPV models of order four to six can be used for simulation or control design since they have a similar model accuracy. For implementation reasons low-order controllers are preferable, hence the models of order four are appealing, since usually full-order LPV controllers are designed, i. e. their order is the order of the model plus the orders of the dynamic weights.

During the optimization it could be noticed that firstly the computation time for the optimization takes considerably more time than the identification and, secondly, it increases

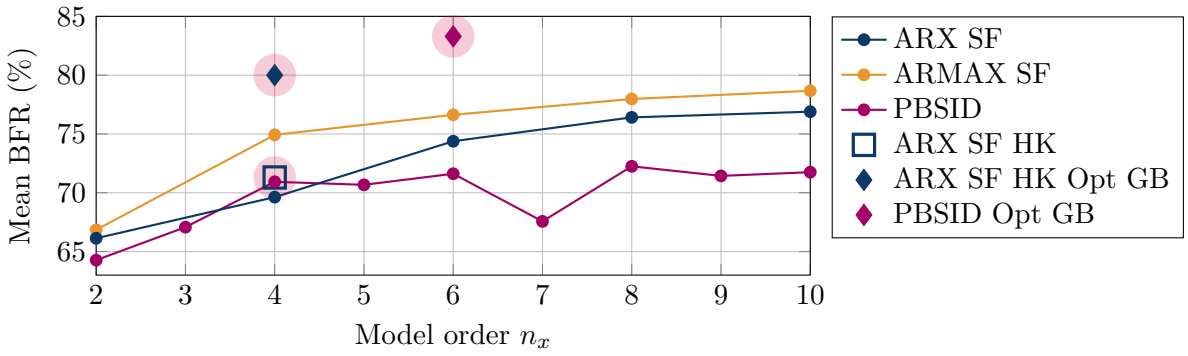


Figure 5.7: Identification results on the air path MIMO system with order reduction and optimization and averaged BFR for both outputs and all data sets (HK: reduced-order model using the quick Ho-Kalman scheme, Opt GB: optimized model using the gradient-based algorithm)

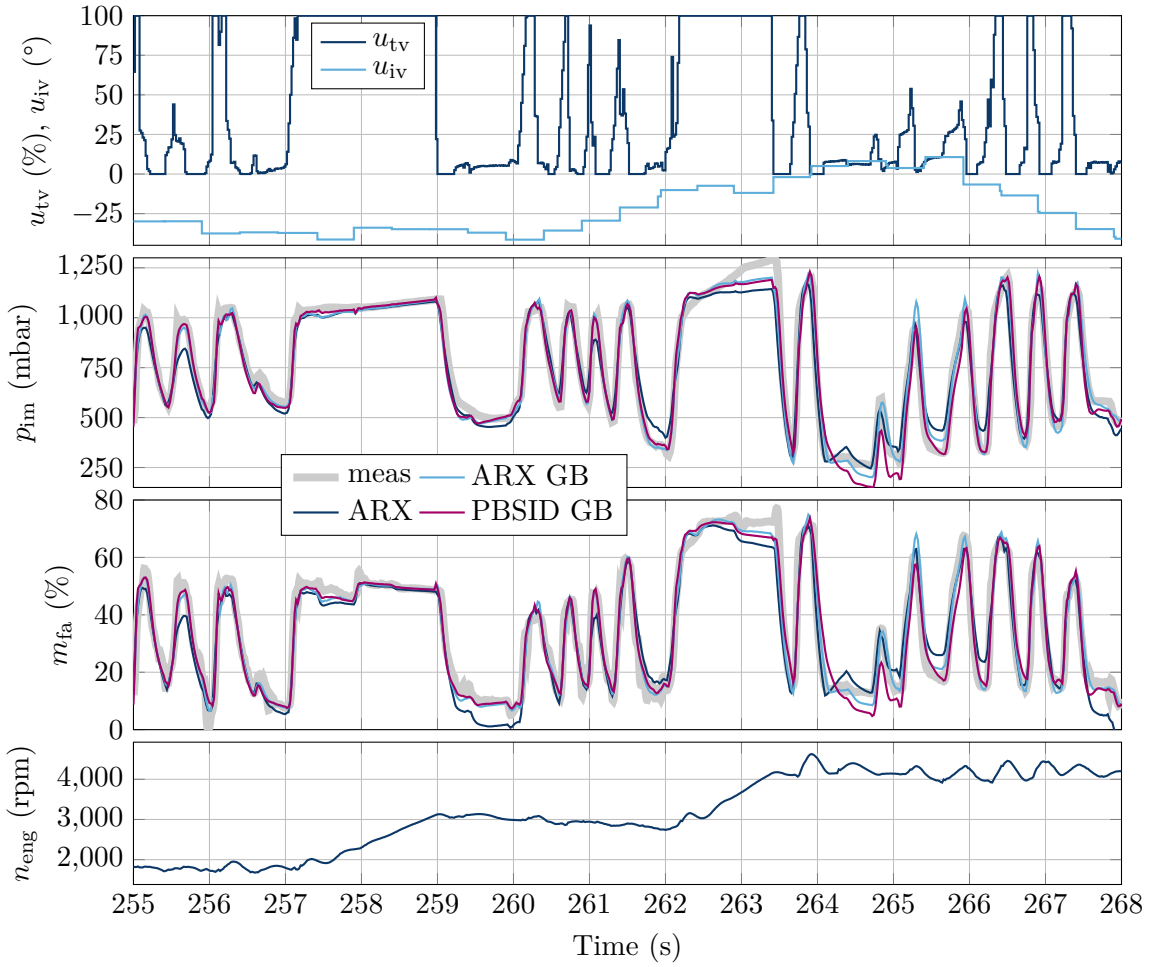


Figure 5.8: Segment of input, measured and modeled output, and engine speed signals for the validation on the air path MIMO system (meas: measured output, GB: optimized models using the gradient-based algorithm)

with the model order. The optimization time for an LPV model of order two is less than five minutes and takes about an hour for models of order ten. The increase of the calculation time corresponds very well to a quadratic function of the model order, which makes sense because the number of optimization parameters increases in the state matrix with n_x^2. Despite the long computation time, the optimization algorithm must be credited with finding a model with an improved BFR, in some cases even for poor initial models.

Finally, the outputs of selected models for a validation data set are shown in Fig. 5.8. For this purpose, the models highlighted by light red circles in Fig. 5.7 are considered. The two ARX models are of fourth order and the PBSID model is of sixth order. Thus, a direct comparison before and after the optimization is possible as well as the comparison with a model where the model order and the identification approach are different. It can be seen that the ARX model obtained with the Ho-Kalman order reduction scheme shows a good fit in the dynamic behavior but deviations in the steady-state accuracy, see 262 s to 263.5 s in Fig. 5.8. With the optimized ARX model, the dynamic behavior is improved even further and the steady-state deviations are also reduced, see 261 s to 262 s and 262 s to 263.5 s in Fig. 5.8. In comparison to the optimized ARX model, the sixth-order PBSID model, which is obtained from nonlinear optimization, provides a very similar match with the measurement.

In conclusion, the identified LPV models perform very well on this nonlinear MIMO real-world air path example of a turbocharged SI engine. However, improvements in the reliability of order reduction are desirable.

5.2.2 LPV SISO Identification in Charged Operation

In this section, the identification procedure for DT LPV SISO models with affine scheduling parameter dependency for the air path in charged operation is presented. Here, a different engine is investigated that is equipped with a wastegate at the turbine instead of a VGT. Thus, the input is the reference position for the wastegate $u_{\mathrm{wg}} = u_{\mathrm{t}}$ and the output is the measured boost manifold pressure p_{bm}. The engine speed n_{eng} is varied and the other actuators of the air path, i. e. the throttle, inlet, and exhaust valve u_{tv}, u_{iv}, and u_{ev} are kept constant. Moreover, the actuators are in a reference position with which a maximal air charge is achieved, hence the throttle valve is, e. g., entirely opened, so $u_{\mathrm{tv}} = 100\,\%$. The general identification procedure is similar to the previous section.

Identification Data Set

It is again difficult to design an excitation signal that both excites the system dynamics and covers the operating range. Especially with high mass flows at the turbine, respectively at high engine speeds, closing the wastegate, i. e. $u_{\mathrm{wg}} = 100\,\%$, can lead to too high turbocharger speed or too high boost manifold pressure, for which the components are not designed and can break. To obtain a suitable excitation signal, the right inverse of the surjective function for the steady-state system behavior can be used again as in the previous example and thus a standard excitation signal can be modified by this right inverse. The steady-state behavior is shown in Fig. 5.9 and the required steady-state actuator position u_{wg} depends on the engine speed and the boost manifold pressure itself. It can also be seen in Fig. 5.9 that (in the stationary case) the sensitivity of the output p_{bm} on the input u_{wg} depends on the operating point, i. e. the engine speed and the boost manifold pressure. At low engine speeds, for example, very high values for the input u_{wg} are required to adjust the boost pressure (the wastegate is almost

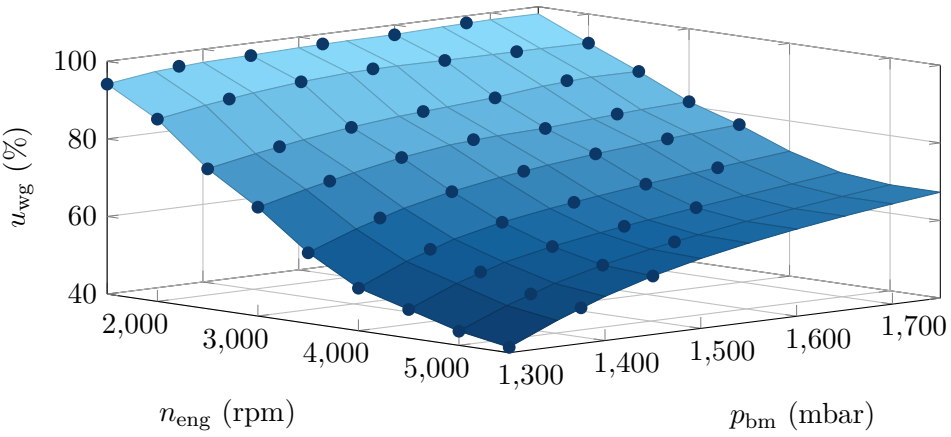


Figure 5.9: Steady-state behavior for the engine in charged operation with wastegate position over engine speed and boost manifold pressure (dots: measured steady-state points)

completely closed) and only small changes in u_{wg} have a large effect on the pressure p_{bm}.

In contrast to the identification setup in Section 5.2.1, this time the subsystem is operated in closed-loop during identification and the boost manifold pressure is controlled by the series production controller on the ECU. Identification in closed-loop has the disadvantage that the input signal correlates with the noise in the output due to the feedback. This leads to a biased estimation, even for IO models in the OE or BJ structure where the system model is estimated independently of the noise model. In [Karimi & Landau 1998], a comparison is made between different approaches and their effect on a closed-loop identification of LTI models. Here, the simple direct method is used, where the measured inputs and outputs of the system are directly taken for the identification. Since the output is barely corrupted by noise, it is assumed that the bias effect is minor because the power spectrum of the excitation signal dominates the one of the noise [Karimi & Landau 1998].

The excitation signal for the wastegate again consists of two components: the signal from the controller plus a known input disturbance in form of a high-frequency PRBS. The boost pressure reference is chosen as a low-frequency signal to cover the entire operating range of the boost pressure and thus to detect the dependency on the boost pressure itself. At the same time, the engine speed is changed, which is controlled by an electric machine. A segment of the resulting input, output and scheduling signals are shown in Fig. 5.10.

Further measurements are carried out for the validation containing similar signals but also, e. g., a stair in the boost pressure reference or large changes in boost pressure and rapid engine speed changes. Thus, again four data sets are available, one estimation data set and three validation data sets.

Scheduling Parameters

From the modeling perspective, candidates for the scheduling parameters are the boost and exhaust manifold pressure p_{bm} and p_{em}, the engine and turbocharger speed n_{eng} and n_{tc}, and the position of the actuator α_{t}, see (5.1). However, due to the physical coupling, some of them are strongly correlated. Thus, different candidates have been tested with the result that the varying system behavior can be captured by the boost manifold pressure and the engine speed, so $\rho^{\mathrm{T}} = [n_{\mathrm{eng}} \quad p_{\mathrm{bm}}]$. The corresponding parameter set is given by the measurement data in the

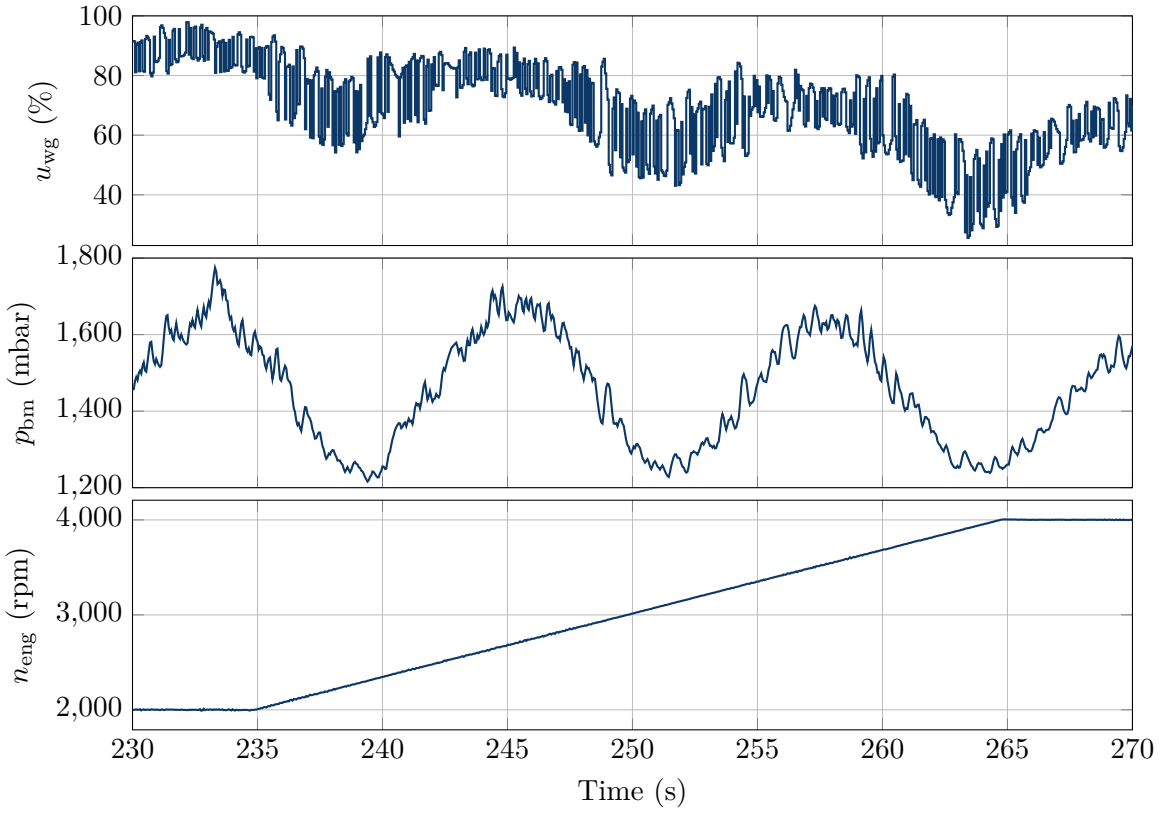


Figure 5.10: Segment of input, output, and engine speed signals for the identification of the engine in charged operation

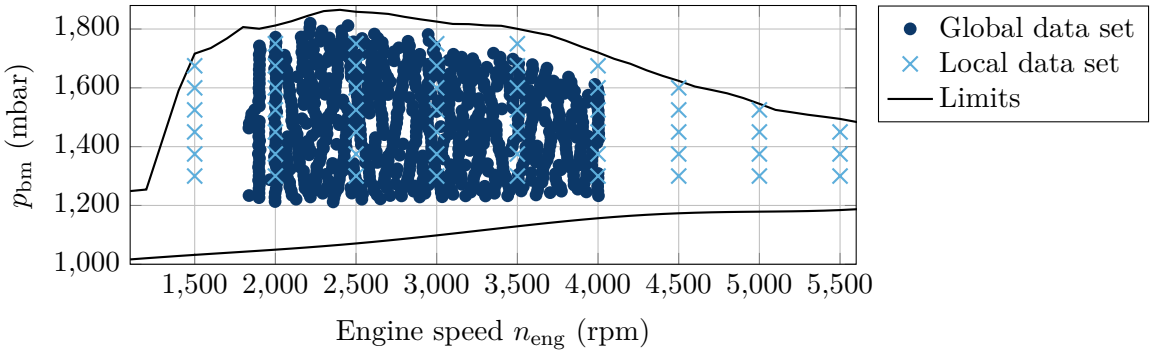


Figure 5.11: Coverage of the parameter set $\mathbb{P}_\rho$ for the identification of the engine in charged operation

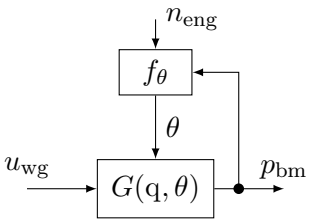


Figure 5.12: Inputs, outputs, and affine scheduling parameter dependence for the identification of the air path SISO subsystem in charged operation

identification data set, hence $\mathbb{P}_\rho \in [1800\,\mathrm{rpm}, 4000\,\mathrm{rpm}] \times [1200\,\mathrm{mbar}, 1800\,\mathrm{mbar}]$. The coverage of the parameter set with the identification data set is shown in Fig. 5.11. In addition to the measurement data, the limits of the minimum and maximum possible boost pressure over the engine speed are plotted. It can be seen that the measurement data cover a wide but not the entire operating range. Furthermore, the system is excited locally at the operating points denoted by the crosses in Fig. 5.11. At each operating point, the system is excited in open-loop using excitation signals with a small amplitude of type PRBS, chirp, and a step sequence.

Based on the general scheduling parameters in ρ, various combinations of scheduling parameters and basis functions are tested to find an appropriate affine scheduling parameter vector. The affine scheduling parameter vector

$$\theta^{\mathrm{T}} = \begin{bmatrix} n_{\mathrm{eng}} & p_{\mathrm{bm}} & p_{\mathrm{bm}}^2 \end{bmatrix} \tag{5.2}$$

provides a good compromise between affine scheduling parameter complexity and LPV model accuracy in initial identification tests. Note that the affine scheduling parameter vector θ contains the output, hence quasi LPV models are identified, see Fig. 5.12.

Identification Results

The identification results are presented in the following, starting with the analysis of the offsets used for scaling the signals and followed by the model order. Again, it is assumed that the data sets contain a sufficient system excitation for the identification in this application example.

The signals are scaled for the identification so that they vary in a similar range, cf. Table 5.1 in Section 5.2.1. Initial identification results using scaled signals that vary approximately in the range from zero to one provide a BFR of 80 % and a VAF around 99 %. This indicates that the output of the identified LPV models contain an offset. Thus, different offsets to scale the input and output are considered, which is motivated by the following idea. Assume that for a chosen equilibrium the steady-state values are taken as offsets for the input and output and assume that the state origin of the LPV system is an attractive equilibrium, hence the state of the system converges to zero as time goes to infinity. When starting from the chosen equilibrium, i. e. the initial state and the input are zero, and only the external scheduling parameter (engine speed in this example) is varied, the output of the LPV model remains zero. In this application example, however, the output changes, i. e. the system moves along an isoline in Fig. 5.9 where u_{wg} remains constant. Consequently, offsets (see Table 5.2) are searched with which the stationary system behavior can be better mapped with the given structure of LPV systems, see Definitions 2.2 and 2.3. Another approach is to assume an unknown disturbance at the input (dynamic offset) and to estimate it, which makes the identification

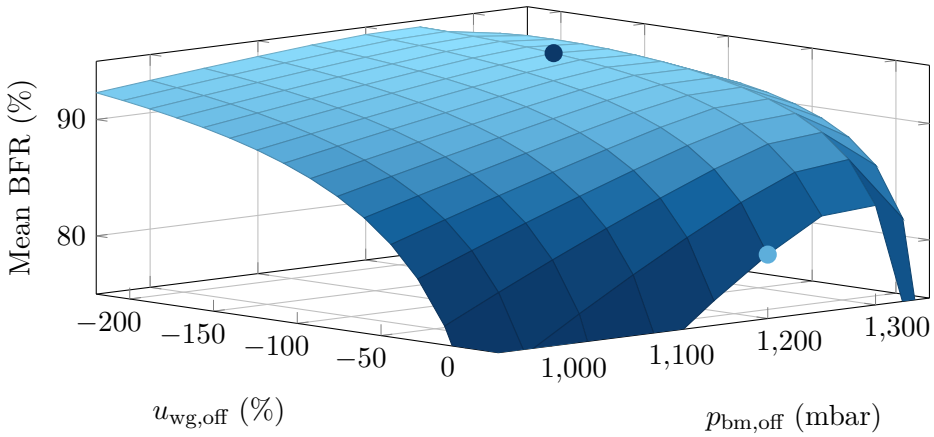


Figure 5.13: Influence of the offsets in the input and output on the identification result of the engine in charged operation (light blue dot: initially chosen offset values, dark blue dot: improved offset values)

problem more complicated[8)]. However, with the simple approach of searching for constant offsets, a satisfactory improvement is achieved, hence the other approach is not pursued further.

To investigate the influence of the offsets on the identification result, the offsets for scaling the input and output data are varied and the BFR of the identified LPV models in the ARX SF with $n_a = 3$ and $n_b = 2$ is examined. The identification results on a grid of offsets are shown in Fig. 5.13 w. r. t. the mean BFR for all four data sets. It can be seen that the offsets have in some range a significant effect on the identification result. Thus, the choice of offsets can be fundamental in LPV system identification. Note that the effect on the VAF is not so pronounced and it only decreases where the BFR also drops sharply (for large values of $u_{\mathrm{wg,off}}$ or $p_{\mathrm{bm,off}}$). Furthermore, the initial offsets (chosen such that the minimal values of the scaled signals are approximately zero) are marked in Fig. 5.13 by a light blue dot and the improved offsets are highlighted by a dark blue dot and they are considered from here on. The improved offsets are not optimal w. r. t. the BFR, but a (minor) further increase of the BFR moves the origin of the scaled input signal $u_{\mathrm{wg,sc}} = 0$ (much) further away from the input operating range. Even for the chosen improved input scaling, a scaled input of $u_{\mathrm{wg,sc}} = 0$ is physically not meaningful. Instead, the scaled input varies in $u_{\mathrm{wg,sc}} \in [2, 3]$. The final scaling coefficients are given in Table 5.2.

Next, models of different order are identified using the scaled signals and the model accuracy is evaluated on the four data sets (estimation and three validation data sets). This results in Fig. 5.14 and it can be seen, that a BFR of 90 % to 95 % can be achieved with the different model structures and identification approaches. Moreover, a very high VAF is achieved of more than 99 % indicating. As the Fig. 5.14 indicates, an improvement from a model order of three can hardly be achieved, which is why models of third or even second order are classified as

[8)] Since in this example the output (boost pressure) depends, among others, on the turbocharger speed or the mass flow over the turbine, one of these two variables can also be used as a disturbance. The turbine mass flow is not known, hence the states in the exhaust manifold (pressure and temperature) can be used under the assumption that the receiver states downstream of the turbine are constant. However, it is unclear whether these disturbances have an additive or multiplicative effect. From the theoretical modeling of the air path it only follows that the states in the exhaust manifold are multiplied by the actuator position to obtain the mass flow, cf. [Daasch et al. 2016, (22) and (23)].

Table 5.2: Scaling coefficients for the signals used in the identification of the engine in charged operation with $x_{\mathrm{sc}} = (x - x_{\mathrm{off}})x_{\mathrm{rng}}^{-1}$ for the signal x (sc: scaled, off: offset, rng: range)

Signal	Offset	Range
u_{wg}	-140 %	80 %
p_{bm}	1250 mbar	500 mbar
n_{eng}	1500 rpm	2500 rpm

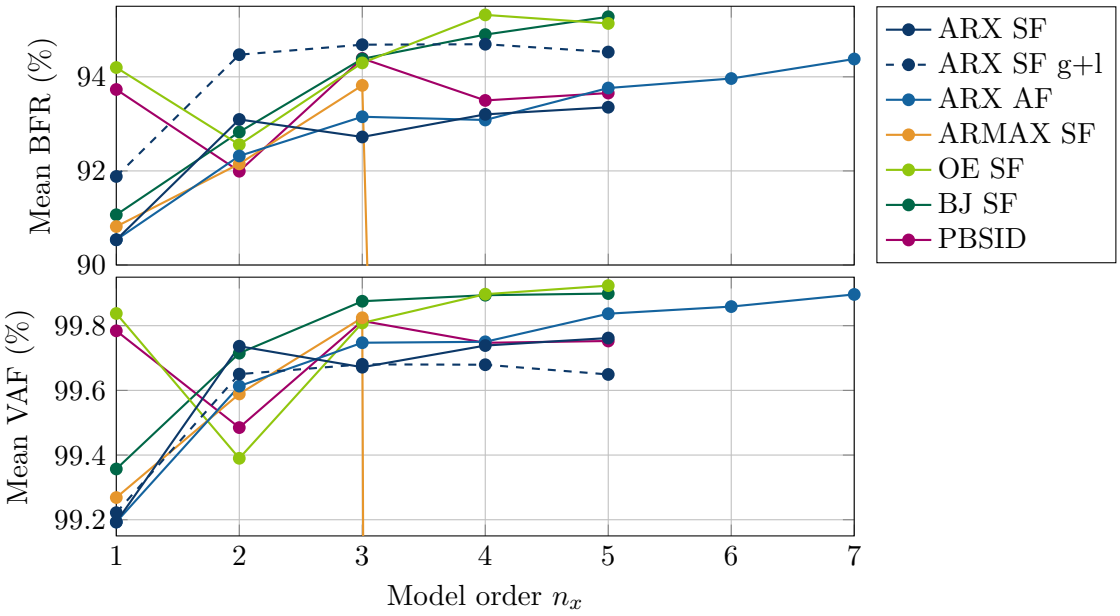


Figure 5.14: Identification results on the air path SISO system with averaged BFR for all data sets

sufficient.

The ARX models in AF have a similar BFR and VAF for model orders up to five, except for model order two. The ARX models in OF and the ARMAX models with an LTI C polynomial provide similar BFRs up to model order two and a poor fit at higher orders, which is why they are not shown here. Using the ARMAX structure with Algorithm 4 (scheduling parameter dependent C polynomial) improves the accuracy at state order $n_x = 3$, however, fails for models of order four and five. The IV method in combination with the ARX model structure and the SF fails again and is not shown. A possible reason for this is that the simulated outputs deviate strongly in an intermediate step and, consequently, the instruments and the fit become poor. In this SISO case, the LPV BJ-RIV algorithm presented in [Laurain et al. 2010] can be used, which, however, only provides a good result for a model of order $n_x = 2$ (BFR of 82.4 % and VAF of 97.2 %). Thus, the newly presented BJ identification approach, see Algorithm 5, seems to be more robust because it provides (BJ and OE models) that achieve the highest BFR. Finally, the PBSID performs for model order of $n_x = 1$ better than the IO identification methods and achieves the best fit at model order three, where the BFR and VAF are comparable to the OE model. Note that also LTI models are estimated, which provide a constant BFR and VAF of 37.5 % and 64.6 %, respectively, from model order $n_x = 2$ onward.

In addition to the global identification measurement, there are also local identification measurements at individual operating points available, as shown in Fig. 5.11. The local data sets can be taken into account during identification. For a more fair comparison with the global data set, only the local models with operating point within the global data set are considered. The extension to include local measurements is simple for models with ARX structure, since for this model structure the data matrices in (3.18)–(3.20) with the outputs and the regressors, i. e. Y_i and Z_i for the local data set i, only need to be stacked to tall data matrices. For all other model structures, an initial value must be estimated for each new data set, which is not followed in here. Thus, further information is available to the LPV ARX identification with integration of the local data sets to the global one, which concerns however only the system dynamics at the operating point but not the dynamics traversing through the operating range. The result of this mixture of global and local identification data is shown in Fig. 5.14 with the dark blue dashed line (g+l in the legend stands for global and local data). In comparison to the ARX SF model only based on global data, it can be seen that the averaged BFR on all four global data sets improves by approximately 1 % and the VAF is slightly reduced. Moreover, only the BJ and OE models (estimated on the global data set) can provide higher model accuracies than the ARX model estimated on the global and local data set.

In the Fig. 5.14, it is noticeable that the PBSID model of model order one provides a good fit. Because the output p_{bm} is included in the scheduling parameter vector, this model is not very trustworthy. The same is true for the other identified models that achieve a good fit. This suspicion is confirmed if the LPV models are simulated self-sufficiently, i. e. their output is fed back to the affine scheduling parameter vector. In this case, only a BFR of around 30 % and a VAF of around 85 % are achieved. Here, nonlinear optimizations can be used to refine the identified LPV, respectively nonlinear, models in their accuracy.

Since local measurements are available, the local behavior of the identified LPV models is analyzed at the operating point with $n_{\mathrm{eng}} = 3000\,\mathrm{rpm}$ and $p_{\mathrm{bm}} = 1450\,\mathrm{mbar}$ that is in the middle of the operating region, see Fig. 5.11. The Bode plot for the frequency response data (FRD) of the local data set and the identified models are shown in Fig. 5.15. Moreover, as a benchmark, LTI ARX models of order $n_x = 2$ and $n_x = 5$ are identified to the local data set. The LPV models have a similar local behavior at this operating point and they fit quite well the FRD. The LPV ARX model in SF has an earlier decrease in amplitude than the FRD, but it matches the phase well up to a frequency of $30\,\mathrm{rad\,s^{-1}}$. Similar behavior is found in the ARX model in SF, where local data are added to the identification (dark blue dashed line). Although this ARX model shows larger deviations in the amplitude at low frequencies, the amplitude in the high frequency range matches very well with the local identified LTI ARX model of second order. Considering the fifth-order BJ model, the amplitude and the phase response are much better approximated. In comparison, the benchmark fifth-order LTI ARX model can only provide a better match with the FRD in the high frequency range in amplitude. Note that the local LPV models considered here are obtained from Jacobian linearization. If instead the scheduling parameter is simply frozen, LTI models are obtained, which, however, are not related to the local system behavior, see for example the orange line in Fig. 5.15.

A comparison between measured validation data and different LPV models is shown in Fig. 5.16, where ARX models of order $n_x = 2$ and the PBSID model of order $n_x = 1$ are considered. As indicated by the high BFR and VAF, it can be seen that the output of the ARX model (light blue) fits very closely the measured data. Since the outputs of the other models in Fig. 5.14 do not differ much from the light blue line and agree very closely with the measured data, they are not presented here.

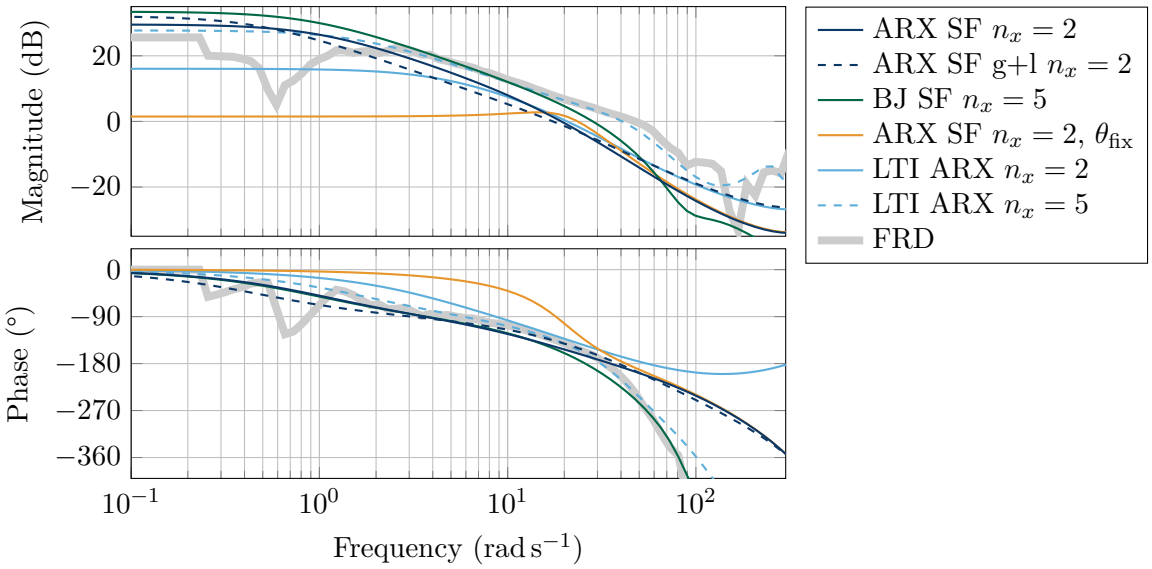


Figure 5.15: Bode plot of local identification results on the air path SISO system at operating point $n_{\text{eng}} = 3000\,\text{rpm}$ and $p_{\text{bm}} = 1450\,\text{mbar}$

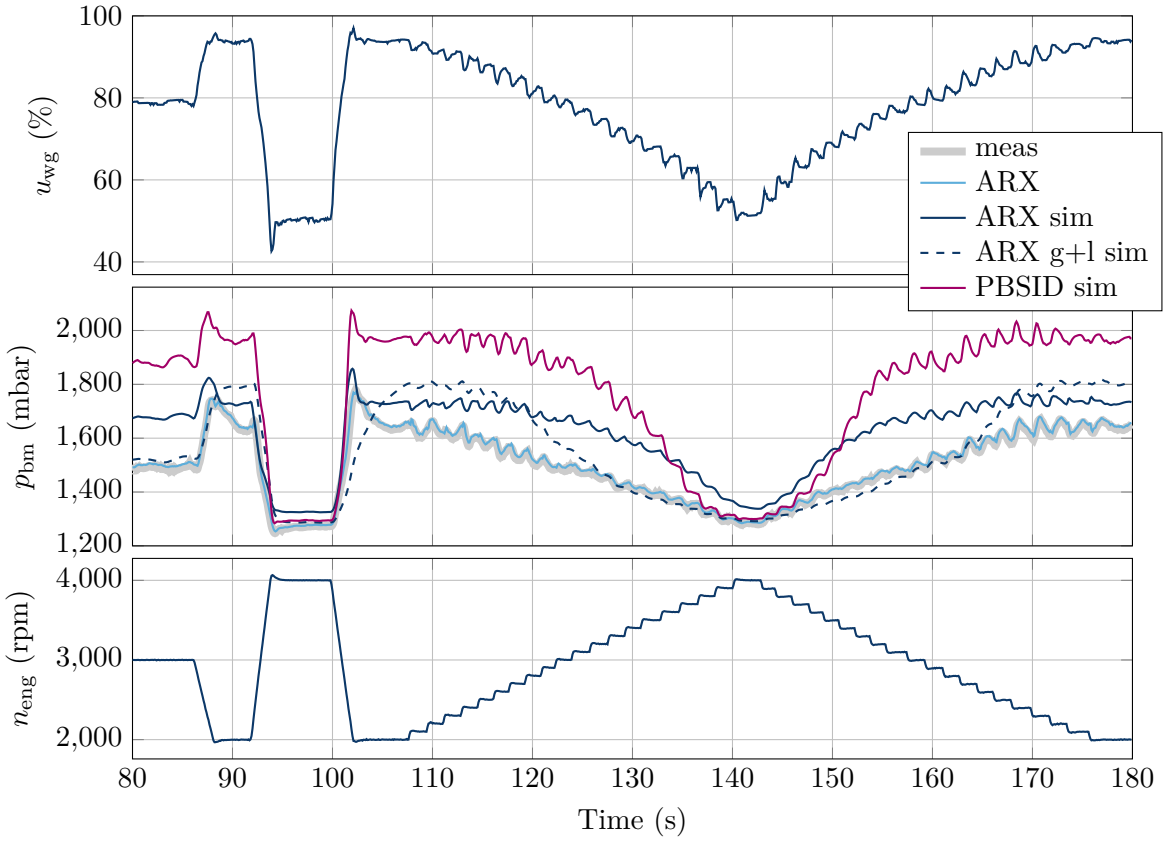


Figure 5.16: Segment of input, measured and modeled output, and engine speed signals for the validation on the air path SISO subsystem (meas: measured output, sim: θ with simulated output)

The very good identification result shown here, however, assumes an external scheduling parameter. In this case, it means that the internal dependence of the affine scheduling parameter θ on the model output is neglected and instead, as in the identification algorithms, the measured output is used to compute θ. For this reason, as an outlook, it is shown in Fig. 5.16 how some identified LPV models perform in simulation when the internal scheduling parameter dependence is considered, that is, when the modeled instead of the measured output is fed to θ. To simulate the quasi LPV models in SS representation, which have a static scheduling parameter dependency, they must be extended as in (2.50) in Section 2.4.1. As a result, the LPV SS models have a one-step delayed dependency on θ. It can be noted that huge pressure changes are simulated quite well with the quasi LPV ARX model, see 100 s in Fig. 5.16. However, there is a well recognizable offset, which seems to depend on the engine speed and/or the boost pressure. The ARX model with the combination of global and local measurement data shows a much slower dynamic, which may be because the local measurements do not contain large pressure changes. The first-order PBSID model, which achieves good results with an external scheduling parameter, shows in simulation similar dynamics as the ARX model, but the offset at high pressures is much more pronounced. However, the outputs of many LPV models are diverging in the simulation, regardless of whether they are identified with IO or subspace approaches. To improve the quasi LPV model accuracy, i. e. if the internal dependence should also be considered during identification, nonlinear optimization algorithms must be used that have not yet been published.

6 Summary and Future Work

In the present thesis, the identification and state order reduction of LPV models is examined with the objective to obtain LPV models that are suitable for established SS model-based LPV control design methods. The focus is on the computationally efficient identification of LPV SS models of low order with static parameter dependence. The developed methods serve this research objective and allow to quickly estimate a large number of LPV models from which the most suitable one can be chosen. A two-step procedure is proposed, i. e. using identification methods for LPV IO models in combination with SS model order reduction based on the Ho-Kalman scheme.

The introduced methods are tested on an industrial application, the air path of a turbocharged SI engine. This system shows nonlinear stationary and dynamic behavior and is the subject of many publications dealing with theoretical modeling, identification, and control design, see [Isermann 2014] and references therein. In order to design a controller for this system using modern model-based synthesis techniques, a sufficiently accurate model of the system is required. Unfortunately, either models of the air path that can be employed for control design are missing or complicated nonlinear models are available, which often originate from a theoretical modeling approach. However, these nonlinear models require, firstly, a time-consuming parameterization effort and, secondly, a sophisticated simplification so that they have a high accuracy and can be used for control design. Therefore, a different approach is pursued in this thesis, the data-driven identification of discrete-time LPV models. Since in the MIMO case, identified LPV models can have a high order, a combination of identification with model order reduction methods is proposed. Real-world experiments show the performance of this approach.

In Chapter 2 of this thesis, preliminary material is presented for LPV systems that is fundamental for the subsequent chapters. The input-output and state-space representations of discrete-time LPV systems are presented in Section 2.1 and different scheduling parameter dependencies are discussed in Section 2.2. In the present thesis, the focus is on LPV systems with static and affine scheduling parameter dependence because the affine dependence simplifies the identification and the static dependence is demanded in the majority of LPV control design techniques. Next, some fundamental system properties such as quadratic stability, observability, and reachability are recapitulated in Section 2.3 for discrete-time LPV SS models. To verify these properties, already known conditions are provided, formulated as LMIs. In this thesis, only constant matrices are allowed as optimization variables in the LMIs to allow a fair comparison with, e. g., the Ho-Kalman model order reduction technique. The transformation of LPV systems is treated in Section 2.4, where particular forms are presented with their input-output and corresponding state-space representations. They become relevant when IO identification methods are combined with techniques that require an LPV SS representation such as control design or state order reduction.

For a model-based LPV control design, a sufficiently accurate LPV model is essential and several identification algorithms for DT LPV model structures are reviewed in Chapter 3. The four key ingredients of system identification, i. e. data collection, model structure selection, optimization of model coefficients, and model validation, are recapitulated in Section 3.1. In

this thesis the focus is on the two steps in the middle, where identification approaches for different model structures are compared w. r. t. the accuracy of the estimated models and the computational effort. The identification of LPV IO models is presented in Section 3.3, where known algorithms are provided and Algorithm 5 is introduced for the identification of SISO and MIMO LPV models in OE or BJ structure. Moreover, the relationship to the identification of LTI models is elaborated and thus the use of highly developed off-the-shelf algorithms are presented, e. g., for the ARMAX model structure in Algorithm 3. Subsequently, the LPV subspace identification method is briefly recapitulated in Section 3.4. This is followed by a detailed comparison in Section 3.5 of IO with subspace identification methods for specific DT LPV model structures. In conclusion of the comparison, the subspace identification of LPV SS models with innovation type noise structure is similar to the identification of finite order LPV IO models in ARMAX structure or infinite order LPV IO models in ARX or MAX structure. The subspace identification approach has fewer restrictions in the model structure, which, however, is associated with a significantly higher computational effort and requires in some steps the neglect of certain matrix structures. Despite their differences, both approaches provide models of similar accuracy in an academic example, see Example 4.2. At the end of this chapter, nonlinear optimization is briefly presented in Section 3.6, which is used to refine identified LPV models.

If the computationally less demanding IO identification methods are chosen, the state order of the associated SS model is not necessarily minimal. This is especially true in the MIMO case and requires model order reduction to obtain low-order models with high accuracy. In Chapter 4, well-known approaches for model order reduction are recapitulated and the balanced truncation is discussed in detail, see Section 4.2. Instead of computing the generalized Gramian matrices via an LMI approach, the Ho-Kalman model order reduction approach is studied in this thesis. This scheme is based on the Hankel matrix $\mathcal{H}_{i,j}$, which also occurs in SID with $i = f$ the future and $j = p$ the past window. In Section 4.3, the computational load of the Ho-Kalman approach is significantly reduced, leading to the quick Ho-Kalman scheme in Lemma 4.1 and Algorithm 7. Thus, model order reduction of DT LPV SS models requires only slightly more computational load than it does in the LTI case and, hence, this approach gains practical relevance. The performance of the quick Ho-Kalman scheme is compared with balanced truncation in the LPV case, where scheduling parameter independent generalized Gramians are used so that the comparison becomes fair. Consequently, an alternative to the SID for the identification of LPV SS models is a two-step procedure consisting of, firstly, IO system identification methods and, secondly, the quick Ho-Kalman order reduction scheme. This two-step procedure is computationally less demanding than the LPV PBSID approach and scales better w. r. t. the number of samples and the input, scheduling parameter, and output dimensions.

In Chapter 5, the LPV identification and model order reduction approaches are applied to a real-world example system from automotive industry: the air path of a turbocharged SI engine. For a better understanding, a physical and phenomenological modeling of the air path is described in Section 5.1. Subsequently, two subsystems of the air path are identified, one in suction and the other in charged operation, see Sections 5.2.1 and 5.2.2. In suction mode, a nonlinear MIMO system is present that is excited with a sophisticated input signal in order to collect as much information as possible about the system in the identification data set. Then, DT LPV models are identified using different IO model structures and the SS model structure with innovation type noise. The identified models are compared and different affine scheduling parameters are tested, since the affine scheduling parameter vector is unknown but

required a priori for the identification methods. Both the IO and the subspace identification methods provide similarly good models, where advanced IO model structures perform slightly better. Note that the order of the polynomials in the IO model structures is not known a priori in these practical applications and many combinations provide SS models of the same order. Similarly, different lengths of past and future windows can be used in the PBSID method to obtain models with the same number of states. In both cases, the number of combinations and/or the computational load rises with the number of states. Followed by the identification, model order reduction is performed for the SS representations of the IO models, where the Ho-Kalman approach is compared with the LMI-Gramian approach. As a result, the quick Ho-Kalman reduction scheme often provides models of better accuracy than the LMI-Gramian approach and it has the shorter computation time. Unfortunately, both approaches often provide models of low accuracy. Possible reasons for this are that either the full-order model cannot be reduced without introducing a large approximation error or the assumption of a constant transformation matrix in the reduction algorithms is too conservative. In addition, numerical problems can occur due to, e. g., very large Hankel singular values in the Ho-Kalman approach. In a last step, a selection of (reduced-order) LPV models are refined by nonlinear optimization. This improves the accuracy of the models in the BFR and VAF by up to 10 %, but the computation time increases significantly by a factor greater than 500, resulting in a calculation time of up to one hour for models of order nine or ten. Next, the ICE is studied in charged mode and a similar procedure is followed, but without order reduction, since it is a SISO system, and without optimization. Instead, the identification of LPV models on a global data set is compared to the identification of LPV IO ARX models based on a data set that contains local excitations at several operating points and a global excitation, cf. [Bachnas et al. 2014]. With this combined global and local approach, the model accuracy, respectively BFR, can be improved in this example. Furthermore, it is noted in this subsystem that the scaling, more precisely the offsets, of the input and output signals have a significant influence on the identification result. This effect is less noticeable in the preceding MIMO subsystem. One reason for the influence of the offsets on the model accuracy can be the change in the output when the external scheduling parameter is varied and the input is kept constant. Here, the identification of LTI and LPV models differs significantly: in the LTI case, local offsets are subtracted depending on the operating point, whereas in the LPV case global offsets for the entire operating range are used. Using improved offsets, the identified LPV models show good agreement with the measured data. Another result is that the identified quasi LPV models show a significantly worse accuracy in the simulation, i. e. with feeding the simulated outputs to the scheduling parameter vector. Moreover, an LPV IO ARX model identified on the global data set performs in simulation better than the same model of same structure identified on the global plus local data set. This can be explained by the fact that the data set with global and local excitation contains relatively fewer samples for the global behavior, which is consequently less well learned during identification. Again, the IO identification methods tend to perform better than the subspace method.

Future Work

In this thesis, it is shown on two industrial example systems that LPV models can be identified that are superior to LTI models. On the one hand, the identified (and order reduced) LPV models reproduce the global system behavior well and on the other hand they are suitable for LPV control design. Nevertheless, the need for future research is recognized, also based on the

practical experience gained.

For the LPV identification methods considered here, it is assumed that the affine scheduling parameters are known a priori, i. e. both the candidates for the scheduling parameters and the functional dependency to derive the affine parameters. However, this is often not the case in practical applications so that it must be searched for the affine scheduling parameters. Here, data-driven methods such as support vector machine or Gaussian process can be used to support the learning of candidates for the scheduling parameters, see [Rizvi et al. 2016], or the functional dependency, see [Darwish et al. 2015; Datar et al. 2018; Tóth et al. 2011]. The estimation of the functional dependence has so far only been developed for SISO IO systems with static scheduling parameter dependence, hence the extension to MIMO systems and dynamic parameter dependence should be investigated. In addition, it can be helpful at this point to incorporate a priori knowledge about the (affine) scheduling parameters. A different approach to learn the scheduling parameter dependency is based on the approach of local linear models [Verdult 2002, Part 3]. Another approach is the local linear model tree or its extension to hierarchical local model tree, where axes-orthogonal or axes-oblique partitioning of the parameter set is iteratively performed and for each subset an LTI model is estimated, which is activated by a weighting function [Nelles 2001, Chapter 17]. However, quasi LPV systems with output dependent scheduling parameter become problematic, since commonly an operating point dependent offset is allowed for the local LTI models and thus the data can be very accurately approximated by these scheduling parameter dependent offsets without learning the local dynamic behavior.

The first step of system identification, collecting an identification data set, is crucial and has a significant impact on the model quality. If only the data set is considered, rough analyses can be performed, such as the coverage of the scheduling parameter set and the dynamics of the signals. For more precise statements, the data set should be considered in combination with the model structure. Then, for example, stochastic analysis of the estimated LPV model coefficients can be performed so that the reliability of the model in relation to the data set can be evaluated. For the stochastic analysis, e. g., the bootstrap method can be used, cf. [Boos 2003], [Ljung 1999, Section 10.4]. It is also desirable to improve the system excitation so that a more reliable LPV model can be identified. Such an investigation and iterative improvement of the excitation is also known as design of experiments. This approach has already been extensively studied, cf. [Chikkula & Lee 1997; Franceschini & Macchietto 2008; Gringard & Kroll 2016; Wahlberg et al. 2010], but not yet in the LPV framework, where an analysis and improvement of the identification experiment can be done regarding the system dynamics and the coverage of the scheduling parameter set. Moreover, if the model is used for control design, iterative procedures for the system identification (operated in closed-loop) and control design be investigated, cf. [Van Den Hof & Schrama 1995].

The nonlinear optimization algorithms, which are introduced in [Cox 2018, Section 7] and applied in here, significantly improve the identified LPV models, see Fig. 5.7. However, they are so far only implemented for LPV systems with external scheduling parameters and future research should involve the computational efficient optimization of quasi LPV systems.

The LPV model order reduction with the quick Ho-Kalman scheme performs well in the academic example in Section 4.3.3. However, in some cases it fails in the MIMO application example in Section 5.2.1 with engine in suction operation although it mostly provides better results than the LMI-Gramian based method. Thus, LPV model order reduction should be further investigated. In addition, the LMI based approach in [Wu 1996] can be compared with the reduction methods considered here. A less conservative, projection-based approach has

recently been published in [Theis et al. 2018]. The extension of this approach to discrete-time systems and the comparison with the approaches shown here is also of interest.

Due to the appealing results shown here, the applicability of LPV identification methods to further industrial systems is of interest. Motivated by the results in Section 5.2.2, constant or scheduling parameter dependent offsets in the inputs and outputs should be considered in the model structure used for the identification of LPV models.

A Appendix

A.1 Matrix Operations

Kronecker Product

The Kronecker product of two matrices $A \in \mathbb{R}^{p\times r}$ and $B \in \mathbb{R}^{q\times s}$ is

$$A \otimes B = \begin{bmatrix} a_{11}B & \dots & a_{1r}B \\ \vdots & \ddots & \vdots \\ a_{p1}B & \dots & a_{pr}B \end{bmatrix} \in \mathbb{R}^{pq\times rs} , \tag{A.1}$$

where a_{ij} denotes the entry in row i and column j.

Khatri-Rao Product

The Khatri-Rao product is a column-wise Kronecker product for two matrices with the same number of columns [Khatri & Rao 1968]. Thus, the Khatri-Rao product of two matrices $A \in \mathbb{R}^{p\times r}$ and $B \in \mathbb{R}^{q\times r}$ is

$$A \odot B = \begin{bmatrix} a_1 \otimes b_1 & a_2 \otimes b_2 & \dots & a_r \otimes b_r \end{bmatrix} \in \mathbb{R}^{pq\times r} , \tag{A.2}$$

where a_i and b_i denotes the i-th column of A and B, respectively.

Further properties of the Khatri-Rao product are given in [Khatri & Rao 1968]. These are for a matrix $C \in \mathbb{R}^{s\times r}$

$$A \odot B \odot C = \begin{bmatrix} a_1 \otimes b_1 \otimes c_1 & a_2 \otimes b_2 \otimes c_2 & \dots & a_r \otimes b_r \otimes c_r \end{bmatrix} \in \mathbb{R}^{pqs\times r} , \tag{A.3}$$

and for two matrices $D_1 \in \mathbb{R}^{m\times p}$ and $D_2 \in \mathbb{R}^{n\times q}$

$$D_1 A \odot D_2 B = (D_1 \otimes D_2)(A \odot B) . \tag{A.4}$$

Product of Linear Parameter Dependent Matrices

The two matrices $A(\mu) \in \mathbb{R}^{p\times r}$ and $B(\mu) \in \mathbb{R}^{r\times s}$ have a linear dependence on the parameter vector $\mu \in \mathbb{R}^{n_\mu}$, i. e.

$$A(\mu) = \sum_{i=1}^{n_\mu} A_i \mu(i) = \begin{bmatrix} A_1 & \dots & A_{n_\mu} \end{bmatrix} (\mu \otimes I_r) = A^{\mathrm{ex}}(\mu \otimes I_r) \tag{A.5}$$

with $\mu(i)$ the i-th entry of the parameter vector and $A_i \in \mathbb{R}^{p\times r}$. The product of the matrices $A(\mu_k)$ and $B(\mu_j)$ for different values of the parameter vector is

$$\begin{aligned} A(\mu_k)B(\mu_j) &= (A_1\mu_k(1) + \ldots + A_{n_\mu}\mu_k(n_\mu))B^{\mathrm{ex}}(\mu_j \otimes I_s) \\ &= \begin{bmatrix} A_1B^{\mathrm{ex}} & \ldots & A_{n_\mu}B^{\mathrm{ex}} \end{bmatrix} \begin{bmatrix} \mu_k(1)(\mu_j \otimes I_s) \\ \vdots \\ \mu_k(n_\mu)(\mu_j \otimes I_s) \end{bmatrix} \\ &= A^{\mathrm{ex}} \begin{bmatrix} B^{\mathrm{ex}} & 0 & \ldots & 0 & \\ 0 & B^{\mathrm{ex}} & 0 & \ldots & 0 \\ \vdots & \vdots & \ddots & \vdots & \\ 0 & 0 & \ldots & B^{\mathrm{ex}} & \end{bmatrix} (\mu_k \otimes \mu_j \otimes I_s) \\ &= A^{\mathrm{ex}}(I_{n_\mu} \otimes B^{\mathrm{ex}})(\mu_k \otimes \mu_j \otimes I_s) \end{aligned} \tag{A.6}$$

with $(\mu_k \otimes \mu_j \otimes I_s) \in \mathbb{R}^{n_\mu^2 s\times s}$. Consequently, the product of f linear parameter dependent matrices is

$$\acute{\mathcal{A}}\grave{\mathcal{P}}_{1|f} = A(\mu_1)A(\mu_2)\ldots A(\mu_f) \tag{A.7}$$

$$\acute{\mathcal{A}} = A^{\mathrm{ex}}(I_{n_\mu} \otimes (A^{\mathrm{ex}}(\ldots(I_{n_\mu} \otimes A^{\mathrm{ex}})))) \in \mathbb{R}^{r\times n_\mu^f r} \tag{A.8}$$

$$\grave{\mathcal{P}}_{1|f} = (\mu_1 \otimes \mu_2 \otimes \ldots \otimes \mu_f \otimes I_r) \in \mathbb{R}^{n_\mu^f r\times r}\ . \tag{A.9}$$

Matrices in calligraphic letters denote matrices that are expanded to allow separation between state-space matrices and scheduling parameters via multiplication. The accents are used to indicate on which side the counterpart to the expanded matrix is located.

A.2 From IO to SS Domain

In this section, the transformation from IO to SS domain for the three different forms with dynamic scheduling parameter dependency, i. e. the shifted form (SF), the augmented form (AF), and the observability form (OF), is shown in more detail.

For the AF and the OF, the states have a simple physical interpretation. They represent in the AF delayed in- and outputs and in the OF predicted outputs.

Note that in the following the transformations are only shown for DT LPV systems. Of course, the methods can also be applied to CT LPV systems, where time shift operations in the DT domain correspond to differentiation or integration in the CT domain. However, these transformation approaches can lead to the fact that the static and affine scheduling parameter dependence in the SS representation can no longer be maintained [Tóth 2010, Section 4.3].

A.2.1 Shifted Form

The DT LPV IO system in SF is given by, see (2.39),

$$y(k) = -\sum_{i=1}^{n_a} a_i(\rho(k-i))y(k-i) + \sum_{j=0}^{n_b} b_j(\rho(k-j))u(k-j)\ . \tag{A.10}$$

Now, the idea is to extract all elements of the current time step from both sums and introduce another state for the remaining sums. Subsequently, the state is multiplied with the forward

time shift operator and again the current inputs and outputs are extracted from the sums. This is repeated until both sums vanish, i. e. no delayed inputs, outputs and scheduling parameters remain. Starting with (A.10) yields

$$y(k) = b_0(\rho(k))u(k) + x_1(k) \tag{A.11}$$

$$x_1(k) = -\sum_{i=1}^{n_a} a_i(\rho(k-i))y(k-i) + \sum_{j=1}^{n_b} b_j(\rho(k-j))u(k-j) \ , \tag{A.12}$$

where the lower index of the sum for the inputs is raised by one and $x_i(k) \in \mathbb{R}^{n_y}$. Eq. (A.11) is the output equation from which the C and D matrix functions of the SS representation are obtained. Now, shifting the state x_1 one time step in the future gives

$$\begin{aligned} \mathrm{q}x_1(k) &= -\sum_{i=1}^{n_a} a_i(\rho(k-i+1))y(k-i+1) + \sum_{j=1}^{n_b} b_j(\rho(k-j+1))u(k-j+1) \\ &= -a_1(\rho(k))y(k) + b_1(\rho(k))u(k) + x_2(k) \end{aligned} \tag{A.13}$$

$$x_2(k) = -\sum_{i=1}^{n_a-1} a_{i+1}(\rho(k-i))y(k-i) + \sum_{j=1}^{n_b-1} b_{j+1}(\rho(k-j))u(k-j) \ . \tag{A.14}$$

Inserting (A.11) in (A.13) to eliminate the output from the state equation results in

$$\mathrm{q}x_1(k) = -a_1(\rho(k))x_1(k) + x_2(k) + \left(b_1(\rho(k)) - a_1(\rho(k))b_0(\rho(k))\right)u(k) \ , \tag{A.15}$$

which is the equation of the first n_y rows in the system matrix of $\mathcal{R}_{\mathrm{SS}}^{\mathrm{DT,SF}}(G(\rho))$. The remaining rows are obtained by continuing this eponymous cut-and-shift technique introduced in [Willems 2007] and [Tóth 2010, Section 4.3] until no delayed in- and/or outputs remain, i. e.

$$\mathrm{q}x_{n_a}(k) = -a_{n_a}(\rho(k))y(k) + b_{n_a}(\rho(k))u(k) \ . \tag{A.16}$$

Due to the sums in (A.12) and (A.14), the states of $\mathcal{R}_{\mathrm{SS}}^{\mathrm{DT,SF}}(G(\rho))$ do not have a simple physical interpretation in contrast to the states of $\mathcal{R}_{\mathrm{SS}}^{\mathrm{DT,AF}}(G(\rho))$ and $\mathcal{R}_{\mathrm{SS}}^{\mathrm{DT,OF}}(G(\rho))$, hence

$$x_{l+1}(k) = -\sum_{i=1}^{n_a-l} a_{i+l}(\rho(k-i))y(k-i) + \sum_{j=1}^{n_b-l} b_{j+l}(\rho(k-j))u(k-j) \tag{A.17}$$

with $l \in \{0, 1, \ldots, n_a - 1\}$. Each state $x_i(k)$ represents the system's dynamic from the most delayed time instant $k - n_a$ up to $k - i$, which in general does not have a meaningful physical interpretation.

Note that the state-space realization of the SF is in the companion observability canonical form, see [Tóth 2010, Section 4.1.3]. However, the state order is reversed, which is achieved by the exchange matrix (one elements on the counterdiagonal and all other elements equal to zero) as a state transformation matrix.

A.2.2 Augmented Form

For the SS AF, the first $n_a n_y$ states represent the (delayed) outputs $y(k) - b_0(\rho(k))u(k)$, $y(k-1)$, up to $y(k - n_a + 1)$ while the last $(n_b - 1)n_u$ states represent the delayed inputs $u(k-1)$ up to

$u(k-n_b+1)$. Hence, shifting (2.41) and introducing states yields

$$\begin{aligned} y(k) =& -\sum_{i=1}^{n_a} a_i\left(\rho(k-1)\right) y(k-i) + b_0(\rho(k))u(k) + \sum_{j=1}^{n_b} b_j\left(\rho(k-1)\right) u(k-j) \\ =& x_1(k) + b_0(\rho(k))u(k) \qquad \text{(A.18)} \\ \mathrm{q}x_1(k) =& -a_1(\rho(k))y(k) - a_2(\rho(k))\underbrace{y(k-1)}_{x_2(k)=\mathrm{q}^{-1}(x_1(k)-b_0(\rho(k))u(k))} - \ldots - a_{n_a}(\rho(k))\underbrace{y(k-n_a+1)}_{x_{n_a}(k)=\mathrm{q}^{-1}x_{n_a-1}(k)} \\ &+ b_1(\rho(k))u(k) + b_2(\rho(k))\underbrace{u(k-1)}_{x_{n_a+1}(k)} + \ldots + b_{n_b}(\rho(k))\underbrace{u(k-n_b+1)}_{x_{n_a+n_b-1}(k)=\mathrm{q}^{-1}x_{n_a+n_b-2}(k)} . \qquad \text{(A.19)} \end{aligned}$$

All the states are connected by $\mathrm{q}x_i(k) = x_{i-1}(k)$ except $x_1(k)$, $x_2(k)$ and $x_{n_a+1}(k)$ that are $y(k) - b_0(\rho(k))u(k)$, $y(k-1)$, and $u(k-1)$, respectively. Moreover, the dimension of the states $x_i(k)$ for $i \in \{1, \ldots, n_a\}$ is n_y while for $i \in \{n_a+1, \ldots, n_a+n_b-1\}$ it is n_u.

A.2.3 Observability Form

The name of this form results from the fact that the state-space realization is in the observability canonical form as given in [Tóth 2010, Section 4.1.1]. This is at first somewhat misleading, since the companion observability canonical form in [Tóth 2010, Section 4.1.3] is in the literature often referred to as observability form.

The states in the SS OF represent the predicted outputs. Shifting the input-output equation of the OF

$$y(k) = -\sum_{i=1}^{n_a} a_i\left(\rho(k-n_a)\right) y(k-i) + b_{n_a}\left(\rho(k-n_a)\right) u(k-n_a) \qquad \text{(A.20)}$$

by n_a time steps to the future and introducing states gives

$$\underbrace{\mathrm{q}^{n_a}y(k)}_{\mathrm{q}x_{n_a}(k)} = -a_1(\rho(k))\underbrace{y(k+n_a-1)}_{x_{n_a}(k)=\mathrm{q}x_{n_a-1}(k)} - \ldots - a_{n_a}(\rho(k))\underbrace{y(k)}_{x_1(k)} + b_{n_a}(\rho(k))u(k) . \qquad \text{(A.21)}$$

Hence, the state $x_i(k)$ represents the predicted output at time instant $k-1+i$. However, this prediction comes at the cost of introducing an input time delay of size n_a, see (A.20). Thus, the OF is constrained to represent only LPV systems with an input delay of the same size as the order n_a, which is a very specific restriction.

The delay can be reduced, if more entries in the B and D matrix functions of the LPV SS representation in OF are allowed. Hence, the SS representation is, see also [Tóth 2010, Section 4.1],

$$\left[\begin{array}{ccccc|c} 0 & I & 0 & \ldots & 0 & b_1(\rho) \\ \vdots & 0 & \ddots & \ddots & \vdots & \vdots \\ \vdots & \vdots & \ddots & \ddots & \vdots & \vdots \\ 0 & 0 & \ldots & 0 & I & b_{n_a-1}(\rho) \\ -a_{n_a}(\rho) & -a_{n_a-1}(\rho) & \ldots & \ldots & -a_1(\rho) & b_{n_a}(\rho) \\ \hline I & 0 & \ldots & \ldots & 0 & b_0(\rho) \end{array}\right] \qquad \text{(A.22)}$$

and the corresponding IO representation is

$$y_k = -\sum_{i=1}^{n_a} a_i(\rho_{k-n_a}) \left(y_{k-i} - \sum_{h=0}^{n_a-i} b_h(\rho_{k-i-h}) u_{k-i-h} \right) + \sum_{j=0}^{n_b} b_j(\rho_{k-j}) u_{k-j} \ . \tag{A.23}$$

Thus, in contrast to the AF, a direct feedthrough (and further time shifts of the input) can be included in the IO representation, which will preserve affine and static parameter dependency in the LPV SS representation. However, the affine dependency is lost in the IO domain. Only if $b_i(\rho)$ is constant for $i \in \{0, 1, \ldots, n_a - 1\}$ the sum over index h vanishes and affine dependence is regained, but then the identification problem in the IO domain is still – due to the multiplication with $a_i(\rho_{k-n_a})$ – nonlinear in the system coefficients.

A.2.4 Extended Augmented Form

For the extended AF, the first $n_a n_y$ states represent the delayed outputs $y(k-1)$ up to $y(k-n_a)$ while the last $n_b n_u$ states represent the delayed inputs $u(k-1)$ up to $u(k-n_b)$. Hence, shifting (2.45) and introducing states yields

$$y(k) = -\sum_{i=1}^{n_a} a_i\left(\rho(k)\right) y(k-i) + \sum_{j=0}^{n_b} b_j\left(\rho(k)\right) u(k-j) = \mathrm{q} x_1(k) \tag{A.24}$$

$$\begin{aligned} \mathrm{q} x_1(k) = &- a_1(\rho(k)) \underbrace{y(k-1)}_{x_1(k)} - a_2(\rho(k)) \underbrace{y(k-2)}_{x_2(k) = \mathrm{q}^{-1} x_1(k)} - \ldots - a_{n_a}(\rho(k)) \underbrace{y(k-n_a)}_{x_{n_a}(k)} \\ &+ b_0(\rho(k)) u(k) + b_1(\rho(k)) \underbrace{u(k-1)}_{x_{n_a+1}(k)} + \ldots + b_{n_b}(\rho(k)) \underbrace{u(k-n_b)}_{x_{n_a+n_b}(k) = \mathrm{q}^{-1} x_{n_a+n_b-1}(k)} \ . \end{aligned} \tag{A.25}$$

All the states are connected by $\mathrm{q} x_i(k) = x_{i-1}(k)$ except $x_1(k)$ and $x_{n_a+1}(k)$ that are $y(k-1)$ and $u(k-1)$, respectively. Moreover, the dimension of the states $x_i(k)$ for $i \in \{1, \ldots, n_a\}$ is n_y while for $i \in \{n_a+1, \ldots, n_a+n_b-1\}$ it is n_u.

A.2.5 Initial State

In order to simulate a state-space representation and compare it, e. g., with a measurement, it is good to know the initial state. Otherwise, the SS model can perform poorly due to a false initial state.

The initial state can be easily reconstructed, if the (past) in-, output, and scheduling parameter signals are known or if they are assumed as constant. Hence, the case is not considered, where no information about the past is available, which is the computational more demanding and system-theoretical more complicated case.

If a comparison with a measurement is desired, the first measured values can be used to construct the approximate initial state. However, this is only correct for an exact model, otherwise the modeled and the measured outputs do not exactly match. In addition, an error occurs in the initial state, especially when the output signal is noisy. However, these disadvantages are accepted here and assumed to cause only a minor error in the initial state construction.

In the IO domain simply the (delayed) in-, outputs, and scheduling parameters can be taken for initialization, the computation of the initial state in the SS domain can be more involved.

For the OF, the AF, and the extended AF, the initial state can be easily constructed, since the state entries are directly related to the in- and outputs, hence

$$x_0^{\mathrm{OF}} = \begin{bmatrix} y_0 \\ y_1 \\ \vdots \\ y_{n_a-1} \end{bmatrix}, \qquad x_0^{\mathrm{AF}} = \begin{bmatrix} y_0 \\ \vdots \\ y_{-n_a+1} \\ u_{-1} \\ \vdots \\ u_{-n_b+1} \end{bmatrix}, \qquad x_0^{\mathrm{eAF}} = \begin{bmatrix} y_{-1} \\ \vdots \\ y_{-n_a} \\ u_{-1} \\ \vdots \\ u_{-n_b} \end{bmatrix}. \tag{A.26}$$

For the SF, each state is given by (A.17) and thus

$$x_0^{\mathrm{SF}} = \begin{bmatrix} -\sum_{i=1}^{n_a} a_i(\rho_{-i})y_{-i} + \sum_{j=1}^{n_b} b_j(\rho_{-j})u_{-j} \\ -\sum_{i=1}^{n_a-1} a_{i+1}(\rho_{-i})y_{-i} + \sum_{j=1}^{n_b-1} b_{j+1}(\rho_{-j})u_{-j} \\ \vdots \\ -a_{n_a}(\rho_{-1})y_{-1} + b_{n_b}(\rho_{-1})u_{-1} \end{bmatrix}. \tag{A.27}$$

Another approach is, to assume that the system is in steady-state at the initial time instant, hence $x(1) = x(0)$, and to compute the initial state by solving the system of linear equations

$$\begin{bmatrix} I - A(\rho_0) \\ -C(\rho_0) \end{bmatrix} x(0) = \begin{bmatrix} B(\rho_0) & 0 \\ D(\rho_0) & -I \end{bmatrix} \begin{bmatrix} u(0) \\ y(0) \end{bmatrix} \tag{A.28}$$

for the initial state $x(0)$.

A.3 LPV Subspace Identification

For reasons of completeness, the problem formulation in LPV subspace identification as well as reduction technique is reviewed. The LPV SID problem formulation was introduced in [Verdult 2002, Section 3] and the reduction technique in [Wingerden & Verhaegen 2009]. Note that the structure of some matrices in the following formulation differ from the formulation in [Verdult 2002, Section 3]. The reason for this is to simplify the comparison between subspace identification and the Ho-Kalman model order reduction technique.

A.3.1 Problem Formulation

The derivation of the LPV SID matrix equation in step one, see Section 3.4.2, is more involved compared to the LTI case. As a start, only the first column of the matrix equation (3.47) is considered. With the LPV model structure in (3.44) it follows that

$$\begin{bmatrix} y_k \\ y_{k+1} \\ \vdots \\ y_{k+f-1} \end{bmatrix} = (\mathsf{O}_f \diamond \mu_k)x_k + (\mathsf{H}_f^u \diamond \mu_k) \begin{bmatrix} u_k \\ u_{k+1} \\ \vdots \\ u_{k+f-1} \end{bmatrix} + (\mathsf{H}_f^e \diamond \mu_k) \begin{bmatrix} e_k \\ e_{k+1} \\ \vdots \\ e_{k+f-1} \end{bmatrix}. \tag{A.29}$$

The matrix functions O_f, H_f^u, and H_f^e are now (dynamically) scheduling parameter dependent, thus

$$\mathsf{O}_f \diamond \mu_k = \begin{bmatrix} C \\ CA(\mu_k) \\ CA(\mu_{k+1})A(\mu_k) \\ \vdots \\ CA(\mu_{k+f-2}) \dots A(\mu_{k+1})A(\mu_k) \end{bmatrix} = \begin{bmatrix} C \\ C\phi_{k|k} \\ C\phi_{k+1|k} \\ \vdots \\ C\phi_{k+f-2|k} \end{bmatrix} \tag{A.30}$$

$$\mathsf{H}_f^u \diamond \mu_k = \begin{bmatrix} D & 0 & \dots & 0 & 0 \\ CB(\mu_k) & D & \dots & 0 & 0 \\ C\phi_{k+1|k+1}B(\mu_k) & CB(\mu_{k+1}) & \dots & 0 & 0 \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ C\phi_{k+f-3|k+1}B(\mu_k) & C\phi_{k+f-3|k+2}B(\mu_{k+1}) & \dots & D & 0 \\ C\phi_{k+f-2|k+1}B(\mu_k) & C\phi_{k+f-2|k+2}B(\mu_{k+1}) & \dots & CB(\mu_{k+f-2}) & D \end{bmatrix} \tag{A.31}$$

and $\mathsf{H}_f^e \diamond \mu_k$ is again similar to $\mathsf{H}_f^u \diamond \mu_k$ only the matrices $K(\mu_k)$ and I must replace $B(\mu_k)$ and D in (A.31), respectively. The matrix $\mathsf{O}_f \diamond \mu_k$ is the state-observability matrix function from Definition 2.8 and $\phi_{k+i|k} = A(\mu_{k+i}) \dots A(\mu_{k+1})A(\mu_k)$ with $i \in \mathbb{N}$ is the transition matrix function[1)]. If the vector equation (A.29) is extended by further columns to obtain the matrix equation (3.47), both the signals and the matrices have to be shifted in time, because in the LPV case both are time-dependent. For this reason and since the objective is to estimate the model matrices, the scheduling parameter dependence is removed from O_f, H_f^u, and H_f^e and included in the signal matrices. For example, the state-observability matrix function in (A.30) can be rewritten as

$$\mathsf{O}_f \diamond \mu_k = \acute{\mathcal{O}}_f(\grave{\mathcal{P}}_{k+f-2|k}(:,1) \otimes I_{n_x}) \tag{A.32}$$

$$\grave{\mathcal{P}}_{k+f-2|k}(:,1) = \mu_{k+f-2} \otimes \dots \otimes \mu_{k+1} \otimes \mu_k \in \mathbb{R}^{n_\mu^{f-1}}, \tag{A.33}$$

where the extended observability matrix $\acute{\mathcal{O}}_f$ is scheduling parameter independent, $\grave{\mathcal{P}}_{k+f-2|k}(:,1)$ denotes the first column of $\grave{\mathcal{P}}_{k+f-2|k}$, and $\otimes$ is the Kronecker product, see Appendix A.1. The calligraphic letters indicate matrices that are expanded in order to convert the scheduling parameter dependency to a matrix product and the accents are used to indicate on which side the counterpart to the expanded matrix is located. More details on how to separate the scheduling parameter from a product of model matrices are given in Appendix A.1. Similar decompositions can be given for $\mathsf{H}_f^u \diamond \mu_k$ and $\mathsf{H}_f^e \diamond \mu_k$.

The insights gained by examining the first column are now used to construct the matrix equation. First, the matrix $\grave{\mathcal{P}}_{k+f-2|k}$ is constructed by replacing in (A.33) the vectors μ_i with matrices P_i and the Kronecker product with the column-wise Kronecker product (Khatri-Rao product, Appendix A.1), i. e.

$$P_k = \begin{bmatrix} \mu_k & \mu_{k+1} & \dots & \mu_{k+N-1} \end{bmatrix} \in \mathbb{R}^{n_\mu \times N} \tag{A.34}$$

$$\grave{\mathcal{P}}_{k+f-2|k} = P_{k+f-2} \odot \dots \odot P_{k+1} \odot P_k \in \mathbb{R}^{n_\mu^{f-1} \times N}. \tag{A.35}$$

[1)] Note that $\phi_{k+i|k}$ is a shorthand notation for the dynamically scheduling parameter dependent transition matrix function $\phi_i \diamond \mu_k$, where the dependency on μ is omitted for reasons of brevity.

Then, the matrix equation (3.47) becomes

$$Y_{k+f-1|k} = \acute{\mathcal{O}}_f \grave{\mathcal{X}}_{k+f-1|k} + \acute{\mathcal{H}}^u_f \grave{\mathcal{U}}_{k+f-1|k} + \acute{\mathcal{H}}^e_f \grave{\mathcal{E}}_{k+f-1|k} \tag{A.36}$$

$$Y_{k+f-1|k} = \begin{bmatrix} y_k & y_{k+1} & \cdots & y_{k+N-1} \\ y_{k+1} & y_{k+2} & \cdots & y_{k+N} \\ \vdots & \vdots & \ddots & \vdots \\ y_{k+f-1} & y_{k+f} & \cdots & y_{k+N+f-2} \end{bmatrix} \in \mathbb{R}^{n_y f \times N} \ . \tag{A.37}$$

The signal matrices $\grave{\mathcal{X}}_{k+f-1|k}$, $\grave{\mathcal{U}}_{k+f-1|k}$, and $\grave{\mathcal{E}}_{k+f-1|k}$ containing the product of states, inputs, and noise, respectively, with the scheduling parameter. They are given by

$$\grave{\mathcal{X}}_{k+f-1|k} = \grave{\mathcal{P}}_{k+f-2|k} \odot X_k \in \mathbb{R}^{n_\mu^{f-1} n_x \times N} \tag{A.38}$$

$$\grave{\mathcal{U}}_{k+f-1|k} = \begin{bmatrix} \grave{\mathcal{P}}_{k+f-2|k} \odot U_k \\ \grave{\mathcal{P}}_{k+f-2|k+1} \odot U_{k+1} \\ \vdots \\ \grave{\mathcal{P}}_{k+f-2|k+f-2} \odot U_{k+f-2} \\ U_{k+f-1} \end{bmatrix} \in \mathbb{R}^{n_u \sum_{i=0}^{f-1} n_\mu^i \times N} \tag{A.39}$$

and $\grave{\mathcal{E}}_{k+f-1|k}$ is similar to $\grave{\mathcal{U}}_{k+f-1|k}$ only the noise E_{k+i} is multiplied by the scheduling parameter matrices $\grave{\mathcal{P}}_{k+f-2|k+i}$. The signal matrices X_k, U_k, and E_k are analogous to P_k in (A.34), only with the state x_k, input u_k, and noise e_k, respectively. Note that in the last n_u rows in $\grave{\mathcal{U}}_{k+f-1|k}$ in (A.39) the inputs are not multiplied by the scheduling parameter, since the last n_u columns of $\mathsf{H}^u_f \diamond \mu_k$ contain only the scheduling parameter independent D matrix. Respectively, the same holds for the last n_y rows in $\grave{\mathcal{E}}_{k+f-1|k}$ since the last n_y columns in $\mathsf{H}^e_f \diamond \mu_k$ contains only the identity matrix in the last n_y rows. The matrices $\acute{\mathcal{O}}_f$ and $\acute{\mathcal{H}}^u_f$ are given by the following recursive equations

$$\acute{\mathcal{O}}_1 = \begin{bmatrix} C & 0 & \dots & 0 \\ CA_0 & CA_1 & \dots & CA_{n_\mu - 1} \end{bmatrix} \tag{A.40a}$$

$$\acute{\mathcal{O}}_i = \begin{bmatrix} \acute{\mathcal{O}}_{i-1} & 0 & \dots & 0 \\ CA_0\acute{\mathcal{A}}_{i-1} & CA_1\acute{\mathcal{A}}_{i-1} & \dots & CA_{n_\mu-1}\acute{\mathcal{A}}_{i-1} \end{bmatrix} \tag{A.40b}$$

$$\acute{\mathcal{H}}^u_1 = \begin{bmatrix} D & 0 & \dots & 0 & 0 \\ CB_0 & CB_1 & \dots & CB_{n_\mu-1} & D \end{bmatrix} \tag{A.40c}$$

$$\acute{\mathcal{H}}^u_i = \begin{bmatrix} \begin{bmatrix} D & 0 \end{bmatrix} & 0 \\ \acute{\mathcal{L}}_i & \acute{\mathcal{H}}^u_{i-1} \end{bmatrix} \tag{A.40d}$$

with auxiliary matrices

$$\acute{\mathcal{A}}_1 = \begin{bmatrix} A_0 & A_1 & \dots & A_{n_\mu-1} \end{bmatrix} \ , \quad \acute{\mathcal{A}}_i = \begin{bmatrix} A_0\acute{\mathcal{A}}_{i-1} & A_1\acute{\mathcal{A}}_{i-1} & \dots & A_{n_\mu-1}\acute{\mathcal{A}}_{i-1} \end{bmatrix} \tag{A.41a}$$

$$\acute{\mathcal{M}}_1 = \begin{bmatrix} B_0 & B_1 & \dots & B_{n_\mu-1} \end{bmatrix} \ , \quad \acute{\mathcal{M}}_i = \begin{bmatrix} A_0\acute{\mathcal{M}}_{i-1} & A_1\acute{\mathcal{M}}_{i-1} & \dots & A_{n_\mu-1}\acute{\mathcal{M}}_{i-1} \end{bmatrix} \tag{A.41b}$$

$$\acute{\mathcal{L}}_1 = C\acute{\mathcal{M}}_1 \ , \qquad \acute{\mathcal{L}}_i = \begin{bmatrix} \begin{bmatrix} \acute{\mathcal{L}}_{i-1} & 0 \end{bmatrix} \\ C\acute{\mathcal{M}}_i \end{bmatrix} \ , \tag{A.41c}$$

which are used to express the transitions of the state and input, respectively. The matrix $\acute{\mathcal{H}}^e_f$ is obtained in the same way as $\acute{\mathcal{H}}^u_f$, but with matrices K_l and I instead of B_l and D, respectively, with $l \in \{0, 1, \ldots, n_\mu - 1\}$.

Remark A.1. *From above equations, it should be clear why C and D are usually considered as scheduling parameter independent in the LPV case. If they are also parameter dependent, they must be treated similarly to A, B, and K and the row dimension of $\grave{\mathcal{X}}_{k+f-1|k}$, $\grave{\mathcal{U}}_{k+f-1|k}$, and $\grave{\mathcal{E}}_{k+f-1|k}$, respectively the column dimension of $\acute{\mathcal{O}}_f$, $\acute{\mathcal{H}}^u_f$, and $\acute{\mathcal{H}}^e_f$, in* (A.36) *increases significantly. For example, the number of columns in $\grave{\mathcal{U}}_{k+f-1|k}$ increases from $n_u(1 + \sum_{i=1}^{f-1} n_\mu^i)$ to $n_u(1 + \sum_{i=1}^{f-1} n_\mu^{2i+1})$. This leads to an identification problem that is computationally even more difficult to solve.*

For the state estimation in step three, it is useful to partition the first matrix in (A.36), since the first n_x columns of $\acute{\mathcal{O}}_f$ are the state-observability matrix Γ_f in the LTI case and the corresponding first n_x rows in $\grave{\mathcal{X}}_{k+f-1|k}$ are X_k, hence

$$\acute{\mathcal{O}}_f \grave{\mathcal{X}}_{k+f-1|k} = \Gamma_f X_k + \acute{\mathcal{O}}^{\text{red}}_f \grave{\mathcal{X}}^{\text{red}}_{k+f-1|k} \tag{A.42}$$

$$\grave{\mathcal{X}}^{\text{red}}_{k+f-1|k} = \grave{\mathcal{P}}^{\text{red}}_{k+f-2|k} \odot X_k \tag{A.43}$$

$$\grave{\mathcal{P}}^{\text{red}}_{k+i|k} = \begin{bmatrix} 0 & I_{n_\mu^{i+1}-1} \end{bmatrix} \grave{\mathcal{P}}_{k+i|k} \tag{A.44}$$

$$\Gamma_f = \begin{bmatrix} C \\ CA_0 \\ \vdots \\ CA_0^{f-1} \end{bmatrix}, \tag{A.45}$$

where $\grave{\mathcal{P}}^{\text{red}}_{k+i|k}$ is obtained by removing the first row of $\grave{\mathcal{P}}_{k+i|k}$, see (A.44).

In the next step, the state equation is used in order to replace the states in (A.36). As a start, only x_k is considered and this time the predictor form (3.46a) is used that yields

$$x_k = \underbrace{\bar{A}(\mu_{k-1})\bar{A}(\mu_{k-2})\ldots\bar{A}(\mu_{k-p})}_{\bar{\phi}_{p-1}\diamond\mu_{k-p}} x_{k-p} + (\tilde{\mathsf{R}}_p \diamond \mu_{k-1}) \begin{bmatrix} z_{k-1} \\ z_{k-2} \\ \vdots \\ z_{k-p} \end{bmatrix}, \tag{A.46}$$

where $\tilde{\mathsf{R}}_p \diamond \mu$ is the state-reachability matrix function from Definition 2.9 and $\bar{\phi}_{k+i|k}$ is the transition matrix function based on $\bar{A}(\mu_k) = A(\mu_k) - K(\mu_k)C(\mu_k)$ and $\bar{B}(\mu_k) = B(\mu_k) - K(\mu_k)D(\mu_k)$. The index p denotes a past window: The current state x_k is obtained from the transition of the past state x_{k-p} and the past input and output sequence from z_{k-p} up to z_{k-1}. Separating signals and model matrices gives the matrix equation

$$X_k = \acute{\bar{\mathcal{A}}}_p \grave{\mathcal{X}}_{k|k-p} + \acute{\bar{\mathcal{R}}}_p \grave{\mathcal{Z}}_{k-p|k-1} \tag{A.47}$$

$$\acute{\bar{\mathcal{R}}}_p = \begin{bmatrix} \acute{\bar{\mathcal{M}}}_1 & \acute{\bar{\mathcal{M}}}_2 & \ldots & \acute{\bar{\mathcal{M}}}_p \end{bmatrix} \tag{A.48}$$

$$\grave{\mathcal{Z}}_{k-p|k-1} = \begin{bmatrix} \grave{\mathcal{P}}_{k-1|k-1} \odot Z_{k-1} \\ \vdots \\ \grave{\mathcal{P}}_{k-1|k-p} \odot Z_{k-p} \end{bmatrix} \tag{A.49}$$

with matrix $\grave{\mathcal{X}}_{k|k-p}$ as in (A.38) and matrix $\acute{\bar{\mathcal{A}}}_p$ is analogous to (A.41a) but with matrices $\bar{A}_l$ instead of A_l for $l \in \{0, 1, \ldots, n_\mu - 1\}$. The p-step extended reachability matrix $\acute{\bar{\mathcal{R}}}_p$ is obtained from matrices $\acute{\mathcal{M}}_i$ that are similar to $\acute{\mathcal{M}}_i$ in (A.41), but based on matrices $\tilde{B}_l$ and $\bar{A}_l$ instead of B_l and A_l for $l \in \{0, 1, \ldots, n_\mu - 1\}$. The signal matrix $\grave{\mathcal{Z}}_{k-p|k-1}$ consists of Z_k that is equivalent to P_k in (A.34), but with $z_k^{\mathrm{T}} = [u_k^{\mathrm{T}} \quad y_k^{\mathrm{T}}]$ instead of μ_k. Note that $\grave{\mathcal{Z}}_{k-p|k-1}$ in (A.49) is filled from top to bottom with (past) values from z_{k-1} backwards in time until z_{k-p}, which is why smaller value precedes a larger value in the index. In contrast, $\grave{\mathcal{U}}_{k+f-1|k}$ in (A.39) is filled with (future) values from u_k forwards in time until u_{k+f-1}.

In the last step of deducing the matrix equation, the state matrix X_k in (A.36) is replaced by (A.47). Thus, using a property of the Khatri-Rao product the matrix equation becomes

$$\begin{aligned} Y_{k+f-1|k} = \Gamma_f \acute{\bar{\mathcal{A}}}_p \grave{\mathcal{X}}_{k|k-p} + (\acute{\mathcal{O}}_f^{\mathrm{red}} \otimes \acute{\bar{\mathcal{A}}}_p)(\grave{\mathcal{P}}_{k+f-2|k}^{\mathrm{red}} \odot \grave{\mathcal{X}}_{k|k-p}) \\ + \Gamma_f \acute{\bar{\mathcal{R}}}_p \grave{\mathcal{Z}}_{k-p|k-1} + (\acute{\mathcal{O}}_f^{\mathrm{red}} \otimes \acute{\bar{\mathcal{R}}}_p)(\grave{\mathcal{P}}_{k+f-2|k}^{\mathrm{red}} \odot \grave{\mathcal{Z}}_{k-p|k-1}) \\ + \acute{\mathcal{H}}_f^u \grave{\mathcal{U}}_{k+f-1|k} + \acute{\mathcal{H}}_f^e \grave{\mathcal{E}}_{k+f-1|k} \end{aligned} \tag{A.50}$$

and the matrices in (3.47) for the DT LPV SID of the innovation type noise structure are

$$Y = Y_{k+f-1|k} \tag{A.51a}$$

$$\Phi_X = \begin{bmatrix} \Gamma_f \acute{\bar{\mathcal{A}}}_p & \acute{\mathcal{O}}_f^{\mathrm{red}} \otimes \acute{\bar{\mathcal{A}}}_p \end{bmatrix} \tag{A.51b}$$

$$Z_X = \begin{bmatrix} \grave{\mathcal{X}}_{k|k-p} \\ \grave{\mathcal{P}}_{k+f-2|k}^{\mathrm{red}} \odot \grave{\mathcal{X}}_{k|k-p} \end{bmatrix} \tag{A.51c}$$

$$\Phi_U = \begin{bmatrix} \Gamma_f \acute{\bar{\mathcal{R}}}_p & \acute{\mathcal{O}}_f^{\mathrm{red}} \otimes \acute{\bar{\mathcal{R}}}_p & \acute{\mathcal{H}}_f^u \end{bmatrix} \tag{A.51d}$$

$$Z_U = \begin{bmatrix} \grave{\mathcal{Z}}_{k-p|k-1} \\ \grave{\mathcal{P}}_{k+f-2|k}^{\mathrm{red}} \odot \grave{\mathcal{Z}}_{k-p|k-1} \\ \grave{\mathcal{U}}_{k+f-1|k} \end{bmatrix} \tag{A.51e}$$

$$\Phi_E = \acute{\mathcal{H}}_f^e \tag{A.51f}$$

$$Z_E = \grave{\mathcal{E}}_{k+f-1|k} \ . \tag{A.51g}$$

The matrix equation (A.50) is similar to this in [Verdult 2002, Section 3.5] for the two-block method, where both windows (past and future) are of equal length, i. e. $p = f$. The name two-block method refers to the fact that the data is split into two sets: the first (past window) is used to approximate the state sequence and the second (future window) is used to compute the model matrices [Verdult 2002, Section 3.3]. In addition, further methods are presented in [Verdult 2002, Section 3.3] with which the state sequence can be more accurately estimated at the expense of computational load.

The second step in SID is to solve for the unknowns in the matrix equation. Following the procedure in [Verdult 2002, Chapter 3], firstly, the transition of the state is neglected at the cost of an approximation error, i. e. $\Phi_X Z_X \approx 0$[2)], and, secondly,

$$\lim_{N \to \infty} \frac{1}{N} Z_E Z_U^{\mathrm{T}} = 0 \tag{A.52}$$

[2)] For an asymptotically stable model $x_{k+1} = \bar{A}(\mu_k) x_k$ and sufficiently large past window p, the initial state x_{k-p} has a negligible effect on the future outputs $Y_{k+f-1|k}$.

holds with probability one, if the noise e is uncorrelated neither with the inputs u nor with the scheduling parameters μ. Consequently, an estimate of the unknowns is

$$\hat{\Phi}_U = Y_{k+f-1|k} Z_U^{\mathrm{T}} (Z_U Z_U^{\mathrm{T}})^{-1} \ . \tag{A.53}$$

The estimation of the states in the third step of SID is performed as follows, see [Verdult 2002, Chapter 3]. The state matrix equation (A.47) is multiplied from the left by Γ_f and an asymptotically stable model is assumed satisfying $\Gamma_f \acute{\mathcal{A}}_p \grave{\mathcal{X}}_{k|k-p} \approx 0$ as in the second step. Then, the state estimate follows from the SVD of the remaining matrix, i. e.

$$\Gamma_f X_k \approx \Gamma_f \acute{\mathcal{R}}_p \grave{\mathcal{Z}}_{k-p|k-1} = \Phi_{U,\mathcal{Z}} \grave{\mathcal{Z}}_{k-p|k-1} \in \mathbb{R}^{n_y f \times N} \tag{A.54}$$

$$\Phi_{U,\mathcal{Z}} \grave{\mathcal{Z}}_{k-p|k-1} \approx \hat{\Phi}_{U,\mathcal{Z}} \grave{\mathcal{Z}}_{k-p|k-1} = \begin{bmatrix} U_1 & U_2 \end{bmatrix} \begin{bmatrix} \Sigma_1 & 0 \\ 0 & \Sigma_2 \end{bmatrix} \begin{bmatrix} V_1^{\mathrm{T}} \\ V_2^{\mathrm{T}} \end{bmatrix} \tag{A.55}$$

$$\hat{X}_k = \Sigma_1^{\frac{1}{2}} V_1^{\mathrm{T}} \ , \tag{A.56}$$

where $\hat{\Phi}_{U,\mathcal{Z}}$ are the first $(n_u + n_y) \sum_{i=1}^{p} n_\mu^i$ columns of $\hat{\Phi}_U$ in (A.53) that are multiplied with $\grave{\mathcal{Z}}_{k-p|k-1}$, the corresponding first rows in Z_U. Note that the estimated states $\hat{X}_k$ in (A.56) can only recover the state sequence X_k up to a similarity transformation.

In the fourth step, the estimated states $\hat{X}_k$ can be used to solve the two LS problems

$$\min_{\acute{A}_1, \acute{\mathcal{R}}_1} \left\| \hat{X}_{k+1} - \acute{A}_1 (P_k \odot \hat{X}_k) - \acute{\mathcal{M}}_1 (P_k \odot U_k) \right\|_{\mathrm{F}}^2 \tag{A.57a}$$

$$\min_{C,D} \left\| Y_k - C \hat{X}_k - D U_k \right\|_{\mathrm{F}}^2 \tag{A.57b}$$

with $\acute{A}_1$ and $\acute{\mathcal{M}}_1$ in (A.41), the signal matrix Y_k that is analogous to P_k in (A.34), and $\|\bullet\|_{\mathrm{F}}$ denotes the Frobenius norm, cf. [Verdult 2002, Section 3.2].

A.3.2 Problem Reduction

The application of subspace methods for LPV models requires, in addition to the solution of two LS problems and the SVD of a matrix with dimensions $n_y f \times N$, the inverse of a square matrix $Z_U Z_U^{\mathrm{T}}$ in (A.53) of dimension

$$(n_y + n_u) \left(\sum_{i=1}^{p} n_\mu^i + (n_\mu^{f-1} - 1) \sum_{j=1}^{p} n_\mu^j \right) + n_u \left(1 + \sum_{i=1}^{f-1} n_\mu^i \right) \ . \tag{A.58}$$

The number of rows in Z_U increases exponentially with window sizes p and f and, in order to achieve full row rank, the number of data samples must be larger than (A.58). Thus, both values p and f should be small. On the other hand, p and f limit the model order (from the state-observability matrix follows $f \geq n_x - 1$) and thus the accuracy[3)].

In [Verdult 2002, Section 3.4] a computational efficient way of implementation is given using QR decomposition. However, Z_U is used in the QR decomposition, which requires full row rank. Therefore, a dimension reduction of the data matrices in Z_U is proposed in [Verdult 2002, Chapter 4].

[3)] Increasing p and f does not automatically mean an improvement in accuracy as discussed in [Verdult 2002, Chapter 6]. Reason for this is the finite data length.

Apart from numerical approaches to problem reduction, structural restrictions can also be imposed. One possibility is to assume some model matrices as scheduling parameter independent, e. g., B or K. Another possibility is full state measurement, which is very rare in practice. However, if the output matrix C has full column rank, the outputs can be used to get an estimate of the states and thus only the two LS problems in (A.57) must be solved.

A different approach for reducing the dimensions is given in [Wingerden & Verhaegen 2009]. It is based on the idea to set the future window to one, i. e. $f = 1$. This means, that only the first n_y rows in (A.50) are considered. Due to the structure of the matrices $\acute{\mathcal{O}}_f$, $\acute{\mathcal{H}}^u_f$, and $\acute{\mathcal{H}}^e_f$, see (A.40), the matrix equation and the matrix Z_U then simplifies to

$$Y_k = C\acute{\mathcal{A}}_p\grave{\mathcal{X}}_{k|k-p} + C\acute{\mathcal{R}}_p\grave{\mathcal{Z}}_{k-p|k-1} + DU_k + E_k \tag{A.59}$$

$$Z_U = \begin{bmatrix} \grave{\mathcal{Z}}_{k-p|k-1} \\ U_k \end{bmatrix} \in \mathbb{R}^{n_u+(n_y+n_u)\sum_{i=1}^{p} n_\mu^i \times N} \; . \tag{A.60}$$

where again the first term $C\acute{\mathcal{A}}_p\grave{\mathcal{X}}_{k|k-p}$ is assumed to be negligible. However, in this case the state estimate in (A.56) obtained from the SVD of the matrix $C\acute{\mathcal{R}}_p$ with n_y rows is limited to a maximum of $n_x = n_y$ states. In order to obtain state estimates with more states than outputs, the matrix $\bar{\Gamma}_p\acute{\mathcal{R}}_p$ is reconstructed from the estimate of $C\acute{\mathcal{R}}_p$. Note that in the previous LPV SID problem formulation the innovation form is used for the future window and therefore Γ_f appears in the matrix equation (A.50) instead. Nevertheless, based on the assumption that $\acute{\mathcal{A}}_p \approx 0$, further multiplications with $\bar{A}_l$ will also tend to zero. Therefore, the lower anti block diagonal entries in $\bar{\Gamma}_p\acute{\mathcal{R}}_p$ are assumed as zero, thus

$$\bar{\Gamma}_p\acute{\mathcal{R}}_p \approx \begin{bmatrix} C\acute{\mathcal{R}}_p & \\ \begin{bmatrix} C\bar{A}_0\acute{\mathcal{R}}_{p-1} & 0 \end{bmatrix} & \\ \vdots & \\ \begin{bmatrix} C\bar{A}_0^{p-1}\acute{\mathcal{R}}_1 & 0 \end{bmatrix} & \end{bmatrix} = \begin{bmatrix} C\acute{\mathcal{M}}_1 & C\acute{\mathcal{M}}_2 & \dots & C\acute{\mathcal{M}}_{p-1} & C\acute{\mathcal{M}}_p \\ C\bar{A}_0\acute{\mathcal{M}}_1 & C\bar{A}_0\acute{\mathcal{M}}_2 & \dots & C\bar{A}_0\acute{\mathcal{M}}_{p-1} & 0 \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ C\bar{A}_0^{p-1}\acute{\mathcal{M}}_1 & 0 & \dots & 0 & 0 \end{bmatrix} . \tag{A.61}$$

The matrices $C\bar{A}_0^i\acute{\mathcal{M}}_j$ with $i \in \{1, 2, \dots, p-1\}$, $j \in \{1, 2, \dots, p-i\}$ can be obtained from the first block in the upper right block matrix $C\bar{A}_0^{i-1}\acute{\mathcal{M}}_{j+1}$, since

$$\begin{aligned} C\bar{A}_0^{i-1}\acute{\mathcal{M}}_{j+1} &= C\bar{A}_0^{i-1}\begin{bmatrix} A_0\acute{\mathcal{M}}_j & A_1\acute{\mathcal{M}}_j & \dots & A_{n_\mu-1}\acute{\mathcal{M}}_j \end{bmatrix} \\ &= \begin{bmatrix} C\bar{A}_0^{i}\acute{\mathcal{M}}_j & C\bar{A}_0^{i-1}A_1\acute{\mathcal{M}}_j & \dots & C\bar{A}_0^{i-1}A_{n_\mu-1}\acute{\mathcal{M}}_j \end{bmatrix} , \end{aligned} \tag{A.62}$$

see the definition of $\acute{\mathcal{M}}_i$ in (A.41). Then, the same procedure for state and model matrices estimation as in the preceding subsection can be followed. This dimension reduction approach requires, however, a sufficient large past window p such that the approximation error of $\acute{\mathcal{A}}_p \approx 0$ becomes small. If the row dimension of Z_U in (A.60) exceed the data samples N (column dimension), it is suggested in [Wingerden & Verhaegen 2009] to solve the dual problem with

$$\min_{\alpha} \|\alpha\|_{\mathrm{F}}^2 \tag{A.63a}$$

$$\text{s. t. } Y_k - \alpha(Z_U^{\mathrm{T}}Z_U) = 0 \; , \tag{A.63b}$$

where $\alpha \in \mathbb{R}^{n_y \times N}$ are the Lagrange multipliers. The solution to this problem is

$$\hat{\alpha} = Y_k (Z_U^{\mathrm{T}} Z_U)^{-1} \ , \tag{A.64}$$

hence the inverse of a quadratic matrix with dimensions N is required. In order to avoid the approximate reconstruction of the huge matrix $\bar{\Gamma}_p \acute{\mathcal{R}}_p$ and the multiplication with matrix $\grave{\mathcal{Z}}_{k-p|k-1}$ another calculation method is introduced in [Wingerden & Verhaegen 2009], in which only square matrices of dimension $(N - p + 1)$ are calculated.

In [Cox & Tóth 2016], an alternative approach for reducing the dimensionality problem is suggested. The SS model is converted into a high-order IO MAX model, which contains the Markov parameters of the model. Thus, the IO model can be identified using the prediction error method and the Markov parameters can be used to construct the Hankel matrix. The Hankel matrix is in the LPV case usually huge (dimensionality problem), so a basis reduced Hankel matrix is generated in [Cox & Tóth 2016]. This matrix is then used to determine the model matrices. Details on Markov parameters, the Hankel matrix, and how to obtain the model matrices from the Hankel matrix are given in Chapter 4.

Bibliography

Abbas, H. S. (2010). "Linear Parameter-Varying Modeling, Identification and Low-Complexity Controller Synthesis". PhD Thesis. Hamburg: Hamburg University of Technology.

Abbas, H. S., R. Tóth, N. Meskin, J. Mohammadpour, & J. Hanema (2015). "An MPC Approach for LPV Systems in Input-Output Form". In: *Proceedings of the Conference on Decision and Control*. IEEE, pp. 91–96.

Abbas, H. S., R. Tóth, & H. Werner (2010). "State-Space Realization of LPV Input-Output Models: Practical Methods for The User". In: *Proceedings of the American Control Conference*. IEEE, pp. 3883–3888.

Abbas, H. S. & H. Werner (2011). "Frequency-Weighted Discrete-Time LPV Model Reduction Using Structurally Balanced Truncation". In: *IEEE Transactions on Control Systems Technology* 19.1, pp. 140–147.

Adegas, F. D. & J. Stoustrup (2011). "Structured Control of Affine Linear Parameter Varying Systems". In: *Proceedings of the American Control Conference*. IEEE, pp. 739–744.

Ali, M., H. S. Abbas, & H. Werner (2010). "Controller Synthesis for Input-Output LPV Models". In: *Proceedings of the Conference on Decision and Control*. IEEE, pp. 7694–7699.

Ali, M., A. Ali, H. S. Abbas, & H. Werner (2011). "Identification of Box-Jenkins Models for Parameter-Varying Spatially Interconnected Systems". In: *Proceedings of the American Control Conference*. IEEE, pp. 145–150.

Antoulas, A. C. (2008). *Approximation of Large-Scale Dynamical Systems*. Society for Industrial and Applied Mathematics.

Antoulas, A. C. & D. C. Sorensen (2001). "Approximation of large-scale dynamical systems: An overview". In: *International Journal of Applied Mathematics and Computer Science* 11, pp. 1093–1121.

Apkarian, P. & R. J. Adams (1998). "Advanced Gain-Scheduling Techniques for Uncertain Systems". In: *IEEE Transactions on Control Systems Technology* 6.1, pp. 21–32.

Apkarian, P., G. Becker, P. Gahinet, & H. Kajiwara (1996). "LMI Techniques in Control Engineering from Theory to Practice". In: *Proceedings of the Conference on Decision and Control - Workshop Notes*. IEEE. Kobe, Japan.

Apkarian, P. & P. Gahinet (1995). "A Convex Characterization of Gain-Scheduled $\mathcal{H}_\infty$ Controllers". In: *IEEE Transactions on Automatic Control* 40.5, pp. 853–864.

Apkarian, P., P. Gahinet, & G. Becker (1995). "Self-scheduled $\mathcal{H}_\infty$ control of linear parameter-varying systems: a design example". In: *Automatica* 31 (9), pp. 1251–1261.

Apkarian, P., P. C. Pellanda, & H. D. Tuan (2000). "Mixed $\mathcal{H}_2/\mathcal{H}_\infty$ multi-channel linear parameter-varying control in discrete time". In: *Systems & Control Letters* 41.5, pp. 333–346.

Bachnas, A. A., R. Tóth, J. H. A. Ludlage, & A. Mesbah (2014). "A review on data-driven linear parameter-varying modeling approaches: A high-purity distillation column case study". In: *Journal of Process Control* 24.4, pp. 272–285.

Bamieh, B. & L. Giarré (2001). "LPV models: Identification for gain scheduling control". In: *Proceedings of the European Control Conference*, pp. 3092–3097.

Bamieh, B. & L. Giarré (2002). "Identification of linear parameter varying models". In: *International Journal of Robust and Nonlinear Control* 12.9, pp. 841–853.

Bartels, M., Q. Liu, G. Kaiser, & H. Werner (2013). "LPV Torque Vectoring for an Electric Vehicle Using Parameter-Dependent Lyapunov Functions". In: *Proceedings of the American Control Conference*. IEEE, pp. 2153–2158.

Barzilai, J. & J. M. Borwein (1988). "Two-Point Step Size Gradient Methods". In: *IMA Journal of Numerical Analysis* 8.1, pp. 141–148.

Beck, C. (2006). "Coprime factors reduction methods for linear parameter varying and uncertain systems". In: *Systems & Control Letters* 55.3, pp. 199–213.

Beck, C. L., J. Doyle, & K. Glover (1996). "Model Reduction of Multidimensional and Uncertain Systems". In: *IEEE Transactions on Automatic Control* 41.10, pp. 1466–1477.

Becker, G., A. Packard, D. Philbrick, & G. Balas (1993). "Control of Parametrically-Dependent Linear Systems: A Single Quadratic Lyapunov Approach". In: *Proceedings of the American Control Conference*. IEEE, pp. 2795–2799.

Beckmann, R. (2015). "Beitrag zur exakten Füllungssteuerung am aufgeladenen Ottomotor". PhD Thesis. Rostock: Universität Rostock.

Besselmann, T., J. Löfberg, & M. Morari (2012). "Explicit MPC for LPV Systems: Stability and Optimality". In: *IEEE Transactions on Automatic Control* 57.9, pp. 2322–2332.

Bodenheimer, B., P. Bendotti, & M. Kantner (1995). "Linear parameter-varying control of a ducted fan engine". In: *International Journal of Robust and Nonlinear Control* 6.9-10, pp. 1023–1044.

Boonto, S. (2011). "Identification of Linear Parameter-Varying Input-Output Models". PhD Thesis. Hamburg: Hamburg University of Technology.

Boos, D. D. (2003). "Introduction to the Bootstrap World". In: *Statistical Science* 18.2, pp. 168–174.

Briat, C. (2015). *Linear Parameter-Varying and Time-Delay Systems: Analysis, Observation, Filtering & Control*. Springer Berlin Heidelberg.

Buchholz, M. & W. E. Larimore (2013). "Subspace Identification of an Aircraft Linear Parameter-Varying Flutter Model". In: *Proceedings of the American Control Conference*. IEEE, pp. 2263–2267.

Bussa, A. (2016). "LPV System Identification for the Air-Path of a Gasoline Engine with I/O and Subspace Identification Methods". Supervised by Schulz, Erik and Werner, Herbert. Master's Thesis. Hamburg, Germany and Gifhorn, Germany: Hamburg University of Technology and IAV GmbH.

Butcher, M., A. Karimi, & R. Longchamp (2008). "On the Consistency of Certain Identification Methods for Linear Parameter Varying Systems". In: *Proceedings of the World Congress*. IFAC, pp. 4018–4023.

Cerone, V., D. Piga, D. Regruto, & R. Tóth (2012). "Input-output LPV model identification with guaranteed quadratic stability". In: *Proceedings of the Symposium on System Identification*. IFAC, pp. 1767–1772.

Chikkula, Y. & J. H. Lee (1997). "Input Sequence Design for Parametric Identification of Nonlinear Systems". In: *Proceedings of the American Control Conference*. Vol. 5. IEEE, pp. 3037–3041.

Cisneros, P. G. & H. Werner (2017). "Fast Nonlinear MPC for Reference Tracking Subject to Nonlinear Constraints via Quasi-LPV Representations". In: *Proceedings of the World Congress*. IFAC, pp. 12106–12111.

Corno, M., M. Tanelli, S. M. Savaresi, & L. Fabbri (2010). “Design and Validation of a Gain-Scheduled Controller for the Electronic Throttle Body in Ride-by-Wire Racing Motorcycles”. In: *IEEE Transactions on Control Systems Technology* 19.1, pp. 18–30.

Cox, P. (2018). “Towards Efficient Identification of Linear Parameter-Varying State-Space Models”. PhD Thesis. Eindhoven: Eindhoven University of Technology.

Cox, P. & R. Tóth (2016). “Alternative Form of Predictor Based Identification of LPV-SS Models with Innovation Noise”. In: *Proceedings of the Conference on Decision and Control*. IEEE, pp. 1223–1228.

Cox, P., R. Tóth, & M. Petreczky (2015). “Estimation of LPV-SS Models with Static Dependency using Correlation Analysis”. In: *Proceedings of the Workshop on Linear Parameter Varying Systems*. IFAC, pp. 91–96.

Cox, P., S. Weiland, & R. Tóth (2018). “Affine Parameter-Dependent Lyapunov Functions for LPV Systems With Affine Dependence”. In: *IEEE Transactions on Automatic Control* 63.11, pp. 3865–3872.

Daafouz, J. & J. Bernussou (2001). “Parameter dependent Lyapunov functions for discrete time systems with time varying parametric uncertainties”. In: *Systems & Control Letters* 43.5, pp. 355–359.

Daasch, A., E. Schulz, & M. Schultalbers (2016). “Intrinsic Performance Limitations of Torque Generation in a Turbocharged Gasoline Engine”. In: *Proceedings of the International Symposium on Advances in Automotive Control*. IFAC, pp. 714–721.

Darwish, M., P. Cox, G. Pillonetto, & R. Tóth (2015). “Bayesian Identification of LPV Box-Jenkins Models”. In: *Proceedings of the Conference on Decision and Control*. IEEE, pp. 66–71.

Datar, A., E. Schulz, & H. Werner (2018). “Identification of Linear Parameter-Varying Models with Unknown Parameter Dependence Using ε-Support Vector Regression”. In: *Proceedings of the American Control Conference*. IEEE, pp. 2011–2016.

De Caigny, J., J. F. Camino, R. C. L. F. Oliveira, P. L. D. Peres, & J. Swevers (2010). “Gain-scheduled $\mathcal{H}_2$ and $\mathcal{H}_\infty$ control of discrete-time polytopic time-varying systems”. In: *IET Control Theory and Applications* 4.3, pp. 362–380.

De Caigny, J., J. F. Camino, R. C. L. F. Oliveira, P. L. D. Peres, & J. Swevers (2012). “Gain-scheduled dynamic output feedback control for discrete-time LPV systems”. In: *International Journal of Robust and Nonlinear Control* 22.5, pp. 535–558.

De Schutter, B. (2000). “Minimal state-space realization in linear system theory: an overview”. In: *Journal of Computational and Applied Mathematics* 121.1, pp. 331–354.

Desoer, C. A. & M. Vidyasagar (1975). *Feedback Systems: Input-Output Properties*. New York: Academic Press.

Dettori, M. (2001). “LMI Techniques for Control: with application to a Compact Disc player mechanism”. PhD Thesis. Delft: Delft University of Technology.

Emedi, Z. & A. Karimi (2014). “Robust Fixed-order Discrete-time LPV Controller Design”. In: *Proceedings of the World Congress*. IFAC, pp. 6914–6919.

Emedi, Z. & A. Karimi (2016). “Fixed-structure LPV discrete-time controller design with induced $\mathcal{L}_2$-norm and $\mathcal{H}_2$ performance”. In: *International Journal of Control* 89.3, pp. 494–505.

Eriksson, L. & L. Nielsen (2014). *Modeling and Control of Engines and Drivelines*. John Wiley & Sons.

Farhood, M. & G. E. Dullerud (2007). “Model Reduction of Nonstationary LPV Systems”. In: *IEEE Transactions on Automatic Control* 52.2, pp. 181–196.

Favoreel, W., B. De Moor, & P. Van Overschee (2000). "Subspace state space system identification for industrial processes". In: *Journal of Process Control* 10.2-3, pp. 149–155.

Feron, E., P. Apkarian, & P. Gahinet (1995). "S-procedure for the analysis of control systems with parametric uncertainties via parameter-dependent Lyapunov functions". In: *Proceedings of the American Control Conference*. IEEE.

Formentin, S., D. Piga, R. Tóth, & S. M. Savaresi (2016). "Direct learning of LPV controllers from data". In: *Automatica* 65, pp. 98–110.

Franceschini, G. & S. Macchietto (2008). "Model-based design of experiments for parameter precision: State of the art". In: *Chemical Engineering Science* 63.19, pp. 4846–4872.

Gahinet, P. & P. Apkarian (1994). "A Linear Matrix Inequality Approach to $\mathcal{H}_\infty$ Control". In: *International Journal of Robust and Nonlinear Control* 4.4, pp. 421–448.

Gahinet, P., P. Apkarian, & M. Chilali (1994). "Parameter-Dependent Lyapunov Functions for Real Parametric Uncertainty". In: *IEEE Transactions on Automatic Control*.

Genç, A. U. (2002). "Linear Parameter-Varying Modelling and Robust Control of Variable Cam Timing Engines". PhD Thesis. Cambridge: University of Cambridge.

Gevers, M., A. S. Bazanella, X. Bombois, & L. Mišković (2009). "Identification and the Information Matrix: How to Get Just Sufficiently Rich?" In: *IEEE Transactions on Automatic Control* 54.12, pp. 2828–2840.

Giarré, L., D. Bauso, P. Falugi, & B. Bamieh (2006). "LPV model identification for gain scheduling control: An application to rotating stall and surge control problem". In: *Control Engineering Practice* 14.4, pp. 351–361.

Glover, K. (1984). "All optimal Hankel-norm approximations of linear multivariable systems and their $\mathcal{L}_\infty$-error bounds". In: *International Journal of Control* 39.6, pp. 1115–1193.

Golub, G. H. & C. F. Van Loan (2013). *Matrix computations*. 4th ed. The Johns Hopkins University Press.

Gorinevsky, D. (1995). "On the Persistency of Excitation in Radial Basis Function Network Identification of Nonlinear Systems". In: *IEEE Transactions on Neural Networks* 6.5, pp. 1237–1244.

Gringard, M. & A. Kroll (2016). "On the parametrization of APRBS and multisine test signals for the identification of nonlinear dynamic TS-models". In: *Symposium Series on Computational Intelligence*. IEEE, pp. 1–8.

Gu, G. (2012). *Discrete-Time Linear Systems: Theory and Design with Applications*. Springer.

Gugercin, S. & A. C. Antoulas (2000). "A Comparative Study of 7 Algorithms for Model Reduction". In: *Proceedings of the Conference on Decision and Control*. Vol. 3. IEEE, pp. 2367–2372.

Gunes, B. (2018). "A Tensor Approach to Linear Parameter Varying System Identification". PhD Thesis. Delft: Delft University of Technology.

Gunes, B., J.-W. van Wingerden, & M. Verhaegen (2018). "Tensor networks for MIMO LPV system identification". In: *International Journal of Control*, pp. 1–15.

Guzzella, L. & C. Onder (2010). *Introduction to Modeling and Control of Internal Combustion Engine Systems*. 2nd ed. Springer Berlin Heidelberg.

He, X. & H. Asada (1993). "A New Method for Identifying Orders of Input-Output Models for Nonlinear Dynamic Systems". In: *Proceedings of the American Control Conference*. IEEE, pp. 2520–2523.

Heeren, L. & H. Werner (2020). "Grid-Free Model Order Reduction for Linear Parameter-Varying Systems via Full Block S-Procedure". In: *Proceedings of the European Control Conference*. IEEE, pp. 1579–1584.

Ho, B. L. & R. E. Kalman (1966). “Effective construction of linear state-variable models from input/output functions”. In: *at-Automatisierungstechnik Methoden und Anwendungen der Steuerungs-, Regelungs- und Informationstechnik* 14.1-12, pp. 545–548.

Hoffmann, C. (2016). “Linear Parameter-Varying Control of Systems of High Complexity”. PhD Thesis. Hamburg: Hamburg University of Technology.

Hoffmann, C. & H. Werner (2015). “A Survey of Linear Parameter-Varying Control Applications Validated by Experiments or High-Fidelity Simulations”. In: *IEEE Transactions on Control Systems Technology* 23.2, pp. 416–433.

Hyde, R. A. & K. Glover (1991). “A comparison of different scheduling techniques for $\mathcal{H}_\infty$ controllers”. In: *Transactions of the Institute of Measurement and Control* 13.5, pp. 227–232.

Hyde, R. A. & K. Glover (1993). “The Application of Scheduled $\mathcal{H}_\infty$ Controllers to a VSTOL Aircraft”. In: *IEEE Transactions on Automatic Control* 38.7, pp. 1021–1039.

Isermann, R. (2014). *Engine Modeling and Control: Modeling and Electronic Management of Internal Combustion Engines*. Springer Berlin Heidelberg.

Isermann, R. & M. Münchhof (2011). *Identification of Dynamic Systems: An Introduction with Applications*. Springer Berlin Heidelberg.

Jaimoukha, I. M., H. El-Zobaidi, D. J. N. Limebeer, & N. Shah (2005). “Controller reduction for linear parameter-varying systems with a priori bounds”. In: *Automatica* 41.2, pp. 273–279.

Jung, M. & K. Glover (2003). “Control-Oriented Linear Parameter-Varying Modelling of a Turbocharged Diesel Engine”. In: *Proceedings of the Conference on Control Applications*. Vol. 1. IEEE, pp. 155–160.

Jung, M. & K. Glover (2006). “Calibratable Linear Parameter-Varying Control of a Turbocharged Diesel Engine”. In: *IEEE Transactions on Control Systems Technology* 14.1, pp. 45–62.

Karimi, A. & I. D. Landau (1998). “Comparison of the closed-loop identification methods in terms of the bias distribution”. In: *Systems & Control Letters* 34.4, pp. 159–167.

Kettlitz, S. (2018). “Identifikation lokal linearer Modelle am Beispiel der Ladedruckstrecke”. Supervised by Schulz, Erik and King, Rudibert and Arnold, Florian. Master’s Thesis. Berlin, Germany and Gifhorn, Germany: Technische Universität Berlin and IAV GmbH.

Khatri, C. G. & C. R. Rao (1968). “Solutions to some functional equations and their applications to characterization of probability distributions”. In: *Sankhyā: The Indian Journal of Statistics, Series A*, pp. 167–180.

Koelewijn, P. J. W., G. S. Mazzoccante, R. Tóth, & S. Weiland (2020). “Pitfalls of Guaranteeing Asymptotic Stability in LPV Control of Nonlinear Systems”. In: *Proceedings of the European Control Conference*. IEEE, pp. 1573–1578.

Koelewijn, P. J. W., R. Tóth, & H. Nijmeijer (2019). “Linear Parameter-Varying Control of Nonlinear Systems based on Incremental Stability”. In: *Proceedings of the Workshop on Linear Parameter Varying Systems*. Vol. 52. 28. IFAC. Elsevier, pp. 38–43.

Kolmanovsky, I., P. Moraal, M. Van Nieuwstadt, & A. Stefanopoulou (1999). “Issues in Modelling and Control of Intake Flow in Variable Geometry Turbocharged Engines”. In: *Proceedings of the Conference on System Modelling and Optimization*. IFIP TC7, pp. 436–445.

Kominek, A., S. Remolina, S. Boonto, H. Werner, M. Garwon, & M. Schultalbers (2012). “Low-Complexity LPV Input-Output Identification and Control of a Turbocharged Combustion Engine”. In: *Proceedings of the Conference on Decision and Control*. IEEE, pp. 4492–4497.

Kulcsár, B., J. Dong, J.-W. van Wingerden, & M. Verhaegen (2009). “LPV Subspace Identification of a DC motor with unbalanced disc”. In: *Proceedings of the Symposium on System Identification*. IFAC, pp. 856–861.

Kung, S.-Y. (1978). “A new identification and model reduction algorithm via singular value decompositions”. In: *Proceedings of the Asilomar Conference on Circuits, Systems and Computers*, pp. 705–714.

Kwiatkowski, A. (2007). “LPV Modeling and Application of LPV Controllers to SI Engines”. PhD Thesis. Hamburg: Hamburg University of Technology.

Kwiatkowski, A. & H. Werner (2008). “PCA-Based Parameter Set Mappings for LPV Models With Fewer Parameters and Less Overbounding”. In: *IEEE Transactions on Control Systems Technology* 16.4, pp. 781–788.

Kwiatkowski, A., H. Werner, J. P. Blath, A. Ali, & M. Schultalbers (2009). “Linear parameter varying PID controller design for charge control of a spark-ignited engine”. In: *Control Engineering Practice* 17.11, pp. 1307–1317.

Lall, S. & C. Beck (2003). “Error-Bounds for Balanced Model-Reduction of Linear Time-Varying Systems”. In: *IEEE Transactions on Automatic Control* 48.6, pp. 946–956.

Laub, A. J., M. T. Heath, C. C. Paige, & R. C. Ward (1987). “Computation of System Balancing Transformations and Other Applications of Simultaneous Diagonalization Algorithms”. In: *IEEE Transactions on Automatic Control* 32.2, pp. 115–122.

Laurain, T. (2017). “Advanced controller design for quasi-LPV systems applied to automotive engine control”. PhD Thesis. Valenciennes: Université de Valenciennes et du Hainaut-Cambrésis.

Laurain, V., M. Gilson, R. Tóth, & H. Garnier (2010). “Refined instrumental variable methods for identification of LPV Box-Jenkins models”. In: *Automatica* 46.6, pp. 959–967.

Laurain, V., R. Tóth, M. Gilson, & H. Garnier (2011). “Direct identification of continuous-time linear parameter-varying input/output models”. In: *IET Control Theory and Applications* 5.7, pp. 878–888.

Lee, L. H. & K. Poolla (1999). “Identification of Linear Parameter-Varying Systems using Nonlinear Programming”. In: *Journal of Dynamic Systems, Measurement, and Control* 121, pp. 71–78.

Leith, D. J. & W. E. Leithead (1999). *Comments On the Prevalence of Linear Parameter Varying Systems*. Tech. rep. University of Strathclyde.

Leith, D. J. & W. E. Leithead (2000a). “On Formulating Nonlinear Dynamics in LPV Form”. In: *Proceedings of the Conference on Decision and Control*. Vol. 4. IEEE, pp. 3526–3527.

Leith, D. J. & W. E. Leithead (2000b). “Survey of gain-scheduling analysis and design”. In: *International Journal of Control* 73.11, pp. 1001–1025.

Ljung, L. (1999). *System Identification: Theory for the User*. 2nd ed. Prentice Hall PTR.

Lu, Q., H. R. Karimi, & K. G. Robbersmyr (2013). “A Data-Based Approach for Modeling and Analysis of Vehicle Collision by LPV-ARMAX Models”. In: *Journal of Applied Mathematics*, pp. 1–9.

Magni, J.-F. (2006). *Linear Fractional Representation toolbox for use with Matlab*. Available with the SMAC Toolbox at `http://w3.onera.fr/smac/lfrt`.

Marcos, A. & G. J. Balas (2004). “Development of Linear-Parameter-Varying Models for Aircraft”. In: *Journal of Guidance, Control, and Dynamics* 27.2, pp. 218–228.

Mohammadpour, J. & C. Scherer (2012). *Control of Linear Parameter Varying Systems with Applications*. Springer US.

Nelles, O. (2001). *Nonlinear System Identification: From Classical Approaches to Neural Networks and Fuzzy Models*. Springer Berlin Heidelberg.

Nelles, O. & R. Isermann (1995). “Identification of Nonlinear Dynamic Systems – Classical Methods versus Radial Basis Function Networks”. In: *Proceedings of the American Control Conference*. Vol. 5. IEEE, pp. 3786–3790.

Nemani, M., R. Ravikanth, & B. Bamieh (1995). “Identification of linear parametrically varying systems”. In: *Proceedings of the Conference on Decision and Control*. Vol. 3. IEEE, pp. 2990–2995.

Nichols, R. A., R. T. Reichert, & W. J. Rugh (1993). “Gain Scheduling for $\mathcal{H}_\infty$ Controllers: A Flight Control Example”. In: *IEEE Transactions on Control systems technology* 1.2, pp. 69–79.

Nocedal, J. & S. J. Wright (2006). *Numerical Optimization*. 2nd ed. Springer New York.

Ogunfunmi, T. (2007). *Adaptive Nonlinear System Identification: The Volterra and Wiener Model Approaches*. Springer.

Oliveira, M. C. de, J. C. Geromel, & J. Bernussou (2002). “Extended $\mathcal{H}_2$ and $\mathcal{H}_\infty$ norm characterizations and controller parametrizations for discrete-time systems”. In: *International Journal of Control* 75.9, pp. 666–679.

Oliveira, R. C. L. F. & P. L. D. Peres (2008). “Robust stability analysis and control design for time-varying discrete-time polytopic systems with bounded parameter variation”. In: *Proceedings of the American Control Conference*. IEEE, pp. 3094–3099.

Overschee, P. van & B. de Moor (1996). *Subspace Identification for Linear Systems: Theory – Implementation – Applications*. Springer US.

Packard, A., G. Becker, D. Philbrick, & G. Balas (1993). “Control of Parameter-Dependent Systems: Applications to $\mathcal{H}_\infty$ Gain-Scheduling”. In: *Proceedings of the Conference on Aerospace Control Systems*. IEEE, pp. 329–333.

Packard, A. & M. Kantner (1996). “Gain Scheduling the LPV Way”. In: *Proceedings of the Conference on Decision and Control*. IEEE, pp. 3938–3941.

Petreczky, M., R. Tóth, & G. Mercère (2016). “Realization Theory for LPV State-Space Representations With Affine Dependence”. In: *IEEE Transactions on Automatic Control* 62.9, pp. 4667–4674.

Pischinger, S. & U. Seiffert, eds. (2016). *Vieweg Handbuch Kraftfahrzeugtechnik*. 8th ed. Springer Vieweg.

Plastino, A. R. & J. C. Muzzio (1992). “On the use and abuse of Newton’s second law for variable mass problems”. In: *Celestial Mechanics and Dynamical Astronomy* 53.3, pp. 227–232.

Proimadis, I., H. J. Bijl, & J.-W. van Wingerden (2015). “A kernel based approach for LPV subspace identification”. In: *Proceedings of the Workshop on Linear Parameter Varying Systems*. Vol. 48. 26. IFAC. Elsevier, pp. 97–102.

Rawlings, J. B. & D. Q. Mayne (2016). *Model Predictive Control: Theory and Design*. 2nd ed. Nob Hill Publishing.

Rizvi, S. Z., F. Abbasi, & J. M. Velni (2018). “Model Reduction in Linear Parameter-Varying Models using Autoencoder Neural Networks”. In: *Proceedings of the American Control Conference*. IEEE, pp. 6415–6420.

Rizvi, S. Z., J. Mohammadpour, R. Tóth, & N. Meskin (2016). “A Kernel-Based PCA Approach to Model Reduction of Linear Parameter-Varying Systems.” In: *IEEE Transactions on Control Systems Technology* 24.5, pp. 1883–1891.

Rugh, W. J. & J. S. Shamma (2000). “Research on gain scheduling”. In: *Automatica* 36.10, pp. 1401–1425.

Safonov, M. G., R. Y. Chiang, & D. J. N. Limebeer (1990). “Optimal Hankel Model Reduction for Nonminimal Systems”. In: *IEEE Transactions on Automatic Control* 35.4, pp. 496–502.

Salcedo, J. V. & M. Martínez (2008). “LPV identification of a turbocharged diesel engine”. In: *Applied Numerical Mathematics* 58.10, pp. 1553–1571.

Sandberg, H. & A. Rantzer (2004). "Balanced Truncation of Linear Time-Varying Systems". In: *IEEE Transactions on Automatic Control* 49.2, pp. 217–229.

Saupe, F. & H. Pfifer (2011). "Applied LPV Control Exploiting the Separation Principle for the Single Axis Positioning of an Industrial Manipulator". In: *International Conference on Control Applications*. IEEE, pp. 1476–1481.

Scherer, C. (2001). "LPV control and full block multipliers". In: *Automatica* 37.3, pp. 361–375.

Scherer, C. (2013). "Gain-scheduled synthesis with dynamic stable strictly positive real multipliers: A complete solution". In: *Proceedings of the European Control Conference*, pp. 3901–3906.

Schoukens, J. & L. Ljung (2019). "Nonlinear System Identification: A User-Oriented Road Map". In: *IEEE Control Systems Magazine* 39.6, pp. 28–99.

Schulz, E., A. Bussa, & H. Werner (2016). "Identification of Linear Parameter-Varying Systems via IO and Subspace Identification – A Comparison". In: *Proceedings of the Conference on Decision and Control*. IEEE, pp. 7147–7152.

Schulz, E., P. Cox, R. Tóth, & H. Werner (2017). "LPV State-Space Identification via IO Methods and Efficient Model Order Reduction in Comparison with Subspace Methods". In: *Proceedings of the Conference on Decision and Control*. IEEE, pp. 3575–3581.

Schulz, E., O. Janda, & M. Schultalbers (2018). "On LPV System Identification Algorithms for Input-Output Model Structures and their Relation to LTI System Identification". In: *Proceedings of the Symposium on System Identification*. IFAC, pp. 1098–1103.

Scorletti, G. & L. E. Ghaoui (1998). "Improved LMI Conditions for Gain Scheduling and Related Control Problems". In: *International Journal of Robust and Nonlinear Control* 8.10, pp. 845–877.

Sehnke, T., M. Schultalbers, & R. Ernst (2018). "Temporal Properties in Component-Based Cyber-Physical Systems". In: *Proceedings of the European Congress Embedded Real Time Software and Systems*.

Shamma, J. S. (1988). "Analysis and Design of Gain-Scheduled Control Systems". PhD Thesis. Cambridge: Massachusetts Institute of Technology.

Shamma, J. S. (2012). "An Overview of LPV Systems". In: *Control of Linear Parameter Varying Systems with Applications*. Springer, pp. 3–26.

Shamma, J. S. & M. Athans (1991). "Guaranteed Properties of Gain Scheduled Control for Linear Parameter-varying Plants". In: *Automatica* 27.3, pp. 559–564.

Shamma, J. S. & M. Athans (1992). "Gain Scheduling: Potential Hazards and Possible Remedies". In: *IEEE Control Systems Magazine* 12.3, pp. 101–107.

Shamma, J. S. & J. R. Cloutier (1993). "Gain-Scheduled Missile Autopilot Design Using Linear Parameter Varying Transformations". In: *Journal of Guidance, Control, and Dynamics* 16.2, pp. 256–263.

Silverman, L. M. (1971). "Realization of Linear Dynamical Systems". In: *IEEE Transactions on Automatic Control* 16.6, pp. 554–567.

Silverman, L. M. & H. E. Meadows (1969). "Equivalent Realizations of Linear Systems". In: *SIAM Journal on Applied Mathematics* 17.2, pp. 393–408.

Sima, D. (2006). "Regularization Techniques in Model Fitting and Parameter Estimation". PhD Thesis. Leuven: Katholieke Universiteit Leuven.

Siraj, M. M., R. Tóth, & S. Weiland (2012). "Joint order and dependency reduction for LPV state-space models". In: *Proceedings of the Conference on Decision and Control*. IEEE, pp. 6291–6296.

Skogestad, S. & I. Postlethwaite (2007). *Multivariable Feedback Control: Analysis and Design*. 2nd ed. Wiley New York.

Söderström, T. & P. Stoica (1989). *System Identification*. Prentice Hall.

Theis, J. (2018). "Robust and Linear Parameter-Varying Control of Aeroservoelastic Systems". PhD Thesis. Hamburg: Hamburg University of Technology.

Theis, J., P. Seiler, & H. Werner (2018). "LPV Model Order Reduction by Parameter-Varying Oblique Projection". In: *IEEE Transactions on Control Systems Technology* 26.3, pp. 773–784.

Thiel, F. M. (2019). "Adaptive Control of Plants with Input Saturation: An Approach for Performance Improvement". PhD Thesis. Rostock: Universität Rostock.

Thiel, M., D. Schwarzmann, M. Schultalbers, & T. Jeinsch (2016). "Indirect Adaptive Pole Placement Control with Performance Orientated Anti-Windup for Electronic Throttle Plates". In: *Proceedings of the Mediterranean Conference on Control and Automation*. IEEE, pp. 461–466.

Tóth, R. (2010). *Modeling and Identification of Linear Parameter-Varying Systems*. Springer Berlin Heidelberg.

Tóth, R., H. S. Abbas, & H. Werner (2012). "On the State-Space Realization of LPV Input-Output Models: Practical Approaches". In: *IEEE Transactions on Control Systems Technology* 20.1, pp. 139–153.

Tóth, R., F. Felici, P. S. C. Heuberger, & P. M. J. Van den Hof (2007). "Discrete time LPV I/O and State Space Representations, Differences of Behavior and Pitfalls of Interpolation". In: *Proceedings of the European Control Conference*. IEEE, pp. 5418–5425.

Tóth, R., P. S. C. Heuberger, & P. M. J. Van den Hof (2012). "Prediction-Error Identification of LPV Systems: Present and Beyond". In: *Control of Linear Parameter Varying Systems with Applications*. Ed. by J. Mohammadpour & C. Scherer. Boston, MA: Springer US. Chap. 2, pp. 27–58.

Tóth, R., V. Laurain, W. X. Zheng, & K. Poolla (2011). "Model Structure Learning: A Support Vector Machine Approach for LPV Linear-Regression Models". In: *Proceedings of the Conference on Decision and Control and the European Control Conference*. IEEE, pp. 3192–3197.

Van Den Hof, P. M. J. & R. J. P. Schrama (1995). "Identification and Control – Closed-loop Issues". In: *Automatica* 31.12, pp. 1751–1770.

Verdult, V. (2002). "Nonlinear System Identification: A State-Space Approach". PhD Thesis. Enschede: University of Twente.

Verdult, V., M. Lovera, & M. Verhaegen (2004). "Identification of linear parameter-varying state-space models with application to helicopter rotor dynamics". In: *International Journal of Control* 77.13, pp. 1149–1159.

Verdult, V. & M. Verhaegen (2000). "Identification of Multivariable Linear Parameter-Varying Systems Based on Subspace Techniques". In: *Proceedings of the Conference on Decision and Control*. Vol. 2. IEEE, pp. 1567–1572.

Verdult, V. & M. Verhaegen (2002). "Subspace identification of multivariable linear parameter-varying systems". In: *Automatica* 38.5, pp. 805–814.

Villemagne, C. d. & R. E. Skelton (1987). "Model reductions using a projection formulation". In: *International Journal of Control* 46.6, pp. 2141–2169.

Wahlberg, B., H. Hjalmarsson, & M. Annergren (2010). "On Optimal Input Design in System Identification for Control". In: *Proceedings of the Conference on Decision and Control*. IEEE, pp. 5548–5553.

Wahlström, J. & L. Eriksson (2011). "Modelling diesel engines with a variable-geometry turbocharger and exhaust gas recirculation by optimization of model parameters for capturing non-linear system dynamics". In: *Proceedings of the Institution of Mechanical Engineers, Part D: Journal of Automobile Engineering* 225.7, pp. 960–986.

Wassink, M. G., M. van de Wal, C. Scherer, & O. Bosgra (2005). "LPV control for a wafer stage: beyond the theoretical solution". In: *Control Engineering Practice* 13.2, pp. 231–245.

Wei, X. & L. del Re (2006). "On Persistent Excitation for Parameter Estimation of Quasi-LPV Systems and its Application in Modeling of Diesel Engine Torque". In: *Proceedings of the Symposium on System Identification*. IFAC, pp. 517–522.

Wei, X. & L. del Re (2007). "Gain Scheduled $\mathcal{H}_\infty$ Control for Air Path Systems of Diesel Engines Using LPV Techniques". In: *IEEE Transactions on Control Systems Technology* 15.3, pp. 406–415.

White, A. P., G. Zhu, & J. Choi (2013). *Linear Parameter-Varying Control for Engineering Applications*. Springer.

Willems, J. C. (2007). "The Behavioral Approach to Open and Interconnected Systems". In: *IEEE Control Systems Magazine* 27.6, pp. 46–99.

Wills, A. & B. Ninness (2011). "System Identification of Linear Parameter Varying State-space Models". In: *Linear Parameter-Varying System Identification: New Developments and Trends*. Ed. by P. L. dos Santos, T. P. A. Perdicoúlis, C. Novara, J. A. Ramos, & D. E. Rivera. World Scientific. Chap. 11, pp. 295–316.

Wingerden, J.-W. van & M. Verhaegen (2009). "Subspace identification of Bilinear and LPV systems for open- and closed-loop data". In: *Automatica* 45.2, pp. 372–381.

Wollnack, S. (2018). "Implicit Representation-Based Fixed-Structure Control of LPV Systems". PhD Thesis. Hamburg: Hamburg University of Technology.

Wollnack, S., H. S. Abbas, H. Werner, & R. Tóth (2013). "Fixed-Structure LPV Controller Synthesis Based on Implicit Input Output Representations". In: *Proceedings of the Conference on Decision and Control*. IEEE, pp. 2103–2108.

Wollnack, S. & H. Werner (2016). "LPV-IO Controller Design: An LMI Approach". In: *Proceedings of the American Control Conference*. IEEE, pp. 4617–4622.

Wood, G. D. (1995). "Control of Parameter-Dependent Mechanical Systems". PhD Thesis. University of Cambridge.

Wood, G. D., P. J. Goddard, & K. Glover (1996). "Approximation of Linear Parameter-Varying Systems". In: *Proceedings of the Conference on Decision and Control*. Vol. 1. IEEE, pp. 406–411.

Wu, F. (1995). "Control of Linear Parameter Varying Systems". PhD Thesis. Berkeley: University of California.

Wu, F. (1996). "Induced $\mathcal{L}_2$ Norm Model Reduction of Polytopic Uncertain Linear Systems". In: *Automatica* 32.10, pp. 1417–1426.

Wu, F. & K. Dong (2006). "Gain-scheduling control of LFT systems using parameter-dependent Lyapunov functions". In: *Automatica* 42.1, pp. 39–50.

Yu, Z., H. Chen, & P.-y. Woo (2002). "Gain Scheduled LPV $\mathcal{H}_\infty$ Control Based on LMI Approach for a Robotic Manipulator". In: *Journal of Robotic Systems* 19.12, pp. 585–593.

Zhang, L. & P. Shi (2008). "$\mathcal{L}_2 - \mathcal{L}_\infty$ Model Reduction for Switched LPV Systems With Average Dwell Time". In: *IEEE Transactions on Automatic Control* 53.10, pp. 2443–2448.

Zhou, K., J. C. Doyle, & K. Glover (1996). *Robust and Optimal Control*. Prentice Hall.

Acronyms

AF	augmented form	25
AR	autoregressive	40
ARMAX	autoregressive moving average with exogenous input	39
ARX	autoregressive with exogenous input	39
BFR	best fit rate	37
BJ	Box-Jenkins	40
CT	continuous-time	16
DT	discrete-time	9
ECU	engine control unit	95
FRD	frequency response data	110
ICE	internal combustion engine	1
IO	input-output	9
IV	instrumental variables	44
LMI	linear matrix inequality	7
LPV	linear parameter-varying	1
LS	least squares	36
LTI	linear time-invariant	1
LTV	linear time-varying	7
MAX	moving average with exogenous input	59
MIMO	multi-input multi-output	2
OE	output error	39
OF	observability form	25
PBSID	predictor-based subspace identification	57
PE	prediction error	51
PRBS	pseudo-random binary signal	34
RIV	refined instrumental variables	51
SF	shifted form	24
SI	spark-ignition	91
SID	subspace identification	38
SISO	single-input single-output	3
SS	state-space	9
SVD	singular value decomposition	57
VAF	variance accounted for	37
VGT	variable-geometry turbocharger	94

List of Symbols

Symbols

α	Convex coordinate Lagrange multiplier Angle, e. g., α_{tv} opening angle of throttle valve
$\mathcal{A}$	Matrix in subspace identification containing products of state matrices
A	State matrix in the state-space domain, i. e. a matrix function in the LPV case Polynomial multiplied by the output in the input-output domain
a	Entry of A polynomial, i. e. a matrix function in the LPV case
B	Input matrix in the state-space domain, i. e. a function in the LPV case Polynomial multiplied by the input in the input-output domain
b	Entry of B polynomial, i. e. a matrix function in the LPV case
C	Output matrix in the state-space domain, i. e. a function in the LPV case Polynomial multiplied by the noise in the input-output domain
c	Entry of C polynomial, i. e. a matrix function in the LPV case
D	Feedthrough matrix in the state-space domain, i. e. a function in the LPV case Polynomial multiplied by the colored noise in the input-output domain Matrix from UDU decomposition
d	Entry of D polynomial, i. e. a matrix function in the LPV case
$\mathcal{E}$	Matrix containing the error and scheduling parameter signal
E	Matrix containing the error signal
e	Error Noise
$\mathcal{F}$	Set of admissible scheduling parameter trajectories
F	Polynomial multiplied by the noise free output in the input-output domain
f	Function, e. g., state function of a nonlinear system Entry of F polynomial, i. e. a matrix function in the LPV case Length of future window
Γ	State-observability matrix (for the LTI part of a system)

γ	Performance index
G	System or model Markov parameter
$\mathcal{H}$	Matrix in subspace identification containing model coefficients Extended Hankel matrix
H	Data matrix function in subspace identification Hankel matrix function
H	Noise system or noise model Matrix containing data, e. g., H_f^u in subspace identification
I	Identity matrix
K	Regularization matrix Controller Observer matrix, i. e. a matrix function in the LPV case
k	Sample instant
ℓ_2	Space of square-summable vector signals
λ	Eigenvalue
$\mathcal{L}$	Matrix in subspace identification containing model coefficients
L	Matrix from Cholesky decomposition
$\mathcal{M}$	Matrix in subspace identification containing products of state and input matrices
μ	Scheduling parameter in case of linear dependency
M	Model
m	Mass
$\mathcal{N}$	Submatrix of observability matrix $\mathcal{O}$
ν	Bound on variation of scheduling parameter between two samples
N	Number of samples
n	Number, e. g., of states Noise Rotational speed, e. g., engine speed
$\mathcal{O}$	Observability matrix
O	State-observability matrix function
o	Entry of state-observability matrix function
$\mathcal{P}$	Matrix containing the scheduling parameter signal

Φ	Matrix containing model coefficients
ϕ	Transition matrix function
P	Lyapunov matrix function Matrix containing the scheduling parameter signal Power
p	External scheduling parameter Length of past window Pressure
Q	Inverse Lyapunov matrix function Matrix from QR decomposition
$\mathcal{R}$	System representation, e. g., $\mathcal{R}_{\mathrm{IO}}^{\mathrm{DT}}(G)$ for the discrete-time, input-output representation of system G Extended reachability matrix
R	State-reachability matrix function
r	Entry of state-reachability matrix function
ρ	Scheduling parameter in case of general functional dependence
R	Matrix from QR decomposition
r	Reference
Σ	Matrix from SVD Diagonal covariance matrix of white noise
σ	Standard deviation
θ	Scheduling parameter in case of affine dependency Model coefficients
T	Duration, e. g., sampling time T_{s} Transformation or projection matrix Temperature
$\mathcal{U}$	Matrix containing the input and scheduling parameter signal
U	Matrix containing the input signal Matrix from UDU decomposition or SVD
u	Input
V	Lyapunov function Matrix from SVD
v	Colored noise

W	Matrix containing the signal w Weight
w	Noise Mass flow
$\mathcal{X}$	Matrix containing the state and scheduling parameter signal
X	Gramian Matrix containing the state signal
x	State
Y	Inverse Gramian Matrix containing the output signal
y	Output
$\mathcal{Z}$	Matrix containing the signal z and the scheduling parameter signal
$\star$	Term of no interest
Z	Matrix containing the signal z Matrix containing data, e. g., Z_U in subspace identification
z	Performance output Transformed state Regressor Combination of inputs and outputs, i. e. $z^{\mathrm{T}} = [u^{\mathrm{T}} \quad y^{\mathrm{T}}]$

Operators

$\lvert x \rvert$	Absolute value of x
$\lVert x \rVert_{\ell_2}$	ℓ_2 signal norm of signal x
$\lVert X \rVert_{\mathrm{F}}$	Frobenius norm of matrix X
$\lVert G \rVert_{\mathrm{i},\ell_2}$	Induced ℓ_2 system norm of system G
$\dot{x}$	Time derivative of x
$\hat{x}$	Estimation or approximation of x
$\breve{x}$	Noise free part of x
X^{T}	Transpose of a matrix X
X^{-1}	Inverse of a matrix X
$X^{\dagger}$	Pseudo-inverse of a matrix X
$X(:, i)$	The i-th column of a matrix X
$\acute{X}$	Expanded matrix with counterpart on the right hand side

$\grave{X}$	Expanded matrix with counterpart on the left hand side
$\bar{X}$	Matrix of a minimal state-space representation Matrix used in predictor form Mean value of X Shifted matrix, e. g., $\bar{H}_{i,j} = O_i A R_j$ is the shifted Hankel matrix
$\tilde{X}$	Scheduling parameter independent term of a polynomial $X(\mathrm{q}, \rho)$ Matrix containing (model coefficients for) inputs, outputs, and possibly scheduling parameters, e. g., $\tilde{b}$ in LPV IO identification using LTI techniques Matrix based on the augmented input matrix in subspace identification, i. e. $\tilde{B} = [B - KD \quad K]$ with transposed input $[u^{\mathrm{T}} \quad y^{\mathrm{T}}]$
$\mathrm{E}\{x\}$	Expectation of x
$\mathrm{cov}\{x\}$	Covariance of x
Δ	Change in a value, e. g., $\Delta\theta$ in the LPV IO identification algorithms
$\prec$	Negative definite
$\succ, \succeq$	Positive definite, positive semi-definite
$\diamond$	Function evaluation along a trajectory
$\otimes$	Kronecker product
$\circledcirc$	Block wise Kronecker product
$\odot$	Khatri-Rao product
lim	Limit
max	Maximum
min	Minimum
sup	Supremum
Ψ	Observability or controllability operator
q	Forward time shift operator in discrete-time
diag	Diagonal matrix operator
trace	Matrix trace
rank	Rank
vec	Matrix vectorization

Sets

$\mathbb{D}$	Data set

$\mathbb{I}$	Set of indexes, e. g., sampling instants
$\mathbb{N}$	Set of natural numbers (including zero)
$\mathbb{N}^+$	Set of natural numbers greater than zero
$\mathbb{P}$	Compact set of scheduling parameter values
$\mathbb{R}$	Set of real numbers
$\mathbb{Z}$	Set of integers
$\mathbb{Z}^-$	Set of integers less than zero

Subscript, Index

0	Initial value/condition, e. g., x_0 Ambiance
$i\|j$	Predictor or matrix containing samples from instant i to instant j
bal	Balanced, e. g., in balanced realization
bm	Boost manifold
c	Controllability Compressor
cyl	Cylinder
em	Exhaust manifold
eng	Engine
ev	Exhaust valve
f	Fuel
fa	Fresh air
fix	Fixed
im	Intake manifold
iv	Inlet valve
o	Observability
off	Offset
r	Random
rg	Residual gas
rng	Range

s	Sampling, e. g., sampling time T_s
sc	Scaled
t	Turbine
tc	Turbocharger
tv	Throttle valve
v	Vertex
wg	Wastegate

Superscript

(i)	Indexing, e. g., $x_j^{(i)}$ in order to index in two dimensions
i	Power or dimension
$i \times j$	Matrix or set of dimension i-by-j, e. g., matrix with i rows and j columns
ex	Extended
o	Observability
r	Reachability
red	Reduced

List of Publications

Lead-Authored

Schulz, E., A. Bussa, & H. Werner (2016). "Identification of Linear Parameter-Varying Systems via IO and Subspace Identification – A Comparison". In: *Proceedings of the Conference on Decision and Control*. IEEE, pp. 7147–7152.

Schulz, E., P. Cox, R. Tóth, & H. Werner (2017). "LPV State-Space Identification via IO Methods and Efficient Model Order Reduction in Comparison with Subspace Methods". In: *Proceedings of the Conference on Decision and Control*. IEEE, pp. 3575–3581.

Schulz, E., O. Janda, & M. Schultalbers (2018). "On LPV System Identification Algorithms for Input-Output Model Structures and their Relation to LTI System Identification". In: *Proceedings of the Symposium on System Identification*. IFAC, pp. 1098–1103.

Schulz, E., M. Schultalbers, & H. Werner (2016). "Linear Parameter-Varying Control for Air/Fuel Ratio in SI Engines with Parameter Dependent Time Delay". In: *Proceedings of the American Control Conference*. IEEE, pp. 3292–3297.

Co-Authored

Daasch, A., E. Schulz, & M. Schultalbers (2016). "Intrinsic Performance Limitations of Torque Generation in a Turbocharged Gasoline Engine". In: *Proceedings of the International Symposium on Advances in Automotive Control*. IFAC, pp. 714–721.

Datar, A., E. Schulz, & H. Werner (2018). "Identification of Linear Parameter-Varying Models with Unknown Parameter Dependence Using ε-Support Vector Regression". In: *Proceedings of the American Control Conference*. IEEE, pp. 2011–2016.

Kettlitz, S., E. Schulz, & M. Schultalbers (2019). "Identifikation lokal linearer Modelle am Beispiel der Ladedruckstrecke eines Ottomotors". In: *VDI/VDE Fachtagung AUTOREG*.

Udupa, N. A., E. Schulz, & M. Schultalbers (2019). "Polytopic LPV Controller Design to Control the Boost Manifold Pressure in an Internal Combustion Engine". In: *VDI/VDE Fachtagung AUTOREG*.