IMT School for Advanced Studies, Lucca

Lucca, Italy

Towards automated data-driven modeling of linear parameter-varying systems

PhD Program in Control Systems

XXX Cycle

By

Manas Mejari

2018

The dissertation of Manas Mejari is approved.

Program Coordinator: Prof. Alberto Bemporad, IMT School for Advanced Studies Lucca, Italy.

Supervisor: Prof. Alberto Bemporad, IMT School for Advanced Studies Lucca, Italy.

Supervisor: Dr. Dario Piga, Dalle Molle Institute for Artificial Intelligence, SUPSI, Lugano, Switzerland.

The dissertation of Manas Mejari has been reviewed by:

Mario Sznaier, Northeastern University, Boston, USA.

Alireza Karimi, École Polytechnique Fédérale de Lausanne, Switzerland.

IMT School for Advanced Studies, Lucca

2018

To my parents,
Ranjana and Dilip

Contents

List of Figures

List of Tables

Acknowledgements

First and foremost I would like to express my sincere gratitude to my supervisor Prof. Alberto Bemporad for giving me the opportunity to be a part of DYSCO research unit in IMT Lucca. Working with him during the three years of my PhD has been a great learning experience and his expertise and vast knowledge has been inspirational. Most importantly, I am extremely grateful to my co-advisor Dr. Dario Piga as this thesis would not have been possible without his constant guidance and tremendous support. I gratefully acknowledge Dario for his directions, ideas, and many brainstorming discussions. It was a pleasure to closely work with both of them and I hope to continue this collaboration in future.

On a personal level, I sincerely thank all my friends in IMT who have become like a family. It was truly a memorable experience being with all of you which is beyond words. Many thanks to Vihang, Soumali, Valerio C, Valerio G, Valentina, Elena, Giovanna, Puya, Vignes, Paolo, Vitaly, Pakhee, Lucia, Andreas, Ajay, Marina, Nilay, Laura, Xu for wonderful memories. I also thank Ankit, Aniket, Mandar, and Vishwanath for crazy talks, laughters and jokes which were a great stress buster during the hard times of PhD.

I am grateful to my entire family, specially my grandfather Mr. Savlaram Sathe who by his own example taught me that curiosity and learning has no age limit. Last but not the least, I would like to sincerely thank my mother Ranjana and my father Dilip for their constant encouragement throughout my career and beyond. Nothing would have been possible without their unconditional support and love.

Vita

Dec. 09, 1989	Born in Mumbai, India
2007 – 2011	B.E. in Electronics & Telecommunication Final mark: Distinction University of Mumbai, India
2011 – 2013	M. Tech in Control Systems Final mark: CGPA 9.8/10 University of Mumbai, India
2013 – 2014	Senior Research Fellow V.J.T.I., Mumbai University of Mumbai, India
2015 – 2018	PhD in Control Systems IMT School for Advanced Studies Lucca Italy
02/18 – 07/18	Visiting researcher Technische Universiteit Eindhoven (TU/e) Netherlands

Publications

Journal papers and Conference proceedings

1. M. Mejari, D. Piga, and A. Bemporad, "A bias-correction method for closed-loop identification of Linear Parameter-Varying systems," in *Automatica*, vol. 87, pp. 128 – 141, 2018. (url)

2. M. Mejari, D. Piga, and A. Bemporad, "LPV model-order selection from noise-corrupted output and scheduling signal measurements," in *Proc. 20th IFAC World Congress*, Toulouse, France, pp. 8685 – 8690, 2017. (url)

3. V.V. Naik, M. Mejari, D. Piga, and A. Bemporad, "Regularized moving-horizon PWA regression using mixed-integer quadratic programming," in *Proc. 25th Mediterranean Control Conference*, Valletta, Malta, pp. 1349 – 1354, 2017. (url)

4. M. Mejari, D. Piga, and A. Bemporad, "Regularized least square support vector machines for order and structure selection of LPV-ARX models," in *Proc. European Control Conf.*, Aalborg, Denmark, pp. 1649 – 1654, 2016. (url)

5. M. Mejari, V.V. Naik, D. Piga, and A. Bemporad, "Regularized moving-horizon PWA regression for LPV system identification," *Accepted, IFAC Symposium on System Identification*, Stockholm, Sweden, 2018.

6. S. Mane, M. Mejari, F. Kazi, and N. Singh, "Improving lifetime of fuel cell in hybrid energy management system by Lure-Lyapunov-based control formulation," in *IEEE Transactions on Industrial Electronics*, vol. 64, no. 8, pp. 6671–6679, 2017. (url)

Submitted papers

1. M. Mejari, D. Piga, R. Tóth, and A. Bemporad, "Identification of linear parameter-varying systems with linear fractional representation," *Submitted, IEEE Conference on Decision and Control*, 2018.
2. M. Mejari, V.V. Naik, D. Piga, and A. Bemporad, "Energy disaggregation using piecewise affine regression and binary quadratic programming," *Submitted, IEEE Conference on Decision and Control*, 2018.

Abstract

The *Linear Parameter-Varying* (LPV) paradigm represents a natural extension of the classical *Linear Time-Invariant* (LTI) framework. By virtue of the so-called *scheduling signal*, LPV models can accurately describe the behavior of a large class of time-varying and nonlinear dynamical systems, while preserving the linearity between input and output signals. Thus, LPV models are of great practical significance for modeling and control of many industrial systems, as the simplicity of the LTI theory is retained. This thesis is concerned with data-driven modeling of LPV systems, addressing some of the well-known issues pertaining to LPV model identification. In particular, the main research questions addressed in this contribution include selection of the model structure, dealing with noisy measurements of the scheduling signal, identification of the plant model from closed-loop data, and identification of linear fractional representations of LPV models which are especially suitable for controller synthesis. The proposed methodologies mainly fall under the framework of parametric and non-parametric LPV identification using machine learning techniques and provide a set of tools for automated selection of LPV modeling parameters. Furthermore, as an alternative to the conventional parametric and non-parametric approaches, a novel algorithm based on mixed-integer programming is devised for identifying LPV models through piecewise affine regression. The framework of this algorithm is applied to the problem of *energy disaggregation* with a real-world benchmark dataset.

Chapter 1

Introduction

In many engineering applications, an accurate mathematical description of the actual physical process is required for analysis and control. The behavior of the system of interest can be modeled from the first principle laws of physics, chemistry, etc. Often, a detailed knowledge of the underlying system is essential to derive a mathematical model from these laws. However, many physical phenomena have complex system dynamics and it is very hard to model them using the first principle laws. As an alternative, a *System Identification* approach can be used to describe the behavior of the system.

System identification aims at building mathematical models of dynamical systems from experimental data. In order to obtain an accurate model of the system, identification procedure consists of multiple design steps which are referred to as an *identification cycle*. A typical identification cycle consists of the following important steps which are summarized in Table 1.

In the first step, an experiment is performed by exciting the system under consideration with input signals and observing its output response. A record of input and output signals over a time interval, termed as *training* dataset, is used for estimating the models. The experiment can be performed either in an open-loop or in a closed-loop setting where the system is in a feedback connection with a controller. The second

Table 1: The identification cycle

Step 1. Collection of data from the system.
Step 2. Selection of model structure.
Step 3. Determination of the identification criterion.
Step 4. Estimation of models which optimize the chosen criterion.
Step 5. Validation of the estimated model on a fresh dataset.

step consists of choosing an appropriate model structure which can adequately describe the recorded input-output dataset. The type of model structures can range from simple static models to complex nonlinear and time-varying dynamical models. Once the model structure is selected, a criterion is chosen in the third step which can quantify the quality of the identified model. This criterion tells how close the model output is to the recorded output measurements. For example, *least-squares* criterion quantifies the fit of the estimated model output to the actual output via square of the euclidean norm of the residuals. The next step is to develop a statistical method to estimate the parameters of the selected model class which are optimal with respect to the identification criterion chosen in the third step. In the final step of the identification cycle, the quality of the estimated model is assessed using a fresh *validation* dataset which is not used in the training phase.

To improve the accuracy of the model estimate so as to describe the behavior of the underlying dynamical system as closely as possible, identification cycle can be performed iteratively exploiting the information from the previous iteration. For instance, staring from a simple model class, the quality of the match between model output and true system output is assessed based on the identification criterion. If the fit to validation data is not satisfactory, a more complex model structure is chosen, its parameters are estimated and validated, etc.

Following the general identification steps outlined in Table 1, in this thesis, the identification problem of a special class of model structures called *Linear Parameter-Varying* (LPV) models is considered. The thesis focused on developing a set of tools for data-driven modeling of LPV systems addressing some of the main questions pertaining to LPV system identification.

1.1 Linear parameter-varying paradigm

The framework of *Linear Time-Invariant* (LTI) system identification has been well established over the years. A vast literature for the identification of LTI models with well founded theoretical tools is available. A comprehensive review of the LTI identification methods is given in [55; 91]. The LTI models have also proven to be useful for controller synthesis in some engineering applications. However, most physical processes are inherently nonlinear and time-varying. In order to accurately describe the behavior of such systems, LTI models are found to be inadequate. Ideally, one can resort to the framework of nonlinear and time-varying models and develop identification tools for this model class. Significant research efforts have been made to develop the theory for nonlinear system identification [12; 42; 54; 76; 80; 84; 88]. However, dealing with nonlinear models is computationally inefficient for identification and control tasks, limiting their use in practical applications [98].

To alleviate this problem and to achieve a balance between accuracy of nonlinear modeling and simplicity of the LTI theory, a new class of model structure termed as *Linear Parameter-Varying* models has been proposed using the concept of *gain scheduling* [87]. The idea of LPV paradigm originates from the fact that a nonlinear system can be approximated by LTI models locally at different operating points. Specifically, the nonlinear dynamics is linearized around multiple operating points resulting in multiple LTI models where each model captures the behavior of the actual system locally. For controller synthesis of a nonlinear system, multiple controllers are designed based on local LTI models which are then interpolated to control the nonlinear dynamics over

the entire region of operation through gain scheduling. The change from one operating point to another can be described via a signal, the so called *scheduling signal*, denoted in this thesis as p. Thus, the parameters of the model vary with the time-varying scheduling variable p, at the same time preserving the linearity between inputs and outputs, hence the name *linear parameter-varying* systems. In this way, the nonlinear and time-varying dynamics are embedded in the scheduling variables. By virtue of scheduling signals, LPV models can accurately describe the dynamic behavior of a large class of nonlinear and time-varying systems. This has led to research initiatives in developing identification and control methods for LPV models with a growing number of applications such as modeling, analysis and control of aircrafts [4; 57], automobiles [22; 69], distillation columns [6], helicopter rotor dynamics [105], wind turbine [31], electronic filter [49], internet web servers [96], etc.

1.2 LPV model representations

In this section, we formalize the description of linear parameter-varying systems via different model representations used in the context of identification.

Input-Output models

Inspired from the Linear Time-Invariant Input-Output (LTI-IO) models, LPV systems can be represented by the following input-output representation which is commonly used in many LPV identification approaches

$$y(k) = -\sum_{i=1}^{n_a} a_i(p(k))y(k-i) + \sum_{j=1}^{n_b} b_j(p(k))u(k-j) + e(k),$$

where p is the scheduling parameter, coefficients $\{a_i\}_{i=1}^{n_a}$ and $\{b_j\}_{j=1}^{n_b}$ are functions of p with static dependency. The noise process is modeled with $e(k)$, which can be colored or white noise depending upon the selected noise structure. For Linear Parameter-Varying Input-Output (LPV-IO) models, the identification problem consists of estimating the unknown

p-dependent coefficient functions and the model orders n_a, n_b from the measurements of inputs, outputs and scheduling variable sequence.

State-Space models

The LPV State-Space (LPV-SS) representation is a direct extension of LTI state space models. The LPV-SS representation is given as

$$\begin{aligned} x(k+1) &= A(p(k))x(k) + B(p(k))u(k) + K_1(p(k))e(k), \\ y(k) &= C(p(k))x(k) + D(p(k))u(k) + K_2(p(k))e(k), \end{aligned}$$

where $x \in \mathbb{R}^{n_x}$ is the state variable, e is the white-noise and A, B, C, D, K_1, K_2 are matrices with static dependency on the scheduling variable p. Often the matrices are considered as a linear function of the scheduling variable, i.e.,

$$A(p) = A_0 + \sum_{j=1}^{n_p} A_j p_j,$$

where $p = \begin{bmatrix} p_1, \ldots, p_{n_p} \end{bmatrix}^\top \in \mathbb{R}^{n_p}$ is the scheduling parameter vector and $A_j \in \mathbb{R}^{n_x \times n_x}$ are the unknown matrices to be estimated. This representation is termed as *affine* dependence.

Linear Fractional Representations

Inspired from the approaches used in the robust control literature [111; 112], *Linear Fractional Representation* (LFR) is a model structure where the scheduling variable dependence is extracted into the feedback path and the forward path is modeled as an LTI system. The forward LTI model in the LFR form is given as

$$\begin{bmatrix} \frac{x(k+1)}{z(k)} \\ y(k) \end{bmatrix} = \left[\begin{array}{c|cc} A & B_1 & B_2 \\ \hline C_1 & D_{11} & D_{12} \\ C_2 & D_{21} & D_{22} \end{array}\right] \begin{bmatrix} \frac{x(k)}{w(k)} \\ u(k) \end{bmatrix},$$

where $x \in \mathbb{R}^{n_x}$ is the state, $z \in \mathbb{R}^{n_z}$ and $w \in \mathbb{R}^{n_w}$ are latent variables, $u \in \mathbb{R}^{n_u}$ and $y \in \mathbb{R}^{n_y}$ are measured inputs and outputs of the forward LTI block, and $\{A, \ \ldots, \ D_{22}\}$ are unknown constant martices of appropriate dimensions. The feedback path is represented by

$$w(k) = \Delta(p(k))z(k),$$

where $\Delta : \mathbb{P} \rightarrow \mathbb{R}^{n_p \times n_p}$ is a function of the scheduling parameter p. This representation is specifically well suited for designing controllers for LPV models.

1.3 Challenges in LPV model identification

Although the LPV paradigm is a natural extension of the well developed LTI theory, classical theoretical tools commonly used for LTI system identification can not be used directly in an LPV setting. This is due to the fact that many of the properties of LTI system theory no longer hold in the case of LPV systems, for instance, transfer function models and commutative properties of the operators. Furthermore, for LPV systems the task of identification is to estimate the model coefficients which are nonlinear functions of the scheduling signals over the scheduling variable space, leading to a more complex problem as compared to static parameter estimation in the case of LTI identification. The main challenges for the identification of LPV models include but are not limited to the following:

1. **Selection of the model structure**
   One of the main issues in LPV system identification is to choose an accurate model structure so as to describe the behavior of the underlying system as closely as possible. The model structure selection problem consists of (i) choosing the type of model (for e.g., input-output or state-space representation), (ii) defining appropriate parameterization of the scheduling variable dependent functions which can accurately capture the underlying nonlinearity, (iii) selecting the dynamical model order in terms of input and output delays etc.

2. **Noise-corrupted scheduling signals**
   In practice, the scheduling variable is a measured exogenous signal which is corrupted by measurement noise. This leads to a more challenging *errors-in-variable* identification problem. Ignoring the effect of noise on the scheduling signal results in a biased estimate of the model parameters. Thus, dedicated methods need to be developed in order to resolve this issue.

3. **Identification from closed-loop data**
   It is well-known that data collected from closed-loop experiments are less informative for identifying plant models. Moreover, due to the presence of feedback, there is a correlation between output noise and inputs to the plant. To handle these issues, the closed-loop identification approaches developed for LTI systems need to be modified appropriately for LPV plant models operating in closed-loop.

4. **Deriving models suitable for controller synthesis**
   In order to use the developed LPV models in practical control applications, it is important to identify models which are suitable for controller synthesis. To this aim, it is necessary to adapt the LPV modeling framework such that it is optimal with respect to control performance, and simple enough for designing controllers without compromising the accuracy in describing the underlying system.

5. **Computationally efficient methods**
   A major challenge in LPV identification is to develop algorithms which are computationally efficient and which can handle the curse of dimensionality for their applicability to large-scale practical systems.

In this thesis, we address these challenges by proposing data-driven LPV model identification methods.

1.4 Contributions and organization

The contributions and main research questions addressed in this thesis are outlined as follows

Chapter 2 : LPV model order selection

In this chapter, we address the problem of data-driven model structure selection of LPV Input-Output (LPV-IO) models. Choosing the model structure in terms of LPV-IO models requires to specify both the model order (i.e., number of input and output delays) and the nonlinear dependence of the model coefficients on the scheduling signal. In most cases, over-parameterized models are used to fit the data in order to have a low bias and to describe the underlying system adequately. However, over-parameterization leads to a large variance of the estimated model parameters and also to a poor generalization to unseen data due to overfitting. Thus, accurate determination of model order becomes essential.

We propose two approaches for data-driven model structure selection of LPV-IO models. In the first approach, the non-parametric framework of Least Squares Support Vector Machines (LS-SVM) [93] to identify LPV-IO models is extended for the purpose of model order selection. The flexibility of LS-SVM is exploited to identify the unknown LPV model coefficients followed by a convex optimization based approach to select the model order.

In the second approach, the conventional *LASSO* method for sparse estimation is adapted for LPV model order selection to deal with the case of noise-corrupted scheduling signal. A bias-corrected cost function is presented which provides an asymptotic bias-free criterion to be optimized when the scheduling variables are corrupted by noise.

The results presented in Chapter 2 are based on the following contributions:

- M. Mejari, D. Piga, and A. Bemporad, "Regularized least square support vector machines for order and structure selection of LPV-ARX models," in *Proc. of the 15th European Control Conference*, Aal-

borg, Denmark, pp. 1649 – 1654, 2016. [60].

- M. Mejari, D. Piga, and A. Bemporad, "LPV model-order selection from noise-corrupted output and scheduling signal measurements," in *Proc. of the 20th IFAC World Congress*, Toulouse, France, pp. 8685 – 8690, 2017. [61]

Chapter 3 : Closed-loop identification of LPV models

In many industrial applications, it is necessary to identify plant models from systems operating in closed-loop, either because of safety constraints or due to unstable open-loop dynamics. The identification of plant models from closed-loop data is more challenging because the data gathered from closed-loop is often less informative. Furthermore, the methods developed for open-loop identification (for e.g., in Chapter 2) may fail as the output noise and the plant inputs are correlated due to the presence of feedback. In this chapter, we addressed this issue by proposing a bias-correction scheme to obtain a consistent estimate of LPV model parameters from the closed-loop data. The proposed bias-correction scheme is further extended to deal with the case of noise-corrupted scheduling signals, which is a more realistic and challenging errors-in-variable problem.

The results presented in Chapter 3 are based on the following contribution:

- M. Mejari, D. Piga, and A. Bemporad, "A bias-correction method for closed-loop identification of Linear Parameter-Varying systems," in *Automatica*, vol. 87, pp. 128 – 141, 2018. [62]

Chapter 4 : PWA regression for identification of LPV models

The LPV identification approaches proposed in Chapter 2 and Chapter 3, follow a conventional framework of parametric and non-parametric methods in terms of characterizing LPV model coefficients. Alternatively, in Chapter 4, we formulate the identification of LPV-IO models as a *Piece-Wise Affine* (PWA) regression problem. The LPV model coefficients are

approximated as PWA maps that introduce additional flexibility instead of restricting to a set of known basis functions as in the case of parametric methods. In this chapter, a regularized moving horizon algorithm is proposed for PWA regression using mixed-integer programming.

Furthermore, the framework of the proposed algorithm based on integer programing is applied to the real world problem of *energy disaggregation*. The goal of energy disaggregation problem is to estimate the power consumption profile of individual household appliances by using only aggregated power measurements. The performance of the proposed disaggregation method is tested on a real benchmark dataset.

The results presented in Chapter 4 are based on the following contributions:

- M. Mejari, V.V. Naik, D. Piga, and A. Bemporad, "Regularized moving horizon PWA regression for LPV system identification," *Submitted, 18th IFAC Symposium on System Identification*, 2017. [58]
- V.V. Naik, M. Mejari, D. Piga, and A. Bemporad, "Regularized moving horizon PWA regression using mixed-integer quadratic programming," in *Proc. of the 25th Mediterranean Control Conference*, Valletta, Malta, pp. 1349 – 1354, 2017. [64]
- M. Mejari, V.V. Naik, D. Piga, and A. Bemporad, "Energy disaggregation using piecewise affine regression and binary quadratic programming," *Submitted, 57th IEEE Conference on Decision and Control*, 2018. [59]

Chapter 5 : Identification of LPV models with linear fractional representation

This chapter is concerned with the identification of LPV models in *Linear Fractional Representation* (LFR). The identification of LFR forms is of particular interest for its suitability to design controllers for LPV models. In this chapter, we propose a two stage approach to identify LPV-LFR models. The first stage consists of estimating the LPV state variable sequence using kernel canonical correlation analysis. In the second stage,

the unknown model parameters are estimated by solving a nonlinear least squares optimization problem using the state estimates obtained from the first stage.

The results presented in Chapter 5 are based on the following contribution:

- M. Mejari, D. Piga, R. Tóth, and A. Bemporad, "Identification of linear parameter varying models with linear fractional representation," *Submitted, 57th IEEE Conference on Decision and Control*, 2018. [63]

Finally, concluding remarks and possible future research directions are outlined in Chapter 6.

Chapter 2

LPV model order selection

In this chapter, two independent approaches for the model order selection of *Linear Parameter-Varying* (LPV) input-output models are presented. In the first approach, we extend the non-parametric framework of least squares support vector machines for identification of sparse LPV models. In the second approach, we introduce a bias-corrected cost function which is to be optimized for accurate model order selection, when the scheduling signal measurements are corrupted by noise.

2.1 Introduction

2.1.1 Motivation

Motivated by the need of accurate and low-complexity LPV models, significant efforts have been spent in the last years for developing efficient approaches for identification of discrete-time LPV models, both in state-space [31; 95; 103; 107] and input-output (IO) representation [9; 51; 78]. As known, a challenging issue in identification is the choice of the model structure. In fact, an under-parameterized model structure might not adequately explain the dynamic behaviour of the system, while an over-parameterized model tends to overfit the data, leading to a poor generalization to unseen data, besides increasing the complexity of the final model and of the estimation procedure.

In terms of LPV identification, choosing the model structure requires to specify both the model order (in terms of number of output lags, input lags, and input delay) and the nonlinear dependence of the model coefficients on the scheduling variable p with the so-called *basis functions*. The basis functions characterize the dependence of the model coefficients on the scheduling signal and thus, in order to avoid the use of under-parametrized models, a large set of basis functions is typically chosen to adequately describe the underlying system. However, there is a risk of over-parametrization of the model which may cause a large variance in the estimate of the model parameters. This is the well known *bias-variance trade-off* problem which highlights the need of an accurate model order selection.

The bias-variance trade-off problem can be partially overcome by using sparse parametric estimation methods like the *Least Absolute Shrinkage and Selection Operator* (LASSO) [97], the *Non-Negative Garrote* (NNG) [17] and *SPARSe Estimation based on VAlidation* (SPARSEVA) [82]. These methods can be used to select a subset of p-dependent nonlinear basis functions by penalizing the ℓ_1-norm of the model parameters, along with the fitting error, thus enforcing sparsity in the final estimate of the parameter vector. Application of the NNG and SPARSEVA for sparse identification of LPV-ARX models in a parametric identification framework is discussed in [102] and [99], respectively . The extension of LASSO for order selection in a non-parametric LPV identification framework is discussed in [79].

2.1.2 Contributions

In this chapter we consider the model order selection of LPV systems using parametric as well as non-parameteric identification approaches. The ideas behind the two approaches are described as follows.

Non-parametric approach

In input-output LPV models, the first possible choice to describe the p-dependent LPV model coefficients is to approximating them as the sum

of a large set of a-priori specified nonlinear basis functions (e.g., polynomials). However, these functions may not be able to describe the underlying nonlinearities accurately and thus, adequate a-priory selection of the basis functions remains an open problem.

In order to deal with the basis functions selection problem, non parametric methods, also termed as kernel-based methods for LPV identification have been recently proposed in [29; 43; 101]. The main idea behind kernel methods is to introduce a feature function that maps the scheduling vector p to a high-dimensional space. The feature maps are not fixed a-priori and can be potentially infinite-dimensional. Only the inner products between the feature maps is specified by the user in terms of nonlinear kernel functions, such as *radial basis functions* or *polynomial kernels*. However, although these methods offer strong flexibility in modeling the nonlinear dependence on the scheduling vector p, they do not address the problem of selecting the LPV model order.

To the best of our knowledge, the only contribution addressing the issue of LPV model order selection in a kernel-based setting is [79], where *Least-Squares Support Vector Machines* (LS-SVM) (i.e., a specific kernel-based method developed in [93]) are reformulated in order to achieve data-driven model order selection along with non-parametric identification of the p-dependent LPV model coefficients. Specifically, the problem is formulated by using an extra penalty term aiming at minimizing the maximum absolute value of the LPV model coefficient functions over a set of grid points in the scheduling vector space. However, the method in [79] suffers from the following drawbacks: 1. since it is necessary to grid the scheduling space, the number of grid points grow exponentially with the dimension of p, thus increasing the computation load of the optimization problem; 2. the LPV model coefficient functions are enforced to be null only at the chosen gridding points, but nothing can be said outside these points.

We present a new method for data-driven order selection along with non-parametric identification of the p-dependent LPV model coefficients, overcoming the main drawbacks of [79]. This is achieved with the following three-step approach:

S1. estimate the coefficients for an over-parameterized LPV model using the non-parametric LS-SVM approach proposed in [101];

S2. scale the estimated coefficients using scheduling-variable dependent polynomial weights, which are then penalised to shrink the previously estimated model coefficients towards zero;

S3. re-estimate the non-null model coefficients.

In this way, an accurate model of the system is determined with low bias and less variance in the estimated coefficients. Overall, the proposed method can be seen as a reformulation of the Non-Negative Garrotte approach in a nonparametric framework. The advantage of this approach over [79] is that it is applicable for identification of LPV systems which are dependent on multi-dimensional scheduling variables, as the scheduling space does not need to be gridded. The polynomial weights also provide flexibility in reshaping and correcting the LPV model coefficients obtained in stage S1.

Parametric approach with noise corrupted scheduling signal

Parametric identification methods characterize the unknown nonlinear p-dependent LPV model coefficients in terms of known a-priory selected basis functions. However, most of the parametric LPV identification and sparse estimation methods available in the literature assume that only the output measurements are corrupted by noise, while the observations of the scheduling signal are noise free. In practice, this is an unrealistic assumption in most cases, as the scheduling variable is often measured by a sensor, and thus inherently affected by measurement noise. To the best of the our knowledge, the only contributions on LPV identification based on noisy measurements of the scheduling signals are given in [20; 23; 78]. The contribution in [23] addresses the identification of LPV systems in the set-membership framework, where an outer bounding box on the feasible parameter set is computed under the assumption that the perturbing noise is bounded. The work in [20] proposes an *instrumental variable* (IV) approach but it is limited to the case in which the

dependence on the scheduling signal is linear. In [78], these limitations are overcome by using a bias-corrected IV approach. The method provides consistent estimates of LPV models with polynomial dependence on the scheduling variable and the instruments are only required to be uncorrelated with the noise corrupting the output observations. However, no penalty function is introduced in [78], and thus ℓ_1-regularization methods (like the LASSO) cannot be used to select the model structure.

We extend the approach presented in [78], to address the issue of model-order selection by introducing a *bias-corrected cost* function. An ℓ_1-regularization term is then added to this cost to achieve accurate model-order selection. Furthermore, by using the introduced bias-corrected cost as a criterion for cross-validation, an unbiased tuning of the hyper-parameters influencing the parameter estimates is achieved.

2.1.3 Outline

The chapter is organized as follows: First, in Section 2.2, we introduce the non-parametric kernel based regularized LS-SVM approach for model order selection of LPV-ARX models. In Section 2.2.1, a short review of LPV-ARX model structure and problem statement are given. The standard LS-SVM method for LPV identification is described in Section 2.2.2. In Section 2.2.3, the mathematical details of the proposed regularized LS-SVM approach for model order selection are discussed. The method is tested on two academic examples and simulation results are presented in Section 2.2.4.

Next, in Section 2.3 the parametric method for model order selection from noise-corrupted scheduling signals is presented for *Output Error* (OE) LPV models. The considered identification problem is formulated in Section 2.3.1. In Section 2.3.2, the instrumental-variable identification method is reviewed, and the asymptotic properties of the estimated parameters are discussed. The bias-corrected version of the instrumental variable method is presented in Section 2.3.3. A simulation example illustrating the capabilities of the proposed bias-correction method is presented in Section 2.3.4. Finally, conclusions are given in Section 2.4.

2.1.4 Notations

Let $\mathbb{R}^n$ be the set of real vectors of dimension n. The ℓ_2-norm of the vector $x \in \mathbb{R}^n$ is denoted by $\|x\|_2$. For matrices $A \in \mathbb{R}^{m\times n}$ and $B \in \mathbb{R}^{p\times q}$, the Kronecker product between A and B is denoted by $A \otimes B \in \mathbb{R}^{mp\times nq}$. Let $\mathbb{I}_a^b$ be the sequence of successive integers $\{a, a+1, \cdots, b\}$, with $b > a$. The floor function is denoted by $\lfloor\cdot\rfloor$, with $\lfloor m\rfloor$ being the largest integer less than or equal to m. The expected value of a random vector $\mathbf{x}$ is denoted by $\mathbb{E}[\mathbf{x}]$.

2.2 Regularized LS-SVM

Least Squares Support Vector Machine (LS-SVM) is a computationally efficient kernel-based regression approach which has been recently applied to nonparametric identification of LPV systems [101]. In contrast to parametric LPV identification approaches, LS-SVM based methods obviate the need to parameterize the scheduling dependence of the LPV model coefficients in terms of a-priori specified basis functions. However, an accurate selection of the underlying model order is still a critical issue in the identification of LPV systems in the LS-SVM setting. We address this issue by extending the LS-SVM method to sparse LPV model identification, which, besides non-parametric estimation of the model coefficients, achieves data-driven model order selection via convex optimization. The main idea of the proposed method is to first estimate the coefficients of an over-parameterized LPV model through LS-SVM. The estimated coefficients are then scaled by polynomial weights, which are shrunk towards zero to enforce sparsity in the final LPV model estimate. In the subsequent sections we formally describe the proposed method.

2.2.1 Problem formulation

The analysis driven in this section is dedicated to LPV models in an I/O form. For clarity of exposition, we consider *single-input single-output* (SISO) LPV systems, described by the *autoregressive with exogenous input*

(ARX) structure:

$$y(t) = \sum_{i=1}^{n_a^o} a_i^o(p(t))y(t-i) + \sum_{j=0}^{n_b^o} b_j^o(p(t))u(t-j) + e^o(t), \tag{2.1}$$

where $t \in \mathbb{N}$ denotes the discrete time; $u(t) \in \mathbb{R}$ and $y(t) \in \mathbb{R}$ are the measured input and output signals of the system, respectively; $e^o(t)$ is an additive zero-mean white noise; $p(t) : \mathbb{N} \to \mathbb{P}$ is the measured n_p-dimensional scheduling vector (which may include also past observations of the scheduling signals) and $\mathbb{P} \subseteq \mathbb{R}^{n_p}$ is a compact set where $p(t)$ is assumed to take values. The p-dependent coefficient functions a_i^o and b_j^o, as well as the parameters n_a^o and n_b^o defining the dynamical order of the system, are unknown and they have to be estimated from an N-length observed sequence $\mathcal{D}_N = \{u(t), y(t), p(t)\}_{t=1}^N$ of data generated by the system in (2.1).

The following model structure is therefore suitable to describe the true LPV data-generating system in (2.1):

$$y(t) = \sum_{i=1}^{n_a} a_i(p(t))y(t-i) + \sum_{j=0}^{n_b} b_j(p(t))u(t-j) + e(t), \tag{2.2}$$

with $e(t)$ denoting the residual term. The parameters n_a and n_b defining the dynamical order of the model in (2.2) are chosen large enough so that $n_a > n_a^o$ and $n_b > n_b^o$ (i.e., the true system belongs to the chosen model class).

In the following section, we briefly describe the LPV LS-SVM identification method proposed in [101], which is used for non-parametric estimation of the model coefficients $\{a_i\}_{i=1}^{n_a}$ and $\{b_j\}_{j=0}^{n_b}$. The main advantage of the LPV LS-SVM identification method is that it obviates the need to specify the underlying dependency of coefficients $\{a_i\}_{i=1}^{n_a}$ and $\{b_j\}_{j=0}^{n_b}$ on the scheduling vector p.

2.2.2 LS-SVM for LPV model identification

Let us consider the LPV-ARX model introduced in (2.2), which is rewritten in the compact form

$$y(t) = \sum_{i=1}^{n_g} c_i(p(t))x_i(t) + e(t), \tag{2.3}$$

where $x_i(t)$ and $c_i(p(t))$ denote the i-th component of the $n_g = n_a+n_b+1$-dimensional vector $x(t)$ and $c(p(t))$, respectively, defined as

$$\begin{aligned} x(t) &= [y(t-1) \ \dots \ y(t-n_a) \ u(t) \ \dots \ u(t-n_b)]^\top, \\ c(p(t)) &= [a_1(p(t)) \ \dots \ a_{n_a}(p(t)) \ b_0(p(t)) \ \dots \ b_{n_b}(p(t))]^\top. \end{aligned}$$

The p-dependent coefficient functions $c_i(p(t))$ are written as

$$c_i(p(t)) = \rho_i^\top \phi_i(p(t)) \quad i = 1, \dots, n_g. \tag{2.4}$$

where $\rho_i \in \mathbb{R}^{n_\mathrm{H}}$ is an unknown vector of parameters and $\phi_i : \mathbb{P} \to \mathbb{R}^{n_\mathrm{H}}$ (with $i = 1, \dots, n_a + n_b + 1$) maps the observed scheduling variable $p(t)$ to an n_H-dimensional space, commonly referred to as the *feature space*. Unlike the LPV parametric identification approaches, neither the maps ϕ_i nor the dimension n_H of the vectors ρ_i and $\phi_i(p(t))$ are explicitly specified by the user. Potentially, ρ_i and $\phi_i(p(t))$ can be infinite-dimensional vectors (i.e., $n_\mathrm{H} = \infty$).

Based on the previously introduced notation, the LPV model (2.3) is rewritten in the linear regression form:

$$y(t) = \sum_{i=1}^{n_g} \rho_i^\top \phi_i(t)x_i(t) + e(t), \tag{2.5}$$

where $\phi_i(t)$ is used as a shorthand notation for $\phi_i(p(t))$.

In the LS-SVM formulation, the following *quadratic programming* (QP) problem with a regularized ℓ_2 loss function is considered to estimate the

LPV model (2.5) from the data observations $\mathcal{D}_N$:

$$\min_{\rho_i, e} \quad \frac{1}{2}\sum_{i=1}^{n_g} \rho_i^\top \rho_i + \frac{\lambda}{2}\sum_{t=1}^{N} e^2(t) \tag{2.6a}$$

$$\text{s.t.} \quad e(t) = y(t) - \sum_{i=1}^{n_g} \rho_i^\top \phi_i(t) x_i(t), \;\; t \in \mathbb{I}_1^N \tag{2.6b}$$

where $\mathbb{I}_1^N$ denotes the set of indexes $\{1, \ldots, N\}$, and $\lambda > 0$ is a tuning hyper-parameter. The term $\sum_{t=1}^{N} e^2(t)$ in the cost function (2.6a) aims at minimizing the prediction error, while the regularization term $\rho_i^\top \rho_i$ is added in (2.6a) to prevent overfitting. The hyper-parameter λ should be then tuned to balance the bias/variance trade-off. Note that the parameters ρ_i minimizing (2.6) cannot be computed since this would require an explicit representation of the feature maps $\{\phi_i(t)\}_{i=1}^{n_g}$. Thus, the dual formulation of Problem (2.6) is considered. The Lagrangian $\mathcal{L}(\rho, e, \alpha)$ associated with the primal Problem (2.6) is given by:

$$\mathcal{L}(\rho, e, \alpha) = \frac{1}{2}\sum_{i=1}^{n_g} \rho_i^\top \rho_i + \frac{\lambda}{2}\sum_{t=1}^{N} e^2(t) - \sum_{t=1}^{N} \alpha_t \left(e(t) - y(t) + \sum_{i=1}^{n_g} \rho_i^\top \phi_i(t) x_i(t) \right), \tag{2.7}$$

where $\alpha = [\alpha_1 \cdots \alpha_N]^\top \in \mathbb{R}^N$ is the vector of Lagrange multipliers associated with the equality constraints (2.6b). The optimal solution of the primal QP problem (2.6) is then achieved when the following KKT conditions are satisfied:

$$\frac{\partial \mathcal{L}}{\partial \rho_i} = 0 \rightarrow \rho_i = \sum_{t=1}^{N} \alpha_t \phi_i(t) x_i(t), \tag{2.8a}$$

$$\frac{\partial \mathcal{L}}{\partial e(t)} = 0 \rightarrow e(t) = \frac{1}{\lambda}\alpha_t, \tag{2.8b}$$

$$\frac{\partial \mathcal{L}}{\partial \alpha_t} = 0 \rightarrow e(t) = y(t) - \sum_{i=1}^{n_g} \rho_i^T \phi_i(t) x_i(t). \tag{2.8c}$$

Substituting equations (2.8a) and (2.8b) into (2.8c) leads to:

$$y(t) = \sum_{i=1}^{n_g} \left(\sum_{t=1}^{N} \alpha_t x_i(t) \phi_i^T(t) \right) \phi_i(t) x_i(t) + \frac{1}{\lambda} \alpha_t \quad t \in \mathbb{I}_1^N. \tag{2.9}$$

Equations in (2.9) can be written in the matrix form

$$Y = \left(\Omega + \lambda^{-1} I_N \right) \alpha, \tag{2.10}$$

where, $Y = [y(1) \cdots y(N)]^\top$, I_N is the identify matrix of size N, and Ω is the *kernel matrix* whose (j,k)-th entry is given by: $[\Omega]_{j,k} = \sum_{i=1}^{n_g} [\Omega^i]_{j,k}$ with

$$[\Omega^i]_{j,k} = x_i(j) \phi_i^\top(p(j)) \phi_i(p(k)) x_i(k) \tag{2.11}$$

$$= x_i(j) K_i(p(j), p(k)) x_i(k), \tag{2.12}$$

where, K_i is a positive definite *kernel function* defining the inner product $\phi_i^\top(p(j))\phi_i(p(k))$. Specification of the kernel instead of the feature maps ϕ is called *kernel trick* [104] and it obviates the need to specify the feature maps explicitly, thus allowing the identification of the coefficient functions c_i by only specifying the kernel functions. A typical choice of kernel, which provides uniformly effective representation of a large class of smooth functions, is the *Radial Basis Function* (RBF) function:

$$K_i(p(j), p(k)) = \exp\left(- \frac{(p(j) - p(k))^2}{\sigma_i^2} \right), \quad i = 1, \ldots, n_g, \tag{2.13}$$

where $\sigma_i > 0$ is a hyper-parameter tuned by the user to control the width of the RBF.

Once the kernel matrix Ω is defined, the Lagrange multipliers α are computed from (2.10), and the estimates of the coefficients c_i are obtained by substituting (2.8a) into (2.4):

$$\hat{c}_i(\cdot) = \rho_i^\top \phi_i(\cdot) = \sum_{t=1}^{N} \alpha_t K_i(p(t), \cdot) x_i(t), \quad i \in \mathbb{I}_1^{n_g} \tag{2.14}$$

2.2.3 Model order selection with LS-SVM

As shown in the previous section, the approach based on LS-SVM allows one to reconstruct the dependence of the coefficients $\{c_i\}_{i=1}^{n_g}$ (or equivalently $\{a_i\}_{i=1}^{n_a}$ and $\{b_j\}_{j=0}^{n_b}$) on the scheduling vector p without the need to explicitly specify the feature maps ϕ_i. Nevertheless, the LPV LS-SVM approach can be further improved for selecting the dynamical structure of the LPV-ARX model in (2.3), which means detecting the non-zero coefficients c_i. Indeed, the final estimate obtained by the LS-SVM approach discussed in Section 2.2.2 strongly depends on the chosen number n_g of coefficients c_i (or equivalently, on the chosen values of the parameters n_a and n_b). As already discussed, low values of n_a and n_b leads to under-parameterized LPV models with poor capabilities of capturing the dynamics of the system. On the other hand, large values of n_a and n_b lead to over-parameterized models which tend to over-fit the noisy training data.

In this section, we present an extension of the LS-SVM method, which aims at estimating a sparse LPV model structure, thus minimizing the number of non-zero p-dependent coefficients $\{c_i\}_{i=1}^{n_g}$. The key idea of the proposed method is to start with an over-parameterized LPV model, whose coefficient functions $\{c_i\}_{i=1}^{n_g}$ are estimated through the LS-SVM approach presented in Section 2.2.2. Then, the estimated coefficients $\{\hat{c}_i\}_{i=1}^{n_g}$ are reshaped, multiplying them by polynomial weights that depend on the scheduling variable p. A regularization term which aims to shrink the polynomial weights towards zero is considered in order to enforce sparsity in the final estimated model.

Let us introduce the polynomial weights

$$w_i(p(t)) = \mathbf{w}_i^\top \varphi(p(t)), \quad i = 1, \dots, n_g, \tag{2.15}$$

where $\varphi(p(t))$ is a vector of monomials in the variable $p(t)$ and $\mathbf{w}_i \in \mathbb{R}^{n_\mathrm{w}}$ is the (unknown) vector of parameters. The monomials in the vector $\varphi(p(t))$ are specified a-priori.

Given the estimates of the coefficients obtained from LS-SVM approach (see eq. (2.14)) and their corresponding values at the training

points $\{p(t)\}_{t=1}^{N}$, i.e.,

$$\hat{c}_i(p(t)) = \sum_{k=1}^{N} \alpha_k K_i(p(k), p(t)) x_i(k), \quad i \in \mathbb{I}_1^{n_g} \tag{2.16}$$

we scale $\hat{c}_i(p(t))$ with the polynomial weights $w_i(p(t))$, i.e.,

$$\tilde{c}_i(p(t)) = w_i(p(t))\hat{c}_i(p(t)) = \mathbf{w}_i^\top \varphi(p(t))\hat{c}_i(p(t)), \quad i \in \mathbb{I}_1^{n_g},$$

where $\tilde{c}_i$ are the scaled LPV coefficients.

With the scaled coefficients, the considered LPV-ARX model becomes:

$$y(t) = \sum_{i=1}^{n_g} \mathbf{w}_i^\top \varphi(p(t))\hat{c}_i(p(t)) x_i(t) + e(t). \tag{2.17}$$

Now, to enforce the sparsity in the estimate of the (scaled) coefficients $\tilde{c}_i(p(t))$, a group LASSO term penalizing the l_∞-norm of the polynomial coefficient vectors $\mathbf{w}_i$ is minimized along with the residual error, leading to the following convex optimization problem:

$$\min_{\{\mathbf{w}_i\}_{i=1}^{n_g}} \sum_{t=1}^{N} \left(y(t) - \sum_{i=1}^{n_g} \mathbf{w}_i^\top \varphi(p(t))\hat{c}_i(p(t)) x_i(t) \right)^2 + \mu \sum_{i=1}^{n_g} \|\mathbf{w}_i\|_\infty \tag{2.18}$$

Note that the group LASSO term in (2.18) penalizes the mixed $\ell_{1,\infty}$-norm (i.e., sum of the infinity norms) of the parameter vectors $\mathbf{w}_i$, with $i = 1, \ldots, n_g$. The infinity norm is considered as a norm of the group so that, at the solution, the vector $\mathbf{w}_i$ is enforced to be either identically zero or full. Indeed, only the component of the vector $\mathbf{w}_i$ with largest absolute value affects the objective function in (2.18). Note that, when $\mathbf{w}_i$ is zero, the polynomial weight $w_i(\cdot) = \mathbf{w}_i^\top \varphi(\cdot)$ is null. Thus, the corresponding scaled coefficient function $\tilde{c}_i$ is also null. The hyper-parameter $\mu \geq 0$ is tuned by the user to balance the trade-off between minimizing the fitting error and minimizing the number of non-zero functions $\{\tilde{c}_i\}_{i=1}^{n_g}$ defining the LPV-ARX model with scaled coefficients reported in (2.17).

Summarizing, the shape of the p-dependent LPV model coefficients $\{c_i\}_{i=1}^{n_g}$ is initially obtained through a standard LS-SVM dedicated to LPV

identification. Then, polynomial weights $\{w_i(p(t))\}_{i=1}^{n_g}$ are used to reshape the estimated model coefficients $\{\hat{c}_i\}_{i=1}^{n_g}$ and, at the same time, to shrink the LPV model coefficients towards zero, thus reducing the complexity of the estimated LPV model (by minimizing the number of nonzero model coefficients $\{\tilde{c}_i\}_{i=1}^{n_g}$).

As the final estimate of the scaled coefficients $\{\tilde{c}_i\}_{i=1}^{n_g}$ will be biased because of the regularization term $\sum_{i=1}^{n_g}\|\mathbf{w}_i\|_\infty$ (see eq. (2.18)), an LPV model with reduced complexity, containing only the coefficients $\{\tilde{c}_i\}_{i=1}^{n_g}$ which have been detected to be nonzero, should be re-identified by using the non-regularized LS-SVM approach discussed in Section 2.2.2.

2.2.4 Simulation examples

This section shows the effectiveness of the proposed regularized LS-SVM method on two simulation examples. To study the statistical properties of the estimation, Monte-Carlo simulations of 100 runs are performed for each example. At each Monte-Carlo run, a new data set of inputs, scheduling variables and noises is generated. The output used in the training phase is corrupted by an additive zero-mean white noise e^{o} with Gaussian distribution. The effect of the noise e^{o} on the output signal is quantified through the Signal-to-Noise Ratio (SNR), defined as

$$\mathrm{SNR} = 10\log\frac{\sum_{t=1}^{N}\left(y(t)-e^{\mathrm{o}}(t)\right)^2}{\sum_{t=1}^{N}\left(e^{\mathrm{o}}(t)\right)^2}, \tag{2.19}$$

Radial basis functions are used as kernels to define the inner product among the feature maps ϕ_i. The values of the hyper-parameters λ, μ (eq. (2.18)) and σ_i (characterizing the RBF K_i in (2.13)) are chosen through a cross-calibration procedure, that is by maximizing (with a grid search) the *Best Fit Rate* (BFR) w.r.t. a calibration data set of length N_{C}. The BFR is defined as

$$\mathrm{BFR} = \max\left\{1-\sqrt{\frac{\sum_{t=1}^{N_{\mathrm{C}}}\left(y(t)-\hat{y}(t)\right)^2}{\sum_{t=1}^{N_{\mathrm{C}}}\left(y(t)-\bar{y}\right)^2}},0\right\} \tag{2.20}$$

with $\hat{y}(t)$ being the simulated model output and $\bar{y}$ being the sample mean of the output over the calibration set. In order to speed up the calibration procedure, the parameters σ_i are chosen to be equal (i.e., $\sigma_i = \sigma$ for all $i = 1, \ldots, n_g$). The results obtained after the training and calibration phase are validated on a noiseless data sequence. The BFR is used to assess the quality of the estimated models. All computations are carried out on a i5 1.7GHz Intel core processor with 4 GB of RAM running MATLAB R2013a. The CVX package [37] is used to solve Problem (2.18).

Example 1

The aim of the academic example reported in this subsection is threefold: 1. showing the capabilities of the presented regularization approach in detecting the correct structure of the underlying LPV data-generating systems; 2. showing that the variance of estimate obtained by non regularized LS-SVM can be reduced by first detecting the model structure; 3. comparing the performance, in terms of computational time, of the presented approach w.r.t. to the regularization approach proposed in [79]. Consider the LPV data-generating system taken from [79]:

$$y(t) = a_1^{\circ}(p(t))y(t-1) + a_2^{\circ}(p(t))y(t-2) + b_5^{\circ}(p(t))u(t-5) + e^{\circ}(t). \quad (2.21)$$

The p-dependent coefficients $a_1^{\circ}(p(t))$, $a_2^{\circ}(p(t)$ and $b_5^{\circ}(p(t))$ are described by the nonlinear functions:

$$\begin{aligned} a_1^{\circ}(p(t)) &= \begin{cases} -0.5, & \text{if } p(t) > 0.5 \\ -p(t), & \text{if } -0.5 \leq p(t) \leq 0.5 \\ 0.5, & \text{if } p(t) < -0.5 \end{cases} \\ a_2^{\circ}(p(t)) &= \sin(2\pi p(t)), \\ b_5^{\circ}(p(t)) &= p^3(t), \end{aligned}$$

The system is estimated from a training data set $\mathcal{D}_N$ of length $N = 500$, while a calibration data set of length $N_{\mathrm{C}} = 200$ is used to tune the hyperparameters λ, μ and σ. To gather data, the input u and the scheduling parameter p are chosen to be white-noise processes independent of each other with uniform distribution $\mathcal{U}(-1, 1)$. The standard deviation of the

noise e^o is 0.3. The average of the SNR over the over 100 Monte-Carlo runs is 7 dB. The identification problem is formulated in the LS-SVM setting by using over-parameterized LPV model structure (2.2) with $n_a = n_b = 10$.

First, the estimates of the coefficients $\{a_i\}_{i=1}^{n_a}$ and $\{b_j\}_{j=0}^{n_b}$ are obtained using the non-regularized LS-SVM approach described in Section 2.2.2. The chosen hyper-parameters, maximizing the BFR w.r.t. the calibration data set, are: $\lambda = 600$ and $\sigma = 0.4$. Then, second-order polynomials $\{w_i(p(t))\}_{i=1}^{n_g}$ are used as weights to re-shape the estimated coefficients $\{\hat{a}_i\}_{i=1}^{n_a}$ and $\{\hat{b}_j\}_{j=0}^{n_b}$, and thus to detect the LPV model structure by solving Problem (2.18). The chosen hyper-parameter μ, maximizing the BFR w.r.t. the calibration data set, is $\mu = 5$.

Table 2 and Table 3 show the maximum absolute values (over the training points $\{p(t)\}_{t=1}^{N}$) of the coefficient functions a_i and b_j obtained from the non-regularized LS-SVM approach, along with ones computed through the regularized LS-SVM version (denoted as R-LS-SVM) proposed in this contribution. The obtained results show that the proposed regularized LS-SVM approach is able to detect the correct underlying structure of the LPV data-generating system. Indeed, the only coefficients a_i and b_j which have been detected to have an (average) maximum absolute value larger than a threshold of 10^{-6} are: a_1, a_2 and b_5. Results in Table 2 and Table 3 also show that, as expected, the estimate of the nonzero coefficients obtained by the regularized LS-SVM is biased, because of the regularization penalty $\sum_{i=1}^{n_g} \|\mathbf{w}_i\|_\infty$ in eq. (2.18) (note that this a well-known problem affecting also parametric regularization methods like the LASSO). Therefore, the nonzero coefficient functions are re-estimated without the regularization term. Specifically, the coefficient functions which are detected to be null are discarded in the description of the LPV model (2.2) and a lower-complexity LPV model is re-identified through non-regularized LPV LS-SVM approach. The estimates of the nonzero coefficients a_1, a_2 and b_5 are plotted in Figure 1, which shows the mean estimate, along with the standard deviation intervals computed over the 100 Monte-Carlo runs. The estimate of the same

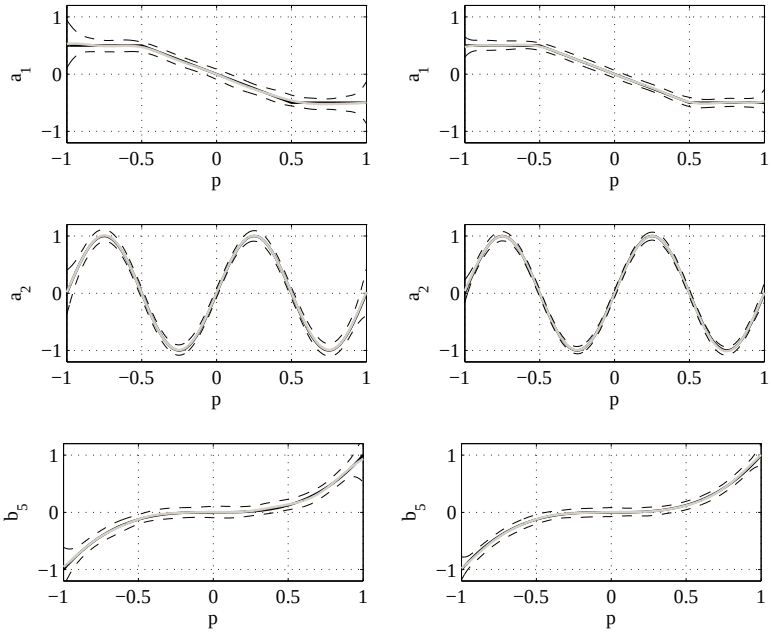

Figure 1: Example 1. Estimate of the non-zero coefficients obtained via non-regularized LS-SVM (left panels) and after model order selection (right panels). True function (solid black line), mean estimate (solid gray line) and standard deviation intervals (dashed black line) over the 100 Monte Carlo runs.

coefficients obtained with the non-regularized LPV LS-SVM approach is also plotted in the same figure. The obtained results show that detecting the LPV model structure is beneficial, in terms of variance reduction, in the final estimate of the coefficient functions. The BFR obtained using standard LS-SVM is 0.73, while a BFR equal to 0.92 is achieved by regularized LS-SVM evaluated on a noise-free validation data set.

For the sake of comparison, the regularization method of [79] is also run, by gridding the scheduling variable space into 20 equidistant points. In terms of model quality, the method in [79] is also able to detect the true LPV model structure. However, the benefits of the proposed method

Table 2: Example 1. Average and standard deviation (over 100 Monte-Carlo runs) of the maximum absolute value of the estimated LPV model coefficients $a_i(p(t))$.

$\lvert a_i(p(t))\rvert$	True Value	Mean (LS-SVM)	Mean (R-LS-SVM)	std (LS-SVM)	std (R-LS-SVM)
a_1	0.5	0.6245	0.4766	8.54e-02	6.63e-02
a_2	1	1.0498	0.9667	3.56e-02	3.81e-02
a_3	0	0.1853	2.61e-11	9.80e-02	5.51e-11
a_4	0	0.1788	2.25e-11	9.12e-02	3.48e-11
a_5	0	0.1858	2.63e-11	8.61e-02	4.01e-11
a_6	0	0.1939	3.29e-11	0.1052	6.14e-11
a_7	0	0.1987	2.47e-11	0.1033	3.78e-11
a_8	0	0.1891	2.24e-11	8.16e-02	3.17e-11
a_9	0	0.1935	2.31e-11	9.06e-02	3.43e-11
a_{10}	0	0.1904	3.02e-11	9.45e-02	5.96e-11

Table 3: Example 1. Average and standard deviation (over 100 Monte-Carlo runs) of the maximum absolute value of the estimated LPV model coefficients $b_j(p(t))$.

$\lvert b_j(p(t))\rvert$	True Value	Mean (LS-SVM)	Mean (R-LS-SVM)	std (LS-SVM)	std (R-LS-SVM)
b_0	0	0.1236	1.79e-11	6.30e-02	3.46e-11
b_1	0	0.1372	1.68e-11	7.10e-02	2.66e-11
b_2	0	0.1292	1.78e-11	5.80e-02	2.63e-11
b_3	0	0.1204	1.73e-11	5.64e-02	3.11e-11
b_4	0	0.1213	1.56e-11	6.32e-02	2.56e-11
b_5	1	1.0133	0.8748	0.1009	7.62e-03
b_6	0	0.1270	1.85e-11	5.62e-02	2.65e-11
b_7	0	0.1350	1.99e-11	5.66e-02	2.71e-11
b_8	0	0.1333	1.96e-11	5.92e-02	3.12e-11
b_9	0	0.1201	1.74e-11	5.66e-02	2.69e-11
b_{10}	0	0.1329	1.71e-11	5.63e-02	2.64e-11

w.r.t. [79] can be appreciated in terms of computational time. For fixed values of the hyper-parameters λ, μ and σ, the average CPU time (over the 100 Monte-Carlo runs) required by the proposed identification algorithm is 2 s. On the other hand, the average overall time required by the method of [79] to solve the same identification problem is 24 s (12x slower than the method proposed here).

Example 2. Multidimensional scheduling variable

The aim of the academic example considered in this section is to show the effectiveness of the proposed regularization scheme in the identification of LPV systems with multidimensional scheduling signals. This represents one of the main advantage of the proposed method over [79]. In fact, the method in [79] requires to grid the scheduling space, thus limiting its applicability to the identification of LPV systems with one/two-dimensional scheduling variables. The considered LPV data-generating system is described by the difference equation:

$$\begin{aligned} y(t) = a_1^{\circ}(p(t))y(t-1) + a_2^{\circ}(p(t))y(t-2) + \\ + b_4^{\circ}(p(t))u(t-4) + b_5^{\circ}(p(t))u(t-5) + e^{\circ}(t), \end{aligned} \tag{2.22}$$

where, $p(t) = [p_1(t) \;\; p_2(t) \;\; p_3(t)]^\top \in \mathbb{R}^3$. The unknown functions are,

$$\begin{aligned} a_1^{\circ}(p(t)) &= 0.3p_1^2(t) + 0.2p_2^2(t) - 0.1p_3^2(t), \\ a_2^{\circ}(p(t)) &= 0.2p_1(t) - 0.3p_2(t) + 0.1p_3(t), \\ b_4^{\circ}(p(t)) &= 0.2\sin(2\pi p_1(t)) + \sin(2\pi p_2(t)), \\ b_5^{\circ}(p(t)) &= 0.4\cos(2\pi p_2(t)) + 0.3\sin(2\pi p_3(t)). \end{aligned}$$

Training data set $\mathcal{D}_N$ of length $N = 3000$ is used for estimation, while a calibration data set of length $N_{\mathrm{C}} = 1000$ is used to tune the hyper-parameters λ, μ and σ. To generate the data set, the input u and the scheduling signals p_1, p_2 and p_3 are chosen to be white-noise processes independent of each other with uniform distribution $\mathcal{U}(-1, 1)$. The standard deviation of the noise e° is 0.08. The average of the SNR over the over 100 Monte-Carlo runs is 15 dB.

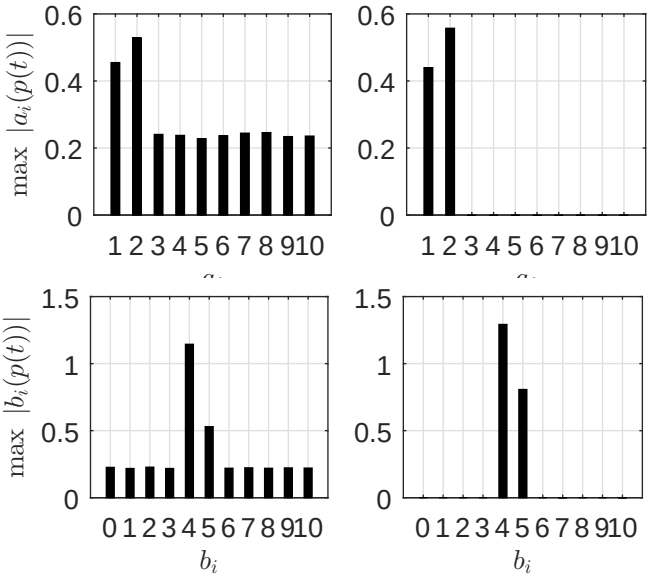

Figure 2: Example 2. Average (over 100 Monte-Carlo runs) of the maximum absolute value of the LPV model coefficients $\{a_i\}_{i=1}^{10}$ and $\{b_j\}_{j=0}^{10}$ obtained through non-regularized LS-SVM (left panels) and through regularized LS-SVM (right panels).

First, the estimates of the coefficients are obtained using the non-regularized LS-SVM approach described in Section 2.2.2 with hyper parameters, $\lambda = 900$ and $\sigma = 0.8$. Then, second-order polynomial functions $\{w_i(p(t))\}_{i=1}^{n_g}$ are used as weights to re-shape the estimated coefficients $\{\hat{a}_i\}_{i=1}^{n_a}$ and $\{\hat{b}_j\}_{j=0}^{n_b}$, and thus to detect the LPV model structure by solving Problem (2.18). The chosen hyper-parameter is $\mu = 25$. The total computational time (including calibration phase) required to solve the estimation problem is 302 s. For fixed values of λ, σ and μ, the average computational time is 15.1 s (i.e., 20 different combinations of the hyper-parameters λ, σ and μ have been tested in calibration). Figure 2 depicts the average maximum absolute values of the estimated coefficient functions over 100 Monte-Carlo runs. Results in Figure 2 show that the proposed regularized LS-SVM detects the true structure of the underlying dynamics, while a non-sparse LPV model is obtained by using

standard LS-SVM. This has an impact on the generalization properties of the model to unseen data. In fact, the BFR (in validation) obtained using standard LS-SVM is 0.62, while a BFR equal to 0.88 is achieved by regularized LS-SVM.

This section concludes the proposed non-parametric method based on regularized LS-SVM for model order selection of LPV-ARX models. In the next section, we consider the model order selection problem in a conventional parametric framework but with a more general *Output Error* (OE) noise structure models and with a more challenging *errors-in-variable* problem where the scheduling signal measurements are also corrupted by noise.

2.3 Parametric approach with noise corrupted scheduling signal

In parametric identification framework, unknown LPV model coefficients are described by a set of known a-priory selected basis functions. It is important to achieve a low variance of the model estimate by limiting the number of parameters to be identified that are associated with each basis function. Moreover, ignoring the effect of noise on the observations of the scheduling signals may lead to a bias in the final estimate and, as a consequence, also to an incorrect selection of the model order. We introduce a *bias-corrected cost* function for the identification of LPV systems from noise-corrupted observations of the output and scheduling signal measurements. The introduced cost function provides a bias-free parameter estimation along with the model order selection. The proposed identification approach has two main advantages: (i) the problem of model order selection can be handled by adding a LASSO-like penalty term to the bias-corrected cost function; (ii) it provides a bias-free cost as a criterion to tune some hyper-parameters influencing the final parameter estimate.

2.3.1 Problem formulation

Data-generating system

As a data-generating system, let us consider a discrete-time *single-input single-output* (SISO) LPV system described by the *Output Error* (OE) structure:

$$y_{\mathrm{o}}(k) = -\sum_{i=1}^{n_a^{\mathrm{o}}} a_i^{\mathrm{o}}(p_{\mathrm{o}}(k))y_{\mathrm{o}}(k-i) + \sum_{j=0}^{n_b^{\mathrm{o}}} b_j^{\mathrm{o}}(p_{\mathrm{o}}(k))u(k-j), \tag{2.23a}$$

$$y(k) = y_{\mathrm{o}}(k) + \nu_{\mathrm{o}}(k), \tag{2.23b}$$

where $u(k) \in \mathbb{R}$ and $y(k) \in \mathbb{R}$ are the measured input and output signals of the system at time k, respectively, $\nu_{\mathrm{o}}(k)$ is an additive zero-mean white noise corrupting the output, $p_{\mathrm{o}}(k)$ is the noise-free scheduling signal, which is assumed to take value in the compact set $\mathbb{P}$. The functions $a_i^{\mathrm{o}}(\cdot)$ and $b_j^{\mathrm{o}}(\cdot)$ are assumed to be polynomials of maximum degree n_g^{o} in the scheduling variable $p_{\mathrm{o}}(k)$, and they have to be estimated along with the parameters $n_a^{\mathrm{o}}, n_b^{\mathrm{o}}, n_g^{\mathrm{o}}$ defining the structure of the system (2.23). For the clarity of exposition, we consider the case of scalar scheduling signal $p_{\mathrm{o}}(k)$. Extension to the multidimensional case is straightforward.

In order to describe the data-generating system (2.23) in a compact form, let us introduce the following matrix notation:

$$\begin{aligned}
\mathbf{p}_{\mathrm{o}}(k) &= \begin{bmatrix} 1 & p_{\mathrm{o}}(k) & p_{\mathrm{o}}^2(k) \cdots p_{\mathrm{o}}^{n_g^{\mathrm{o}}}(k) \end{bmatrix}^\top, \\
\bar{a}_i^{\mathrm{o}} &= \begin{bmatrix} \bar{a}_{i,0}^{\mathrm{o}} & \bar{a}_{i,1}^{\mathrm{o}} \cdots & \bar{a}_{i,n_g^{\mathrm{o}}}^{\mathrm{o}} \end{bmatrix}^\top, \quad a_i^{\mathrm{o}}(p_{\mathrm{o}}(k)) = (\bar{a}_i^{\mathrm{o}})^\top \mathbf{p}_{\mathrm{o}}(k), \\
\bar{b}_j^{\mathrm{o}} &= \begin{bmatrix} \bar{b}_{j,0}^{\mathrm{o}} & \bar{b}_{j,1}^{\mathrm{o}} \cdots & \bar{b}_{j,n_g^{\mathrm{o}}}^{\mathrm{o}} \end{bmatrix}^\top, \quad b_j^{\mathrm{o}}(p_{\mathrm{o}}(k)) = (\bar{b}_j^{\mathrm{o}})^\top \mathbf{p}_{\mathrm{o}}(k), \\
\theta_{\mathrm{o}} &= \begin{bmatrix} (\bar{a}_1^{\mathrm{o}})^\top \cdots \left(\bar{a}_{n_a^{\mathrm{o}}}^{\mathrm{o}}\right)^\top & (\bar{b}_0^{\mathrm{o}})^\top \cdots \left(\bar{b}_{n_b^{\mathrm{o}}}^{\mathrm{o}}\right)^\top \end{bmatrix}^\top, \\
\chi_{\mathrm{o}}(k) &= [-y_{\mathrm{o}}(k-1) \cdots - y_{\mathrm{o}}(k-n_a^{\mathrm{o}}) \;\; u(k) \cdots u(k-n_b^{\mathrm{o}})]^\top, \\
\phi_{\mathrm{o}}(k) &= \chi_{\mathrm{o}}(k) \otimes \mathbf{p}_{\mathrm{o}}(k).
\end{aligned}$$

The data-generating system (2.23) can be then rewritten as:

$$y(k) = \phi_{\mathrm{o}}^\top(k)\theta_{\mathrm{o}} + \nu_{\mathrm{o}}(k). \tag{2.24}$$

Scheduling signal observations

The measurements $p(k)$ of the scheduling signal are assumed to be corrupted by an additive noise $\eta_{\text{o}}(k)$, i.e.,

$$p(k) = p_{\text{o}}(k) + \eta_{\text{o}}(k), \tag{2.25}$$

where $\eta_{\text{o}}(k) \sim \mathcal{N}(0, \sigma_\eta^2)$ is a zero mean white Gaussian noise uncorrelated with the noise corrupting the output signal, i.e., $\mathbb{E}\left[\eta_{\text{o}}(k)\nu_{\text{o}}(t)\right] = 0$ for all time indexes k and t.

LPV model structure

The following model structure ($\mathcal{M}$) is considered to estimate the data-generating system (2.23):

$$y(k) = -\sum_{i=1}^{n_a} a_i(p(k))y(k-i) + \sum_{j=0}^{n_b} b_j(p(k))u(k-j) + \epsilon(k), \tag{2.26}$$

with $\epsilon(k)$ denoting the residual term, modeling the mismatch between the true system and the model output. For the true system to belong to the model class $\mathcal{M}$, the parameters n_a and n_b defining the dynamical order of the model in (2.26) are chosen large enough so that $n_a \geq n_a^{\text{o}}$ and $n_b \geq n_b^{\text{o}}$. In other words, an over-parametrized model structure is used. Moreover, the functions $a_i : \mathbb{R} \to \mathbb{R}$ and $b_j : \mathbb{R} \to \mathbb{R}$ are parametrized with polynomial basis functions as follows:

$$a_i(p(k)) = \bar{a}_{i,0} + \sum_{s=1}^{n_g} \bar{a}_{i,s}p^s(k) = (\bar{a}_i)^\top \mathbf{p}(k), \tag{2.27a}$$

$$b_j(p(k)) = \bar{b}_{j,0} + \sum_{s=1}^{n_g} \bar{b}_{j,s}p^s(k) = (\bar{b}_j)^\top \mathbf{p}(k), \tag{2.27b}$$

where the degree n_g of the polynomials in (2.27) is also chosen large enough so that $n_g \geq n_g^{\text{o}}$ and $\mathbf{p}(k) = \begin{bmatrix}1 & p(k) & p^2(k) \cdots p^{n_g}(k)\end{bmatrix}^\top$.

Using the matrix notations introduced in Section 2.3.1, model (2.26) can be compactly written as:

$$y(k) = \phi^\top(k)\theta + \epsilon(k), \tag{2.28}$$

where $\theta = [\bar{a}_1^\top \cdots \bar{a}_{n_a}^\top \;\; \bar{b}_0^\top \cdots \bar{b}_{n_b}^\top]^\top \in \mathbb{R}^{n_\theta}$ is the vector of model parameters to be identified, $\bar{a}_i$ and $\bar{b}_i$ are defined similarly to $\bar{a}_i^{\mathrm{o}}$ and $\bar{b}_i^{\mathrm{o}}$, and $\phi(k)$ is the regressor with measured (thus, noise-corrupted) outputs and scheduling signals at time k, defined as $\phi(k) = \chi(k) \otimes \mathbf{p}(k)$, with

$$\chi(k) = [-y(k-1) \cdots -y(k-n_a) \;\; u(k) \cdots u(k-n_b)]^\top .$$

The identification problem addressed in this section aims at obtaining an asymptotically unbiased estimate of the "true" parameter vector θ_{o} along with the unknown parameters n_a^{o}, n_b^{o} and n_g^{o} from an N-length observed data sequence $\mathcal{D}_N = \{u(k), y(k), p(k)\}_{k=1}^N$ generated by (2.23).

2.3.2 Instrumental-variable estimate

As proposed in [51] and [78], an *instrumental variable* (IV) approach can be used to handle the bias due to the noise $\nu_{\mathrm{o}}(k)$ affecting the output signal measurements. This bias can be removed by choosing the instruments $z(k) \in \mathbb{R}^{n_\theta}$ in such a way that they are uncorrelated with the output noise $\nu_{\mathrm{o}}(k)$, i.e., $\mathbb{E}[z(k)\nu_{\mathrm{o}}(k)] = 0$ for all k. In the following, we show that, because of the effect of the noise on the scheduling signal, using only instrumental variables is not enough to achieve a consistent parameter estimate and an accurate selection of the model structure.

Consider the following IV-LASSO optimization problem for estimating a sparse model parameter vector θ:

$$\hat{\theta}_{\mathrm{IV}} = \underset{\theta}{\operatorname{argmin}} \;\; \mathcal{J}_{\mathrm{IV}}(\theta, \lambda, N), \tag{2.29}$$

with

$$\mathcal{J}_{\mathrm{IV}}(\theta, \lambda, N) = \left\| \frac{1}{N} \left(Z^\top Y - Z^\top \Phi \theta \right) \right\|_2^2 + \lambda \left\| \theta \right\|_1 , \tag{2.30}$$

where $Z = [z(1) \cdots z(N)]^\top$ is the matrix of instrumental variables, $\Phi = [\phi(1) \cdots \phi(N)]^\top$ is the regressor matrix, $Y = [y(1) \cdots y(N)]^\top$ is the noise-corrupted output observation vector. Note that the quadratic term in the definition of $\mathcal{J}_{\mathrm{IV}}$ is the loss function minimized in standard IV identification schemes [90]. The second term is used to enforce sparsity in the estimate of θ, and the hyper-parameter $\lambda \geq 0$ is tuned to balance the

trade-off between model complexity and data fitting. The chosen instruments $z(k)$ must be independent of the output noise realization $\nu_{\mathrm{o}}(k)$. Thus, a possible choice of $z(k)$ is

$$z(k) = [-\hat{y}(k-1)\cdots-\hat{y}(k-n_a)\ \ u(k)\cdots u(k-n_b)]^\top \otimes \mathbf{p}(k),$$

where $\hat{y}$ is an approximation of the noise-free output, independent of the noise ν_{o}, which can be obtained from an estimated (not necessarily unbiased) model of the system. An iterative algorithm can be implemented to 'refine' the instruments [51], by computing an estimate θ of the model parameters at each iteration, based on which the simulated output $\hat{y}$ is generated and used as an instrument at the next run.

Due to the noise in the measurements of p, the quadratic cost in equation (2.30) is asymptotically biased, in the sense that, asymptotically, its minimum is not achieved at the true system parameter vector θ_{o} (see Appendix 2.5 for a proof). In order to overcome this drawback, instead of the cost in equation (2.30), a bias-corrected cost function achieving a consistent estimate of θ_{o}, along with an accurate model order selection, is introduced in the following section.

2.3.3 Bias-corrected LASSO for sparse LPV identification

In this section, we formulate a bias-corrected version of the IV-LASSO cost $\mathcal{J}_{\mathrm{IV}}(\theta, \lambda, N)$ to obtain a consistent estimate of the model parameters as well as an accurate model order selection. It will be proved that the proposed biased-corrected cost function converges asymptotically (as $N \to \infty$) to the "true cost" function (i.e., a non-negative loss function which achieves its minimum at the true parameter vector θ_{o}). Such a bias-corrected cost function is also used as an optimal criterion to tune the regularization parameter λ via cross-validation. To perform model order selection, let us solve the optimization problem:

$$\hat{\theta}_{\mathrm{CIV}} = \underset{\theta}{\operatorname{argmin}}\ \ \mathcal{J}_{\mathrm{CIV}}(\theta, \lambda, N), \tag{2.31}$$

with

$$\mathcal{J}_{\mathrm{CIV}}(\theta, \lambda, N) = \left\| \frac{1}{N}\left(Z^\top Y - \Psi\theta\right) \right\|_2^2 + \lambda \left\|\theta\right\|_1. \tag{2.32}$$

The function $\mathcal{J}_{\mathrm{CIV}}(\theta, \lambda, N)$ will be referred to as *bias-corrected cost*. In case $\lambda = 0$, $\mathcal{J}_{\mathrm{CIV}}(\theta, \lambda, N)$ will be referred to as *non-regularized bias-corrected cost*.

The matrix Ψ appearing in the definition of $\mathcal{J}_{\mathrm{CIV}}(\theta, \lambda, N)$ was originally introduced in [78] and it is given by $\Psi = \sum_{k=1}^{N} \Psi_k$, where each matrix Ψ_k is constructed in such a way so as to satisfy the following conditions:

C1. the matrix Ψ_k only depends on the noise-corrupted observations of the data $\{u(k), p(k), y(k)\}_{k=1}^{N}$ and on the variance σ_η^2 of the noise $\eta_{\mathrm{o}}(k)$ corrupting the scheduling variable measurement $p(k)$.

C2. let $\Omega_k = z(k)[\chi(k) \otimes \mathbf{p}_{\mathrm{o}}(k)]^\top$. Then,

$$\lim_{N\to\infty} \frac{1}{N} \sum_{k=1}^{N} \Omega_k = \lim_{N\to\infty} \frac{1}{N} \sum_{k=1}^{N} \Psi_k \ \text{ w.p.1.}$$

Construction of Ψ_k

As explained in [78], the matrices Ψ_k satisfying conditions C1 and C2 can be constructed through the procedure outlined below (inspired by [80]), under the assumption that the variance σ_η^2 of the noise corrupting the scheduling observations $p(k)$ is known.

1. Compute the analytic expression of the conditional expectation matrix $\mathbb{E}\left\{\Omega_k = z(k)[\chi(k) \otimes \mathbf{p}_{\mathrm{o}}(k)]^\top | Y\right\}$. By construction, since each element of the vector $\mathbf{p}_{\mathrm{o}}(k)$ is a polynomial in $p_{\mathrm{o}}(k)$, the entries of $\mathbb{E}\{\Omega_k | Y\}$ are described by an affine combination of the monomials $p_{\mathrm{o}}(k)$, $p_{\mathrm{o}}^2(k)$, $p_{\mathrm{o}}^3(k)$, ...

2. Express the n^{th}-order monomial of the noise-free scheduling signal $p_{\mathrm{o}}^n(k)$ in terms of the expected value of the noise-corrupted monomial $p^n(k)$ and the noise variance σ_η^2 in terms of the *probabilists' Hermite polynomial*[1]:

$$p_{\mathrm{o}}^n(k) = \mathbb{E}\left\{ (n!) \sum_{m=0}^{\lfloor n/2 \rfloor} \frac{(-1)^m \sigma_\eta^{2m}}{m!(n-2m)!} \frac{p^{n-2m}(k)}{2^m} \right\} \tag{2.33}$$

[1] See Appendix 2.5.2 for details of the Hermite polynomial expression

3. Compute the matrix Ψ_k by replacing each of the monomials $p_{\mathrm{o}}(k)$, $p_{\mathrm{o}}^2(k)$, $p_{\mathrm{o}}^3(k)$, appearing in the analytic expression of $\mathbb{E}\left[\Omega_k|Y\right]$, with the term inside the expectation operator in (2.33). In this way, the matrix Ψ_k satisfies the following condition:

$$\mathbb{E}\left\{\frac{1}{N}\Omega_k|Y\right\} = \mathbb{E}\left\{\frac{1}{N}\Psi_k|Y\right\} \quad \forall k \in \mathbb{I}_1^N. \tag{2.34}$$

Property 1 *The computed matrices Ψ_k satisfy conditions C1 and C2 under the assumption that the amplitude of the measured output and of the scheduling signals is bounded.*

Proof: Condition C1 is satisfied from the construction of the matrix Ψ_k. From a direct application of Ninness' strong law of large numbers [68], it can be proved that Ψ_k also satisfies condition C2. The sketch of the proof detailed in [78, Appendix A2] is as follows :

Let $\{v(t)\}$ denote a sequence of random variables characterized by a constant C such that $\sum_{t=1}^{N}\sum_{s=1}^{N}\mathbb{E}\left\{v(t)v(s)\right\} < CN$. Then from the strong law of large numbers we have, $\frac{1}{N}\sum_{t=1}^{N} v(t) \to 0$ as $N \to \infty$.

Consider the variable $v_{i,j}(k) = [\Psi_k - \Omega_k]_{i,j}$ where $[\cdot]_{i,j}$ denotes the (i,j)-th entry of the matrix. As the noise corrupting scheduling signal is white, we have $\mathbb{E}\left\{v_{i,j}(k)v_{i,j}(t)|Y\right\} = 0$ for all $k, t \geq 0,\ k \neq t$. Moreover, since the output $y(k)$ and scheduling signal $p_{\mathrm{o}}(k)$ are bounded there exists a positive scalar $G_{i,j}$ such that $\mathbb{E}\left\{v_{i,j}(k)v_{i,j}(k)|Y\right\} < G_{i,j}$ for all $k > 0$. Thus, based on these considerations,

$$\sum_{k=1}^{N}\sum_{t=1}^{N}\mathbb{E}\left\{v_{i,j}(k)v_{i,j}(t)|Y\right\} = \sum_{k=1}^{N}\mathbb{E}\left\{v_{i,j}(k)v_{i,j}(k)|Y\right\} < G_{i,j}N$$

Thus, it follows that $\frac{1}{N}\sum_{t=1}^{N} v_{i,j}(t) \to 0$ as $N \to \infty$, or equivalently,

$$\lim_{N\to\infty}\frac{1}{N}\left[\sum_{k=1}^{N}\Psi_k\right]_{i,j} = \lim_{N\to\infty}\frac{1}{N}\left[\sum_{k=1}^{N}\Omega_k\right]_{i,j} \quad \text{w.p.1.}$$

Remark 1 *For clarity of exposition, the procedure outlined above to construct Ψ_k is demonstrated via an example in Appendix 2.5.2. The expression of $p_{\mathrm{o}}^n(k)$*

in terms of the expected value of the noise-corrupted observation $p^n(k)$ and noise variance σ_η^2 is not reported in [78] in terms of the Hermite polynomial (2.33), but in terms of recursive constructions which can be proved to have the compact expression in (2.33). See Appendix 2.5.2 for details.

Consistency of the bias-corrected cost function

In this section, we prove that minimizing the non-regularized bias corrected cost $\mathcal{J}_{\mathrm{CIV}}(\theta, 0, N)$ leads to a consistent estimate of the system parameters θ_{o}. This basically means that the bias on the estimated parameters $\hat{\theta}_{\mathrm{CIV}}$ due to the noise on the data vanishes as the length N of the training dataset increases.

In the following, we prove that the non-regularized bias-corrected cost $\mathcal{J}_{\mathrm{CIV}}(\theta, 0, N)$ converges pointwise (as $N \to \infty$) to the "true" cost $\mathcal{V}_{\mathrm{o}}(\theta, N)$ defined as:

$$\mathcal{V}_{\mathrm{o}}(\theta, N) = \left\| \frac{1}{N} \sum_{k=1}^{N} z(k) \left(y_{\mathrm{o}}(k) - [\chi_{\mathrm{o}}(k) \otimes \mathbf{p}_{\mathrm{o}}(k)]^\top \theta \right) \right\|_2^2, \tag{2.35}$$

which achieves its minimum $\mathcal{V}_{\mathrm{o}}(\theta, N) = 0$ at $\theta = \theta_{\mathrm{o}}$. In the following, we will assume that θ_{o} is the only minimizer of $\mathcal{V}_{\mathrm{o}}(\theta, N)$.

Property 2 *Let us define the cost function:*

$$\mathcal{J}_{\mathrm{o}}(\theta, N) = \left\| \frac{1}{N} \sum_{k=1}^{N} z(k) \left(y(k) - [\chi(k) \otimes \mathbf{p}_{\mathrm{o}}(k)]^\top \theta \right) \right\|_2^2. \tag{2.36}$$

Then, for any compact set $\Theta \subset \mathbb{R}^{n_\theta}$, the following property holds:

$$\lim_{N \to \infty} \mathcal{J}_{\mathrm{o}}(\theta, N) = \lim_{N \to \infty} \mathcal{V}_{\mathrm{o}}(\theta, N) \quad \forall \theta \in \Theta. \tag{2.37}$$

Proof: Consider the limit of the argument of the ℓ_2-norm in the cost (2.36):

$$\lim_{N\to\infty}\frac{1}{N}\sum_{k=1}^{N} z(k)\left(y(k)-\left[\chi(k)\otimes\mathbf{p}_{\mathrm{o}}(k)\right]^{\top}\theta\right) \tag{2.38a}$$

$$=\lim_{N\to\infty}\frac{1}{N}\sum_{k=1}^{N} z(k)y_{\mathrm{o}}(k) \tag{2.38b}$$

$$+\lim_{N\to\infty}\frac{1}{N}\sum_{k=1}^{N} z(k)\left(\nu_{\mathrm{o}}(k)-\left[(\chi(k)-\chi_{\mathrm{o}}(k))\otimes\mathbf{p}_{\mathrm{o}}(k)\right]^{\top}\theta\right) \tag{2.38c}$$

$$-\lim_{N\to\infty}\frac{1}{N}\sum_{k=1}^{N}\left(z(k)\left[\chi_{\mathrm{o}}(k)\otimes\mathbf{p}_{\mathrm{o}}(k)\right]^{\top}\theta\right), \tag{2.38d}$$

where the decomposition above is obtained by splitting $y(k) = y_{\mathrm{o}}(k) + \nu_{\mathrm{o}}(k)$ and $\chi(k) = \chi(k) - \chi_{\mathrm{o}}(k) + \chi_{\mathrm{o}}(k)$ in (2.38a). We now analyse the asymptotic behaviour of (2.38). The term (2.38c) converges to zero as N tends to infinity. This follows from the fact that $\nu_{\mathrm{o}}(k)$ and $(\chi(k) - \chi_{\mathrm{o}}(k))$ are zero mean noises and they are uncorrelated with the instrument $z(k)$. The remaining term is the sum of (2.38b) and (2.38d), which is equal to the argument of the ℓ_2-norm of the true cost $\mathcal{V}_{\mathrm{o}}(\theta, N)$ (2.35). Thus, since the argument of the ℓ_2-norm in (2.35) converges to the argument of the ℓ_2-norm in (2.36), because of continuity of the ℓ_2-norm, it follows that, $\lim_{N\to\infty}\mathcal{J}_{\mathrm{o}}(\theta, N) = \lim_{N\to\infty}\mathcal{V}_{\mathrm{o}}(\theta, N)$ for any $\theta \in \Theta$. ■

Next step is to prove that the non-regularized cost $\mathcal{J}_{\mathrm{CIV}}(\theta, 0, N)$ (2.32) converges asymptotically to $\mathcal{J}_{\mathrm{o}}(\theta, N)$, or equivalently, because of Property 2, to the true cost function $\mathcal{V}_{\mathrm{o}}(\theta, N)$. This also implies that minimizing the non-regularized bias-corrected cost $\mathcal{J}_{\mathrm{CIV}}(\theta, 0, N)$ provides a consistent parameter estimate $\hat{\theta}_{\mathrm{CIV}}$ which asymptotically converges to true parameter vector θ_{o}.

Property 3 *For any compact set $\Theta \subset \mathbb{R}^{n_\theta}$, the following property holds:*

$$\lim_{N\to\infty}\mathcal{J}_{\mathrm{CIV}}(\theta, 0, N) = \lim_{N\to\infty}\mathcal{J}_{\mathrm{o}}(\theta, N) \quad \forall\theta \in \Theta. \tag{2.39}$$

Proof: Let us rewrite the argument of the ℓ_2-norm of the bias-corrected

cost $\mathcal{J}_{\mathrm{CIV}}(\theta, 0, N)$ as:

$$\frac{1}{N}\sum_{k=1}^{N} z(k)y(k) - \frac{1}{N}\sum_{k=1}^{N} \Psi_k \theta. \tag{2.40}$$

By construction of matrix Ψ_k, we have (see Condition C2):

$$\lim_{N\to\infty} \frac{1}{N}\sum_{k=1}^{N} \Psi_k = \lim_{N\to\infty} \frac{1}{N}\sum_{k=1}^{N} \Omega_k. \tag{2.41}$$

Taking the limit of (2.40) and substituting (2.41) into (2.40) we obtain:

$$\lim_{N\to\infty} \frac{1}{N}\sum_{k=1}^{N} z(k)y(k) - \frac{1}{N}\sum_{k=1}^{N} \Psi_k \theta \tag{2.42a}$$

$$= \lim_{N\to\infty} \frac{1}{N}\sum_{k=1}^{N} z(k)y(k) - \lim_{N\to\infty} \frac{1}{N}\sum_{k=1}^{N} \Omega_k \theta = \tag{2.42b}$$

$$= \lim_{N\to\infty} \frac{1}{N}\sum_{k=1}^{N} z(k)y(k) - z(k)\left[\chi(k) \otimes \mathbf{p}_{\mathrm{o}}(k)\right]^{\top} \theta. \tag{2.42c}$$

Note that the argument of the limit in (2.42c) is the argument of the ℓ_2-norm of the cost $\mathcal{J}_{\mathrm{o}}(\theta, N)$. Thus, because of continuity of the ℓ_2-norm function, eq. (2.39) follows. ■

Combining Property 2 and Property 3, we obtain:

$$\lim_{N\to\infty} \mathcal{J}_{\mathrm{CIV}}(\theta, 0, N) = \lim_{N\to\infty} \mathcal{V}_{\mathrm{o}}(\theta, N) \;\; \forall \theta \in \Theta. \tag{2.43}$$

Condition (2.43) implies that the minimum $\hat{\theta}_{\mathrm{CIV}}$ of the non regularized bias-corrected cost function $\mathcal{J}_{\mathrm{CIV}}(\theta, 0, N)$ converges to the minimum of the true cost $\mathcal{V}_{\mathrm{o}}(\theta, N)$, i.e.,

$$\lim_{N\to\infty} \hat{\theta}_{\mathrm{CIV}} = \theta_{\mathrm{o}}. \tag{2.44}$$

The advantages of introducing the bias-corrected cost $\mathcal{J}_{\mathrm{CIV}}(\theta, 0, N)$ are twofold:

- it allows us to perform model order selection by adding an ℓ_1-regularization term and it gives a 'correct' quadratic error-fitting term in order to remove the bias (asymptotically) due to the noise affecting the scheduling variable observations.

- the bias-corrected cost $\mathcal{J}_{\mathrm{CIV}}(\theta, 0, N)$ is an unbiased criterion which can be used to assess the performance of the estimated model, and thus to tune the hyper-parameter λ via cross validation.

Note that, in the bias-corrected LASSO cost (2.32), the quadratic fitting-error term is asymptotically unbiased and the ℓ_1 regularization enforces sparsity in the final estimate, shrinking the component of the vector θ towards zero. Then, the finale estimate $\hat{\theta}_{\mathrm{CIV}}$ is actually biased. Nevertheless, once the model order is selected based on the regularized cost $\mathcal{J}_{\mathrm{CIV}}(\theta, \lambda, N)$, the zero components of the estimated parameter vector $\hat{\theta}_{\mathrm{CIV}}$ are discarded and a lower complex model is re-identified by minimizing the non-regularized bias-corrected cost $\mathcal{J}_{\mathrm{CIV}}(\theta, 0, N)$, thus obtaining a consistent estimate.

Estimation with unknown noise variance

In computing the bias correcting matrix Ψ_k (Section 2.3.3), the variance of the noise σ_η^2 corrupting the scheduling signal measurements is assumed to be known. This is quite a restrictive assumption. Nevertheless, an exhaustive grid search over the scalar σ_η^2 can be performed, and the non-regularized bias-corrected cost $\mathcal{J}_{\mathrm{CIV}}(\hat{\theta}_{\mathrm{CIV}}(\sigma_\eta^2), 0, N)$ can be used as a performance metric, on calibration data, to tune the "optimal" variance σ_η^2 via cross-validation. This tuning of σ_η^2 is simpler than the one used in [78], which requires to solve a set of nonlinear equations.

2.3.4 Simulation examples

This section illustrates the effectiveness of the proposed method on a simulation example. The main objective of the example is to show how the noise on the scheduling signal can deteriorate the performance of the instrumental variable (IV) scheme detailed in Section 2.3.2, which instead can be significantly improved if the proposed bias correction method is used.

Data-generating system and model structure

The considered data-generating system is of the form:

$$\begin{aligned} y_{\mathrm{o}}(k) &= a^{\mathrm{o}}_{1,2} p^2_{\mathrm{o}}(k) y_{\mathrm{o}}(k-1) + b^{\mathrm{o}}_{0,2} p^2_{\mathrm{o}}(k) u(k), \\ y(k) &= y_{\mathrm{o}}(k) + \nu_{\mathrm{o}}(k), \\ p(k) &= p_{\mathrm{o}}(k) + \eta_{\mathrm{o}}(k), \end{aligned}$$

with true parameters $a^{\mathrm{o}}_{1,2} = -0.8$ and $b^{\mathrm{o}}_{0,2} = 0.4$. The input $u(k)$ is taken as a white-noise with uniform distribution in the interval $[0 \quad 1]$. The noise-free scheduling variable is given by $p_{\mathrm{o}}(k) = \sin(0.1k) + 0.2\delta(k)$, with $\delta(k)$ being a random variable with Gaussian distribution $\mathcal{N}(0,1)$. The noise signals η_{o} and ν_{o} are white Gaussian noise processes with standard deviation $\sigma_\eta = 0.25$ and $\sigma_\nu = 0.06$. The influence of the noise on the signal measurements is quantified in terms of the signal-to-noise ratios:

$$\begin{aligned} \mathrm{SNR}_{\mathrm{y}} &= 10 \log \frac{\sum_{k=1}^{N} \left(y_{\mathrm{o}}(k) - \bar{y}_{\mathrm{o}}(k)\right)^2}{\sum_{k=1}^{N} \left(\nu_{\mathrm{o}}(k)\right)^2} = 9 \ \ \mathrm{dB}, \\ \mathrm{SNR}_{\mathrm{p}} &= 10 \log \frac{\sum_{k=1}^{N} \left(p_{\mathrm{o}}(k) - \bar{p}_{\mathrm{o}}(k)\right)^2}{\sum_{k=1}^{N} \left(\eta_{\mathrm{o}}(k)\right)^2} = 10 \ \ \mathrm{dB}, \end{aligned}$$

with $\bar{y}_{\mathrm{o}}$ and $\bar{p}_{\mathrm{o}}$ being the sample mean of y_{o} and p_{o}.

The following over-parametrized LPV model structure $\mathcal{M}$ is used to describe the behaviour of the system:

$$y(k) = \sum_{i=1}^{n_a} a_i(p(k), \theta) y(k-i) + \sum_{j=0}^{n_b} b_j(p(k), \theta) u(k-j) + \epsilon(k),$$

with $n_a = 4$ and $n_b = 2$. Each coefficient $a_i(.)$ and $b_j(.)$ is parametrized as a second order polynomial in p, i.e.,

$$a_i(p(k), \theta) = a_{i,0} + a_{i,1} p(k) + a_{i,2} p^2(k), \tag{2.45a}$$

$$b_j(p(k), \theta) = b_{j,0} + b_{j,1} p(k) + b_{j,2} p^2(k). \tag{2.45b}$$

The IV-LASSO method (based on the minimization of $\mathcal{J}_{\mathrm{IV}}(\theta, \lambda, N)$ in (2.30)) and the Bias-corrected IV-LASSO method (based on the minimization of $\mathcal{J}_{\mathrm{CIV}}(\theta, \lambda, N)$ in (2.31)) are compared. The model parameters are

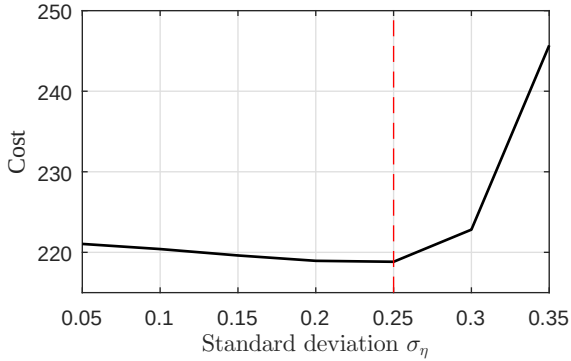

Figure 3: Non-regularized bias-corrected cost $\mathcal{J}_{\mathrm{CIV}}(\hat{\theta}_{\mathrm{CIV}}(\sigma_\eta), 0, N_c)$ vs noise standard deviation σ_η.

estimated based on a training set with $N = 8000$ samples. A recursive scheme is used to refine the instruments $z(k)$ as described in [78], in order to maximize the accuracy of the estimated parameters. To study the statistical properties of the algorithms, a Monte-Carlo study with 100 runs is performed. At each run, new inputs, scheduling variables and noise signals are generated.

A calibration set with $N_c = 1000$ samples is used to calibrate the hyper-parameter λ and to estimate the noise-variance σ_η^2 (see Section 2.3.3) through cross-validation. A grid search over the λ and σ_η is then performed, and the combined values of λ and σ_η which provide the best data fit over the calibration dataset are selected. Specifically, the non-regularized IV cost $\mathcal{J}_{\mathrm{IV}}(\hat{\theta}_{\mathrm{IV}}(\sigma_\eta), 0, N_c)$ and the bias-corrected cost $\mathcal{J}_{\mathrm{CIV}}(\hat{\theta}_{\mathrm{CIV}}(\sigma_\eta), 0, N_c)$, computed on the calibration dataset and for different values of λ and σ_η, are used to assess the performance of the estimated models. The computed value of λ is 9.2 for the IV scheme and 22 for the bias-corrected IV. As far as the estimate of the noise variance σ_η^2 is concerned, Figure 3 shows the non-regularized bias-corrected cost $\mathcal{J}_{\mathrm{CIV}}(\hat{\theta}_{\mathrm{CIV}}(\sigma_\eta), 0, N_c)$ (computed w.r.t. the calibration dataset) as a function of σ_η and for $\lambda = 22$. Note that the minimum of the bias-corrected cost is achieved for $\sigma_\eta = 0.25$, which is exactly the true value of the noise

Table 4: Average and standard deviation (over 100 Monte-Carlo runs) of the LPV model coefficients $a_i(p(t))$ estimated through IV-LASSO and bias-corrected (BC) IV-LASSO.

a	True Value	Mean (IV)	Mean (BC-IV)	std (IV)	std (BC-IV)
$a_{1,0}$	0	-0.1715	-0.0302	0.0279	0.0358
$a_{1,1}$	0	0	0	0.0050	0.0019
$a_{1,2}$	-0.8	-0.3882	-0.7120	0.0241	0.0457
$a_{2,0}$	0	0.1619	0.0026	0.0313	0.0066
$a_{2,1}$	0	0	0	0.0040	0.0031
$a_{2,2}$	0	-0.0040	0.0107	0.0129	0.0152
$a_{3,0}$	0	0.0044	0	0.0264	0
$a_{3,1}$	0	0	0	0.0027	0.0029
$a_{3,2}$	0	0	0	0.0041	0.0016
$a_{4,0}$	0	0.0146	0	0.0117	0.0012
$a_{4,1}$	0	0	0	0.0022	0.0014
$a_{4,2}$	0	-0.0011	0.0015	0.0057	0.0051

standard deviation. This shows that the bias-corrected cost function provides an efficient criterion to tune the parameter σ_η.

The estimates of the model parameters computed through the IV-LASSO and the bias-corrected IV-LASSO approach are reported in Table 4, which shows the mean and standard deviations of the estimated parameters $a_{i,j}$ over the Monte Carlo runs. The obtained results show that the bias-corrected IV-LASSO provides a parameter estimate close to the true parameters and it detects the structure of the underlying system quite accurately. On the other hand, the noise on the scheduling variable deteriorates the performance of the IV-LASSO scheme, which provides biased parameter estimates and, as a consequence, also a systematic error in the selection of the model structure. Similar results are obtained for the parameters $b_{i,j}$.

The performance of the IV-LASSO and the biased corrected IV-LASSO methods is tested on a validation dataset of length $N_{\text{val}} = 1000$. The Best Fit Rates (BFR) are reported in Table 5, and the true output and the esti-

Table 5: BFR computed on validation data.

Method	BFR
IV method	0.6061
Bias-Corrected IV	0.9146

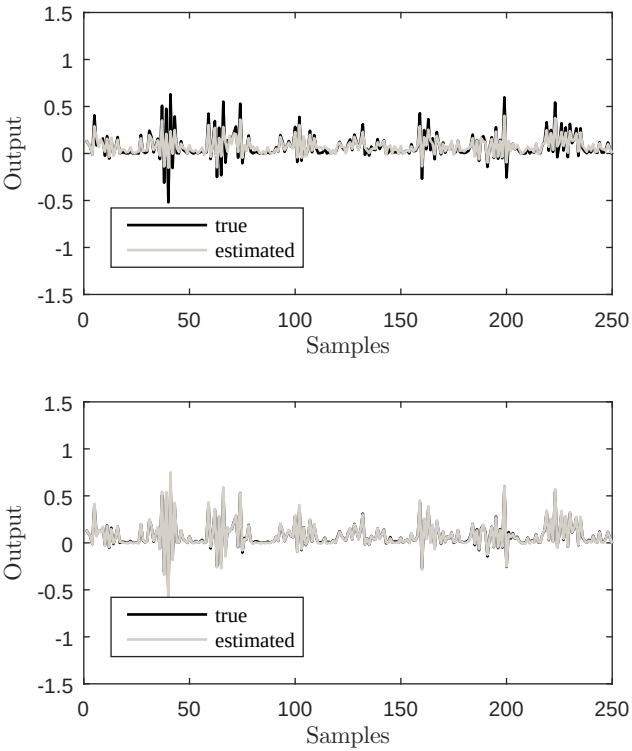

Figure 4: Validation data. Results achieved by IV-LASSO (top panel) and by the bias-corrected IV-LASSO (bottom panel).

mated model output $\hat{y}$ are plotted in Figure 4. The obtained results show the better performance of the bias-corrected IV-LASSO with respect to the classical IV-LASSO approach.

2.4 Conclusions

In this chapter, we have proposed two methods for LPV model order selection in non-parametric and parametric framework.

In the non-parametric setting, we have proposed a regularized LS-SVM method for sparse identification of LPV-ARX models. The dependence of the LPV model coefficients on the scheduling variable is not a-priori parameterized, and it is estimated, at the first stage, by using a standard (i.e., non-regularized) LS-SVM based regression approach. The obtained coefficients are then re-shaped by polynomial weights, and a penalty term is minimized in order to shrink the scaled model coefficients towards zero, thus enforcing a sparse structure in the estimated LPV model. The proposed method exploits the flexibility of the LS-SVM to reconstruct the underlying dependence of the LPV model coefficients on the scheduling signal, while the parametric structure of the scaling weights allows to select the dynamical structure of the model through a group-LASSO based approach. The reported simulation results show the capabilities of the proposed approach and its advantages in terms of computation time with respect to an other regularized LS-SVM approach available in the literature.

In the parametric setting, we have presented an extension of the bias-corrected IV method [78] for sparse identification of LPV-OE models from noisy scheduling variable measurements. The LPV model-structure selection problem is solved along with the asymptotically bias-free parameter estimation, through the formulation of a bias-corrected cost function. An ℓ_1-regularization term is used to enforce sparsity in the parameter vector estimate, and the trade-off between model complexity and data-fitting is balanced through cross-validation, using the introduced bias-corrected cost as a performance criterion.

2.5 Appendix

2.5.1 Biased instrumental variable cost

We prove that the quadratic term in (2.30) is an asymptotically biased cost, in the sense that the true parameter θ_{o} is not guaranteed to be the minimizer of $\mathcal{J}_{\mathrm{IV}}(\theta, 0, N)$ as $N \to \infty$. Let us rewrite the quadratic term in (2.30) as:

$$\begin{aligned}
\mathcal{J}_{\mathrm{IV}}(\theta,0,N) &= \left\| \frac{1}{N}\sum_{k=1}^{N} z(k)\Big(y(k) - [\chi(k)\otimes \mathbf{p}(k)]^{\top}\theta\Big)\right\|_2^2 \\
&= \left\| \frac{1}{N}\sum_{k=1}^{N} z(k)\left(y_{\mathrm{o}}(k) - [\chi_{\mathrm{o}}(k)\otimes \mathbf{p}_{\mathrm{o}}(k)]^{\top}\theta\right)\right. && \text{(2.46a)} \\
&+ \frac{1}{N}\sum_{k=1}^{N} z(k)\left(\nu_{\mathrm{o}}(k) - [(\chi(k)-\chi_{\mathrm{o}}(k))\otimes \mathbf{p}_{\mathrm{o}}(k)]^{\top}\theta\right) && \text{(2.46b)} \\
&- \left.\frac{1}{N}\sum_{k=1}^{N} z(k)\left([\chi(k)\otimes(\mathbf{p}(k)-\mathbf{p}_{\mathrm{o}}(k))]^{\top}\theta\right)\right\|_2 . && \text{(2.46c)}
\end{aligned}$$

As $N \to \infty$, (2.46b) converges to zero with probability 1 since $z(k)$ is uncorrelated with the output noise $\nu_{\mathrm{o}}(k)$ and $\chi(k) - \chi_{\mathrm{o}}(k)$ depends linearly on the past samples $\nu_{\mathrm{o}}(k)$. The term (2.46a) is the ideal cost, since it quantifies the error between the noise-free and the estimated output. For $\theta = \theta_{\mathrm{o}}$, (2.46a) is equal to 0. The asymptotically non-zero term (2.46c) causes the optimum of the asymptotic cost $\mathcal{J}_{\mathrm{IV}}(\theta, 0, N)$ to deviate from the optimum of the ideal cost.

2.5.2 Construction of Ψ_k

Hermite polynomial

The Hermite polynomial expression inside the expectation operator in (2.33) is a compact form to express noise-free monomials $p_{\mathrm{o}}^{n}(k)$ in terms of the expected value of the noise-corrupted observation $p^{n}(k)$ and noise variance σ_{η}^{2}, instead of the recursive relations reported in [78].

For example, for each monomial $p_o(k),\ p_o^2(k),\ p_o^3(k),\ldots$, compute the coefficients $\alpha_1(k),\ \alpha_2(k),\ \alpha_3(k),\ldots$ satisfying the following properties:

$$\begin{aligned} p_o(k) &= \mathbb{E}\{p(k) + \alpha_1(k)\} \\ p_o^2(k) &= \mathbb{E}\{p^2(k) + \alpha_2(k)\} \\ p_o^3(k) &= \mathbb{E}\{p^3(k) + \alpha_3(k)\} \\ &\vdots \end{aligned}$$

The coefficients $\alpha_1(k), \alpha_2(k), \alpha_3(k), \ldots$, should only depend on the observations of $p(k)$ and can be computed in a recursive way as follows (Assuming $\eta_o(k) \sim \mathcal{N}(0, \sigma_\eta^2)$) :

Consider $\alpha_1(k)$. Then,

$$\begin{aligned} p_o(k) &= \mathbb{E}\left\{p(k) + \alpha_1(k)\right\} = && \text{(2.47a)} \\ &= \mathbb{E}\left\{p_o(k) + \eta_o(k) + \alpha_1(k)\right\} = && \text{(2.47b)} \\ &= p_o(k) + \mathbb{E}\left\{\alpha_1(k)\right\}. && \text{(2.47c)} \end{aligned}$$

Equation (2.47c) implies that $\mathbb{E}\left\{\alpha_1(k)\right\} = 0$. Thus, a possible choice of $\alpha_1(k)$ is $\alpha_1(k) = 0$, and $p_o(k) = \mathbb{E}\left\{p(k)\right\}$.

Similarly, for α_2:

$$\begin{aligned} p_o^2(k) &= \mathbb{E}\left\{p^2(k) + \alpha_2(k)\right\} = \\ &= \mathbb{E}\left\{(p_o(k) + \eta_o(k))^2 + \alpha_2(k)\right\} \\ &= p_o^2(k) + \sigma_\eta^2 + \mathbb{E}\left\{\alpha_2(k)\right\}. \end{aligned}$$

Therefore, $\alpha_2(k)$ can be chosen as $\alpha_2(k) = -\sigma_\eta^2$, $p_o^2(k) = \mathbb{E}\left\{p^2(k) - \sigma_\eta^2\right\}$.

For $n \geq 2$, the values of $\alpha_n(k)$ can be recursively computed on the basis of the previously obtained estimates of $p_o(k),\ p^2(k),\ p_o^{n-1}(k),\ldots$.

For example, for $n = 3$,

$$\begin{aligned} p_o^3(k) &= \mathbb{E}\left\{p^3(k) + \alpha_3(k)\right\} \\ &= \mathbb{E}\left\{(p_o(k) + \eta_o(k))^3 + \alpha_3(k)\right\} \\ &= p_o^3(k) + 3p_o\sigma_\eta^2 + \mathbb{E}\left\{\alpha_3(k)\right\} \end{aligned}$$

This equation implies $\alpha_3(k)$ should be selected in such a way that $\mathbb{E}\{\alpha_3(k)\} = -3p_{\mathrm{o}}(k)\sigma_\eta^2$. From the previous computations, we have that $p_{\mathrm{o}}(k) = \mathbb{E}\{p(k)\}$. Thus we get, $\mathbb{E}\{\alpha_3(k)\} = \mathbb{E}\{-3p(k)\sigma_\eta^2\}$ giving a possible choice of $\alpha_3(k)$ as $\alpha_3(k) = -3p(k)\sigma_\eta^2$. Thus, $p_{\mathrm{o}}^3(k) = \mathbb{E}\{p^3(k) - 3p(k)\sigma_\eta^2\}$.

These relations (for all n) have closed-form expression in terms of the probabilists' Hermite polynomials as reported in (2.33).

Illustrative example to construct Ψ_k :

Consider the following LPV data generating system:

$$\begin{aligned} y_{\mathrm{o}}(k) &= a_{1,2}^{\mathrm{o}} p_{\mathrm{o}}^2(k) y_{\mathrm{o}}(k-1) + b_{0,2}^{\mathrm{o}} p_{\mathrm{o}}^2(k) u(k), \\ y(k) &= y_{\mathrm{o}}(k) + \nu_{\mathrm{o}}(k), \\ p(k) &= p_{\mathrm{o}}(k) + \eta_{\mathrm{o}}(k), \end{aligned}$$

According to the definitions introduced in Section 2.3.1, we have,

$$\chi(k) = [-y(k-1) \;\; u(k)]^\top, \;\; \mathbf{p}(k) = p^2(k), \;\; \mathbf{p}_{\mathrm{o}}(k) = p_{\mathrm{o}}^2(k),$$

and

$$\begin{aligned} \chi(k) \otimes \mathbf{p}_{\mathrm{o}}(k) &= \left[-y(k-1)p_{\mathrm{o}}^2(k) \;\; u(k)p_{\mathrm{o}}^2(k)\right]^\top, \\ z(k) &= \left[-\hat{y}(k-1)p^2(k) \;\; u(k)p^2(k)\right]^\top, \end{aligned}$$

where $\hat{y}(k-1)$ is the simulated output used as an instrument which is uncorrelated with the output noise $\nu_{\mathrm{o}}(k)$.

Then, the matrix $\Omega_k = z(k)(\chi(k) \otimes \mathbf{p}_{\mathrm{o}}(k))^\top$ (defined in condition C2) is given by,

$$\begin{bmatrix} \hat{y}(k-1)y(k-1)p^2(k)p_{\mathrm{o}}^2(k) & \hat{y}(k-1)u(k)p^2(k)p_{\mathrm{o}}^2(k) \\ -u(k)y(k-1)p^2(k)p_{\mathrm{o}}^2(k) & u^2(k)p^2(k)p_{\mathrm{o}}^2(k) \end{bmatrix}$$

It follows that, by taking the expectation w.r.t. scheduling variable noise $\eta_{\mathrm{o}}(k)$, the expectation matrix $\mathbb{E}\left[\Omega_k | Y\right]$ is given by,

$$(p_{\mathrm{o}}^4(k) + p_{\mathrm{o}}^2(k)\sigma_\eta^2) \begin{bmatrix} \hat{y}(k-1)y(k-1) & \hat{y}(k-1)u(k) \\ -u(k)y(k-1) & u^2(k) \end{bmatrix}$$

Now, since from (2.33) we have,

$$p_{\text{o}}^2(k) \;=\; \mathbb{E}\left\{p^2(k) - \sigma_\eta^2\right\} \text{ and } p_{\text{o}}^4(k) \;=\; \mathbb{E}\left\{p^4(k) + 3\sigma_\eta^4 - 6\sigma_\eta^2 p^2(k)\right\},$$

then, the matrix Ψ_k for $k = 1, \ldots, N$, is given by,

$$\Psi_k = (p^4(k) - 5p^2(k)\sigma_\eta^2 + 2\sigma_\eta^4) \begin{bmatrix} \hat{y}(k-1)y(k-1) & \hat{y}(k-1)u(k) \\ -u(k)y(k-1) & u^2(k) \end{bmatrix}$$

Chapter 3

Closed-loop identification of LPV models

In the previous chapter, identification of LPV input-output models was addressed in an open-loop setting, i.e., using the data collected by performing an open-loop experiment. The underlying assumption was that the open-loop system is stable for all input and scheduling signal trajectories. However, due to safety constraints and unstable open-loop dynamics, system identification of many real-world processes often require gathering data from closed-loop experiments. Identification of plant models from closed-loop data imposes more challenges as the data collected from the closed-loop is less informative and input signals are correlated with noise corrupting output.

In this chapter, we present a bias-correction scheme for closed-loop identification of *Linear Parameter-Varying Input-Output* (LPV-IO) models, which aims at correcting the bias caused by the correlation between the input signal exciting the process and output noise. Furthermore, we extend the bias-correction framework for the case where the scheduling signals are also corrupted by noise. The proposed identification algorithm provides a consistent estimate of the open-loop model parameters by correcting the bias stemming from the noise corrupting the output as well as the scheduling signal measurements.

3.1 Introduction

3.1.1 Motivation

Many real world systems must be identified based on data collected from closed-loop experiments. This is typical for open-loop unstable plants, where a feedback controller is necessary to perform the experiments, and in many applications in which a controller is needed to keep the system at certain operating points. Safety, performance, and economic requirements are further motivations to operate in closed-loop.

From the system identification point of view, one of the main issues which makes identification from closed-loop experiments more challenging than in the open-loop setting is due to the correlation between the plant input and output noise. If such a correlation is not properly taken into account, approaches that work in open-loop may fail when closed-loop data is used [55]. Several remedies have been proposed in the literature to overcome this problem, especially for the *Linear Time-Invariant* (LTI) case (see [27] and [34] for an overview). These approaches can be classified in: *direct methods*, which neglect the existence of the feedback loop and apply prediction error methods directly on the input-output data after properly parametrizing the noise model; *indirect methods*, where the closed-loop system is identified first and the model of the open-loop plant is then extracted exploiting the knowledge of the controller and of the feedback structure; *joint input-output methods*, which treat the measured input and output signals as the outputs of an augmented multi-variable system driven by external disturbances. The model of the open-loop process is then extracted based on the estimate of different transfer functions of the augmented system. Unlike indirect methods, an exact knowledge of the controller is not needed.

Unfortunately, the extension of these approaches to the Linear Parameter Varying (LPV) case is not straightforward, mainly because the classical theoretical tools which are commonly used in closed-loop LTI identification no longer hold in the LPV setting [98], such as transfer functions and commutative properties of operators. Therefore, only few contributions addressing identification of LPV systems from closed-loop

data are available in the literature. A subspace method, which can be applied both for open- and closed-loop identification of LPV models was proposed in [103]. The idea of this method is to construct a matrix approximating the product between the extended time-varying observability and controllability matrices, and later use an LPV extension of the predictor subspace approach originally proposed in [25]. As far as the identification of *LPV Input-Output* (LPV-IO) models is concerned, the closed-loop output error approach proposed in [47; 48] in the LTI setting is extended in [15] to the identification of LPV-IO models, whose parameters are estimated recursively through a parameter adaptation algorithm. *Instrumental-Variable* (IV) based methods are proposed in [2; 3; 100]. The contribution in [2] is mainly focused on the identification of *quasi*-LPV systems, where the scheduling variable is a function of the output. The main idea in [2] is to recursively estimate the output signal (and thus the scheduling variable) through recursive least-squares and later use the estimated signals (instead of the measurements) to obtain a consistent estimate of the open-loop model parameters through IV methods. An indirect approach is used in [3], where IV methods are used to estimate a model of the closed-loop system based on pre-filtered external reference and output signals. The plant parameters are later extracted from the estimated closed-loop model using plant-controller separation methods. In [100], an iterative *Refined Instrumental Variable* (RIV) approach is proposed for closed-loop identification of LPV-IO models with Box-Jenkins noise structures. At each iteration of the IV algorithm, the signals are pre-filtered by stable LTI filters constructed using the parameters estimated at the previous iteration. The filtered signals are then used to build the instruments, which are used to recompute an (improved) estimate of the model parameters. Unlike the methods in [2; 3], which are restricted to the case of LTI controllers, the approach in [100] can handle both LTI and LPV controllers.

3.1.2 Contributions

In this chapter, we present a bias-correction approach for closed-loop identification of LPV systems. The main idea underlying bias-correction methods is to eliminate the bias from ordinary *Least Squares* (LS) to obtain a consistent estimate of the model parameters. Bias-correction methods have been used in the past for the identification of LTI systems both in the open-loop [41; 109] and closed-loop setting [36; 110], as well as for open-loop identification of nonlinear [80] and LPV systems from noisy scheduling variable observations [78]. The main idea behind the closed-loop identification algorithm proposed in this chapter is to quantify the asymptotic bias due to the correlation between the plant input and the measurement noise, based on the available measurements. Recursive relations are derived to compute the asymptotic bias based on the knowledge of the controller and of the closed-loop structure of the system. Furthermore, in order to handle a more realistic scenario where not only the output signal, but also the scheduling variables are corrupted by a measurement noise, the proposed approach is combined with the ideas presented in [78], with the following improvements:

- an analytic expression, in terms of Hermite polynomials, is provided to compute the bias-correcting term used to handle the noise on the scheduling variable;
- as the bias-correcting term depends on the variance of the noise corrupting the scheduling variable, a bias-corrected cost function is introduced. This cost function serves as a tuning criterion to determine the value of the unknown noise variance via cross-validation.

Overall, the proposed closed-loop LPV identification approach offers a computationally low-demanding algorithm which:

(i) provides a consistent estimate of the model parameters;

(ii) can be applied under LTI or LPV controller structures;

(iii) does not require to identify the closed-loop LPV system;

(iv) can handle noisy observations of the scheduling signal.

3.1.3 Outline

The considered identification problem is formulated in Section 3.2. Section 3.3 describes the proposed closed-loop bias-correction approach that is extended in Section 3.4 to handle the case of identification from noisy measurements of the scheduling signal. Two case studies are reported in Section 3.5 to show the effectiveness of the presented method.

3.1.4 Notation

The notation used throughout the chapter is introduced in this section. Let $\mathbb{R}^n$ be the set of real vectors of dimension n. The i-th element of a vector $x \in \mathbb{R}^n$ is denoted by x_i and $\|x\|^2 = x^\top x$ denotes the square of the ℓ_2-norm of x. For matrices $A \in \mathbb{R}^{m\times n}$ and $B \in \mathbb{R}^{p\times q}$, the Kronecker product between A and B is denoted by $A \otimes B \in \mathbb{R}^{mp\times nq}$. Given a matrix A, the symbol $[A]_{n\times m}$ means that A is a matrix of dimension $n \times m$. Let $\mathbb{I}_a^b$ be the sequence of successive integers $\{a, a+1, \cdots, b\}$, with $a < b$. The floor function is denoted by $\lfloor\cdot\rfloor$, where $\lfloor m \rfloor$ is the largest integer less than or equal to m. The expected value of a function f w.r.t. the random vector $x \in \mathbb{R}^n$ is denoted by $\mathbb{E}_{x_1,\cdots,x_n}\{f(x)\}$. The subscript $x_1, \cdots, x_n$ is dropped from $\mathbb{E}_{x_1,\cdots,x_n}$ when its meaning is clear from the context.

3.2 Problem formulation

3.2.1 Data generating system

By referring to Figure 5, consider the LPV data-generating closed-loop system $\mathcal{S}_\text{o}$. We assume that the plant $\mathcal{G}_\text{o}$ is described by the LPV difference equations with output-error noise

$$\mathcal{G}_\text{o} : \left\{ \begin{array}{rl} \mathcal{A}_\text{o}(q^{-1}, p_\text{o}(k))x(k) & = \mathcal{B}_\text{o}(q^{-1}, p_\text{o}(k))u(k), \\ y(k) & = x(k) + e(k), \end{array} \right. \tag{3.1}$$

and that the controller $\mathcal{K}_\text{o}$ is a *known* LPV or LTI system described by

$$\mathcal{K}_\text{o} : \mathcal{C}_\text{o}(q^{-1}, p_\text{o}(k))u(k) = \mathcal{D}_\text{o}(q^{-1}, p_\text{o}(k))\left(r(k) - y(k)\right), \tag{3.2}$$

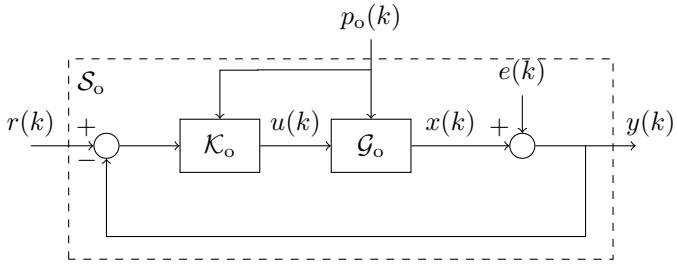

Figure 5: Closed-loop LPV data-generating system

where $r(k)$ is a bounded reference signal of the closed-loop system $\mathcal{S}_{\mathrm{o}}$, $u(k) \in \mathbb{R}$ and $y(k) \in \mathbb{R}$ are the measured input and output signals of the plant $\mathcal{G}_{\mathrm{o}}$, respectively; $x(k)$ is noise-free output; $e(k) \sim \mathcal{N}(0, \sigma_e^2)$ is an additive zero-mean white Gaussian noise with variance σ_e^2 corrupting the output signal; $p_{\mathrm{o}}(k) : \mathbb{N} \to \mathbb{P}$ is the measured (noise-free) scheduling signal and $\mathbb{P} \subseteq \mathbb{R}^{n_p}$ is a compact set where $p_{\mathrm{o}}(k)$ is assumed to take values. In order not to make the notation too complex, from now on we assume that $p_{\mathrm{o}}(k)$ is scalar (i.e., $n_p = 1$). The operator q denotes the time shift (i.e., $q^{-i}x(k) = x(k-i)$), and $\mathcal{A}_{\mathrm{o}}(q^{-1}, p_{\mathrm{o}}(k))$, $\mathcal{B}_{\mathrm{o}}(q^{-1}, p_{\mathrm{o}}(k))$, $\mathcal{C}_{\mathrm{o}}(q^{-1}, p_{\mathrm{o}}(k))$ and $\mathcal{D}_{\mathrm{o}}(q^{-1}, p_{\mathrm{o}}(k))$ are polynomials in q^{-1} of degree n_a, n_b, n_c and $n_d - 1$, respectively, defined as follows:

$$\mathcal{A}_{\mathrm{o}}(q^{-1}, p_{\mathrm{o}}(k)) = 1 + \sum_{i=1}^{n_a} a_i^{\mathrm{o}}(p_{\mathrm{o}}(k))q^{-i},$$

$$\mathcal{B}_{\mathrm{o}}(q^{-1}, p_{\mathrm{o}}(k)) = \sum_{i=1}^{n_b} b_i^{\mathrm{o}}(p_{\mathrm{o}}(k))q^{-i},$$

$$\mathcal{C}_{\mathrm{o}}(q^{-1}, p_{\mathrm{o}}(k)) = 1 + \sum_{i=1}^{n_c} c_i^{\mathrm{o}}(p_{\mathrm{o}}(k))q^{-i},$$

$$\mathcal{D}_{\mathrm{o}}(q^{-1}, p_{\mathrm{o}}(k)) = \sum_{i=0}^{n_d-1} d_{i+1}^{\mathrm{o}}(p_{\mathrm{o}}(k))q^{-i},$$

where the coefficient functions $a_i^{\mathrm{o}}, b_i^{\mathrm{o}}, c_i^{\mathrm{o}}, d_i^{\mathrm{o}}$ are supposed to be polynomials in $p_{\mathrm{o}}(k)$, i.e.,

$$a_i^{\mathrm{o}}(p_{\mathrm{o}}(k)) = \bar{a}_{i,0}^{\mathrm{o}} + \sum_{s=1}^{n_g} \bar{a}_{i,s}^{\mathrm{o}} p_{\mathrm{o}}^s(k), \tag{3.3a}$$

$$b_i^{\mathrm{o}}(p_{\mathrm{o}}(k)) = \bar{b}_{i,0}^{\mathrm{o}} + \sum_{s=1}^{n_g} \bar{b}_{i,s}^{\mathrm{o}} p_{\mathrm{o}}^s(k), \tag{3.3b}$$

$$c_i^{\mathrm{o}}(p_{\mathrm{o}}(k)) = \bar{c}_{i,0}^{\mathrm{o}} + \sum_{s=1}^{n_g} \bar{c}_{i,s}^{\mathrm{o}} p_{\mathrm{o}}^s(k), \tag{3.3c}$$

$$d_i^{\mathrm{o}}(p_{\mathrm{o}}(k)) = \bar{d}_{i,0}^{\mathrm{o}} + \sum_{s=1}^{n_g} \bar{d}_{i,s}^{\mathrm{o}} p_{\mathrm{o}}^s(k), \tag{3.3d}$$

with $\bar{a}_{i,s}^{\mathrm{o}} \in \mathbb{R}$ and $\bar{b}_{i,s}^{\mathrm{o}} \in \mathbb{R}$ being *unknown* real constants to be identified, while $\bar{c}_{i,s}^{\mathrm{o}} \in \mathbb{R}$ and $\bar{d}_{i,s}^{\mathrm{o}} \in \mathbb{R}$ are *known* coefficients characterizing the controller $\mathcal{K}_{\mathrm{o}}$. In order not to burden the notation, the polynomials in (3.3) are assumed to have the same degree n_g.

The following assumptions are made for the closed-loop data generating system:

A1. the measurement noise $e(k)$ is uncorrelated with the scheduling signal $p_{\mathrm{o}}(k)$ and with the external reference signal $r(k)$;

A2. to avoid algebraic loops, the open-loop plant is strictly causal, i.e., $b_0^{\mathrm{o}}(p_{\mathrm{o}}(k)) = 0$;

A3. the controller ensures closed-loop stability of the system $\mathcal{S}_{\mathrm{o}}$ for any scheduling trajectory $p_{\mathrm{o}}(k) \in \mathbb{P}$.

In order to describe the plant $\mathcal{G}_{\mathrm{o}}$ in a compact form, the following matrix notation is introduced:

$$\bar{a}_i^{\mathrm{o}} = \begin{bmatrix} \bar{a}_{i,0}^{\mathrm{o}} & \bar{a}_{i,1}^{\mathrm{o}} \cdots & \bar{a}_{i,n_g}^{\mathrm{o}} \end{bmatrix}^\top,$$

$$\bar{b}_j^{\mathrm{o}} = \begin{bmatrix} \bar{b}_{j,0}^{\mathrm{o}} & \bar{b}_{j,1}^{\mathrm{o}} \cdots & \bar{b}_{j,n_g}^{\mathrm{o}} \end{bmatrix}^\top,$$

$$\theta_{\mathrm{o}} = \begin{bmatrix} \left(\bar{a}_1^{\mathrm{o}}\right)^\top \cdots \left(\bar{a}_{n_a}^{\mathrm{o}}\right)^\top & \left(\bar{b}_1^{\mathrm{o}}\right)^\top \cdots \left(\bar{b}_{n_b}^{\mathrm{o}}\right)^\top \end{bmatrix}^\top,$$

$$
\begin{aligned}
&\mathbf{p}_{\mathrm{o}}(k) = \begin{bmatrix} 1 & p_{\mathrm{o}}(k) & p_{\mathrm{o}}^2(k) \cdots p_{\mathrm{o}}^{n_g}(k) \end{bmatrix}^\top, \\
&\chi_{\mathrm{o}}(k) = [-x(k-1) \cdots - x(k-n_a), \;\; u(k-1) \cdots u(k-n_b)]^\top, \\
&\phi_{\mathrm{o}}(k) = \chi_{\mathrm{o}}(k) \otimes \mathbf{p}_{\mathrm{o}}(k).
\end{aligned} \tag{3.4}
$$

Based on the above notation, the plant $\mathcal{G}_{\mathrm{o}}$ in (3.1) can be rewritten as follows:

$$
\mathcal{G}_{\mathrm{o}} : y(k) = \phi_{\mathrm{o}}^\top(k)\theta_{\mathrm{o}} + e(k). \tag{3.5}
$$

3.2.2 Model structure for identification

The following parametrized model structure $\mathcal{M}_\theta$ is considered to describe the true LPV plant $\mathcal{G}_{\mathrm{o}}$ in (3.1):

$$
\mathcal{M}_\theta : y(k) = -\sum_{i=1}^{n_a} a_i(p_{\mathrm{o}}(k)) y(k-i) + \sum_{j=1}^{n_b} b_j(p_{\mathrm{o}}(k)) u(k-j) + \epsilon(k), \tag{3.6}
$$

where $\epsilon(k)$ is the residual term.

The functions $a_i : \mathbb{R} \to \mathbb{R}$ and $b_j : \mathbb{R} \to \mathbb{R}$ are parametrized as follows:

$$
a_i(p_{\mathrm{o}}(k)) = \bar{a}_{i,0} + \sum_{s=1}^{n_g} \bar{a}_{i,s} p_{\mathrm{o}}^s(k) = \bar{a}_i^\top \mathbf{p}_{\mathrm{o}}(k), \tag{3.7a}
$$

$$
b_j(p_{\mathrm{o}}(k)) = \bar{b}_{j,0} + \sum_{s=1}^{n_g} \bar{b}_{j,s} p_{\mathrm{o}}^s(k) = \bar{b}_i^\top \mathbf{p}_{\mathrm{o}}(k). \tag{3.7b}
$$

Note that, since the contribution in this chapter aims at presenting a consistent closed-loop identification algorithm, the problem of model structure selection is not addressed. Thus, we assume that both the true plant $\mathcal{G}_{\mathrm{o}}$ and the model $\mathcal{M}_\theta$ share the same parameters n_a, n_b and n_g.

By using a similar matrix notation already introduced to describe the true plant $\mathcal{G}_{\mathrm{o}}$ in (3.5), the LPV model $\mathcal{M}_\theta$ in (3.6) can be written in the linear regression form:

$$
\mathcal{M}_\theta : y(k) = \phi^\top(k)\theta + \epsilon(k), \tag{3.8}
$$

where

$$
\theta = [\bar{a}_1^\top \cdots \bar{a}_{n_a}^\top \;\; \bar{b}_1^\top \cdots \bar{b}_{n_b}^\top]^\top, \tag{3.9}
$$

is the vector of model parameters to be identified and $\phi(k)$ is the regressor with measured outputs and scheduling signals at time k, defined as

$$\phi(k) = \chi(k) \otimes \mathbf{p}_{\mathrm{o}}(k), \tag{3.10}$$

with

$$\chi(k) = [-y(k-1), \ldots, -y(k-n_a),\ u(k-1), \ldots, u(k-n_b)]^{\top}. \tag{3.11}$$

Problem 1 *The identification problem addressed in this chapter aims at computing a consistent estimate of the true system parameter vector θ_{o}, given the model orders n_a, n_b and n_g and an N-length observed data sequence $\mathcal{D}_N = \{u(k), y(k), p_{\mathrm{o}}(k), r(k)\}_{k=1}^{N}$ of data generated by the closed-loop system $\mathcal{S}_{\mathrm{o}}$ in Figure 5. To this aim, a novel identification algorithm based on asymptotic bias-corrected least squares is described in the next sections.*

3.3 Bias-corrected least squares

It is well known that ordinary least squares give an asymptotically biased estimate of the model parameters due to the feedback structure [91]. In this section we quantify this bias and show how to remove it to give a consistent estimate of the model parameter vector θ.

3.3.1 Bias in the least-squares estimate

Consider the LS estimate $\hat{\theta}_{\mathrm{LS}}$ given by:

$$\hat{\theta}_{\mathrm{LS}} = \underbrace{\left(\frac{1}{N}\sum_{k=1}^{N}\phi(k)\phi^{\top}(k)\right)}_{\Gamma_N}^{-1}\frac{1}{N}\sum_{k=1}^{N}\phi(k)y(k), \tag{3.12}$$

under the assumption that matrix Γ_N is invertible. In order to compute the difference between the LS estimate $\hat{\theta}_{\mathrm{LS}}$ and true system parameters

θ_{o}, the output signal (3.5) is rewritten as follows:

$$\begin{aligned} y(k) &= \phi_{\mathrm{o}}^{\top}(k)\theta_{\mathrm{o}} + e(k) \\ &= \left[\chi_{\mathrm{o}}(k) \otimes \mathbf{p}_{\mathrm{o}}(k)\right]^{\top} \theta_{\mathrm{o}} + e(k) \\ &= \left[\chi(k) \otimes \mathbf{p}_{\mathrm{o}}(k)\right]^{\top} \theta_{\mathrm{o}} + \left[(\chi_{\mathrm{o}}(k) - \chi(k)) \otimes \mathbf{p}_{\mathrm{o}}(k)\right]^{\top} \theta_{\mathrm{o}} + e(k) \\ &= \phi^{\top}(k)\theta_{\mathrm{o}} + \Delta\phi(k)\theta_{\mathrm{o}} + e(k), \end{aligned} \tag{3.13}$$

with

$$\Delta\phi(k) = \left[(\chi_{\mathrm{o}}(k) - \chi(k)) \otimes \mathbf{p}_{\mathrm{o}}(k)\right]^{\top} = \phi_{\mathrm{o}}^{\top}(k) - \phi^{\top}(k). \tag{3.14}$$

Based on the representation of $y(k)$ in (3.13), the difference between the least-square estimate $\hat{\theta}_{\mathrm{LS}}$ and the true system parameter vector θ_{o} can be expressed as follows:

$$\begin{aligned} \hat{\theta}_{\mathrm{LS}} - \theta_{\mathrm{o}} &= \Gamma_N^{-1} \sum_{k=1}^{N} \frac{1}{N} \phi(k) y(k) - \theta_{\mathrm{o}} \\ &= \Gamma_N^{-1} \frac{1}{N} \sum_{k=1}^{N} \phi(k) \left(\phi^{\top}(k)\theta_{\mathrm{o}} + \Delta\phi(k)\theta_{\mathrm{o}} + e(k)\right) - \theta_{\mathrm{o}} \\ &= \underbrace{\Gamma_N^{-1} \frac{1}{N} \sum_{k=1}^{N} \phi(k)\phi^{\top}(k)}_{\Gamma_N} \theta_{\mathrm{o}} + \Gamma_N^{-1} \frac{1}{N} \sum_{k=1}^{N} \phi(k)\Delta\phi(k)\theta_{\mathrm{o}} \\ &\quad + \Gamma_N^{-1} \frac{1}{N} \sum_{k=1}^{N} \phi(k) e(k) - \theta_{\mathrm{o}} \\ &= \underbrace{\Gamma_N^{-1} \frac{1}{N} \sum_{k=1}^{N} \phi(k)\Delta\phi(k)\theta_{\mathrm{o}}}_{B_{\Delta}(\theta_{\mathrm{o}}, \phi(k), \Delta\phi(k))} + \underbrace{\Gamma_N^{-1} \frac{1}{N} \sum_{k=1}^{N} \phi(k) e(k)}_{B_e} \end{aligned} \tag{3.15}$$

Because of strict causality of the plant $\mathcal{G}_{\mathrm{o}}$ (see Assumption A2), the regressor $\phi(k)$ is uncorrelated with the current value of the noise $e(k)$. Thus, the term B_e in (3.15) asymptotically (as $N \to \infty$) converges to zero with probability 1 ($w.p.$ 1). This is due to the fact that the components of the regressor $\phi(k)$ are uncorrelated with the zero-mean white noise

$e(k)$ (in fact, because of Assumption A1, the regressor $\phi(k)$ is a function of only past values of the noise $e(k-1), e(k-2), \cdots$). Therefore, asymptotically, the bias in the LS estimate $\hat{\theta}_{\mathrm{LS}}$ is only due to the term $B_\Delta(\theta_{\mathrm{o}}, \phi(k), \Delta\phi(k))$, i.e.,

$$\lim_{N\to\infty} \hat{\theta}_{\mathrm{LS}} - \theta_{\mathrm{o}} = \lim_{N\to\infty} B_\Delta(\theta_{\mathrm{o}}, \phi(k), \Delta\phi(k)).$$

Note that, since the bias term $B_\Delta(\theta_{\mathrm{o}}, \phi(k), \Delta\phi(k))$ depends on the true system parameters θ_{o} as well as on the noise-free regressor $\phi_{\mathrm{o}}(k)$, it cannot be computed and thus it cannot be simply removed from the LS estimate $\hat{\theta}_{\mathrm{LS}}$.

In order to overcome the first difficulty due to the dependence of $B_\Delta(\theta_{\mathrm{o}}, \phi(k), \Delta\phi(k))$ on θ_{o}, the following estimate, inspired by [78], is introduced:

$$\tilde{\theta}_{\mathrm{CLS}} = \hat{\theta}_{\mathrm{LS}} - B_\Delta(\tilde{\theta}_{\mathrm{CLS}}, \phi(k), \Delta\phi(k)), \tag{3.16}$$

with

$$B_\Delta(\tilde{\theta}_{\mathrm{CLS}}, \phi(k), \Delta\phi(k)) = \Gamma_N^{-1} \frac{1}{N} \sum_{k=1}^{N} \phi(k)\Delta\phi(k)\tilde{\theta}_{\mathrm{CLS}}.$$

The main idea behind (3.16) is to correct the least-squares estimate $\hat{\theta}_{\mathrm{LS}}$ by removing the bias term B_Δ, which is evaluated at the parameter estimate $\tilde{\theta}_{\mathrm{CLS}}$ instead of at the unknown system parameters θ_{o}. Note that (3.16) provides an implicit expression for the estimate $\tilde{\theta}_{\mathrm{CLS}}$, as the term B_Δ depends on $\tilde{\theta}_{\mathrm{CLS}}$ itself. By simple algebraic manipulations, (3.16) can be rewritten as follows:

$$\begin{aligned}
\tilde{\theta}_{\mathrm{CLS}} =& \hat{\theta}_{\mathrm{LS}} - \Gamma_N^{-1} \frac{1}{N} \sum_{k=1}^{N} \phi(k)\Delta\phi(k)\tilde{\theta}_{\mathrm{CLS}} \\
=& \hat{\theta}_{\mathrm{LS}} - \Gamma_N^{-1} \frac{1}{N} \sum_{k=1}^{N} \phi(k)\phi_{\mathrm{o}}^\top(k)\tilde{\theta}_{\mathrm{CLS}} + \Gamma_N^{-1}\Gamma_N\tilde{\theta}_{\mathrm{CLS}} \\
=& \Gamma_N^{-1} \left(\frac{1}{N} \sum_{k=1}^{N} \phi(k)y(k) \right) - \Gamma_N^{-1} \frac{1}{N} \sum_{k=1}^{N} \phi(k)\phi_{\mathrm{o}}^\top(k)\tilde{\theta}_{\mathrm{CLS}} + \tilde{\theta}_{\mathrm{CLS}}.
\end{aligned}$$

Thus,

$$\tilde{\theta}_{\mathrm{CLS}} = \left(\frac{1}{N} \sum_{k=1}^{N} \phi(k) \phi_{\mathrm{o}}^{\top}(k) \right)^{-1} \left(\frac{1}{N} \sum_{k=1}^{N} \phi(k) y(k) \right). \tag{3.17}$$

Using the definition $\Delta\phi(k) = \phi_{\mathrm{o}}^{\top}(k) - \phi^{\top}(k)$, (3.17) can be written as

$$\tilde{\theta}_{\mathrm{CLS}} = \mathbf{R}_N^{-1} \left(\frac{1}{N} \sum_{k=1}^{N} \phi(k) y(k) \right), \tag{3.18}$$

where

$$\mathbf{R}_N = \frac{1}{N} \left(\sum_{k=1}^{N} \phi(k) \phi^{\top}(k) + \sum_{k=1}^{N} \phi(k) \Delta\phi(k) \right).$$

Property 4 *Assuming that the following limit exists:*

$$\lim_{N \to \infty} \mathbf{R}_N^{-1},$$

then $\tilde{\theta}_{\mathrm{CLS}}$ *is a consistent estimate of true system parameters* θ_{o}*, i.e.,*

$$\lim_{N \to \infty} \tilde{\theta}_{\mathrm{CLS}} = \theta_{\mathrm{o}} \ \ w.p.\ 1. \tag{3.19}$$

Proof: By substituting (3.13) into (3.18), we obtain

$$\tilde{\theta}_{\mathrm{CLS}} = \mathbf{R}_N^{-1} \underbrace{\frac{1}{N} \left(\sum_{k=1}^{N} \phi(k) (\phi^{\top}(k) + \Delta\phi(k)) \right)}_{\mathbf{R}_N} \theta_{\mathrm{o}} + \mathbf{R}_N^{-1} \left(\frac{1}{N} \sum_{k=1}^{N} \phi(k) e(k) \right).$$

Since the regressor $\phi(k)$ is uncorrelated with the current value of the noise $e(k)$, the term $\frac{1}{N} \sum_{k=1}^{N} \phi(k) e(k)$ asymptotically converges to zero $w.p.\ 1$. Thus,

$$\lim_{N \to \infty} \tilde{\theta}_{\mathrm{CLS}} = \theta_{\mathrm{o}} \ \ w.p.\ 1.$$

As $\Delta\phi(k)$ depends on the unknown noise-free regressors $\phi_{\mathrm{o}}(k)$ the estimate $\tilde{\theta}_{\mathrm{CLS}}$ in (3.18) cannot be computed. To overcome this problem, the term $\phi(k)\Delta\phi(k)$ is replaced by a bias-eliminating matrix Ψ_k, which is constructed (as explained in the following section) in such a way that

it only depends on the available measurements $\mathcal{D}_N$ and it satisfies the following property:

$$\mathbf{C1}: \lim_{N\to\infty} \frac{1}{N} \sum_{k=1}^{N} \phi(k)\Delta\phi(k) = \lim_{N\to\infty} \frac{1}{N} \sum_{k=1}^{N} \Psi_k \;\; w.p.\ 1.$$

3.3.2 Construction of the bias-eliminating Ψ_k

A bias-eliminating matrix Ψ_k satisfying condition **C1** is constructed by evaluating the expected value of the matrix $\mathbb{E}\{\phi(k)\Delta\phi(k)\}$, as follows:

$$\begin{aligned}
\Psi_k &= \mathbb{E}\{\phi(k)\Delta\phi(k)\} \\
&= \mathbb{E}\{(\chi(k) \otimes \mathbf{p}_{\mathrm{o}}(k))([(\chi_{\mathrm{o}}(k) - \chi(k)) \otimes \mathbf{p}_{\mathrm{o}}(k)]^{\top})\} \\
&= \mathbb{E}\{(\chi(k) \otimes \mathbf{p}_{\mathrm{o}}(k))(\left[(\chi_{\mathrm{o}}(k) - \chi(k))^{\top} \otimes (\mathbf{p}_{\mathrm{o}}^{\top}(k))\right])\} \\
&= \mathbb{E}\{\left[\chi(k)(\chi_{\mathrm{o}}(k) - \chi(k))^{\top}\right] \otimes \left[\mathbf{p}_{\mathrm{o}}(k)\mathbf{p}_{\mathrm{o}}^{\top}(k)\right]\} \\
&= \mathbb{E}\{\Upsilon_k \otimes \mathbf{P}_{\mathrm{o}}(k)\} \\
&= \mathbb{E}\{\Upsilon_k\} \otimes \mathbf{P}_{\mathrm{o}}(k),
\end{aligned} \tag{3.20}$$

with

$$\Upsilon_k = \chi(k)(\chi_{\mathrm{o}}(k) - \chi(k))^{\top}, \tag{3.21a}$$

$$\mathbf{P}_{\mathrm{o}}(k) = \mathbf{p}_{\mathrm{o}}(k)\mathbf{p}_{\mathrm{o}}^{\top}(k). \tag{3.21b}$$

The derivations reported above follow from the mixed-product property of the Kronecker product

$$(A \otimes B)\,(C \otimes D) = (AC) \otimes (BD)\,. \tag{3.22}$$

Property 5 *The matrix $\mathbb{E}\{\Upsilon_k\}$ is given by*

$$[\mathbb{E}\{\Upsilon_k\}]_{(n_a+n_b)\times(n_a+n_b)} = \Lambda_k = \left[\begin{array}{c|c} (\Upsilon_k^y)_{n_a\times n_a} & \mathbf{0}_{n_a\times n_b}, \\ \hline (\Upsilon_k^u)_{n_b\times n_a} & \mathbf{0}_{n_b\times n_b}, \end{array}\right] \tag{3.23}$$

where Υ_k^y and Υ_k^u are upper triangular matrices,

$$\Upsilon_k^y = \begin{bmatrix} f_1(k-1) & f_2(k-2) & \cdots & f_{n_a}(k-n_a) \\ 0 & f_1(k-2) & \ddots & \vdots \\ \vdots & \ddots & \ddots & f_2(k-n_a) \\ 0 & \cdots & 0 & f_1(k-n_a) \end{bmatrix}, \tag{3.24a}$$

$$\Upsilon_k^u = \begin{bmatrix} g_1(k-1) & g_2(k-2) & \cdots & \cdots & g_{n_a}(k-n_a) \\ 0 & g_1(k-2) & \ddots & & \vdots \\ \vdots & \ddots & \ddots & & \vdots \\ 0 & \cdots & 0 & \ddots & g_{n_a-n_b+1}(n_a-n_b+1) \end{bmatrix}, \tag{3.24b}$$

and

$$\begin{aligned} f_m(k-j) &= \mathbb{E}\{-y(k-j+m-1)e(k-j)\}, \\ g_m(k-j) &= \mathbb{E}\{u(k-j+m-1)e(k-j)\} \quad \forall m = \mathbb{I}_1^{n_a}, \end{aligned}$$

and

$$f_m(k) = g_m(k) = 0 \quad \text{for} \quad k \leq 0. \tag{3.25}$$

Proof: See Appendix 3.7.1.

Property 6 *The relation between $f_m(k)$ and $g_m(k)$ can be expressed by the following recursion, initialized with $f_1(k) = -\sigma_e^2$ for all $k = 1, \ldots, N$,*

$$g_m(k) = -\sum_{i=1}^{\min(n_c, m-1)} c_i(p_o(k+m-1))g_{m-i}(k) \tag{3.26a}$$

$$+\sum_{j=1}^{\min(n_d, m)} d_j(p_o(k+m-1))f_{m-j+1}(k), \tag{3.26b}$$

$$f_m(k) = -\sum_{i=1}^{m-2} a_i^o(p_o(k+m-1))f_{m-i}(k) \tag{3.26c}$$

$$-\sum_{j=1}^{\min(n_b, m-1)} b_j^o(p_o(k+m-1))g_{m-j}(k). \tag{3.26d}$$

Proof: See Appendix 3.7.2 .

Remark 2 *In the case of open-loop data, the input signal is uncorrelated with the measurement noise affecting the output, i.e.,* $\mathbb{E}\{u(k-i)e(k-j)\} = 0$, $\forall i \neq j$. *Moreover, as the measurement noise is assumed to be white, i.e.,* $\mathbb{E}\{y(k-i)e(k-j)\} = 0 \quad \forall i \neq j$, *we have that*

1. $\Upsilon_k^u = \mathbf{0}_{n_b \times n_a}$,
2. Υ_k^y *is a diagonal matrix with the diagonal entries* $[\Upsilon_k^y]_{i,i} = -\sigma_e^2$, *and thus it does not depend on the true system parameter vector* θ_{o}.

The above matrices can be used to remove the bias in the identification of open-loop LPV models with an output-error type noise structure. ■

3.3.3 Bias corrected estimate

The matrix Ψ_k, which actually depends on the true system parameter θ_{o}, is constructed using Property 5 and Property 6 (namely, (3.20), (3.23) and (3.26)) using an estimated parameter vector $\hat{\theta}_{\mathrm{CLS}}$ instead of the unknown θ_{o}. Specifically, an implicit expression for the final bias-corrected estimate is given by:

$$\hat{\theta}_{\mathrm{CLS}} = \left(\frac{1}{N}\sum_{k=1}^{N}\left(\phi(k)\phi^{\top}(k) + \Psi_k(\hat{\theta}_{\mathrm{CLS}})\right)\right)^{-1}\left(\frac{1}{N}\sum_{k=1}^{N}\phi(k)y(k)\right) \tag{3.27}$$

The main properties enjoyed by the estimate $\hat{\theta}_{\mathrm{CLS}}$ in (3.27) are reported in the following.

Property 7 *Assume that the following limit*

$$\lim_{N\to\infty}\left(\frac{1}{N}\sum_{k=1}^{N}\left(\phi(k)\phi^{\top}(k) + \Psi_k(\theta_{\mathrm{o}})\right)\right)^{-1} \tag{3.28}$$

exists. Then, asymptotically, the true system parameter vector θ_{o} *is a solution of* (3.27), *namely, for* $\theta = \theta_{\mathrm{o}}$,

$$\theta = \lim_{N\to\infty}\left(\frac{1}{N}\sum_{k=1}^{N}\left(\phi(k)\phi^{\top}(k) + \Psi_k(\theta)\right)\right)^{-1}\left(\frac{1}{N}\sum_{k=1}^{N}\phi(k)y(k)\right), \tag{3.29}$$

where the limit in (3.29) *holds* $w.p.\ 1$. *Thus, if* θ_{o} *is the unique solution of* (3.29)*, then the estimate* $\hat{\theta}_{\mathrm{CLS}}$ *in* (3.27) *is consistent, i.e.,*

$$\lim_{N\to\infty} \hat{\theta}_{\mathrm{CLS}} = \theta_{\mathrm{o}}. \tag{3.30}$$

Proof: By construction, $\mathbb{E}\{\Psi_k(\theta_{\mathrm{o}})\} = \mathbb{E}\{\phi(k)\Delta\phi(k)\}$, then condition **C1** follows from Ninness' strong law of large numbers [68]. See Property 1 in Chapter 2 for a detailed proof. By substituting

$$y(k) = \left(\phi^{\top}(k) + \Delta\phi(k)\right)\theta_{\mathrm{o}} + e(k),$$

into the right-hand side of (3.29), we obtain

$$\left(\frac{1}{N}\sum_{k=1}^{N}\phi(k)\phi^{\top}(k)+\Psi_k(\theta_{\mathrm{o}})\right)^{-1}\left(\frac{1}{N}\sum_{k=1}^{N}\phi(k)\left(\phi^{\top}(k)+\Delta\phi(k)\right)\right)\theta_{\mathrm{o}} \tag{3.31a}$$

$$+\left(\frac{1}{N}\sum_{k=1}^{N}\left(\phi(k)\phi^{\top}(k)+\Psi_k(\theta_{\mathrm{o}})\right)\right)^{-1}\left(\frac{1}{N}\sum_{k=1}^{N}\phi(k)e(k)\right). \tag{3.31b}$$

As the regressor $\phi(k)$ is uncorrelated with the white noise $e(k)$, (3.31b) converges to zero $w.p.\ 1$ as $N \to \infty$. Furthermore, from condition **C1**, it follows that (3.31a) converges to θ_{o} as $N \to \infty$. Thus, (3.29) holds for $\theta = \theta_{\mathrm{o}}$. Furthermore, taking the limit of the left- and right-hand side of (3.27), (3.30) follows from (3.29) and uniqueness assumption. ■

Note that (3.27) provides an implicit expression for the bias-corrected estimate $\hat{\theta}_{\mathrm{CLS}}$. In order to overcome this problem, (3.27) is solved iteratively as detailed in Algorithm 1. The main idea of Algorithm 1 is to compute the bias-eliminating matrix Ψ_k at each step τ, using the estimate $\hat{\theta}_{\mathrm{CLS}}^{(\tau-1)}$ obtained at step $\tau - 1$ and then to compute $\hat{\theta}_{\mathrm{CLS}}^{(\tau)}$ based on (3.27). Algorithm 1 can be initialized with a random vector $\hat{\theta}_{\mathrm{CLS}}^{(0)}$ or, for instance, with the LS estimates $\hat{\theta}_{\mathrm{LS}}$ in (3.12). Although convergence of this algorithm is not theoretically proven, and its final solution may depend on the chosen initial condition, Algorithm 1 seems to be quite insensitive to initial conditions and its convergence has been empirically observed from numerical tests (cf. Section 3.5.1).

Algorithm 1 Iterative bias-correction algorithm

Input: noise variance σ_e^2; tolerance ϵ; maximum number $\tau^{\max}$ of iterations; initial condition $\hat{\theta}_{\text{CLS}}^{(0)}$.

1. **let** $\tau \leftarrow 0$;
2. **while:** $\tau \leq \tau^{\max}$
   2.1. **let** $\tau \leftarrow \tau + 1$;
   2.2. **compute** $\Psi_k(\hat{\theta}_{\text{CLS}}^{(\tau-1)})$ using Eqs. (3.20), (3.23) and (3.26);
   2.3. **calculate** the bias corrected estimates $\hat{\theta}_{\text{CLS}}^{(\tau)}$ in (3.27);
   2.4. **if** $\left\| \hat{\theta}_{\text{CLS}}^{(\tau)} - \hat{\theta}_{\text{CLS}}^{(\tau-1)} \right\|_2 \leq \epsilon$
      2.4.1. **exit** while;
   2.5. **end if**
3. **end while**

Output: Bias-corrected estimate $\hat{\theta}_{\text{CLS}}$.

3.3.4 Estimate with unknown noise variance

In computing the bias-correcting matrix Ψ_k (and thus the bias-corrected estimate $\hat{\theta}_{\text{CLS}}$ in (3.27)), the variance σ_e^2 of the noise corrupting the output signal measurements is assumed to be known. This is a restrictive assumption which may limit the applicability of the proposed identification approach. However, the unknown noise variance can be simply tuned via cross-validation. Specifically, the following cost can be minimized through a grid search over $\sigma_{e,i}^2$:

$$\mathcal{J}(\hat{\theta}_{\text{CLS}}^i, \sigma_{e,i}^2) = \frac{1}{N_{\text{c}}} \sum_{k=1}^{N_{\text{c}}} \left(y(k) - \hat{x}^i(k) \right)^2, \tag{3.32}$$

where N_{c} is the length of the calibration set. The sequence $\hat{x}^i$ denotes the open-loop simulated output of the model with parameters $\hat{\theta}_{\text{CLS}}^i$ estimated from Algorithm 1 using a given value of $\sigma_{e,i}^2$ as a guess for σ_e^2.

The simulated output is defined as:

$$\hat{x}^i(k) = \hat{\phi}_{\text{cal}}^\top(k)\hat{\theta}^i_{\text{CLS}},$$

where the regressor $\hat{\phi}_{\text{cal}}(k)$ (as defined in (3.4)) is given by

$$\begin{aligned}\hat{\chi}(k) &= \left[-\hat{x}^i(k-1)\cdots-\hat{x}^i(k-n_a)\;\; u(k-1)\cdots u(k-n_b)\right]^\top,\\ \hat{\phi}_{\text{cal}}(k) &= \hat{\chi}(k)\otimes \mathbf{p}_{\text{o}}(k).\end{aligned}$$

It is worth stressing that the cost $\mathcal{J}$ in (3.32) is minimized only with respect to the scalar parameter σ_e. Specifically, once $\sigma_e^2 = \sigma_{e,i}^2$ is fixed, the corresponding $\hat{\theta}^i_{\text{CLS}}$ (which depends on the chosen $\sigma_{e,i}^2$) is given by (3.27) and the corresponding cost $\mathcal{J}$ can be computed. Among the considered values of $\sigma_{e,i}^2$, the one minimizing $\mathcal{J}$ is taken.

3.4 Bias-correction with noisy scheduling signal measurements

So far we have assumed that noise-free measurements of the scheduling variable $p_{\text{o}}(k)$ are available. However, in many real applications, this might not be a realistic assumption, as the scheduling signal is often related to a measured signal and thus inherently corrupted by measurement noise (e.g., velocity and lateral acceleration in vehicle lateral dynamics modelling [22], gate-source voltage of a transistor in the description of an electronic filter [49], air speed and flight altitude in aircraft control [4]). This noise induces a bias in the final parameter estimate $\hat{\theta}_{\text{CLS}}$ (3.27). Starting from the results presented in Section 3.3 and in [78] (where open-loop LPV identification from noisy scheduling variable measurements is addressed), in this section we show how to compute an asymptotically bias-free estimate of the LPV model parameters from closed-loop data with noisy measurements of the scheduling signal.

In particular, we consider the closed-loop data-generating system $\mathcal{S}_{\text{o}}$ in Figure 5, and we assume that the noise-free scheduling signal $p_{\text{o}}(k)$ is corrupted by an additive zero-mean white Gaussian noise with variance

σ_η^2, independent of the output noise $e(k)$, i.e.,

$$p(k) = p_{\mathrm{o}}(k) + \eta(k), \quad \mathbb{E}\left\{\eta(k)e(t)\right\} = 0, \ \ \forall k, t.$$

Following the same ideas described in Section 3.3, we quantify the bias in the LS estimate stemming from the output noise $e(k)$ and from the scheduling signal noise $\eta(k)$.

3.4.1 Bias-corrected least squares

By defining the "observed" regressor vector as

$$\phi_p(k) = \chi(k) \otimes \mathbf{p}(k),$$

with $\chi(k)$ defined in (3.11) and

$$\mathbf{p}(k) = \begin{bmatrix} 1 & p(k) & p^2(k) \cdots p^{n_g}(k) \end{bmatrix}^\top, \tag{3.33}$$

the standard least-squares estimate is given by

$$\hat{\theta}_{\mathrm{LS}}^p = \underbrace{\left(\frac{1}{N} \sum_{k=1}^{N} \phi_p(k) \phi_p^\top(k) \right)}_{\Gamma_N^p}^{-1} \frac{1}{N} \sum_{k=1}^{N} \phi_p(k) y(k). \tag{3.34}$$

By similar algebraic manipulations used in (3.15), the asymptotic bias in the LS estimate (3.34) is expressed as

$$\begin{aligned} \lim_{N\to\infty} \hat{\theta}_{\mathrm{LS}}^p - \theta_{\mathrm{o}} = & \lim_{N\to\infty} \underbrace{(\Gamma_N^p)^{-1} \frac{1}{N} \sum_{k=1}^{N} \phi_p(k) \Delta\phi(k) \theta_{\mathrm{o}}}_{B_\Delta(\theta_{\mathrm{o}}, \phi_p(k), \Delta\phi(k))} \\ & + \lim_{N\to\infty} \underbrace{(\Gamma_N^p)^{-1} \frac{1}{N} \sum_{k=1}^{N} \phi_p(k) \Delta\phi_p(k) \theta_{\mathrm{o}}}_{B_p(\theta_{\mathrm{o}}, \phi_p(k), \Delta\phi_p(k))}, \end{aligned} \tag{3.35}$$

with $\Delta\phi(k)$ as defined in (3.14) and

$$\Delta\phi_p(k) = \left[\chi(k) \otimes (\mathbf{p}_{\mathrm{o}}(k) - \mathbf{p}(k))\right]^\top.$$

Following the same rationale used to define $\tilde{\theta}_{\text{CLS}}$ in (3.16), let us introduce the bias-corrected estimate

$$\tilde{\theta}^p_{\text{CLS}} = \hat{\theta}^p_{\text{LS}} - B_\Delta(\tilde{\theta}^p_{\text{CLS}}, \phi_p(k), \Delta\phi(k)) - B_p(\tilde{\theta}^p_{\text{CLS}}, \phi_p(k), \Delta\phi_p(k)). \quad (3.36)$$

Remark 3 *In the case of noise-free scheduling signal observations, i.e.,* $\mathbf{p}_{\text{o}}(k) = \mathbf{p}(k)$, $\phi_p(k) = \phi(k)$ *and* $B_p(\tilde{\theta}^p_{\text{CLS}}, \phi_p(k), \Delta\phi_p(k)) = 0$. *Thus,* (3.36) *coincides with* (3.16). ■

By algebraic manipulations, the estimate $\tilde{\theta}^p_{\text{CLS}}$ in (3.36) can be rewritten explicitly as:

$$\tilde{\theta}^p_{\text{CLS}} = \underbrace{\left(\frac{\sum_{k=1}^N \phi_p(k)\phi_{\text{o}}^\top(k)}{N}\right)}_{R(\mathbf{p}_0)}^{-1} \left(\frac{1}{N}\sum_{k=1}^N \phi_p(k)y(k)\right), \quad (3.37)$$

or equivalently as in (3.38)[1].

Then, a bias-corrected estimate $\hat{\theta}^p_{\text{CLS}}$ can be obtained from (3.38d) as follows:

- replace the matrix $\chi(k)\chi^\top(k) \otimes \left[\mathbf{p}(k)\mathbf{p}_{\text{o}}^\top(k)\right]$ by following matrix : $\chi(k)\chi^\top(k) \otimes \Psi^p_k$ which depends only on the available dataset $\mathcal{D}^p_N = \{u(k), y(k), p(k), r(k)\}_{k=1}^N$, satisfying condition:

$$\mathbf{C2}: \lim_{N\to\infty} \frac{1}{N}\sum_{k=1}^N \mathbf{p}(k)\mathbf{p}_{\text{o}}^\top(k) = \lim_{N\to\infty} \frac{1}{N}\sum_{k=1}^N \Psi^p_k \ \ w.p.\ 1. \quad (3.39)$$

- replace the matrix $[\chi(k)\Delta\chi(k)] \otimes \left[\mathbf{p}(k)\mathbf{p}_{\text{o}}^\top(k)\right]$ by a matrix Ω_k depending only on the available dataset $\mathcal{D}^p_N$ and satisfying condition:

$$\mathbf{C3}: \lim_{N\to\infty} \frac{1}{N}\sum_{k=1}^N [\chi(k)\Delta\chi(k)] \otimes \left[\mathbf{p}(k)\mathbf{p}_{\text{o}}^\top(k)\right] = \lim_{N\to\infty} \frac{1}{N}\sum_{k=1}^N \Omega_k \ \ w.p.\ 1. \quad (3.40)$$

The procedure to construct the matrices Ψ^p_k and Ω_k satisfying conditions **C2** and **C3** is outlined in the following section.

[1] Eq. (3.38b) follows from (3.38a) and the Kronecker product property (3.22).

$$\tilde{\theta}_{\mathrm{CLS}} = \left(\frac{\sum_{k=1}^{N} \left[\chi(k) \otimes \mathbf{p}(k)\right] \left[\chi_{\mathrm{o}}(k) \otimes \mathbf{p}_{\mathrm{o}}(k)\right]^{\top}}{N} \right)^{-1} \left(\frac{1}{N} \sum_{k=1}^{N} \phi_p^{\top}(k) y(k) \right) \tag{3.38a}$$

$$= \left(\frac{\sum_{k=1}^{N} \left[\chi(k)\chi_{\mathrm{o}}^{\top}(k)\right] \otimes \left[\mathbf{p}(k)\mathbf{p}_{\mathrm{o}}^{\top}(k)\right]}{N} \right)^{-1} \left(\frac{1}{N} \sum_{k=1}^{N} \phi_p^{\top}(k) y(k) \right) \tag{3.38b}$$

$$= \left(\frac{\sum_{k=1}^{N} \left[\chi(k)(\chi_{\mathrm{o}}(k) - \chi(k))^{\top}\right] \otimes \left[\mathbf{p}(k)\mathbf{p}_{\mathrm{o}}^{\top}(k)\right] + \left[\chi(k)\chi^{\top}(k)\right] \otimes \left[\mathbf{p}(k)\mathbf{p}_{\mathrm{o}}^{\top}(k)\right]}{N} \right)^{-1} \times \left(\frac{1}{N} \sum_{k=1}^{N} \phi_p^{\top}(k) y(k) \right) \tag{3.38c}$$

$$= \left(\frac{\sum_{k=1}^{N} \left[\chi(k)\Delta\chi(k)\right] \otimes \left[\mathbf{p}(k)\mathbf{p}_{\mathrm{o}}^{\top}(k)\right] + \left[\chi(k)\chi^{\top}(k)\right] \otimes \left[\mathbf{p}(k)\mathbf{p}_{\mathrm{o}}^{\top}(k)\right]}{N} \right)^{-1} \times \left(\frac{1}{N} \sum_{k=1}^{N} \phi_p^{\top}(k) y(k) \right). \tag{3.38d}$$

3.4.2 Construction of the bias-eliminating matrices

Construction of Ψ_k^p

Inspired by [78], the bias-correction matrix Ψ_k^p satisfying **C2** in (3.39) is constructed as follows:

1. compute the analytic expression of $\mathbb{E}\{\mathbf{p}(k)\mathbf{p}_{\mathrm{o}}^{\top}(k)\}$. Note that, since $\mathbf{p}_{\mathrm{o}}(k)$ and $\mathbf{p}(k)$ are polynomials in $p_{\mathrm{o}}(k)$ and $p(k)$ (see (3.4) and (3.33)), the entries of $\mathbb{E}\{\mathbf{p}(k)\mathbf{p}_{\mathrm{o}}^{\top}(k)\}$ are polynomials in $p_{\mathrm{o}}(k)$;

2. express the $n-$th order monomial $p_{\mathrm{o}}^n(k)$ in terms of the expected value of the noise-corrupted observation $p^n(k)$ and noise variance

σ_η^2 as the "probabilists" Hermite polynomial: [2]

$$p_{\mathrm{o}}^n(k) = \mathbb{E}\left\{ (n!) \sum_{m=0}^{\lfloor n/2 \rfloor} \frac{(-1)^m \sigma_\eta^{2m}}{m!(n-2m)!} \frac{p^{n-2m}(k)}{2^m} \right\}; \tag{3.41}$$

3. compute the matrix Ψ_k^p by replacing each of the monomials $p_{\mathrm{o}}(k)$, $p_{\mathrm{o}}^2(k)$, $p_{\mathrm{o}}^3(k), \ldots$ appearing in the analytic expression of the matrix $\mathbb{E}\{\mathbf{p}(k)\mathbf{p}_{\mathrm{o}}^\top(k)\}$, with the term inside the expectation operator in (3.41).

By construction, the matrix Ψ_k^p satisfies

$$\mathbb{E}\left\{\Psi_k^p\right\} = \mathbb{E}\left\{\mathbf{p}(k)\mathbf{p}_{\mathrm{o}}^\top(k)\right\}. \tag{3.42}$$

Based on (3.42) and Ninness' strong law of large numbers [68], Ψ_k^p satisfies Condition **C2**. An example of construction of matrix Ψ_k^p is reported in Appendix 3.7.3.

Construction of Ω_k

The matrix Ω_k satisfying condition **C3** can be constructed by properly combing the ideas used to construct the bias-eliminating matrices Ψ_k^p (see Section 3.4.2) and Υ_k^y and Υ_k^u (introduced in (3.24)). Specifically, matrix Ω_k is constructed in such a way that the following equality holds:

$$\mathbb{E}_{e,\eta}\left\{\Omega_k\right\} = \mathbb{E}_{e,\eta}\left\{\left[\chi(k)\Delta\chi(k)\right] \otimes \left[\mathbf{p}(k)\mathbf{p}_{\mathrm{o}}^\top(k)\right]\right\}. \tag{3.43}$$

Since, $\chi(k)\Delta\chi(k)$ does not depend on the noise $\eta(k)$ and $\mathbf{p}(k)\mathbf{p}_{\mathrm{o}}^\top(k)$ does not depend on the output noise e, and since the random variables $e(k)$ and $\eta(k)$ are independent, (3.43) is equivalent to

$$\mathbb{E}_{e,\eta}\left\{\Omega_k\right\} = \mathbb{E}_e\left\{\left[\chi(k)\Delta\chi(k)\right]\right\} \otimes \mathbb{E}_\eta\left\{\left[\mathbf{p}(k)\mathbf{p}_{\mathrm{o}}^\top(k)\right]\right\}. \tag{3.44}$$

Note that $\chi(k)\Delta\chi(k)$ is equal to Υ_k as defined in (3.21a). Thus, the matrix $\mathbb{E}_e\left\{\left[\chi(k)\Delta\chi(k)\right]\right\}$ is equal to Λ_k (see (3.23)) and it can be constructed using the results in Property 5. However, Λ_k defined in (3.23)

[2] The expression of $p_{\mathrm{o}}^n(k)$ in terms of the expected value of the noise-corrupted observation $p^n(k)$ and noise variance σ_η^2 is not reported in [78] in terms of the Hermite polynomial (3.41), but in terms of recursive constructions which can be proved to have the compact expression in (3.41). See Appendix 2.5.2 of Chapter 2 for detailed explanation.

depends on the noise-free scheduling signal p_{o}, and thus its expression can be only derived analytically, but it cannot be constructed based on the available dataset $\mathcal{D}_N^p$. Nevertheless, as $\Lambda_k(\mathbf{p}_{\mathrm{o}})$ does not depend on the random variable η, condition (3.44) becomes

$$\begin{aligned}\mathbb{E}_{e,\eta}\left\{\Omega_k\right\} =&\Lambda_k(\mathbf{p}_{\mathrm{o}}) \otimes \mathbb{E}_{\eta}\left\{\left[\mathbf{p}(k)\mathbf{p}_{\mathrm{o}}^{\top}(k)\right]\right\}\\ =&\mathbb{E}_{\eta}\left\{\Lambda_k(\mathbf{p}_{\mathrm{o}}) \otimes \left[\mathbf{p}(k)\mathbf{p}_{\mathrm{o}}^{\top}(k)\right]\right\}. \end{aligned} \tag{3.45}$$

Thus, Ω_k can be constructed based on the same procedure outlined in Section 3.4.2 to construct Ψ_k^p, replacing the term $\mathbb{E}\left\{\left[\mathbf{p}(k)\mathbf{p}_{\mathrm{o}}^{\top}(k)\right]\right\}$ in Section 3.4.2 with the term $\mathbb{E}_{\eta}\left\{\Lambda_k(\mathbf{p}_{\mathrm{o}}) \otimes \left[\mathbf{p}(k)\mathbf{p}_{\mathrm{o}}^{\top}(k)\right]\right\}$ in (3.45).

As the matrix $\Lambda_k(\mathbf{p}_{\mathrm{o}})$ has a dynamic dependence on p_{o} (i.e., it is a function of $p_{\mathrm{o}}(k)$, $p_{\mathrm{o}}(k-1)$, ...), the analytic expression of $\Lambda_k(\mathbf{p}_{\mathrm{o}}) \otimes \left[\mathbf{p}(k)\mathbf{p}_{\mathrm{o}}^{\top}(k)\right]$ has product terms such as $p_{\mathrm{o}}^n(k)$, $p_{\mathrm{o}}^n(k-1)$. Nevertheless, as the noise terms $\eta(k)$ and $\eta(k-t)$ are uncorrelated, $\forall t \neq 0$, we have that $\mathbb{E}_{\eta}\left\{p^n(k)p^n(k-1)\right\} = \mathbb{E}_{\eta}\left\{p^n(k)\right\}\mathbb{E}_{\eta}\left\{p^n(k-1)\right\}$, and the Hermite polynomial expression defined in (3.41) can be used to construct Ω_k.

3.4.3 Bias-corrected estimate

Based on (3.38d) and the ideas introduced in the previous sections, the final bias-corrected estimate $\hat{\theta}_{\mathrm{CLS}}^p$ is given by

$$\hat{\theta}_{\mathrm{CLS}}^p = \left(\frac{1}{N}\sum_{k=1}^{N}\Omega_k(\hat{\theta}_{\mathrm{CLS}}^p) + \left[\chi(k)\chi^{\top}(k)\right] \otimes \Psi_k^p\right)^{-1}\left(\frac{1}{N}\sum_{k=1}^{N}\phi_p^{\top}(k)y(k)\right). \tag{3.46}$$

Note that, as in the case of noise-free scheduling signal, the matrix Ω_k depends on the true system parameter vector θ_{o} and the estimate $\hat{\theta}_{\mathrm{CLS}}^p$ should be computed based on an iterative approach similar to Algorithm 1.

Property 8 *Assume that the following limit*

$$\lim_{N\to\infty}\left(\frac{1}{N}\sum_{k=1}^{N}\Omega_k(\hat{\theta}_{\mathrm{CLS}}^p) + \left[\chi(k)\chi^{\top}(k)\right] \otimes \Psi_k^p\right)^{-1}. \tag{3.47}$$

exists. Then, asymptotically, the true system parameter vector θ_{o} is a solution of (3.46), namely, for $\theta = \theta_{\mathrm{o}}$,

$$\theta = \lim_{N\to\infty} \left(\frac{1}{N}\sum_{k=1}^{N} \Omega_k(\theta) + \left[\chi(k)\chi^\top(k)\right] \otimes \Psi_k^p \right)^{-1} \left(\frac{1}{N}\sum_{k=1}^{N} \phi_p^\top(k) y(k) \right), \quad (3.48)$$

where the limit in (3.48) *holds w.p.* 1. *Thus, if* θ_o *is the unique solution of* (3.48)*, then the estimated* $\hat{\theta}^p_{\rm CLS}$ *in* (3.46) *is consistent, i.e.,*

$$\lim_{N\to\infty} \hat{\theta}^p_{\rm CLS} = \theta_o. \quad (3.49)$$

Proof: Property 8 follows from conditions **C2** and **C3** and from the same rationale used in the proof of Property 7. ■

3.4.4 Estimation with unknown variances σ_e^2 and σ_η^2

In computing the bias correcting matrices Ω_k and Ψ_k^p, the noise variances σ_e^2 and σ_η^2 are assumed to be known. In the case of noise-free scheduling variable, the open-loop simulation error was used in Section 3.3.4 as a performance criterion to tune σ_e^2 via cross validation. However, in the noisy p scenario, a cross-validation procedure will fail, as a model with the "true" system parameters θ_o will not provide the "true" output due the fact that the scheduling variable $p(k)$ used to simulate the output of the model is not the "true" one. In order to overcome this problem, we propose next a novel procedure based on a bias-free tuning criterion.

Let us introduce the simulated regressor

$$\hat{\chi}(k) = \left[-\hat{y}(k-1)\cdots-\hat{y}(k-n_a), u(k-1), \cdots u(k-n_b)\right]^\top, \quad (3.50)$$

where $\hat{y}(k)$ is the bias-corrected simulated model output at time k given by

$$\hat{y}(k) = \left[\hat{\chi}(k) \otimes \mathbf{p}^{\rm C}(k)\right]^\top \hat{\theta}^p_{\rm CLS} \quad (3.51)$$

and $\mathbf{p}^{\rm C}(k)$ being the vector of bias-corrected monomials[3].

Given an estimate $\hat{\theta}^p_{\rm CLS}$, computed through (3.46) for fixed values of σ_e and σ_η, and a calibration dataset of length N_c not used to compute $\hat{\theta}^p_{\rm CLS}$, define the cost

[3]The vector of bias-corrected monomials $\mathbf{p}^{\rm C}(k)$ is such that it only depends on $\mathbf{p}(k)$ and σ_η and satisfies the condition $\mathbb{E}\left\{\mathbf{p}^{\rm C}(k)\right\} = \mathbf{p}_o(k)$. Thus, it can constructed using the Hermite polynomial (3.41). For instance, when $\mathbf{p}_o(k) = \left[1 \;\; p_o(k) \;\; p_o^2(k)\right]^\top$, then $\mathbf{p}^{\rm C}(k) = \left[1 \;\; p(k) \;\; p^2(k) - \sigma_\eta^2\right]^\top$.

$$\mathcal{J}_{\mathrm{BC}}\left(\hat{\theta}^{p}_{\mathrm{CLS}}(\sigma_e,\sigma_\eta)\right) = \Bigg\| \frac{1}{N_c}\sum_{k=1}^{N_c}\left[\chi(k)\hat{\chi}^{\top}(k)\otimes\Psi_k^p\right]\hat{\theta}^{p}_{\mathrm{CLS}}(\sigma_e,\sigma_\eta) - \frac{1}{N_c}\sum_{k=1}^{N_c}\left(\Omega_k(\hat{\theta}^{p}_{\mathrm{CLS}}) + \left[\chi(k)\chi^{\top}(k)\right]\otimes\Psi_k^p\right)\hat{\theta}^{p}_{\mathrm{CLS}}(\sigma_e,\sigma_\eta)\Bigg\|^2. \tag{3.52}$$

The cost $\mathcal{J}_{\mathrm{BC}}$ will be referred to as bias-corrected cost and, as discussed in the following, it should be used as a criterion to tune the unknown noise variances σ_e^2 and σ_η^2.

Property 9 *The bias-corrected cost* (3.52) *asymptotically achieves its minimum at* $\hat{\theta}^{p}_{\mathrm{CLS}} = \theta_{\mathrm{o}}$*, i.e.,*

$$\theta_{\mathrm{o}} = \arg\min_{\theta}\lim_{N_c\to\infty}\mathcal{J}_{\mathrm{BC}}(\theta) \;\; w.p.\ 1. \tag{3.53}$$

Proof: See Appendix 3.7.4.

Property 9 proves that, if $\mathcal{J}_{\mathrm{BC}}(\theta)$ has asymptotically a unique minimizer, then its minimum is achieved at the true system parameter vector θ_{o}. Thus, $\mathcal{J}_{\mathrm{BC}}(\theta)$ is an asympotically bias-free criterion which can be used to assess the quality of a given model parameter vector $\hat{\theta}^{p}_{\mathrm{CLS}}$. Therefore, the hyper-parameters σ_e and σ_η can be tuned through a grid search using $\mathcal{J}_{\mathrm{BC}}\left(\hat{\theta}^{p}_{\mathrm{CLS}}(\sigma_e,\sigma_\eta)\right)$ as a performance metric on a calibration dataset.

3.5 Case studies

In order to show the effectiveness of the proposed identification method, we consider two examples. In the first example, we focus on the effect of the measurement noise on the final parameter estimate, hence the model structure of the true LPV data-generating system is assumed to be exactly known. As a more realistic case study, the second example addresses the identification of a nonlinear two-tank system. All the simulations are carried out on an i5 2.40-GHz Intel core processor with 4 GB of RAM running MATLAB R2015b.

The performance of the identified models is assessed on a noiseless validation dataset not used for training through the *Best Fit Rate* (BFR) index, defined as

$$\mathrm{BFR} = \max\left\{1 - \sqrt{\frac{\sum_{k=1}^{N_{\mathrm{val}}} \left(y(k) - \hat{y}(k)\right)^2}{\sum_{k=1}^{N_{\mathrm{val}}} \left(y(k) - \bar{y}\right)^2}}, 0\right\}, \tag{3.54}$$

with N_{val} being the length of the validation set and $\hat{y}$ being the estimated model output and $\bar{y}$ the sample mean of the output signal.

3.5.1 Example 1

Data-generating system

The considered closed-loop data-generating system $\mathcal{S}_{\mathrm{o}}$ is taken from [1], and it consists of an (unknown) LPV plant $\mathcal{G}_{\mathrm{o}}$ described by (3.1), with

$$\mathcal{A}_{\mathrm{o}}(q^{-1}, p_k) = 1 + a_1^{\mathrm{o}}(p_{\mathrm{o}}(k))q^{-1} + a_2^{\mathrm{o}}(p_{\mathrm{o}}(k))q^{-2}, \tag{3.55a}$$
$$\mathcal{B}_{\mathrm{o}}(q^{-1}, p_k) = b_1^{\mathrm{o}}(p_{\mathrm{o}}(k))q^{-1} + b_2^{\mathrm{o}}(p_{\mathrm{o}}(k))q^{-2}, \tag{3.55b}$$

where,

$$a_1^{\mathrm{o}}(p_{\mathrm{o}}(k)) = 1.0 - 0.5p_{\mathrm{o}}(k) - 0.1p_{\mathrm{o}}^2(k), \tag{3.56a}$$
$$a_2^{\mathrm{o}}(p_{\mathrm{o}}(k)) = 0.5 - 0.7p_{\mathrm{o}}(k) - 0.1p_{\mathrm{o}}^2(k), \tag{3.56b}$$
$$b_1^{\mathrm{o}}(p_{\mathrm{o}}(k)) = 0.5 - 0.4p_{\mathrm{o}}(k) + 0.01p_{\mathrm{o}}^2(k), \tag{3.56c}$$
$$b_2^{\mathrm{o}}(p_{\mathrm{o}}(k)) = 0.2 - 0.3p_{\mathrm{o}}(k) - 0.02p_{\mathrm{o}}^2(k). \tag{3.56d}$$

The noise term $e(k)$ corrupting the output observations is a white Gaussian noise with standard deviation $\sigma_e = 0.05$. This corresponds to a *Signal-to-Noise Ratio* (SNR) of 12.5 dB, where the SNR on the output channel is defined as

$$\mathrm{SNR}_{\mathrm{y}} = 10\log\frac{\sum_{k=1}^{N}\left(x(k) - \bar{x}\right)^2}{\sum_{k=1}^{N} e^2(k)}, \tag{3.57}$$

with $\bar{x}$ denoting the mean of the noise free output.

The controller $\mathcal{K}_{\mathrm{o}}$ is LTI and known, and it is described by (3.2) with

$$\mathcal{C}_{\mathrm{o}}(q^{-1}, p_k) = 1 + c_1^{\mathrm{o}}(p_{\mathrm{o}}(k))q^{-1} + c_2^{\mathrm{o}}(p_{\mathrm{o}}(k))q^{-1},$$
$$\mathcal{D}_{\mathrm{o}}(q^{-1}, p_k) = d_1^{\mathrm{o}}(p_{\mathrm{o}}(k)) + d_2^{\mathrm{o}}(p_{\mathrm{o}}(k))q^{-1} + d_3^{\mathrm{o}}(p_{\mathrm{o}}(k))q^{-2},$$

with

$$c_1^{\mathrm{o}}(p_{\mathrm{o}}(k)) = -0.28, \;\; c_2^{\mathrm{o}}(p_{\mathrm{o}}(k)) = 0.5,$$
$$d_1^{\mathrm{o}}(p_{\mathrm{o}}(k)) = 0.35, \;\; d_2^{\mathrm{o}}(p_{\mathrm{o}}(k)) = -0.28, \;\; d_3^{\mathrm{o}}(p_{\mathrm{o}}(k)) = 0.1.$$

The scheduling signal trajectory is described by:

$$p_{\mathrm{o}}(k) = 1.1(0.5\sin(0.35\pi k) + 0.05).$$

With the given controller $\mathcal{K}_{\mathrm{o}}$, the closed-loop system is stable in the whole operating range of the scheduling variable $p_{\mathrm{o}}(k)$. The reference $r(k)$ is a white noise signal with uniform distribution in the interval $[-1 \;\; 1]$. A training data set $\mathcal{D}_N$ of length $N = 20,000$ is used to estimate the plant $\mathcal{G}_{\mathrm{o}}$ and, in order to assess the statistical properties of the proposed identification approach, a Monte-Carlo study with 100 runs is performed. At each Monte-Carlo run, a new data set of inputs $u(k)$, scheduling variables $p_{\mathrm{o}}(k)$, reference signal $r(k)$ and noise $e(k)$ is generated.

Model structure

As a model structure for the plant $\mathcal{G}_{\mathrm{o}}$, we consider the second-order LPV model

$$y(k) = -\sum_{i=1}^{2} a_i(p_{\mathrm{o}}(k))y(k-i) + \sum_{j=1}^{2} b_j(p_{\mathrm{o}}(k))u(k-j),$$

where the coefficient functions $a_i(p_{\mathrm{o}}(k))$ and $b_j(p_{\mathrm{o}}(k))$ are parametrized as second order-polynomials:

$$a_1(p_{\mathrm{o}}(k)) = a_{1,0} + a_{1,1}p_{\mathrm{o}}(k) + a_{1,2}p_{\mathrm{o}}^2(k),$$
$$a_2(p_{\mathrm{o}}(k)) = a_{2,0} + a_{2,1}p_{\mathrm{o}}(k) + a_{2,2}p_{\mathrm{o}}^2(k),$$
$$b_1(p_{\mathrm{o}}(k)) = b_{1,0} + b_{1,1}p_{\mathrm{o}}(k) + b_{1,2}p_{\mathrm{o}}^2(k),$$
$$b_2(p_{\mathrm{o}}(k)) = b_{2,0} + b_{2,1}p_{\mathrm{o}}(k) + b_{2,2}p_{\mathrm{o}}^2(k).$$

Identification from noise-free scheduling signal

First, we assume that the observations of the scheduling variable $p_{\mathrm{o}}(k)$ are not corrupted by a measurement noise. The following two cases are considered:

1. the variance σ_e^2 of the noise $e(k)$ on the output signal $y(k)$ is known;

2. σ_e^2 is unknown.

Furthermore, since Algorithm 1 depends on the initial guess $\hat{\theta}_{\text{CLS}}^{(0)}$ used to iteratively compute the bias-correcting matrix $\Psi_k(\hat{\theta}_{\text{CLS}}^{(\tau-1)})$ (see Step 2.2), we test its sensitivity w.r.t. different initial conditions $\hat{\theta}_{\text{CLS}}^{(0)}$.

Identification with known variance σ_e^2

The identification results obtained through standard least-squares and the closed-loop bias-correction approach presented in Algorithm 1 are compared in Table 6, which shows the averages and the standard deviations of the estimated model parameters over 100 Monte-Carlo runs. The average CPU time for computing the estimate for a given value of noise variance is 2.5 sec.

In order to further assess the performance of the developed identification scheme, we also compute the BFR on a noise-free validation data set of length $N_{\text{val}} = 10,000$, which is reported in Table 7. The obtained results shows that, unlike the least squares, the proposed approach provides a consistent estimate of the system parameters. This leads to a higher BFR (namely, better reconstruction of the output signal on the validation set) w.r.t. least squares.

In order to analyze the sensitivity of Algorithm 1 w.r.t. the initial condition $\hat{\theta}_{\text{CLS}}^{(0)}$, we initialize Algorithm 1 with 100 different random values of $\hat{\theta}_{\text{CLS}}^{(0)}$. The initial values of each component of $\hat{\theta}_{\text{CLS}}^{(0)}$ are chosen randomly from a uniform distribution in the interval $[0\ \ 1]$. The iterative algorithm is stopped when no change in the final estimate is observed or when a maximum number of iterations $\tau^{\max} = 50$ is reached. The same training data-set is used in all runs. We observe that the algorithm is insensitive to the initial conditions and it provides the same model estimate, resulting in an equal BFR for all the 100 different initial conditions $\hat{\theta}_{\text{CLS}}^{(0)}$ (see Figure 6).

The proposed method is also compared with a *prediction-error method* (PEM). In the prediction-error identification framework, the unknown plant parameters θ are obtained by minimizing the one-step ahead prediction error: $\epsilon_\theta(k) = y(k) - \hat{y}(k \mid k-1) = \hat{\phi}^\top(k)\theta$, resulting in the

Table 6: Example 1. Identification from noise-free scheduling signal measurements: means and standard deviations (over 100 Monte-Carlo runs) of the estimated parameters using Least Squares, the proposed closed-loop bias-correction method and prediction-error method (PEM).

True Value	Least squares	Bias-correction	PEM
1.00	0.7542 ± 0.0085	0.9992 ± 0.0138	0.9976 ± 0.0082
-0.50	-0.2117 ± 0.0194	-0.4785 ± 0.0425	-0.4999 ± 0.0188
-0.10	-0.9288 ± 0.0525	-0.1245 ± 0.1229	-0.0873 ± 0.0606
0.50	0.3449 ± 0.0057	0.5000 ± 0.0088	0.4986 ± 0.0057
-0.70	-0.7288 ± 0.0099	-0.6961 ± 0.0181	-0.7016 ± 0.0088
-0.10	-0.1685 ± 0.0295	-0.0994 ± 0.0609	-0.0898 ± 0.0403
0.50	0.5001 ± 0.0037	0.5008 ± 0.0041	0.4996 ± 0.0023
-0.40	-0.4007 ± 0.0070	-0.3997 ± 0.0081	-0.4007 ± 0.0027
0.01	-0.0266 ± 0.0194	0.0063 ± 0.0235	0.0109 ± 0.0108
0.20	0.0671 ± 0.0058	0.1995 ± 0.0082	0.1986 ± 0.0043
-0.30	-0.0788 ± 0.0136	-0.2887 ± 0.0267	-0.2999 ± 0.0118
-0.02	-0.4697 ± 0.0352	-0.0337 ± 0.0680	-0.0147 ± 0.0328

minimization of the following non-convex loss function:

$$\mathcal{W}(\mathcal{D}_N, \theta) = \frac{1}{N} \sum_{k=1}^{N} \epsilon_\theta^2(k)$$

where, the regressor $\hat{\phi}(k)$ (as defined in (3.4)) is given by

$$\begin{aligned}
\hat{\chi}(k) &= \left[-\hat{y}(k-1) \cdots - \hat{y}(k-n_a) \;\; u(k-1) \cdots u(k-n_b)\right]^\top, \\
\hat{\phi}(k) &= \hat{\chi}(k) \otimes \mathbf{p}_{\mathrm{o}}(k).
\end{aligned}$$

The average CPU time taken by the PEM to find the estimate is 2.5 sec. The estimated model parameters and the achieved BFR are reported in Table 6 and Table 7, respectively. Similar results are obtained by the bias-correction approach and PEM. However, unlike PEM, the proposed bias-correction approach leads to a consistent parameter estimate also in the case of noisy scheduling variable observations (as shown in the results reported in Section 3.5.1).

Table 7: Example 1. Identification from noise-free scheduling signal measurements: *Best Fit Rates* (BFRs) over (noise-free) validation data.

Method	BFR
Least-squares	0.8202
Bias-correction	0.9964
PEM	0.9984

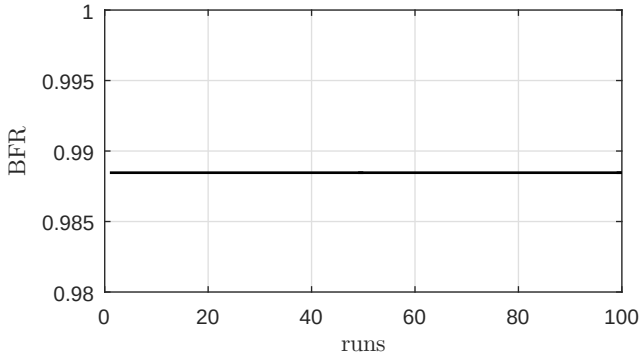

Figure 6: Example 1. Best Fit Rate for different initial conditions $\hat{\theta}_{\text{CLS}}^{(0)}$ of Algorithm 1.

Identification with unknown noise variance σ_e^2

We now consider the case where the variance σ_e^2 of the noise corrupting the output signal is not known a priori, but recovered through the cross-validation procedure described in Section 3.3.4. Figure 7 shows the cost function $\mathcal{J}(\sigma_e)$ (multiplied by N_c for a better visualization) defined in (3.32) against different values of the hyper-parameter σ_e. Note that the minimum of $\mathcal{J}$ is achieved exactly at the true value of the noise standard deviation (i.e., at $\sigma_e = 0.05$). Thus, since the true value of σ_e^2 is exactly recovered, the estimated model parameters coincide with the ones obtained in the case of known variance σ_e^2 (and already provided in Table 6).

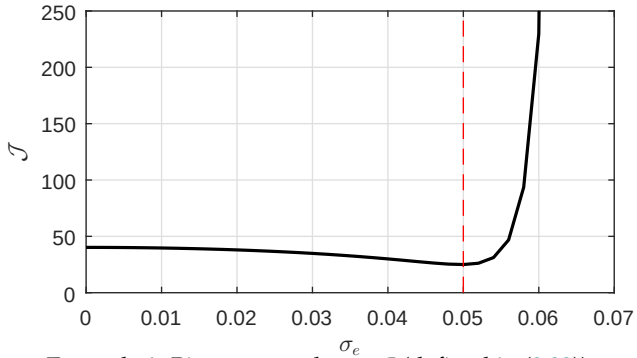

Figure 7: Example 1. Bias-corrected cost $\mathcal{J}$ (defined in (3.32)) vs noise standard deviation σ_e.

Identification from noisy scheduling signal

In this paragraph, the proposed closed-loop identification algorithm is tested for the case of noisy measurements of the scheduling signal. To this aim, the scheduling variable observations are corrupted by an additive zero-mean white Gaussian noise $\eta_o(k)$ with standard deviation $\sigma_\eta = 0.12$. This corresponds to a *Signal-To-Noise Ratio* SNR_p equal to 10 dB[4].

The unknown model parameters are computed through the following three approaches:

1. Least Squares;
2. Bias Correction 1: closed-loop bias-correction without handling the bias due to the noise on p. The model parameters are estimated using Algorithm 1, correcting only the bias due to the output noise e.
3. Bias Correction 2: closed-loop bias-correction correcting both the bias due to the noise on the scheduling signal observations and the bias due to the feedback structure.

First, we consider the case when the noise variances σ_e^2 and σ_η^2 are known. The estimated model parameters are provided in Table 8. The norm

[4]The *Signal-To-Noise Ratio* SNR_p on scheduling variable observations is defined similarly to (3.57).

Table 8: Example 1. Identification with noisy scheduling signal measurements: means and standard deviations (over 100 Monte-Carlo runs) of the estimated parameters using Least Squares, Bias Correction 1 and Bias Correction 2. For the sake of simplicity, the coefficients multiplying the quadratic terms in (3.56) are set to 0.

True Value	Least Squares	Bias Correction 1	Bias Correction 2
1.0	0.6908 ± 0.0070	1.0161 ± 0.0087	0.9965 ± 0.0087
-0.5	-0.3337 ± 0.0160	-0.4524 ± 0.0311	-0.4970 ± 0.0358
0.5	0.3297 ± 0.0044	0.5003 ± 0.0049	0.4948 ± 0.0051
-0.7	-0.6123 ± 0.0082	-0.6274 ± 0.0152	-0.6906 ± 0.0169
0.5	0.4970 ± 0.0021	0.5002 ± 0.0023	0.5155 ± 0.0023
-0.4	-0.3769 ± 0.0062	-0.3662 ± 0.0067	-0.4221 ± 0.0075
0.2	0.0357 ± 0.0043	0.2083 ± 0.0055	0.2058 ± 0.0057
-0.3	-0.1727 ± 0.0105	-0.2765 ± 0.0176	-0.3095 ± 0.0204

Table 9: Example 1. Identification with noisy scheduling signal observations: Best Fit Rates (BFRs) over validation data achieved by: Least-squares; Bias Correction 1 and Bias Correction 2.

Method	$\left\|\theta_o - \hat{\theta}\right\|_2$	BFR
Least Squares	0.4513	0.7784
Bias Correction 1	0.0977	0.9641
Bias Correction 2	0.0314	0.9710

$\left\|\theta_o - \hat{\theta}\right\|_2$ of the difference between the true system parameters θ_o and the estimate parameters $\hat{\theta}$ is reported in Table 9, along with the BFRs on validation data. The obtained results show that correcting the bias due to the noise on the scheduling signal observations further improves the final model parameter estimate.

Finally, we present the results of the proposed method when no information is available a priori about the variance of the noise corrupting the output and the scheduling signal measurements. As detailed in Section 3.4.4, the standard deviations of the noise signals is estimated by cross-validation using the bias corrected cost function $\mathcal{J}_{\text{BC}}$ in (3.52) as a

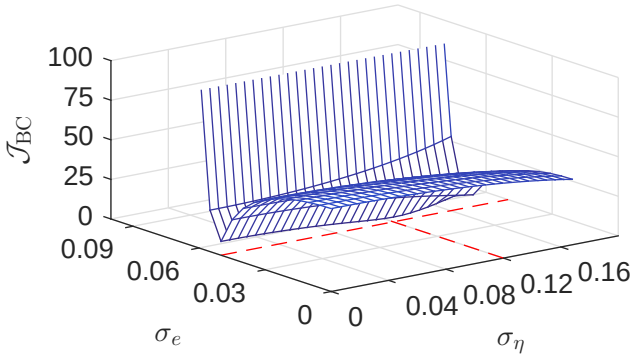

Figure 8: Bias corrected cost $\mathcal{J}_{\mathrm{BC}}$ vs σ_e and σ_η.

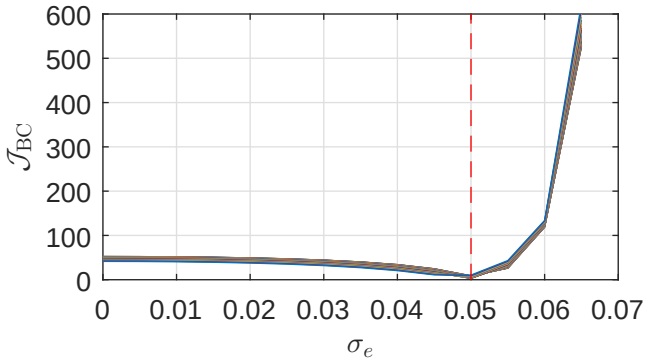

Figure 9: $\mathcal{J}_{\mathrm{BC}}$ vs σ_e for different values of σ_η.

performance criterion. Figure 8 shows the bias-corrected cost $\mathcal{J}_{\mathrm{BC}}$ plotted against range of values of σ_e and σ_η. For clarity, we have shown the 2-D plot of $\mathcal{J}_{\mathrm{BC}}$ versus σ_e for different values of σ_η in Figure 9. The cost $\mathcal{J}_{\mathrm{BC}}$ as a function of σ_η for fixed value of σ_e at which the minimum is achieved (i.e., at $\sigma_e = 0.05$) is plotted in Figure 10. It can be seen from the figures that the minimum is achieved at the true values of σ_η and σ_e (i.e., $\sigma_e = 0.05$ and $\sigma_\eta = 0.12$).

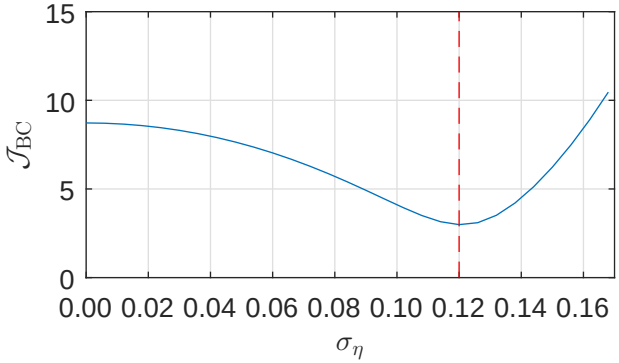

Figure 10: $\mathcal{J}_{\rm BC}$ vs σ_η for fixed value of $\sigma_e = 0.05$.

3.5.2 LPV identification of a nonlinear two-tank system

As a second case study, we consider the identification of the nonlinear two-tank system reported in [89]. The physical system consists of two tanks, placed one above the other. The upper tank receives the liquid inflow through a pump. The voltage applied to the pump is the input $u(t)$, which controls the inflow of the liquid in the upper tank. The lower tank gets the liquid inflow via a small hole at the bottom of the upper tank. The output $y(t)$ is the liquid level of the lower tank. The following nonlinear equations are used to simulate the behaviour of the system:

$$\dot{x_1}(t) = (1/A_1)(ku(t) - a_1\sqrt{2gx_1(t)}), \tag{3.59a}$$

$$\dot{x_2}(t) = (1/A_2)(a_1\sqrt{2gx_1(t)} - a_2\sqrt{2gx_2(t)}), \tag{3.59b}$$

$$y(t) = x_2(t), \tag{3.59c}$$

where $A_1 = 0.5\,\mathrm{m}^2$, $A_2 = 0.25\,\mathrm{m}^2$ are the cross-section areas of tank 1 and 2, respectively, $a_1 = 0.02\,\mathrm{m}^2$, $a_2 = 0.015\,\mathrm{m}^2$ are the cross-section areas of the holes in the two tanks, $g = 9.8\,\mathrm{m/s}^2$ is the acceleration due to gravity, $x_1(t)$ and $x_2(t)$ are the liquid levels in tank 1 and tank 2, respectively. The reader is referred to [89] for a more detailed description of the considered two-tank system.

The plant is controlled by a proportional controller $u = Kx_2(t)$, with $K = 1$, and the output $y(t)$ is measured with a sampling time of 0.3 sec. To gather data, the closed-loop system is excited with a discrete-time

Table 10: Example 2. Best fit rates over validation data achieved by Least-Squares and closed-loop bias-correction.

Method	BFR
Least Squares	0.3517
Bias-correction	0.7748

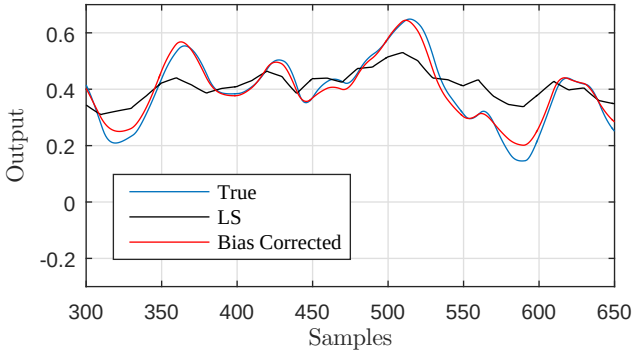

Figure 11: Example 2. Validation dataset: true output, simulated output of the LS model, simulated output of the bias-corrected model.

zero-mean white noise reference signal $r(k)$ uniformly distributed in the interval $[2\ 15]$ followed by a zero-order hold block. The measured output $y(k)$ is corrupted by a white Gaussian noise $\mathcal{N}(0, \sigma_e^2)$ with $\sigma_e = 0.01$, which corresponds to an SNR of 20 dB.

To estimate the plant, we consider the LPV model structure $\mathcal{M}_\theta$ described in (3.6) and (3.7), with $n_a = 2$, $n_b = 1$ and polynomial degree $n_g = 2$. The input $u(k-1)$ is used as a scheduling signal $p(k)$. Thus, the considered model is actually *quasi*-LPV. $N = 20,000$ and $N_{\text{val}} = 5,000$ samples are used for training and validation, respectively. The actual and simulated outputs of the models estimated through standard least-squares and the proposed bias-correction method are plotted in Figure 11. For the sake of visualization, only a subset of validation data is plotted. Furthermore, the BFRs of the estimated models are reported in Table 10. Note that, although the true system (3.59) does not belong to the model class $\mathcal{M}_\theta$, the proposed bias-correction approach outperforms

standard least squares in estimating the dynamics of the two-tank system.

3.6 Conclusions

This chapter has introduced a novel bias-correction approach for closed-loop identification of LPV systems. Starting from a least-square estimate, the proposed method exploits the knowledge of the controller to recursively compute an estimate of the asymptotic bias in the model parameters due to the feedback loop. This bias is then eliminated in order to obtain a consistent estimate of the open-loop plant. Based on a similar rationale, the bias caused by the noise corrupting the scheduling variable observations is also corrected, thus extending the applicability of the approach to realistic scenarios where not only the output signal, but also the scheduling signal observations are affected by a measurement noise. The computation of the bias strongly depends on the noise variance. In case this is not available or it cannot be estimated through dedicated experiments, a bias-corrected cost serves as a performance criterion for tuning the noise variance. The reported examples point out that the proposed method outperforms least-squares in terms of achieving a consistent estimate of the open-loop model parameters, provided that the true system belongs to the chosen model class. Although the latter assumption is barely achieved in practice, correcting the bias due to the measurement noise also leads to a significant improvement in the final model estimate when an under-parametrized model structure is considered, as shown in the second case study. Future activities will be devoted to the extension of the presented approach under more general controller structures, like linear model-predictive controllers, which are characterized by piecewise-affine state-feedback control laws. Furthermore, conditions to guarantee convergence of the iterative Algorithm 1 will be sought.

3.7 Appendix

3.7.1 Proof of Property 5

The a-priori known controller $\mathcal{K}_o$ and the closed-loop structure $\mathcal{S}_o$ in Figure 5 is exploited to construct the matrix Ψ_k, taking into account that the input signals depend on the measurement noise $e(k)$ due to the presence of feedback. Property 5 can be proved as follows. According to (3.20),

$$\Psi_k = \mathbb{E}\{\phi(k)\Delta\phi(k)\} = \mathbb{E}\{\Upsilon_k\} \otimes \mathbf{P}_o(k),$$

with

$$\mathbb{E}\{\Upsilon_k\} = \mathbb{E}\left\{\chi(k)(\chi_o(k) - \chi(k))^\top\right\}. \tag{3.60}$$

By definition of $\chi(k)$ and $\chi_o(k)$, we have:

$$\chi_o(k) - \chi(k) = [e(k-1) \;\; \cdots \;\; e(k-n_a) \;\; \mathbf{0}_{1\times n_b}]^\top.$$

Then,

$$\mathbb{E}\{\Upsilon_k\} = \mathbb{E}\left\{\chi(k)(\chi_o(k) - \chi(k))^\top\right\} =$$

$$\mathbb{E}\left\{\left[\begin{array}{ccc|c} -y(k-1)e(k-1) & \cdots & -y(k-1)e(k-n_a) & \\ \vdots & -y(k-i)e(k-i) & \vdots & \mathbf{0}_{n_a\times n_b} \\ -y(k-n_a)e(k-1) & \cdots & -y(k-n_a)e(k-n_a) & \\ \hline u(k-1)e(k-1) & \cdots & u(k-1)e(k-n_a) & \\ \vdots & \ddots & \vdots & \mathbf{0}_{n_b\times n_b} \\ u(k-n_b)e(k-1) & \cdots & u(k-n_b)e(k-n_a) & \end{array}\right]\right\} \tag{3.61}$$

The following observations are made to compute $\mathbb{E}\{\Upsilon_k\}$ explicitly. The value of input and output at time k does not depend on the future values of the measurement noise e, i.e.,

$$\begin{aligned} &\mathbb{E}\{y(k-i)e(k-j)\} = 0, \\ &\mathbb{E}\{u(k-i)e(k-j)\} = 0 \quad \forall i > j. \end{aligned}$$

This implies that the matrices Υ_k^y and Υ_k^u are upper triangular as in (3.24a) and (3.24b).

3.7.2 Proof of Property 6

The recurrence relations in Property 6 can be proved with the following observations:

1. Due to the strict causality of the plant $\mathcal{G}_{\mathrm{o}}$ and since e is white, the noise-free output $x(k)$ does not depend on the current and future values of the measurement noise, i.e.,

$$\mathbb{E}\left\{x(k-i)e(k-j)\right\} = 0 \quad \forall i \geq j.$$

Thus, for $i = j$,

$$\begin{aligned}
&-\mathbb{E}\left\{y(k-i)e(k-i)\right\} \\
=&-\mathbb{E}\left\{(x(k-i)+e(k-i))e(k-i)\right\} \\
=&-\mathbb{E}\left\{x(k-i)e(k-i)\right\} - \mathbb{E}\left\{e(k-i)e(k-i)\right\} \\
=&0-\sigma_e^2 = -\sigma_e^2 = f_1(k-i). \qquad (3.62)
\end{aligned}$$

2. The terms $f_m(k)$ and $g_m(k)$ can be computed in a recursive manner as described in the following.

Let us first consider the term $f_m(k)$. By definition:

$$f_m(k) = -\mathbb{E}\left\{y(k+m-1)e(k)\right\}.$$

By writing $y(k)$ as $x(k)+e(k)$, we have

$$\begin{aligned}
f_m(k) &= -\mathbb{E}\left\{x(k+m-1)e(k)+e(k+m-1)e(k)\right\} \\
&= -\mathbb{E}\left\{x(k+m-1)e(k)\right\} \\
&= -\mathbb{E}\left\{-\sum_{i=1}^{n_a} a_i^{\mathrm{o}}(p_{\mathrm{o}}(k+m-1))x(k+m-1-i)e(k)\right. \\
&\qquad \left. +\sum_{i=1}^{n_b} b_j^{\mathrm{o}}(p_{\mathrm{o}}(k+m-1))u(k-m-1-j)e(k)\right\} \\
&= -\sum_{i=1}^{n_a} a_i^{\mathrm{o}}(p_{\mathrm{o}}(k+m-1))(-\mathbb{E}\left\{x(k+m-1-i)e(k)\right\}) \\
&\quad -\sum_{i=1}^{n_b} b_j^{\mathrm{o}}(p_{\mathrm{o}}(k+m-1))(\mathbb{E}\left\{u(k-m-1-j)e(k)\right\})
\end{aligned}$$

with $f_1(k) = -\sigma_e^2$ (see (3.62)). Note that,

$$\begin{aligned}
-\mathbb{E}\left\{x(k+m-1-i)e(k)\right\} &= f_{m-i}(k), \\
\mathbb{E}\left\{u(k-m-1-j)e(k)\right\} &= g_{m-j}(k).
\end{aligned}$$

Thus,

$$f_m(k) = -\sum_{i=1}^{n_a} a_i^{\mathrm{o}}(p_{\mathrm{o}}(k+m-1))f_{m-i}(k) - \sum_{i=1}^{n_b} b_j^{\mathrm{o}}(p_{\mathrm{o}}(k+m-1))g_{m-j}(k).$$

Since $f_m = 0$ and $g_m = 0$, for $m \leq 0$, we have

$$f_m(k) = -\sum_{i=1}^{m-2} a_i^{\mathrm{o}}(p_{\mathrm{o}}(k+m-1))f_{m-i}(k) - \sum_{j=1}^{\min(n_b,m-1)} b_j^{\mathrm{o}}(p_{\mathrm{o}}(k+m-1))g_{m-j}(k).$$

Consider now the term $g_m(k)$. Since the reference signal $r(k)$ is uncorrelated with the measurement noise $e(k)$, i.e., $\mathbb{E}(r(k)e(k')) = 0, \ \forall k, k'$, the terms $g_m(k)$ (for $m = 1, \cdots, n_a$) can be computed as

$$\begin{aligned}
g_m(k) &= \mathbb{E}\{u(k+m-1)e(k)\} \\
&= \mathbb{E}\left\{-\sum_{i=1}^{n_c} c_i(p_{\mathrm{o}}(k+m-1))u(k+m-1-i)e(k) \right. \\
&\quad \left. +\sum_{j=0}^{n_d-1} d_{j+1}(p_{\mathrm{o}}(k+m-1))(-x(k+m-1-j)e(k))\right\} \\
&= -\sum_{i=1}^{n_c} c_i(p_{\mathrm{o}}(k+m-1))(\mathbb{E}\{u(k+m-1-i)e(k)\}) \\
&\quad +\sum_{j=0}^{n_d-1} d_{j+1}(p_{\mathrm{o}}(k+m-1))(-\mathbb{E}\{x(k+m-1-j)e(k)\}) \\
&= -\sum_{i=1}^{n_c} c_i(p_{\mathrm{o}}(k+m-1))g_{m-i}(k) \\
&\quad + \sum_{j=0}^{n_d-1} d_{j+1}(p_{\mathrm{o}}(k+m-1))f_{m-j}(k).
\end{aligned}$$

Since $f_m = 0$ and $g_m = 0$, for $m \leq 0$, it follows that

$$g_m(k) = -\sum_{i=1}^{\min(n_c, m-1)} c_i(p_o(k+m-1))g_{m-i}(k) + \sum_{j=1}^{\min(n_d, m)} d_j(p_o(k+m-1))f_{m-j+1}(k).$$

Thus, the recurrence relations in Property 6 are proved.

3.7.3 Construction of Ψ_k^p

For clarity of exposition, the procedure outlined in Section 3.4.2 to construct Ψ_k^p is shown via the following example.

Consider the following vector of monomials

$$\mathbf{p}_o(k) = [1 \;\; p_o(k) \;\; p_o^2(k)]^\top, \quad \mathbf{p}(k) = [1 \;\; p(k) \;\; p^2(k)]^\top.$$

Then,

$$\mathbf{p}(k)\mathbf{p}_o^\top(k) = \begin{bmatrix} 1 & p_o(k) & p_o^2(k) \\ p(k) & p(k)p_o(k) & p(k)p_o^2(k) \\ p^2(k) & p^2(k)p_o(k) & p^2(k)p_o^2(k) \end{bmatrix}$$

By writing $p(k)$ as $p_o(k) + \eta(k)$ and taking the expectation of $\mathbf{p}(k)\mathbf{p}_o^\top(k)$ w.r.t. the random variable $\eta(k)$, we get

$$\mathbb{E}[\mathbf{p}(k)\mathbf{p}_o^\top(k)] = \begin{bmatrix} 1 & p_o(k) & p_o^2(k) \\ p_o(k) & p_o^2(k) & p_o^3(k) \\ p_o^2(k) + \sigma_\eta^2 & p_o^3(k) + \sigma_\eta^2 p_o(k) & p_o^4(k) + \sigma_\eta^2 p_o^2(k) \end{bmatrix}.$$

Then, matrix Ψ_k^p is constructed by replacing each monomial $p_o^n(k)$ with the Hermite polynomial in (3.41), that is

$$\Psi_k^p = \begin{bmatrix} 1 & p(k) & p^2(k) - \sigma_\eta^2 \\ p(k) & p^2(k) - \sigma_\eta^2 & p^3(k) - 3\sigma_\eta^2 p(k) \\ p^2(k) & p^3(k) - 2\sigma_\eta^2 p(k) & p^4(k) - 5\sigma_\eta^2 p^2(k) + 2\sigma_\eta^4 \end{bmatrix}.$$

3.7.4 Proof of Property 9

Because of conditions C2 and C3, we have:

$$\lim_{N_c\to\infty}\frac{1}{N_c}\sum_{k=1}^{N_c}\Omega_k(\theta_{\rm o})+\left[\chi(k)\chi^\top(k)\right]\otimes\Psi_k^p=$$
$$\lim_{N_c\to\infty}\frac{1}{N_c}\sum_{k=1}^{N_c}\phi_p(k)\phi_{\rm o}^\top(k)\ \ w.p.\ 1. \tag{3.63}$$

Let us now focus on the term $\chi(k)\hat{\chi}^\top(k)\otimes\Psi_k^p$ appearing in the bias-corrected cost (3.52).

For the sake of simplicity, let us assume that the initial condition $\hat{\chi}(1)$ used to simulate the bias-corrected output $\hat{y}(1)$ is known, i.e., $\hat{\chi}(1)=\chi_{\rm o}(1)$. This means

$$\mathbb{E}_\eta\left\{\hat{y}(i)\right\}=y_{\rm o}(i)\ \ \forall i=-n_{\rm a}+1,\dots,0. \tag{3.64}$$

Let us now prove, by induction, that for $\hat{\theta}^p_{\rm CLS}=\theta_{\rm o}$,

$$\mathbb{E}_\eta\left\{\hat{y}(k)\right\}=y_{\rm o}(k)\ \ \forall\, k>0. \tag{3.65}$$

Suppose that the above equation holds for $k-n_{\rm a},\dots,k-1$, i.e.,

$$\mathbb{E}_\eta\left\{\hat{y}(k-i)\right\}=y_{\rm o}(k-i)\ \ \forall i=1,\dots,n_{\rm a}. \tag{3.66}$$

Note that, for $\hat{\theta}^p_{\rm CLS}=\theta_{\rm o}$,

$$\begin{aligned}
\mathbb{E}_\eta\left\{\hat{y}(k)\right\}&=\theta_{\rm o}^\top\left(\mathbb{E}_\eta\left\{\hat{\chi}(k)\otimes\mathbf{p}^{\rm C}(k)\right\}\right) && \text{(3.67a)}\\
&=\theta_{\rm o}^\top\left(\mathbb{E}_\eta\left\{\hat{\chi}(k)\right\}\otimes\mathbb{E}_\eta\left\{\mathbf{p}^{\rm C}(k)\right\}\right) && \text{(3.67b)}\\
&=\theta_{\rm o}^\top\left(\mathbb{E}_\eta\left\{\hat{\chi}(k)\right\}\otimes\mathbf{p}_{\rm o}(k)\right) && \text{(3.67c)}\\
&=\theta_{\rm o}^\top\left[\chi_{\rm o}(k)\otimes\mathbf{p}_{\rm o}(k)\right] && \text{(3.67d)}\\
&=y_{\rm o}(k), && \text{(3.67e)}
\end{aligned}$$

where (3.67b) follows from white noise assumption on η and (3.67d) follows from (3.66) and construction of the bias-corrected monomials $\mathbf{p}^{\rm C}(k)$. Thus, from (3.64), (3.66) and (3.67), (3.65) follows by induction.

Eq. (3.65) also implies that

$$\mathbb{E}_\eta\left\{\hat{\chi}(k)\right\}=\chi_{\rm o}(k)\ \ \forall\, k>0. \tag{3.68}$$

Thus,

$$\begin{aligned}
\mathbb{E}_\eta\left\{\chi(k)\hat{\chi}^\top(k)\otimes\Psi_k^p\right\} &= \mathbb{E}_\eta\left\{\chi(k)\hat{\chi}^\top(k)\right\}\otimes\mathbb{E}_\eta\left\{\Psi_k^p\right\} && \text{(3.69a)}\\
&= \chi(k)\chi_{\mathrm{o}}^\top(k)\otimes\mathbb{E}_\eta\left\{\mathbf{p}(k)\mathbf{p}_{\mathrm{o}}^\top(k)\right\} && \text{(3.69b)}\\
&= \mathbb{E}_\eta\left\{\chi(k)\chi_{\mathrm{o}}^\top(k)\otimes\mathbf{p}(k)\mathbf{p}_{\mathrm{o}}^\top(k)\right\} && \text{(3.69c)}\\
&= \mathbb{E}_\eta\left\{\left(\chi(k)\otimes\mathbf{p}(k)\right)\left(\chi_{\mathrm{o}}(k)\otimes\mathbf{p}_{\mathrm{o}}(k)\right)^\top\right\} && \text{(3.69d)}\\
&= \mathbb{E}_\eta\left\{\phi_p(k)\phi_{\mathrm{o}}^\top(k)\right\}, && \text{(3.69e)}
\end{aligned}$$

where (3.69b) follows from (3.68) and (3.42). Then, because of (3.69) and Ninness' strong law of large numbers [68], at $\hat{\theta}_{\mathrm{CLS}}^p = \theta_{\mathrm{o}}$, we have

$$\begin{aligned}
&\lim_{N_c\to\infty}\frac{1}{N_c}\sum_{k=1}^{N_c}\chi(k)\hat{\chi}^\top(k)\otimes\Psi_k^p\\
&=\lim_{N_c\to\infty}\frac{1}{N_c}\sum_{k=1}^{N_c}\phi_p(k)\phi_{\mathrm{o}}^\top(k)\ \ w.p.\ 1.
\end{aligned}\tag{3.70}$$

Thus, from (3.63) and (3.70), we obtain:

$$\lim_{N_c\to\infty}\mathcal{J}_{\mathrm{BC}}\left(\theta_{\mathrm{o}}\right)=0\ \ w.p.\ 1.\tag{3.71}$$

Property 9 follows from (3.71) and because of non-negativity of the cost $\mathcal{J}_{\mathrm{BC}}$. This completes the proof.

Note that, even if the initial conditions are not exactly known (i.e., assumption (3.64) is not satisfied), Property 9 still holds since the error due to the mismatch between the true initial conditions and the ones used to simulate the output $\hat{y}$ vanishes asymptotically, under the assumption that the system is asymptotically stable.

Chapter 4

PWA regression for identification of LPV models

The methodologies presented so far in the previous chapters for the identification of *Linear Parameter-Varying* (LPV) models fall under the conventional framework of parametric and non-parametric approaches. In parametric methods, the nonlinear LPV model coefficients are parameterized in terms of a-priory specified basis functions whereas in non-paramtric approaches these coefficients are characterized implicitly via kernel maps.

Alternative to the conventional methods, in this chapter, the LPV identification problem is transformed into *PieceWise Affine* (PWA) regression problem. Specifically, the nonlinear LPV model coefficients are approximated with PWA maps. A novel regularized moving-horizon algorithm is devised for PWA regression based on mixed-integer programming.

Furthermore, the framework of the proposed algorithm is applied to the real-world problem of *energy disaggregation* where the goal is to estimate the power consumption profiles of individual appliances by using only the aggregated power measurements. The behavior of the power consumption profiles of individual appliance is described using PWA AutoRegressive (PWA-AR) models. The proposed PWA regression algorithm is used to estimate the device models using a short du-

ration disaggregated training dataset. Once the device models are estimated, energy disaggregation is performed by solving a binary quadratic programming problem. The dynamic PWA-AR modeling of the appliances significantly improves the estimation of power consumption profiles compared to the approaches relying on static models.

4.1 Introduction

4.1.1 Motivation

PieceWise Affine (PWA) models are heterogeneous systems which exhibit both continuous and discrete dynamics. PWA models are simple and flexible model structures and thanks to their universal approximation properties, any nonlinear function can be modeled with arbitrary accuracy by a PWA map [16]. These aspects make PWA functions an attractive model class for nonlinear regression and black-box identification of nonlinear and LPV dynamical systems. Furthermore, due to the equivalence between PWA models and several classes of hybrid models [40], available tools for modeling, analysis and control of hybrid systems can be also applied to PWA systems [11; 13].

PWA regression is an NP-hard problem [50], where both the regressor space partition and the submodel parameters have to be estimated from a set of training data. Several algorithms/heuristics for PWA regression and for the identification of hybrid systems, have been proposed in the last two decades (see the survey papers [35; 73]). Among these algorithms, we mention the set-membership approaches [12; 71], the sparse-optimization based approaches [7; 70] and the mixed-integer programming method [83]. In the latter approach, the estimation of hinging-hyperplane ARX models and piecewise affine Wiener models is formulated as a mixed-integer linear or quadratic programming problem, and then solved through a branch-and-bound algorithm. As the number of integer variables is proportional to the number of modes and the number of training samples, the approach in [83] is limited to medium-scale problems. The contributions [8; 12; 18; 19; 32; 45; 66] fall in the class of cluster-based two-stage methods. At the first stage, the training samples are clustered by assigning each datapoint to a submodel according to a certain criterion and, at the same time, the parameters of the affine submodels are estimated. In the second stage, the polyhedral partition of the regressor space is computed by linear separation techniques. Un-

like the mixed-integer programming approach [83], which can be solved for the global optimum, sub-optimal solutions are obtained by the aforementioned two-stage methods.

4.1.2 Contribution

PWA regresion algorithm

We introduce a regularized moving-horizon PWA regression algorithm, which can be seen as a mix between the mixed-integer programming approach [83] and the cluster-based algorithm [19]. We show how the presented method can be applied for the identification of discrete-time multi-input multi-output LPV and PWA-AutoRegressive with eXogenous input (PWA-ARX) models. Specifically, in the case of LPV models, unknown LPV model coefficients are approximated by PWA maps, defined over polyhedral partition of the scheduling variable space. In this way, LPV model identification is recast into PWA regression problem and PWA approximation of unknown LPV model coefficients along with the polyhedral partition of the scheduling variable space is estimated via proposed algorithm.

The key idea of the proposed two-stage moving horizon PWA regression algorithm can be summarized as follows: At the first stage, a *Mixed-Integer Quadratic-Programming* (MIQP) problem is formulated to compute both the optimal sequence of active modes within the considered horizon as well as the parameters of the affine sub-models. Moreover, a regularization term is included in the cost of the formulated MIQP problem, to exploit the information from the past training samples outside the considered time window. Thus, the length N_p of the horizon acts as a knob to combine the advantages of the two-stage algorithm [19] (namely, computational efficiency and iterative processing of the training samples) and the advantages of the mixed-integer programming approach [83] (namely, non-decoupled optimization over the active modes and the sub-model parameters). According to a moving-horizon strategy, only the active mode at current sampling time is extracted from the computed optimal sequence of active modes, and the next training sample is processed by shifting forward the estimation horizon. At the second stage, the regressor space is partitioned using computationally efficient multi-class linear separation methods proposed in [19].

Application to energy disaggregation problem

The proposed two-stage moving-horizon algorithm for PWA regression is employed for the energy disaggregation problem. The energy disaggregation problem is to estimate the power consumption profile of each device based on aggregated power measurements. We propose a novel approach that relies on *PieceWise Affine AutoRegressive* (PWA-AR) dynamical models to describe the behavior of the individual appliances' power consumption patterns. Using a set of disaggregated data collected over a short intrusive period, the PWA-AR models are first estimated off-line using the proposed moving-horizon PWA regression algorithm. Once the appliance models are estimated, energy disaggregation is formulated as a binary quadratic programming problem. Specifically, based on the measurements of the aggregated power, the active operating mode of each appliance (and thus its power consumption) is determined at each time instance in an iterative way. The developed disaggregation algorithm is tested on a real world benchmark dataset. For comparison, an approach relying on static models (i.e., models defined based on the average power ratings of the devices) is also tested using the same dataset. The obtained results show that the proposed approach with dynamic PWA-AR models describing power consumption of individual appliances significantly improves the performance compared to the approach relying on static models.

4.1.3 Outline

The chapter is organized as follows: In Section 4.2, the identification of PWA-ARX and LPV-ARX models is formulated as a PWA regression problem. The proposed moving-horizon PWA regression algorithm is described in Section 4.3. The identification results for PWA-ARX and LPV-ARX models are reported in Section 4.4.

In the appendix of the chapter, the framework of the proposed algorithm is applied for energy disaggregation problem described in Section 4.6.1. In Section 4.6.4, we formally state the problem of energy disaggregation and present the proposed approach in Section 4.6.5. Specifically, we briefly explain the moving-horizon PWA regression algorithm for estimating individual appliance models in Section 4.6.6, the binary quadratic programming formulation to perform energy disaggregation in Section 4.6.7. Tests on benchmark data set are discussed in Section 4.6.8.

4.2 Problem formulation

Let us consider a data-generating system in the form

$$y(k) = f_{\mathrm{o}}(x(k)) + e_{\mathrm{o}}(k), \tag{4.1}$$

where $k \in \mathbb{N}$ is the time index, $y(k) \in \mathbb{R}^{n_y}$ is the measured output at time k, $e_{\mathrm{o}}(k) \in \mathbb{R}^{n_y}$ is an additive random noise, $x(k) \in \mathbb{R}^{n_x}$ is the regressor vector which is assumed to take values in a set $\mathcal{X} \subset \mathbb{R}^{n_x}$, and $f_{\mathrm{o}} : \mathcal{X} \to \mathbb{R}^{n_y}$ is an unknown and possible discontinuous function.

In this chapter, we address a PWA regression problem, which amounts to computing a PWA function $f : \mathcal{X} \to \mathbb{R}^{n_y}$ approximating the regression function f_{o} based on a set of N observations of the regressor/output pairs $\{x(k), y(k)\}_{k=1}^{N}$. The PWA vector-valued function f is described as:

$$f(x) = \begin{cases} \Theta_1 \left[\begin{smallmatrix} 1 \\ x \end{smallmatrix}\right] & \text{if } x \in \mathcal{X}_1, \\ \vdots & \\ \Theta_s \left[\begin{smallmatrix} 1 \\ x \end{smallmatrix}\right] & \text{if } x \in \mathcal{X}_s, \end{cases} \tag{4.2}$$

where $s \in \mathbb{N}$ denotes the number of modes, $\Theta_i \in \mathbb{R}^{n_y \times (n_x+1)}$ are parameter matrices, and $\mathcal{X}_i$, with $i = 1, \ldots, s$, are polyhedra ($\mathcal{H}_i x \leq \mathcal{D}_i$) that form a complete polyhedral partition[1] of the regressor space $\mathcal{X}$.

Estimation of the PWA function f in (4.2) thus requires: (i) selecting the number of modes s; (ii) estimating the parameter matrices Θ_i; and (iii) finding the polyhedra $\mathcal{X}_i$ (i.e., the matrices $\mathcal{H}_i$ and $\mathcal{D}_i$) defining the partition of the regressor space $\mathcal{X}$. Tradeoff between data fitting and model complexity should be taken into account while choosing the number of modes s. If the number of modes s is small, then the PWA map f may not be flexible enough to capture the shape of the underlying nonlinear data-generating function f_{o} (4.1). On the contrary, considering the high number of modes results in a more accurate description of the PWA map f with more degrees of freedom. However, this may cause overfitting and poor generalization to unseen data (not used in the training phase) as the final estimate is sensitive to the noise corrupting the observations, besides increasing the complexity of the estimation procedure and of the resulting PWA model. Here, we assume that s is fixed by the user, and chosen via cross-calibration. This is done by evaluating the performance of the estimated model for different values of s, on a fresh data set which is different from the training data set.

[1] A collection $\{\mathcal{X}_i\}_{i=1}^{s}$ is a complete partition of the regressor domain $\mathcal{X}$ if $\bigcup_{i=1}^{s} \mathcal{X}_i = \mathcal{X}$ and $\mathring{\mathcal{X}}_i \cap \mathring{\mathcal{X}}_j = \emptyset, \forall i \neq j$, with $\mathring{\mathcal{X}}_i$ denoting the interior of $\mathcal{X}_i$.

4.2.1 Identification of PWA-ARX models

The model (4.2) represents discrete-time PWA-ARX dynamical systems if the regressor vector $x(k)$ is defined by the collection of past values inputs and outputs, i.e.,

$$x(k) = \begin{bmatrix} y^\top(k-1) \dots y^\top(k-n_a) & u^\top(k-1) \dots u^\top(k-n_b) \end{bmatrix}^\top$$

where, $y(k) \in \mathbb{R}^{n_y}$ and $u(k) \in \mathbb{R}^{n_u}$ are measured output and input signals at time k respectively.

4.2.2 Identification of LPV-ARX models

The model (4.2) can represent the LPV-ARX model structure if the scheduling variable dependent LPV model coefficients are approximated by PWA functions. The scheduling-variable space is partitioned into polyedral regions, where each region is assigned to a PWA function describing the local affine dependence of the underlying LPV model coefficients on the scheduling variable.

In this section, we formulate the LPV system identification problem in terms of PWA regression. Let us consider the LPV-ARX model structure

$$y(k) = a_0(p(k)) + \sum_{j=1}^{n_a} a_j(p(k))y(k-j) + \sum_{j=1}^{n_b} a_{j+n_a}(p(k))u(k-j) \quad (4.3)$$

where $p(k) \in \mathcal{P} \subseteq \mathbb{R}^{n_p}$ is the measurement of the scheduling signal at time k, $u(k) \in \mathbb{R}^{n_u}$ and $y(k) \in \mathbb{R}^{n_y}$ are the model input and output vectors, respectively.

Each coefficient function $a_j(p)$ is parametrized by the PWA map:

$$a_j(p) = \begin{cases} \Theta^0_{1,j} + \sum_{h=1}^{n_p} \Theta^h_{1,j} p_h & \text{if } p \in \mathcal{P}_1, \\ \vdots & \\ \Theta^0_{s,j} + \sum_{h=1}^{n_p} \Theta^h_{s,j} p_h & \text{if } p \in \mathcal{P}_s, \end{cases} \quad (4.4)$$

where p_h denotes the h-th element of scheduling vector p, number of modes are denoted by $s \in \mathbb{N}$, (i.e., the number of affine local functions defining a_j), $\Theta^h_{i,j}$ ($h = 1, \dots n_p$) are parameter matrices of proper dimension, and $\mathcal{P}_i$, with $i = 1, \dots, s$, are polyhedra that form a complete polyhedral partition of the scheduling-variable domain $\mathcal{P}$.

In order to introduce flexibility in the LPV model (4.3)–(4.4), the polyhedral partition $\{\mathcal{P}_i\}_{i=1}^{s}$ is not fixed a priori and will be directly reconstructed from data. This represents one of the main advantages with respect to widely used parametric LPV identification approaches which need parameterization of $a_j(p(k))$ as a linear combination of some known basis functions (e.g. polynomial functions).

Let us now stack the parameter matrices as:

$$\Theta_i = \left[\Theta_{i,0}^0 \ \cdots \ \Theta_{i,0}^{n_p} \ \cdots \ \Theta_{i,n_a+n_b}^0 \ \cdots \ \Theta_{i,n_a+n_b}^{n_p}\right], \tag{4.5}$$

for all modes $i = 1, \ldots, s$, and let us introduce the regressor vector $x(k)$

$$x(k) = \begin{bmatrix} 1 \\ y(k-1) \\ \vdots \\ y(k-n_a) \\ u(k-1) \\ \vdots \\ u(k-n_b) \end{bmatrix} \otimes \begin{bmatrix} 1 \\ p(k) \end{bmatrix}, \tag{4.6}$$

where $\otimes$ denotes the Kronecker product. Substituting (4.4) into (4.3), and based on the above notation, the LPV model (4.3) can be written in the compact PWA-LPV form

$$y(k) = \begin{cases} \Theta_1 x(k) & \text{if } p(k) \in \mathcal{P}_1, \\ \vdots & \\ \Theta_s x(k) & \text{if } p(k) \in \mathcal{P}_s. \end{cases} \tag{4.7}$$

Thus, the LPV-ARX model identification has been recasted into the PWA regression problem. The estimation of the PWA-LPV model (4.7) consists of, (i) estimation of the model parameter matrices Θ_i and (ii) computation of the polyhedra $\mathcal{P}_i$ defining the partition of the scheduling variable space $\mathcal{P}$.

Remark: In the rest of the chapter, for the sake of uniformity, with a slight abuse of notation, we denote the polyhedral partition of the scheduling space by $\mathcal{X}$ (i.e. by the notation used to define the polyhedral partition of the regressor space, introduced to define the model (4.2)), instead of $\mathcal{P}$. It should be clear from the context that for LPV models, polyhedra $\mathcal{P}_i$ are computed defining the partition of the scheduling variable space $\mathcal{P}$, where each scheduling vector $p(k)$ at time k is assigned to a specific partition $\mathcal{P}_i$.

4.3 PWA regression algorithm

The developed algorithm for PWA regression consists of the following two stages:

S1. Recursive *estimation* of the model parameters Θ_i and simultaneous *clustering* of the regressors $\{x(k)\}_{k=1}^{N}$.

S2. *Computation of a polyhedral partition* of the regressor space $\mathcal{X}$ using computationally efficient multi-category linear separation methods already available in the literature. This can be computed either offline or online (recursively) and is executed after S1.

4.3.1 Recursive clustering and parameter estimation

Stage **S1** is carried out through a regularized moving-horizon identification algorithm. The training regressor/output pairs $\{x(k), y(k)\}$ are processed iteratively. At each time sample k, a moving-horizon window of length N_p containing regressor/output pairs from time $k - N_p + 1$ to time k is considered. The model parameters Θ_i and the active mode $\sigma(k) \in \{1, \dots, s\}$ at time k (characterized by binary variables $\delta(k)$) are estimated simultaneously by solving the mixed-integer programming problem:

$$\min_{\substack{\{\Theta_i\}_{i=1}^{s} \\ \{\delta_i(k-t)\}_{i=1,t=0}^{s,N_p-1}}} \sum_{i=1}^{s}\sum_{t=0}^{N_p-1} \left\| \left(y(k-t) - \Theta_i \begin{bmatrix} 1 \\ x(k-t) \end{bmatrix}\right) \delta_i(k-t) \right\|_2^2 \tag{4.8a}$$

$$+ \sum_{t=1}^{k-N_p} \left\| y(t) - \Theta_{\sigma(t)} \begin{bmatrix} 1 \\ x(t) \end{bmatrix} \right\|_2^2 \tag{4.8b}$$

$$+ \sum_{t=0}^{N_p-1}\sum_{i=1}^{s} \|(x(k-t) - c_i)\delta_i(k-t)\|_2^2 \tag{4.8c}$$

$$\text{s.t. } \delta_i(k-t) \in \{0,1\}, \quad \sum_{i=1}^{s} \delta_i(k-t) = 1, \quad t = 0, \dots, N_p - 1. \tag{4.8d}$$

The active mode $\sigma(k) \in \{1, \dots, s\}$ represents the cluster $\mathcal{C}_{i^*}$ (whose centroid is denoted by c_{i^*}) that the regressor $x(k)$ is assigned to, and it is

extracted from the optimizer of problem (4.8), i.e.,

$$\sigma(k) = i^*, \quad \text{with } i^* : \delta_{i^*}(k) = 1. \tag{4.9}$$

According to a moving-horizon estimation strategy, only the active mode $\sigma(k)$ at time k is kept, and the N_p-length time window is shifted forward to process the next pair $\{x(k+1), y(k+1)\}$. Problem (4.8) aims at searching for the optimal sequence of active modes within the considered time window and the model parameters Θ_i which best match the available observations up to time k. Note that the term (4.8a) aims at finding both the model parameters and the sequence of active modes $\{\sigma(t)\}_{t=k-N_p+1}^{k}$ which best match the observations within the N_p-step time horizon. The term (4.8b) acts as a regularization term on the parameters Θ_i and it takes into account the time history of the observations outside the considered time window. More specifically, in (4.8b), the sequence of active modes is not optimized from time 1 to time $k-N_p$, but it is fixed to the estimates $\{\sigma(t)\}_{t=1}^{k-N_p}$ obtained from the previous iterations of the moving-horizon estimation algorithm.

The sequence of active modes is optimized only within the considered time horizon in (4.8a) and in (4.8c). The term (4.8c) penalizes the distance of the regressor $x(k)$ vector (scheduling vector $p(k)$, in case of LPV models) from the centroid c_i of the cluster $\mathcal{C}_i$. Overall, the active mode $\sigma(k)$ is selected based on the trade-off between the fitting error term (4.8a) and the penalty on the distance of the current regressor sample $x(k)$ from the centroids of each clusters.

Increasing N_p increases the information used to cluster the regressor $x(k)$ and to estimate the model parameters Θ_i. On the one hand, increasing N_p increases the number of binary decision variables δ_i in (4.8). Thus, the length N_p of the horizon provides a trade off between complexity of the optimization problem (4.8), and accuracy in estimating the model parameters Θ_i and in clustering the regressor $x(k)$.

Recursive update of the objective function

Note that, at a first glance, the regularization cost (4.8b) requires to use, and thus to store, the whole time-history of observations up to time $k - N_p$ (i.e., the sequence of regressor/output pairs $\{x(k), y(k)\}_{k=1}^{k-N_p}$). Nevertheless, once a new observation is available at time k, the term (4.8b) can be recursively updated, as described in the following.

Let us rewrite the regularization term (4.8b) as

$$\sum_{t=1}^{k-N_p} \mathrm{tr}\Big(\big(y(t) - \Theta_{\sigma(t)} \left[\begin{smallmatrix} 1 \\ x(t) \end{smallmatrix}\right]\big) \big(y(t) - \Theta_{\sigma(t)} \left[\begin{smallmatrix} 1 \\ x(t) \end{smallmatrix}\right]\big)^\top \Big) =$$
$$\mathrm{tr}\left(\sum_{t=1}^{k-N_p} \Theta_{\sigma(t)} \left[\begin{smallmatrix} 1 \\ x(t) \end{smallmatrix}\right] \left[\begin{smallmatrix} 1 \\ x(t) \end{smallmatrix}\right]^\top \Theta_{\sigma(t)}^\top \right) -$$
$$2\mathrm{tr}\left(\sum_{t=1}^{k-N_p} \Theta_{\sigma(t)} \left[\begin{smallmatrix} 1 \\ x(t) \end{smallmatrix}\right] y^\top(t) \right) + \mathrm{tr}\left(\sum_{t=1}^{k-N_p} y(t) y^\top(t) \right), \tag{4.10}$$

with $\mathrm{tr}(\cdot)$ denoting the matrix trace. Let us now define the matrices

$$H_i(k-N_p) = \sum_{t=1}^{k-N_p} \left[\begin{smallmatrix} 1 \\ x(t) \end{smallmatrix}\right] \left[\begin{smallmatrix} 1 \\ x(t) \end{smallmatrix}\right]^\top h_i(t), \tag{4.11a}$$

$$F_i(k-N_p) = \sum_{t=1}^{k-N_p} \left[\begin{smallmatrix} 1 \\ x(t) \end{smallmatrix}\right] y^\top(t) h_i(t), \tag{4.11b}$$

with $h_i(t) = 1$, if $\sigma(t) = i$ or $h_i(t) = 0$, otherwise.

Substituting (4.11) into the cost (4.10), we can represent (4.8b) as

$$\sum_{t=1}^{k-N_p} \big\| y(t) - \Theta_{\sigma(t)} \left[\begin{smallmatrix} 1 \\ x(t) \end{smallmatrix}\right] \big\|_2^2 = \mathrm{tr}\left(\sum_{i=1}^{s} \Theta_i H_i(k-N_p) \Theta_i^\top \right)$$
$$- 2\mathrm{tr}\left(\sum_{i=1}^{s} \Theta_i F_i(k-N_p) \right) + \mathrm{tr}\left(\sum_{t=1}^{k-N_p} y(t) y^\top(t) \right). \tag{4.12}$$

Note that the matrices $H_i(k-N_p)$ and $F_i(k-N_p)$ can be recursively computed as

$$H_i(k-N_p) = H_i(k-N_p-1) + \left[\begin{smallmatrix} 1 \\ x(k-N_p) \end{smallmatrix}\right] \left[\begin{smallmatrix} 1 \\ x(k-N_p) \end{smallmatrix}\right]^\top h_i(k-N_p), \tag{4.13a}$$

$$F_i(k-N_p) = F_i(k-N_p-1) + \left[\begin{smallmatrix} 1 \\ x(k-N_p) \end{smallmatrix}\right] y^\top(k-N_p)\, h_i(k-N_p). \tag{4.13b}$$

Thus, processing the observation $\{x(k), y(k)\}$ just requires to update the matrices $H_i(k-N_p-1)$ and $F_i(k-N_p-1)$ through (4.13), with no need to store the time history of observations $\{x(k), y(k)\}_{t=1}^{k-N_p-1}$.

MIQP formulation

To reformulate (4.8) as an MIQP problem, let us define the vector $z_i(k) \in \mathbb{R}^{n_y}$ with $\delta_i(k) \in \{0,1\}$ as

$$z_i(k) = \left(y(k) - \Theta_i \begin{bmatrix} 1 \\ x(k) \end{bmatrix}\right) \delta_i(k). \tag{4.14a}$$

Note that:

$$z_i(k) = \begin{cases} y(k) - \Theta_i \begin{bmatrix} 1 \\ x(k) \end{bmatrix} & \text{if } \delta_i(k) = 1 \\ 0 & \text{if } \delta_i(k) = 0 \end{cases} \tag{4.14b}$$

Let M and m be an arbitrary large (resp. small) upper (resp. lower) bound of the elements of the vector $y(k) - \Theta_i \begin{bmatrix} 1 \\ x(k) \end{bmatrix}$,

$$m \leq y(k) - \Theta_i \begin{bmatrix} 1 \\ x(k) \end{bmatrix} \leq M. \tag{4.14c}$$

Based on conditions (4.12) and (4.14), problem (4.8) can be equivalently written as the MIQP problem

$$\min_{\substack{\{\Theta_i\}_{i=1}^{s} \\ \{\delta_i(k-t)\}_{i=1,t=0}^{s,N_p-1} \\ \{z_i(k-t)\}_{i=1,t=0}^{s,N_p-1}}} \sum_{t=0}^{N_p-1} \sum_{i=1}^{s} z_i^2(k-t) + \tag{4.15a}$$

$$\operatorname{tr}\left(\sum_{i=1}^{s} \Theta_i H_i(k-N_p) \Theta_i^\top\right) - 2\operatorname{tr}\left(\sum_{i=1}^{s} \Theta_i F_i(k-N_p)\right), \tag{4.15b}$$

$$+ \sum_{t=0}^{N_p-1} \sum_{i=1}^{s} \|(x(k-t) - c_i)\delta_i(k-t)\|_2^2, \tag{4.15c}$$

$$\text{s.t. } z_i(k-t) \leq M\delta_i(k-t), \tag{4.15d}$$

$$z_i(k-t) \geq m\delta_i(k-t), \tag{4.15e}$$

$$z_i(k-t) \leq y(k-t) - \Theta_i \begin{bmatrix} 1 \\ x(k-t) \end{bmatrix} - m(1 - \delta_i(k-t)), \tag{4.15f}$$

$$z_i(k-t) \geq y(k-t) - \Theta_i \begin{bmatrix} 1 \\ x(k-t) \end{bmatrix} - M(1 - \delta_i(k-t)), \tag{4.15g}$$

$$\sum_{i=1}^{s} \delta_i(k-t) = 1, \; t = 0, \ldots, N_p - 1, \tag{4.15h}$$

$$\delta_i(k-t) \in \{0,1\}, \; i = 1, \ldots, s, \tag{4.15i}$$

To solve the MIQP problem (4.15), the accelerated dual gradient projection (GPAD) [75] coupled with *Branch and Bound* (B&B) method (GPAD-B&B) [65] can be used.

Summary and iterative refinement

The steps described so far for recursive clustering of the regressors $\{x(k)\}_{k=1}^{N}$ and for model parameters Θ_i estimation are summarized in Algorithm 2. At the beginning of Algorithm 2, a mini-batch identification problem is solved to estimate the sequence of active modes $\sigma(t)$ from time 1 up to time N_p and to assign the regressor $\{x(t)\}_{t=1}^{N_p}$ to the cluster $\{\mathcal{C}_{\sigma(t)}\}_{t=1}^{N_p}$ (stages 2–6). Then, the observations $\{x(k), y(k)\}$ are processed iteratively. Besides updating the model parameters Θ_i at each time k (stage 7.3), the active mode $\sigma(k)$ is estimated (stages 7.4–7.6), the regressor $x(k)$ is consequently assigned to cluster $\mathcal{C}_{\sigma(k)}$ (stage 7.7) and the cluster's centroid $c_{\sigma(k)}$ is updated (step 7.8–7.9).

It is worth pointing out that, at the first iterations of Algorithm 2 (i.e., for $\bar{k} \ll N$), the observations $\{x(t), y(t)\}_{t=1}^{\bar{k}}$ may be wrongly classified, as the learning phase is based on a "small" set of observations. An error in the classification of the pairs $\{x(t), y(t)\}_{t=1}^{\bar{k}}$ (i.e., an incorrect estimate of the mode sequence $\{\sigma(t)\}_{t=1}^{\bar{k}}$) also influences the estimate of the active modes and of the model parameters Θ_i at the next time samples $k > \bar{k}$, as the regularization cost (4.8b) depends on the estimated sequence $\{\sigma(t)\}_{t=1}^{\bar{k}}$. When working in a batch mode, the effect of the initial misclassification may be reduced by running Algorithm 2 multiple times, including the sequences of active modes estimated at the previous runs, in the regularization term (4.8b). More specifically, at the n_q-th run of Algorithm 2, the following cost is considered instead of (4.8a)–(4.8b):

$$\sum_{i=1}^{s} \sum_{t=0}^{N_p-1} \left\| \left(y(k-t) - \Theta_i \begin{bmatrix} 1 \\ x(k-t) \end{bmatrix} \right) \delta_i(k-t) \right\|_2^2 + \tag{4.16a}$$

$$\sum_{t=1}^{k-N_p} \left\| y(t) - \Theta_{\sigma(t,n_q)} \begin{bmatrix} 1 \\ x(t) \end{bmatrix} \right\|_2^2 + \tag{4.16b}$$

$$\sum_{q=1}^{n_q-1} \lambda^{n_q-q-1} \sum_{t=1}^{N-N_p} \left\| y(t) - \Theta_{\sigma(t,q)} \begin{bmatrix} 1 \\ x(t) \end{bmatrix} \right\|_2^2, \tag{4.16c}$$

$$\sum_{t=0}^{N_p-1} \sum_{i=1}^{s} \| (x(k-t) - c_i) \delta_i(k-t) \|_2^2, \tag{4.16d}$$

with $\sigma(t, q)$ ($q = 1, \ldots, n_q$) being the estimate of the active mode at time t obtained at the q-th run of Algorithm 2. Note that (4.16c) is a regular-

ization term based on the past runs of Algorithm 2, while (4.16b) plays the same role of (4.8b), as it regularizes the parameters Θ_i based on the estimate $\{\sigma(t)\}_{t=1}^{k-N_p}$ obtained at the current run of Algorithm 2. The influence of the past runs of Algorithm 2 is controlled by including a forgetting factor $\lambda \in \mathbb{R} : 0 < \lambda \leq 1$ in (4.16c), which exponentially down-weights the estimates $\{\sigma(t,q)\}_{t=1}^{N}$ obtained at the previous runs.

Similar to the original regularization term (4.8b), the cost (4.16b)–(4.16c) can be also recursively updated as a new sample $\{x(k), y(k)\}$ is processed, without the need to store the whole time history of estimates $\{\sigma(k,q)\}_{k=1,q=1}^{N,n_q-1}$ obtained at the previous runs of Algorithm 2. As a matter of fact, the terms (4.16b)–(4.16c) can be written as

$$\left\{ \mathrm{tr}\left(\sum_{i=1}^{s} \Theta_i H_i(k-N_p, n_q) \Theta_i^\top \right) - \tag{4.17a}$$

$$2\mathrm{tr}\left(\sum_{i=1}^{s} \Theta_i F_i(k-N_p, n_q) \right) + \tag{4.17b}$$

$$\mathrm{tr}\left(\sum_{t=1}^{k-N_p} y(t) y^\top(t) \right) \right\} + \tag{4.17c}$$

$$\sum_{q=1}^{n_q-1} \mathrm{tr}\left(\sum_{i=1}^{s} \Theta_i \lambda^{n_q-q-1} H_i(N-N_p, q) \Theta_i^\top \right) - \tag{4.17d}$$

$$2 \sum_{q=1}^{n_q-1} \mathrm{tr}\left(\sum_{i=1}^{s} \Theta_i \lambda^{n_q-q-1} F_i(N-N_p, q) \right) + \tag{4.17e}$$

$$\sum_{q=1}^{n_q-1} \mathrm{tr}\left(\lambda^{n_q-q-1} \sum_{t=1}^{N-N_p} y(t) y^\top(t) \right), \tag{4.17f}$$

where $H_i(N-N_p, q)$ and $F_i(N-N_p, q)$ $(q = 1, \ldots, n_q)$ are defined similarly to (4.11) and computed based on the estimates $\sigma(t,q)$ computed at the q-th run of Algorithm 2. Specifically,

$$H_i(N-N_p, q) = \sum_{t=1}^{N-N_p} \begin{bmatrix} 1 \\ x(t) \end{bmatrix} \begin{bmatrix} 1 \\ x(t) \end{bmatrix}^\top h_i(t,q),$$

$$F_i(N-N_p, q) = \sum_{t=1}^{N-N_p} \begin{bmatrix} 1 \\ x(t) \end{bmatrix} y^\top(t) h_i(t,q),$$

with $h_i(t,q) = 0$, if $\sigma(t,q) = i$ or $h_i(t,q) = 1$, otherwise.

When the pair $\{x(k), y(k)\}$ is processed, the matrices $H_i(k - N_p, n_q)$ and $F_i(k-N_p, n_q)$ in (4.17a)–(4.17b) can be recursively updated through (4.13), while only the matrices $H_i(N - N_p, q)$ and $F_i(N - N_p, q)$ (with $q = 1, \ldots, n_q - 1$) are needed to construct the terms (4.17d)–(4.17e).

4.3.2 Construction of the state partition

The partition $\{\mathcal{X}_i\}_{i=1}^s$ of the regressor space $\mathcal{X}$ (or the scheduling variable space) can be found along with the estimation of the model parameters $\{\Theta_i\}_{i=1}^s$ and the sequence of active modes $\{\sigma(k)\}_{k=1}^N$. This is done by separating the computed clusters $\{\mathcal{C}_i\}_{i=1}^s$ using linear multicategory discrimination.

In the following subsection, we briefly describe the algorithms recently presented in [19], which are suited both for offline and online (i.e., recursive) computation of the state partition.

Linear multicategory discrimination: problem formulation

According to the formulation introduced in [14], the linear multicategory discrimination problem is tackled by searching for a convex piecewise affine separator function $\phi : \mathbb{R}^{n_x} \to \mathbb{R}$ discriminating between the clusters $\mathcal{C}_1, \ldots, \mathcal{C}_s$. The separator function ϕ is defined as

$$\phi(x) = \max_{i=1,\ldots,s} \left(\begin{bmatrix} x' & -1 \end{bmatrix} \begin{bmatrix} \omega^i \\ \gamma^i \end{bmatrix} \right), \tag{4.19}$$

where $\omega^i \in \mathbb{R}^{n_x}$ and $\gamma^i \in \mathbb{R}$ are the parameters to be computed. Let m_i denote the cardinality of the cluster $\mathcal{C}_i$ and let $M_i \in \mathbb{R}^{m_i \times n_x}$, for $i = 1, \ldots, s$, which is obtained by stacking the regressors $x^\top(k)$ belonging to $\mathcal{C}_i$ in its rows.

If the clusters $\{\mathcal{C}_i\}_{i=1}^s$ are linearly separable, then the separator function ϕ satisfies the following conditions:

$$\begin{bmatrix} M_i & -\mathbf{1}_{m_i} \end{bmatrix} \begin{bmatrix} \omega^i \\ \gamma^i \end{bmatrix} \geq \begin{bmatrix} M_i & -\mathbf{1}_{m_i} \end{bmatrix} \begin{bmatrix} \omega^j \\ \gamma^j \end{bmatrix} + \mathbf{1}_{m_i}, \quad i, j = 1, \ldots, s, \; i \neq j, \tag{4.20}$$

where $\mathbf{1}_{m_i}$ is an m_i-dimensional vector of ones.

The piecewise-affine separator ϕ thus satisfies the conditions:

$$\begin{cases} \phi(x) = \quad [x' \quad -1] \begin{bmatrix} \omega^i \\ \gamma^i \end{bmatrix}, \ \forall x \in \mathcal{C}_i, \ \ i = 1, \ldots, s \\ \phi(x) \geq \quad [x' \quad -1] \begin{bmatrix} \omega^j \\ \gamma^j \end{bmatrix} + 1, \ \forall x \in \mathcal{C}_i, \ i \neq j \end{cases} \tag{4.21}$$

From (4.21), the polyhedra $\{\mathcal{X}_i\}_{i=1}^s$ are defined as

$$\mathcal{X}_i = \left\{ x \in \mathbb{R}^{n_x} : [x' \quad -1] \begin{bmatrix} \omega^i - \omega^j \\ \gamma^i - \gamma^j \end{bmatrix} \geq 1, \ j = 1, \ldots, s, \ j \neq i \right\}.$$

Off-line multicategory discrimination

The parameters $\{\omega^i, \gamma^i\}_{i=1}^s$ are calculated by solving the optimization problem which is convex [19] (instead of solving a *robust linear programming* (RLP) problem as in [14]),

$$\begin{aligned} &\min_{\xi} \frac{\kappa}{2} \sum_{i=1}^{s} \left(\|\omega^i\|_2^2 + (\gamma^i)^2 \right) + \\ &\sum_{i=1}^{s} \sum_{\substack{j=1 \\ j \neq i}}^{s} \frac{1}{m_i} \left\| \left(\begin{bmatrix} M_i & -\mathbf{1}_{m_i} \end{bmatrix} \begin{bmatrix} \omega^j - \omega^i \\ \gamma^j - \gamma^i \end{bmatrix} + \mathbf{1}_{m_i} \right)_+ \right\|_2^2, \end{aligned} \tag{4.22}$$

with $\xi = \begin{bmatrix} (\omega^1)^\top & \ldots & (\omega^s)^\top & \gamma^1 & \ldots & \gamma^s \end{bmatrix}^\top$. Problem (4.22) minimizes the averaged squared 2-norm of the violation of the inequalities in (4.20). The regularization parameter $\kappa > 0$ guarantees that the objective function (4.22) is strongly convex. Problem (4.22) is then solved through the *Regularized Piecewise-Smooth Newton* (RPSN) method explained in [19] and originally proposed in [10].

Recursive multicategory discrimination

An online approach can be used, either in place of the off-line approach or to refine the partition ϕ online based on streaming data. A recursive approach using on-line convex programming can be used to solve (4.22).

Considering the data-points $x \in \mathbb{R}^{n_x}$ as random vectors and let us assume that there exists an oracle function $i : \mathbb{R}^{n_x} :\to \{1, \ldots, s\}$ that assigns the corresponding mode $i(x) \in \{1, \ldots, s\}$ to a given $x \in \mathbb{R}^{n_x}$. By definition, function i describes the clusters in the data-point space $\mathbb{R}^{n_x}$. Let us also assume that the following probabilities

$$\pi_i = \text{Prob}[i(x) = i] = \int_{\mathbb{R}^{n_x}} \delta(i, i(x)) p(x) dx,$$

are known for all $i = 1, \dots, s$, where $\delta(i,j) = 1$ if $i = j$, zero otherwise. Problem (4.22) can be the generalized as the following convex regularized stochastic optimization problem

$$\xi^* = \min_{\xi} E_{x \in \mathbb{R}^{n_x}} \left[\ell(x, \xi) \right] + \frac{\kappa}{2} \|\xi\|_2^2 \tag{4.23}$$
$$\ell(x, \xi) = \sum_{\substack{j=1 \\ j \neq i(x)}}^{s} \frac{1}{\pi_{i(x)}} \left(x^\top (\omega^j - \omega^{i(x)}) - \gamma^j + \gamma^{i(x)} + 1 \right)_+^2,$$

where $E_x[\cdot]$ denotes the expected value w.r.t. x. Problem (4.23) aims at violating the least, on average over x, the condition in (4.20) for $i = i(x)$. The coefficients π_i can be estimated offline from a data subset, specifically $\pi_i = \frac{m_i}{N}$, and can be updated iteratively. Nevertheless, numerical experiments have shown that uniform coefficients $\pi = \frac{1}{s}$ work equally well. Problem (4.23) can be solved online using convex optimization algorithm, *Averaged Stochastic Gradient Descent* (ASGD) method described in [19].

To summarize, the proposed PWA regression algorithm consists of the following two stages.

- At the first stage, the training samples are processed iteratively, and a *Mixed-Integer Quadratic-Programming* (MIQP) problem is solved to find the sequence of active modes and the model parameters which best match the training data, within a relatively short time window in the past. According to a moving-horizon strategy, only the last element of the optimal sequence of active modes is kept, and the next sample is processed by shifting forward the estimation horizon. A regularization term on the model parameters is included in the cost of the formulated MIQP problem, to partly take into account also the past training data outside the considered time horizon.

- At the second stage, linear multi-category discrimination techniques are used to compute a polyhedral partition of the regressor space based on the estimated sequence of active modes.

4.4 Simulation examples

In this section, the performance of the proposed PWA regression algorithm is shown via identification of PWA-ARX and LPV models. The output sequence used for training is corrupted by a zero mean white Gaussian noise process e_{o}. The *Signal-to-Noise Ratio* (SNR) index

$$\mathrm{SNR} = 10 \log \frac{\sum_{t=1}^{N} (y(t) - e_{\mathrm{o}}(t))^2}{\sum_{t=1}^{N} (e_{\mathrm{o}}(t))^2}, \tag{4.24}$$

quantifies the effect of the measurement noise on the output. The quality of the identified models is assessed on a noiseless validation dataset (not used for training) through the *Best Fit Rate* (BFR) index

$$\mathrm{BFR} = \max \left\{ 1 - \sqrt{\frac{\sum_{k=1}^{N_{\mathrm{val}}} \left(y(k) - \hat{y}(k)\right)^2}{\sum_{k=1}^{N_{\mathrm{val}}} \left(y(k) - \bar{y}\right)^2}}, 0 \right\}, \tag{4.25}$$

with N_{val} being the length of the validation set and $\hat{y}$ being the estimated model output and $\bar{y}$ the sample mean of the output signal. All the simulations are carried out using a desktop computer with MATLAB R2015a, Intel Core i7-4700MQ CPU with 2.40 GHz and 8 GB of RAM.

4.4.1 Identification of PWA-ARX models

SISO case

As a first example, we consider a *single-input single-output* (SISO) PWA-ARX system for the data generation, described by the difference equation

$$\begin{aligned} y(k) =& 0.8y(k-1) + 0.4u(k-1) - 0.1 + \max\{-0.3y(k-1) \\ &+ 0.6u(k-1) + 0.3, 0\} + e_{\mathrm{o}}(k), \end{aligned}$$

with $\bar{s} = 2$ modes, based on the possible combinations generated by the sign of the "max" operator. To gather the data, the system is excited by an input $u(k)$ which is chosen to be white noise with uniform distribution $\mathcal{U}(-1, 1)$ and length $N = 1000$, $e_{\mathrm{o}}(k) \in \mathbb{R}$ is a zero-mean white Gaussian noise with variance $\sigma_e^2 = 6.25 \cdot 10^{-4}$. This corresponds to a SNR on the output channel equal to 20 dB.

For identification, Algorithm 2 is run for 3 iterations (i.e. $n_q = 3$) with $s = \bar{s} = 2$ and with prediction horizon $N_p = 5$, forgetting factor

$\lambda = 0.01$. In (4.8), the terms (4.8a) and (4.8b) are weighted by factors $\gamma_1 = 10$, $\gamma_2 = 1$ respectively. The resultant MIQP problem (4.15) consists of 26 variables out of which 10 are binary, 40 inequality and 5 equality constraints. The MIQP problem is solved with the recently proposed GPAD-B&B algorithm [65] and the performance is compared with the commercial solver GUROBI. In the second stage, off-line multicategory discrimination algorithm (Section 4.3.2) is executed for the partitioning of the regressor space, with parameter $\kappa = 10^{-5}$. The BFR on the noise-free validation data-set of length $N_{\text{val}} = 300$ are summarized in Table 11. The mean time taken to process a single training sample by GPAD-B&B

Table 11: BFR on the validation data set for SISO PWARX system.

runs(n_q)	GUROBI	GPAD-B&B
1	0.86	0.86
2	0.90	0.89
3	0.92	0.90

is 0.13 sec with the feasibility tolerance $\epsilon_G = 1 \cdot 10^{-3}$, optimality tolerance $\epsilon_V = 1 \cdot 10^{-3}$, infeasibility detection tolerance $\epsilon_I = 1 \cdot 10^{-3}$, whereas GUROBI with default settings takes 0.09 sec. GPAD-B&B makes a trade off between the execution time and quality of solution, by selecting appropriate tolerance values. It is simple library-free solver, yet rendered comparable performance with respect to the commercial solver for the given problem. The effect of increasing the prediction horizon N_p on the

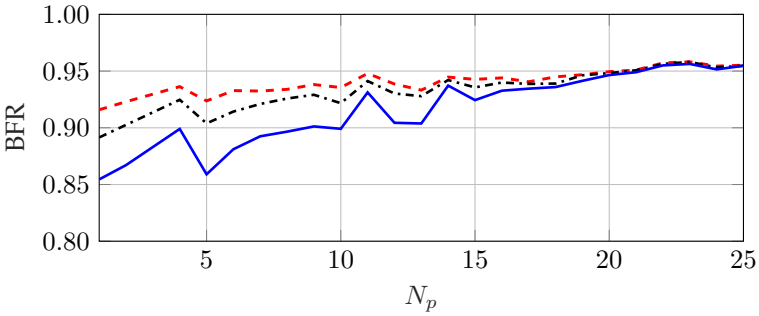

Figure 12: BFR vs N_p : $n_q = 1$, $n_q = 2$, $n_q = 3$.

BFR is shown in Figure 12, for $n_q = 1, 2$ and 3 runs respectively.

MIMO case

As a second example, the following *Multiple-Input Multiple-Output* (MIMO) PWA-ARX data generating system, taken from [18], is considered

$$\begin{bmatrix} y_1(k) \\ y_2(k) \end{bmatrix} = \begin{bmatrix} -0.83 & 0.20 \\ 0.30 & -0.52 \end{bmatrix} \begin{bmatrix} y_1(k-1) \\ y_2(k-1) \end{bmatrix} + \begin{bmatrix} -0.34 & 0.45 \\ -0.30 & 0.24 \end{bmatrix} \begin{bmatrix} u_1(k-1) \\ u_2(k-1) \end{bmatrix} + \begin{bmatrix} 0.20 \\ 0.15 \end{bmatrix} + \max \left\{ \begin{bmatrix} 0.20 & -0.90 \\ 0.10 & -0.42 \end{bmatrix} \begin{bmatrix} y_1(k-1) \\ y_2(k-1) \end{bmatrix} + \begin{bmatrix} 0.42 & 0.20 \\ 0.50 & 0.64 \end{bmatrix} \begin{bmatrix} u_1(k-1) \\ u_2(k-1) \end{bmatrix} + \begin{bmatrix} 0.40 \\ 0.30 \end{bmatrix}, \begin{bmatrix} 0 \\ 0 \end{bmatrix} \right\} + e_{\mathrm{o}}(k),$$

described by $\bar{s} = 4$ modes, based on the possible combinations generated by the sign of the vector-valued "max" operator. To gather the data, the input sequence $u(k)$ is chosen to be a white noise process of length $N = 3000$, having uniform distribution in the boxes $[-2\ \ 2] \times [-2\ \ 2]$. The noise signal $e_{\mathrm{o}} \in \mathbb{R}^2$ is a white Gaussian noise with covariance matrix $\Lambda_e = \begin{bmatrix} 2.5 \cdot 10^{-3} & 0 \\ 0 & 2.5 \cdot 10^{-3} \end{bmatrix}$. This results in SNRs equal to 27 dB and 26 dB for the first and the second output channel, respectively.

Algorithm 2 is run for one iteration (i.e. $n_q = 1$) with $s = \bar{s} = 4$ and with a prediction horizon $N_p = 10$. GUROBI is used to solve the MIQP problem (4.15). In the second stage, off-line multicategory discrimination algorithm (reported in Section 4.3.2) is run with parameter $\kappa = 10^{-5}$, for the partitioning of the regressor space.
The achieved results in terms of BFR for output channel 1 and 2 on a noise-free validation data set of length $N_{\mathrm{val}} = 500$ are reported in Table 12.

Table 12: BFRs on the validation data set for MIMO PWARX system.

Output channel	BFR
y_1	0.95
y_2	0.94

4.4.2 Identification of LPV models

SISO LPV model

Consider the SISO-LPV ARX data generating system:

$$y(k) = a_1^{\circ}(p(k))y(k-1) + a_2^{\circ}(p(k))y(k-2) + b_1^{\circ}(p(k))u(k-1) + e_{\circ}(k).$$

The p-dependent coefficients $a_1^{\circ}(p(k))$, $a_2^{\circ}(p(k)$ and $b_1^{\circ}(p(k))$ are described by the nonlinear functions:

$$a_1^{\circ}(p(k)) = \begin{cases} -0.5, & \text{if } p(k) > 0.5 \\ -p(k), & \text{if } -0.5 \leq p(k) \leq 0.5 \\ 0.5, & \text{if } p(k) < -0.5 \end{cases} \tag{4.26a}$$

$$a_2^{\circ}(p(k)) = p^3(k) \tag{4.26b}$$

$$b_1^{\circ}(p(k)) = \sin(\pi p(k)) \tag{4.26c}$$

A training data and a validation dataset of length $N = 6000$ and $N_{\text{val}} = 2000$, respectively, are generated. The training input u and the scheduling signal p are considered as independent white-noise processes with uniform distribution $\mathcal{U}(-1, 1)$. The standard deviation of the noise e is 0.05, which corresponds to SNR of 20 dB.

A PWA model with $s = 6$ modes is considered. Algorithm 2 is run with prediction horizon $N_p = 6$. Each MIQP sub-problem (4.15), solved with GPAD-B&B, contains 36 binary and 72 continuous variables, 144 inequality and 6 equality constraints. The average time to solve this problem is 0.13 sec for GPAD-B&B and 0.09 sec for GUROBI with default settings. In the second stage, off-line multicategory discrimination algorithm is executed for partitioning the scheduling space, and the estimated model parameters are refined based on the computed partition, using simple least-squares for each sub-model.

The best fit rate on the noise-free validation dataset is 0.95. The estimated LPV model coefficient functions $a_1(p(k))$, $a_2(p(k))$, $b_1(p(k))$ are shown in Figure 13.

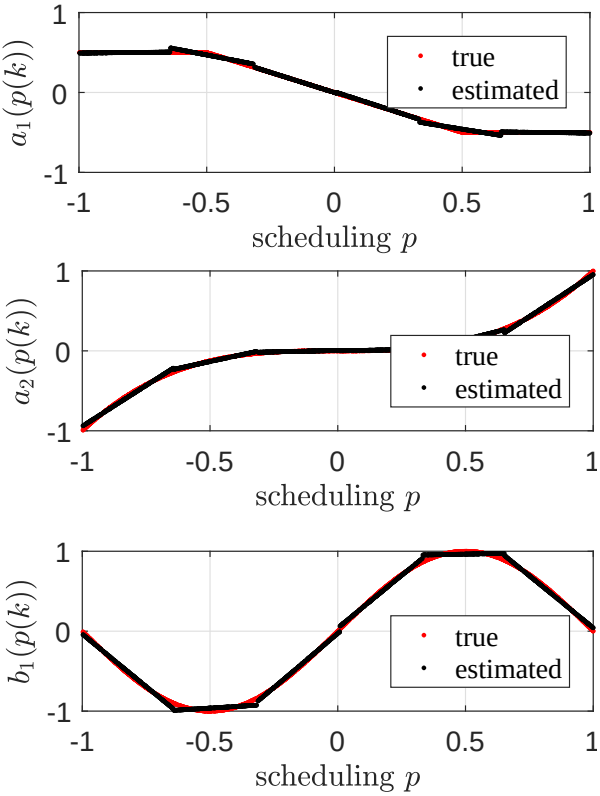

Figure 13: PWA estimates of the LPV coefficient functions

MIMO LPV models

Consider the MIMO LPV-ARX data generating system:

$$\begin{bmatrix} y_1(k) \\ y_2(k) \end{bmatrix} = \begin{bmatrix} \bar{a}_{1,1}(p(k)) & \bar{a}_{1,2}(p(k)) \\ \bar{a}_{2,1}(p(k)) & \bar{a}_{2,2}(p(k)) \end{bmatrix} \begin{bmatrix} y_1(k-1) \\ y_2(k-1) \end{bmatrix} + \begin{bmatrix} \bar{b}_{1,1}(p(k)) & \bar{b}_{1,2}(p(k)) \\ \bar{b}_{2,1}(p(k)) & \bar{b}_{2,2}(p(k)) \end{bmatrix} \begin{bmatrix} u_1(k-1) \\ u_2(k-1) \end{bmatrix} + e(k), \tag{4.27}$$

with

$$\bar{a}_{1,1}(p(k)) = \begin{cases} -0.3 & \text{if} \quad 0.4\,(p_1(k)+p_2(k)) \leq -0.3, \\ 0.3 & \text{if} \quad 0.4\,(p_1(k)+p_2(k)) \geq 0.3, \\ 0.4\,(p_1(k)+p_2(k)) & \text{otherwise}, \end{cases}$$

$$\bar{a}_{1,2}(p(k)) = 0.5\,(|p_1(k)| + |p_2(k)|)\,,$$

$$\bar{a}_{2,1}(p(k)) = p_1(k) - p_2(k),$$

$$\bar{a}_{2,2}(p(k)) = \begin{cases} 0.5 & \text{if} \quad p_1(k) < 0, \\ 0 & \text{if} \quad p_1(k) = 0, \\ -0.5 & \text{if} \quad p_1(k) > 0, \end{cases}$$

$$\bar{b}_{1,1}(p(k)) = 3p_1(k) + p_2(k),$$

$$\bar{b}_{1,2}(p(k)) = \begin{cases} 0.5 & \text{if} \quad 2\left(p_1^2(k)+p_2^2(k)\right) \geq 0.5, \\ 2\left(p_1^2(k)+p_2^2(k)\right) & \text{otherwise}, \end{cases}$$

$$\bar{b}_{2,1}(p(k)) = 2\sin\left\{p_1(k) - p_2(k)\right\},$$

$$\bar{b}_{2,2}(p(k)) = 0.$$

The training and the validation dataset consist of $N = 6000$ and $N_{\text{val}} = 2000$ samples, respectively. The input $u(k)$ and the scheduling vector $p(k)$ are white noise sequences generated drawn from a uniform distribution in the range $[-0.5 \;\; 0.5] \times [-0.5 \;\; 0.5]$ and $[-1 \;\; 1] \times [-1 \;\; 1]$, respectively. The covariance matrix of $e(k) \in \mathbb{R}^2$ is $\Lambda_e = \begin{bmatrix} 0.25e^{-2} & 0 \\ 0 & 0.25e^{-2} \end{bmatrix}$ which corresponds to signal-to-noise ratios on the first and on the second output channel equal to $\text{SNR}_1 = 23$ dB and $\text{SNR}_2 = 23.5$ dB, respectively.

A PWA-LPV model with $s = 10$ modes is considered. Algorithm 2 is run for $n_{\text{q}} = 2$ iterations with the parameters: prediction horizon $T = 10$, forgetting factor $\lambda = 0.25$.

In the second stage, problem (4.22) is solved to compute the partition of the scheduling space, with parameter $\kappa = 10^{-10}$ in (4.22). The obtained partition of the scheduling space is depicted in Figure 14. The BFRs on the noise-free validation dataset are reported in Table 13 for both

Table 13: BFR on the validation data

runs(n_q)	BFR y_1	BFR y_2
1	0.8793	0.8108
2	0.8817	0.8146

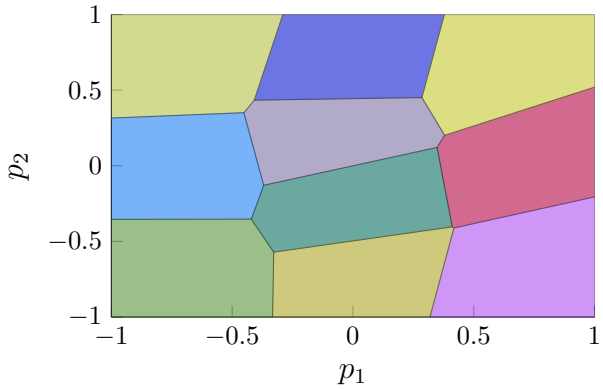

Figure 14: Partition of the scheduling space

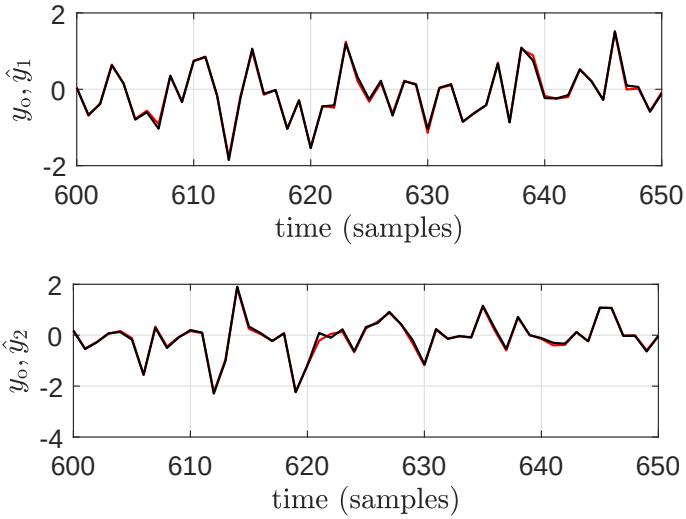

Figure 15: Estimated (black) vs true (red) outputs

the output channels, and the output trajectories are shown in Figure 15. For the sake of visualization, only part of the validation data are plotted. The obtained results show the capabilities of the estimated PWA-LPV model in reproducing the behaviour of the true LPV system.

4.5 Conclusions

A novel moving-horizon algorithm for *PieceWise Affine* regression has been described in this chapter. The proposed method combines the advantages of the mixed-integer programming method [83] (namely, simultaneous choice of the model parameters and of the optimal sequence of active modes within a relatively short time horizon) and the recursive algorithm [19] (namely, computational efficiency and iterative processing of the training samples). The effectiveness of the proposed method is tested via identification of LPV and PWA-ARX models.

In the appendix of the chapter, the framework of the proposed algorithm is applied to real world energy disaggregation problem. A small set of training data consisting of disaggregated power profiles for individual appliances is used to estimate PWA autoregressive models describing the consumption patterns of individual appliances. Once the model parameters are estimated for each appliance, the energy disaggregation problem is formulated as a binary quadratic program. The dynamic modeling of the power profiles of individual appliances leads to better energy disaggregation results compared to the same approach relying on static models. This is due to the fact that the dynamic models are able to capture the transient behavior, thus providing vital information to distinguish between the appliances having similar power signatures. The proposed energy disaggregation method is computationally efficient as the appliance models can be estimated off-line only once, while energy disaggregation is performed online with low computational complexity. Thus, the approach proposed in this chapter is promising for embedded implementation in smart meters.

4.6 Appendix

4.6.1 Energy Disaggregation

In this section, we apply the framework of the proposed algorithm for the problem of energy disaggregation commonly referred in the literature as non-intrusive load monitoring. The problem is to estimate the end-use power consumption profiles of individual household appliance using only aggregated power measurements. We propose a two-stage supervised approach. At the first stage, dynamical models of individual appliances are estimated using disaggregated training data gathered over a short intrusive period. The consumption profiles of individual appliances are described by *PieceWise Affine AutoRegressive* (PWA-AR) models with multiple operating modes, which are estimated via a moving horizon PWA regression Algorithm 2. Once the model of each appliance is identified, a binary quadratic programming problem is solved at the second stage to determine the set of active appliances which contribute to the instantaneous aggregated power, along with their operating modes. A benchmark dataset is used to assess the performance of the presented disaggregation approach.

4.6.2 Motivation

Retrieving power consumptions at the single-appliance level provides useful information to energy suppliers, municipalities, and consumers to design and assess efficiency of energy management strategies, increase consumers' awareness on their habits, detect malfunctioning, etc. One can acquire this information via hardware, by attaching a smart meter or a smart plug to every individual appliance. However, this is not economical when there are many devices to monitor. Alternatively, a software-based solution can be used to decompose the aggregate power reading gathered from a single-point smart meter into the individual consumption of each appliance. This approach is known as *Non-Intrusive Load Monitoring* (NILM) or *energy disaggregation*. The advantages of the software-based solution are reduction of intrusiveness into consumers' houses and lower costs for installation, maintenance and replacement of the monitoring system.

A first energy disaggregation algorithm was proposed by Hart in [39], where the aggregate power signal is decomposed to match the individual appliances' typical power demand curves (commonly referred to as

signatures). The limitation of Hart's approach is that it cannot detect appliances with multiple operating modes and it is neither able to decompose power signals made of simultaneous on/off events on multiple appliances. Thereafter, the NILM problem has been extensively studied in the literature (see [108], [113], [26] and references therein). The main idea behind most of the available methods is to characterize the typical consumption signatures of the appliances using a small set of disaggregated data gathered during a short intrusive period. Once the appliances' signatures are available, disaggregation is performed. Among the available approaches, we mention the ones based on sparse coding [33], blind identification [28], pattern recognition [30], hidden Markov models and its variants [44; 46; 74], deep learning [85; 92], integer programming [21; 94], and convex optimization [77].

4.6.3 Contribution

We propose a novel approach for energy disaggregation that relies on *PieceWise Affine AutoRegressive* (PWA-AR) dynamical models to describe the behavior of the individual appliances. Using a set of disaggregated data collected over a short intrusive period, the PWA-AR models are first estimated off-line using the moving-horizon PWA regression Algorithm 2. Once the appliance models are estimated, energy disaggregation is formulated as an integer programming problem. Specifically, based on the measurements of the aggregated power, the active operating mode of each appliance (and thus its power consumption) is determined at each time instance in an iterative way. The developed disaggregation algorithm is tested on a benchmark dataset, using PWA-AR dynamic models and also static models defined based on the average power ratings of the devices. The obtained results show that using dynamic models for the individual appliances instead of static models significantly improves the performance.

4.6.4 Problem formulation

Consider a household with n different electric appliances connected to the power line. The energy consumption of each appliance is described by *PieceWise Affine AutoRegressive* (PWA-AR) model with $s_i \in \mathbb{N}$ (with $i = 1, \ldots, n$) operating modes. Although the appliances may have different operating modes, to simplify the notation we consider the case in which all appliances have equal number of operating modes $s \in \mathbb{N}$ described

by PWA-AR models having the same dynamical order $n_a \in \mathbb{N}$. More specifically, the power $y_i(k)$ consumed by the i-th appliance at time k is modeled by

$$y_i(k) = \begin{cases} \Theta_{i,1} \begin{bmatrix} 1 \\ x_i(k) \end{bmatrix} & \text{if } \delta_{i,1}(k) = 1, \\ \vdots & \\ \Theta_{i,s} \begin{bmatrix} 1 \\ x_i(k) \end{bmatrix} & \text{if } \delta_{i,s}(k) = 1, \end{cases} \tag{4.28}$$

where $\delta_{i,j}(k) \in \{0,1\}$ (with $j = 1, \ldots, s$) is a binary variable which is used to characterize the active operating mode of the appliance (i.e., the j-th mode is active if and only if $\delta_{i,j}(k) = 1$), $\Theta_{i,j}$ is a set of parameters describing the behavior of the i-th appliance at the j-th operating mode and $x_i(k)$ denotes the regressor vector containing the past values of the outputs

$$x_i(k) = [y_i(k-1), \ldots, y_i(k-n_a)]^\top . \tag{4.29}$$

At a given time k, one and only one mode of the appliance is active, i.e., $\sum_{j=1}^{s} \delta_{i,j}(k) = 1$. The measured aggregated power reading is

$$y(k) = \sum_{i=1}^{n} y_i(k) + e(k), \tag{4.30}$$

with $e(k)$ taking into account the measurement noise and unmodeled appliances.

Problem 1 (Energy disaggregation problem) *Given an N-length data sequence $\mathcal{D} = \{y(k)\}_{k=1}^{N}$ of aggregated power signals $y(k)$, estimate the end-use power consumption profiles $y_i(k)$ (with, $i = 1, \ldots, n$ and $k = 1, \ldots, N$).*

4.6.5 Energy disaggregation algorithm

We detail a supervised disaggregation algorithm which consists of two stages:

S1. The PWA-AR model in (4.28) describing the behavior of individual appliances is estimated via the PWA regression Algorithm 2, using disaggregated training data collected over a short intrusive period. It is a common practice to use a small set of disaggregated data to learn the signature of the appliances [77; 108; 113].

S2. Using the PWA-AR models obtained from stage **S1**, an integer programming problem is solved iteratively to determine the active devices contributing to the instantaneous total power, along with their corresponding operating modes.

In the following sections, stages **S1** and **S2** are described in detail.

4.6.6 Stage S1: Training appliance models

For each appliance, consider a set of training data of length $\bar{N}$ consisting of the disaggregated power consumption $\{y_i(k)\}_{k=1}^{\bar{N}}$. The training regressor/output pairs $\{x_i(k), y_i(k)\}$, with $x_i(k)$ defined in (4.29), are processed iteratively. At each time sample k, a moving-horizon window of length $N_p \ll \bar{N}$ containing regressor/output pairs from time $k - N_p + 1$ to time k is considered. The model parameters $\Theta_{i,j}$ and the binary variables $\delta_{i,j}(k)$ (for $j = 1, \dots, s$) at time k are estimated simultaneously by solving the mixed-integer quadratic programming problem (as given in (4.8))

$$\min_{\substack{\Theta_{i,j} \\ \delta_{i,j}(k-t)}} \sum_{j=1}^{s} \sum_{t=0}^{N_p-1} \left\| \left(y_i(k-t) - \Theta_{i,j} \begin{bmatrix} 1 \\ x_i(k-t) \end{bmatrix}\right) \delta_{i,j}(k-t) \right\|_2^2 \tag{4.31a}$$

$$+ \sum_{t=1}^{k-N_p} \left\| y_i(t) - \Theta_{i,\sigma(t)} \begin{bmatrix} 1 \\ x_i(t) \end{bmatrix} \right\|_2^2 \tag{4.31b}$$

$$\text{s.t. } \delta_{i,j}(k-t) \in \{0,1\}, \sum_{j=1}^{s} \delta_{i,j}(k-t) = 1,\ t = 0, \dots, N_p - 1. \tag{4.31c}$$

The objective of problem (4.31) is to determine the optimal sequence of active modes

$$\sigma_i(k-t) = j^* \Leftrightarrow \delta_{i,j^*}(k-t) = 1, \quad t = 0, \dots, N_p - 1,$$

within the considered time window, along with the model parameters $\Theta_{i,j}$ (for each appliance i and mode j), which best match the available power consumption up to time k. The term (4.31b) is a regularization term on the parameters $\Theta_{i,j}$, which takes into account the past data outside the considered horizon. As described in Section 4.3, in (4.31b) the active mode sequence is not optimized from time 1 to $k - N_p$, but it is fixed

to the estimates $\{\sigma_i(t)\}_{t=1}^{k-N_p}$ computed from the previous iterations of the moving-horizon estimation algorithm. In turn, the sequence of active modes is optimized only within the considered time horizon in (4.31a). At time k, only the active mode $\sigma_i(k)$ is kept, and the N_p-length time window is shifted forward to process the next pair $\{x_i(k+1), y_i(k+1)\}$ in a moving-horizon fashion. Increasing N_p provides more data to estimate the model parameters $\Theta_{i,j}$ and the sequence of active modes $\sigma_i(k)$, which improves the accuracy of the estimates. On the other hand, increasing N_p increases the number of binary decision variables $\delta_{i,j}$. Thus, the parameter N_p acts as a tuning knob to trade off accuracy vs complexity.

At the end of the training phase, the signature of the i-th appliance is captured by the estimated model parameters $\Theta_{i,j}$ for all modes $j = 1, \ldots, s$.

4.6.7 Stage S2: Energy disaggregation

Once the model parameters $\Theta_{i,j}$ for each appliance are estimated, the energy disaggregation problem is to determine the operating mode of each appliance based on aggregated power measurements. To this end we solve the following binary quadratic program

$$\min_{\{\delta_{i,j}(k)\}_{i,j=1}^{n,s}} \left\| y(k) - \sum_{i=1}^{n}\sum_{j=1}^{s} \Theta_{i,j} \begin{bmatrix} 1 \\ \hat{x}_i(k) \end{bmatrix} \delta_{i,j}(k) \right\|_2^2, \tag{4.32a}$$

$$\text{s.t. } \delta_{i,j}(k) \in \{0,1\}, \ \sum_{j=1}^{s} \delta_{i,j}(k) = 1, \tag{4.32b}$$

at each time instance k, where $y(k)$ is the measurement of the aggregated power, $\Theta_{i,j}$ are model parameters estimated at stage **S1**, $\hat{x}_i(k)$ is the simulated regressor vector[2] defined as

$$\hat{x}_i(k) = [\hat{y}_i(k-1), \ldots, \hat{y}_i(k-n_\mathrm{a})]^\top,$$

where $\hat{y}_i(k)$ is the estimate of the disaggregated power for the i-th appliance constructed as follows.

[2]The true regressor $x_i(k)$ defined in (4.29) can not be constructed as it depends on the past values of the individual appliance power y_i which are not available at the stage **S2**.

At each time instance $k = 1, \ldots, N$ the binary quadratic program (4.32) is solved iteratively using an estimate $\hat{x}_i(k)$ of the regressor obtained from the previous iterations. Specifically, at each iteration k the active operating mode j^* of each appliance is determined by the solution of problem (4.32), namely

$$j^* : \delta_{i,j^*}(k) = 1.$$

The power of each individual appliance is thus given by

$$\hat{y}_i(k) = \Theta_{i,j^*} \begin{bmatrix} 1 \\ \hat{x}_i(k) \end{bmatrix}, \tag{4.33}$$

which is used to construct the regressor $\hat{x}_i(k+1)$.

4.6.8 Application to real data

The proposed disaggregation algorithm is tested on a benchmark AMPds dataset [56], which consists of power consumptions of 19 appliances monitored from April 1, 2012 to March 31, 2013 at one minute read intervals in a house located in Canada. In our analysis we consider only the aggregate power consumption given by the sum of the power consumption readings of the following three electric appliances: 1) fridge (FGE); 2) dish washer (DWE); 3) heat pump (HPE). Moreover, for a realistic scenario the aggregated power is corrupted by a fictitious white noise $e(k)$ with Gaussian distribution $\mathcal{N}(0, \sigma_{\mathrm{e}}^2)$ having standard deviation $\sigma_{\mathrm{e}} = 4$ W.

The computations are carried out on an Intel i5 1.7 GHz running MATLAB R2016b.

4.6.9 Performance measures

The quality of the energy disaggregation results is assessed via the following performance measures [77]:

1. *Energy Fraction Index* (EFI)
   The EFI index

   $$\hat{h}_i = \frac{\sum_{k=1}^{N} \hat{y}_i(k)}{\sum_{i=1}^{n} \sum_{k=1}^{N} \hat{y}_i(k)}$$

   quantifies the estimated fraction of total energy consumed by the $i-th$ appliance. This index provides an important information to

the user for potential savings. In order to asses the effectiveness of the algorithm the index $\hat{h}_i$ is compared with an analogous index defined based on the actual disaggregated profiles, namely

$$h_i = \frac{\sum_{k=1}^{N} y_i(k)}{\sum_{i=1}^{n} \sum_{k=1}^{N} y_i(k)}.$$

We remark that the true disaggregated power profiles y_i are not used in the disaggregation algorithm (stage **S2**) but only to evaluate estimation performance.

2. *Relative Square Error (RSE) and R^2 coefficient*
The normalized error between the actual and the estimated power consumption is quantified for the i-th appliance by the RSE_i index defined as

$$\mathrm{RSE}_i = \frac{\sum_{k=1}^{N} \left(y_i(k) - \hat{y}_i(k)\right)^2}{\sum_{k=1}^{N} y_i^2(k)},$$

and the R_i^2 coefficient defined as

$$R_i^2 = 1 - \frac{\sum_{k=1}^{N} \left(y_i(k) - \hat{y}_i(k)\right)^2}{\sum_{k=1}^{N} \left(y_i(k) - \bar{y}_i\right)^2},$$

with $\bar{y}_i = \frac{1}{N} \sum_{k=1}^{N} y_i(k)$. Both RSE_i and R_i^2 measure the match between the actual and the estimated power profiles over time. Indeed, low values of RSE_i (or equivalently high values of R_i^2) imply an accurate estimate of the index $\hat{h}_i$. A precise estimate of the power consumption profiles gives the information to the consumer about the use of household appliances over time. This is essential for potential power savings as well as monetary benefits as the consumer can differ the use of some appliances to off-peak hours.

4.6.10 Supervised training phase

At stage **S1**, PWA-AR models (4.28) describing the behavior for each appliance are estimated as discussed in Section 4.6.6 using data of a one-day (April 19, 2012) intrusive period. PWA-AR models with $s = 3$ modes and order $n_a = 2$ are considered. The moving-horizon mixed-integer quadratic programming problem (4.31) is solved with horizon length $N_p = 5$ using Gurobi [38]. The average computation time to solve (4.31)

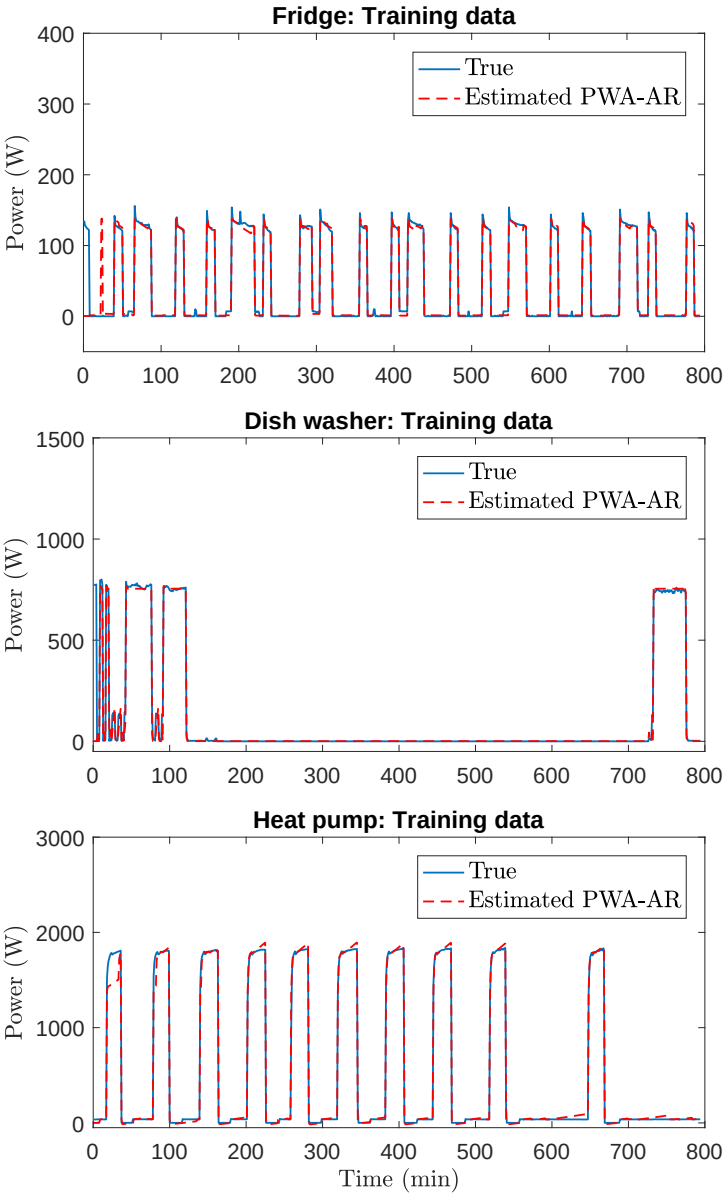

Figure 16: Supervised learning: True vs estimated power consumption.

is 90 ms. The results of the training phase are reported in Figure 16, which shows that the estimated PWA-AR models accurately capture the behavior of the individual appliances' power consumption profiles.

For comparison, the use of static device models for energy disaggregation is reported in the next section. Static models are a special case of PWA-AR models (4.28) with $n_a = 0$ and $y_i(k) = \Theta_{i,j}$. The parameter $\Theta_{i,j}$ thus models the appliance power consumption at the j-th mode and is chosen via simple visual inspection of the training data. The selected values of the parameters $\Theta_{i,j}$ are

a. fridge: $[\Theta_{1,1}\ \Theta_{1,2}\ \Theta_{1,3}] = [0\ 128\ 200]$ W;

b. dish washer: $[\Theta_{2,1}\ \Theta_{2,2}\ \Theta_{2,3}] = [0\ 120\ 800]$ W;

c. heat pump: $[\Theta_{3,1}\ \Theta_{3,2}\ \Theta_{3,3}] = [0\ 39\ 1900]$ W.

4.6.11 Energy disaggregation

Once the PWA-AR models of individual appliances are estimated, the power measurements of one month (from 1st to 30th June, 2012) are disaggregated by solving the binary quadratic program (4.32) iteratively using Gurobi. The average CPU time taken to solve problem (4.32) is 8 ms.

The results obtained by using PWA-AR models (estimated in stage **S1**) and by using static models (introduced in Section 4.6.10), are reported in Table 14 and Table 15. The disaggregated power consumption profiles of fridge, dish washer and heat pump are shown in Figure 17, Figure 18, and Figure 19, respectively.

For visualization purpose, only a portion of the disaggregated power profiles is plotted in Figure 17, Figure 18, and Figure 19. From the figures, it can be seen that the proposed algorithm using PWA-AR models accurately estimates the power consumption trajectories of each individual appliance over time. The efficiency of the method is also reflected in the performance measures reported in Table 14 and Table 15.

We remark that for fridge and dish washer, the second operating mode of both the appliances have similar static models with parameters 128 W and 120 W respectively. From the obtained results it can be observed that using only static models it is difficult to distinguish between these two devices. On the contrary, thanks to their dynamic nature PWA-AR models are able to resolve such a conflict. Overall, PWA-AR models outperform static models in the task of energy disaggregation.

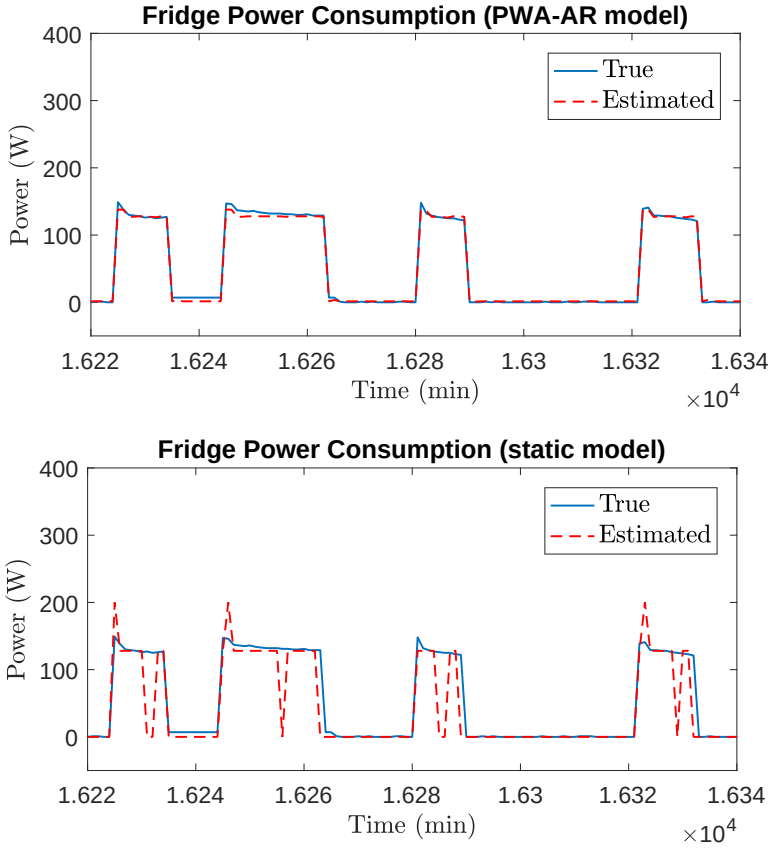

Figure 17: Disaggregated power consumption profile of fridge. Results obtained using PWA-AR models and static models.

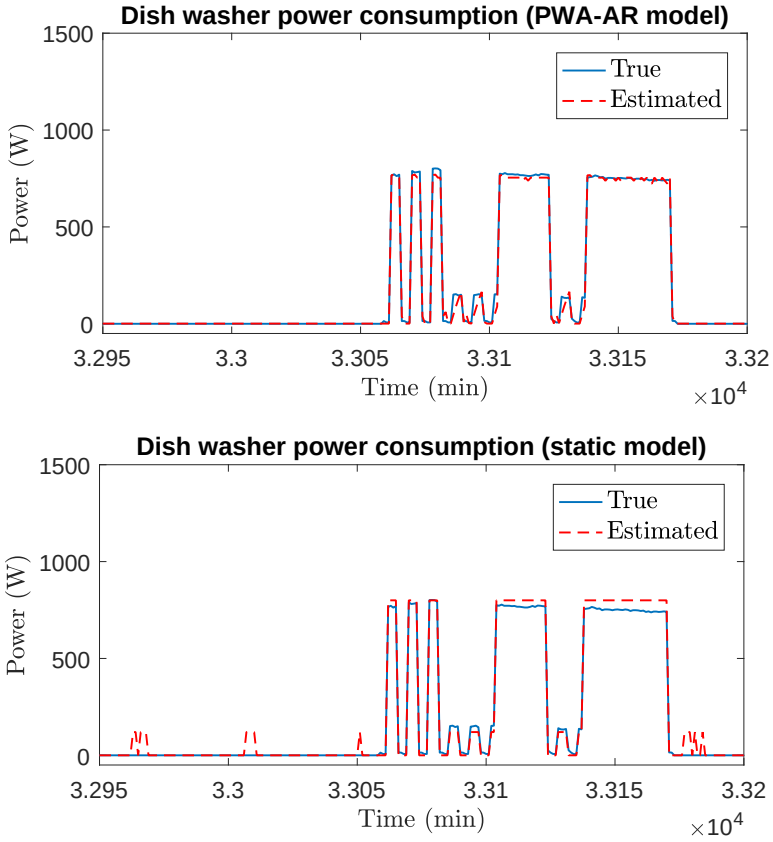

Figure 18: Disaggregated power consumption profile of dish washer. Results obtained using PWA-AR models and static models.

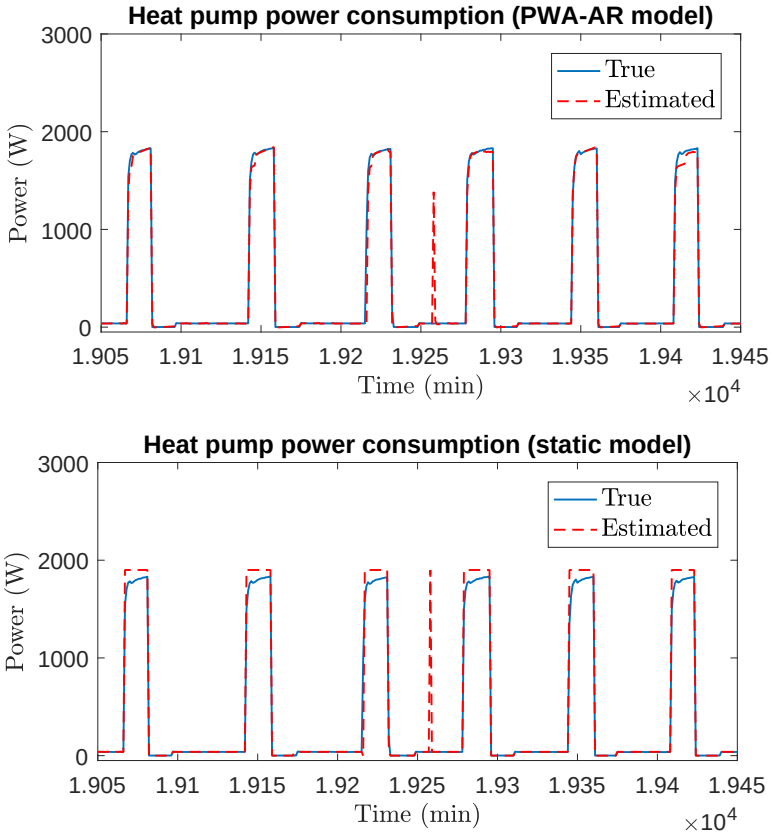

Figure 19: Disaggregated power consumption profile of heat pump. Results obtained using PWA-AR models and static models.

Table 14: Estimated Energy Fraction Index $\hat{h}_i$ and Actual Energy Fraction Index h_i. Results obtained using PWA-AR models and static models.

	PWA-AR models $\hat{h}_i$	static models $\hat{h}_i$	ground truth h_i
Fridge	29.9 %	20.7 %	31.0 %
Dish washer	9.3 %	16.6 %	7.4 %
Heat pump	60.8 %	62.7 %	61.6 %

Table 15: Relative Square Errors and R^2 coefficients. Results obtained using PWA-AR models and static models.

	PWA-AR models		static models	
	RSE_i	R_i^2	RSE_i	R_i^2
Fridge	13.9 %	78.9 %	35.3 %	46.4 %
Dish washer	10.5 %	89.3 %	30.4 %	69.0 %
Heat pump	0.6 %	99.3 %	2.0 %	97.8 %

Algorithm 2 Recursive clustering of the regressors and model parameters estimation

Input: Observations sequence $\{x(k), y(k)\}_{k=1}^{N}$; number of modes s; horizon N_p, initial clusters $\mathcal{C}_i$ and centroids c_i.

1. **let** $H_i(0) \leftarrow 0$, $F_i(0) \leftarrow 0$, $\mathcal{C}_i \leftarrow \emptyset$, $i = 1, \dots, s$;
2. **let** $k \leftarrow N_p$;
3. **solve** the MIQP problem (4.15);
4. **let** $\{\delta_i^*(t)\}_{i=1,t=1}^{s,N_p}$ be the optimal parameters minimizing (4.15);
5. **for** $t = 1, \dots, N_p$ **do**
   - 5.1. **let** $i^*(t)$ be the index such that $\delta_{i^*}^*(t) = 1$;
   - 5.2. **let** $\sigma(t) \leftarrow i^*(t)$;
   - 5.3. **let** $\mathcal{C}_{\sigma(t)} \leftarrow \mathcal{C}_{\sigma(t)} \cup \{x(t)\}$;
6. **end for**
7. **for** $k = N_p + 1, \dots, N$ **do**
   - 7.1. **update** the matrices $H_i(k - N_p)$ and $F_i(k - N_p)$ through (4.13);
   - 7.2. **solve** the MIQP problem (4.15);
   - 7.3. **let** $\Theta_i^*(k)$ be the optimal solution of (4.15), $i = 1, \dots, s$;
   - 7.4. **let** $\{\delta_i^*(k - t)\}_{t=0}^{N_p-1}$ be the optimal parameters minimizing (4.15), $i = 1, \dots, s$;
   - 7.5. **let** i^* be the index such that $\delta_{i^*}^*(k) = 1$;
   - 7.6. **let** $\sigma(k) \leftarrow i^*$;
   - 7.7. **let** $\mathcal{C}_{\sigma(k)} \leftarrow \mathcal{C}_{\sigma(k)} \cup x(k)$;
   - 7.8. **let** $\delta c_{\sigma(k)} \leftarrow \frac{1}{|\mathcal{C}_{\sigma(k)}|}(x(k) - c_{\sigma(k)})$;
   - 7.9. **update** the centroid $c_{\sigma(k)}$ of cluster $\mathcal{C}_{\sigma(k)}$: $c_{\sigma(k)} \leftarrow c_{\sigma(k)} + \delta c_{\sigma(k)}$;
8. **end for**;

Output: Estimated parameters $\Theta_1^*(N), \dots, \Theta_s^*(N)$; clusters $\mathcal{C}_1, \dots, \mathcal{C}_s$; sequence of active modes $\{\sigma(k)\}_{k=1}^{N}$.

Chapter 5

Identification of LPV models with linear fractional representation

This chapter presents a method for identification of *Linear Parameter-Varying* (LPV) systems in *Linear Fractional Representation* (LFR) which corresponds to a *Linear Time-Invariant* (LTI) model connected to scheduling variable dependent block via feedback. We propose a two stage identification approach. In the first stage, *Kernelized Canonical Correlation Analysis* (KCCA) is formulated to estimate the state sequence of the underlying LPV model. In the second stage, a nonlinear least squares cost function is minimized employing a coordinate descent algorithm to estimate latent variables characterizing the LFRs and the unknown model matrices of the LTI block by using the state estimates obtained from the first stage. Here, it is assumed that the structure of the scheduling variable dependent block in the feedback path is known. For a special case of affine dependence of the model on the feedback block, it is shown that the optimization problem in the second stage reduces to ordinary least-squares followed by a singular value decomposition.

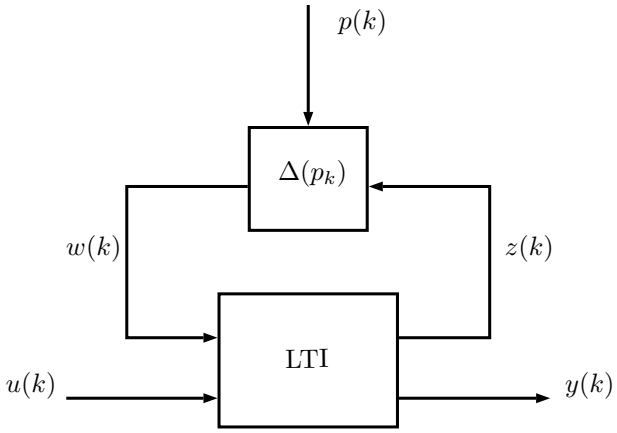

Figure 20: Linear fractional representations of LPV system

5.1 Introduction

5.1.1 Motivation

In the existing literature, many methods have been proposed for the identification of LPV models, both in state-space (SS) [31; 95; 103; 107] and input-output (IO) representations [9; 51; 60; 62]. A detailed summary of the available LPV identification approaches can be found in [98].

The main drawback of these approaches is that the obtained models are not well suited for controller synthesis. The controller design approaches for LPV models ([86; 111]) often require the LPV models to be in *Linear Fractional Representation* (LFR) depicted in Figure 20. Moreover, motivated by the need to design controllers for many real world problems where uncertainty or scheduling parameters enter in a feedback path, identification of LPV-LFR model structures is of significant practical importance. For such models, very efficient *self-scheduled* controllers can be synthesized with low computational complexity [72].

Most of the methods available in the literature for the identification LPV models assume input-output (for eg., the methods presented in the previous chapters) or state space model structures and very few works have addressed the problem of identifying LPV-LFR models. To mention

a few, in [67], it is shown that under the assumption of full state measurement, the problem of single input single output (SISO) LPV-LFR model identification can be solved by recursive least-squares. A prediction error method is proposed in [52; 53], where the mean-squared prediction error cost function is minimized using gradient and Hessian based nonlinear programming. In [24], a local identification approach is presented for SISO LPV-LFR models with scalar scheduling variable. By suitably manipulating the scheduling signal trajectory, it is shown that, for the noise-free case, the model can be identified using convex optimization. For noisy measurements, tractable convex relaxations are proposed under the assumption that a bound on the measurement noise is known.

5.1.2 Contributions

In this chapter, we present a two-stage method which can be seen as a step towards direct identification LPV models with LFRs without structural assumptions. The first stage consists of estimating the state sequence of the underlying LPV models using Kernel Canonical Correlation Analysis (KCCA), which has been recently introduced for LPV-SS models in [81]. It can be proved that the state of the LPV model is a minimal interface between its past and future data. Using this fact, in a KCCA approach, the correlation between past and future data samples is maximized in order to estimate the state sequence up to a similarity transformation. In the second stage, a nonlinear least squares cost function is minimized with coordinate descent algorithm to estimate the latent variables z characterizing LFRs and the unknown model matrices in the forward LTI part, using the state estimates obtained from the first stage. It is shown that under the assumption of affine parametric dependency and diagonal structure of the feedback block, the unknown LTI model matrices can be computed using ordinary least square followed by a singular value decomposition.

5.1.3 Outline

The chapter is organized as follows. In Section 5.1.4, notations used throughout the chapter are introduced. The identification problem of linear fractional representations is formulated in Section 5.2. The proposed method is described in Section 5.3. In Section 5.4, numerical examples are reported to show the effectiveness of the proposed method. Finally, conclusions of the chapter are given in Section 5.5.

5.1.4 Notation

Let $\mathbb{R}^n$ be the set of real vectors of dimension n. The i-th element of a vector $x \in \mathbb{R}^n$ is denoted by $[x]_i$ and $\|x\|_Q^2 = x^\top Q x$ denotes the squared weighted ℓ_2-norm of x. For matrices $A \in \mathbb{R}^{m\times n}$ and $B \in \mathbb{R}^{p\times q}$, the Kronecker product between A and B is denoted by $A \otimes B \in \mathbb{R}^{mp\times nq}$. Let $\mathbb{I}_a^b$ be the sequence of successive integers $\{a, a+1, \cdots, b\}$, with $a < b$. The Moore-Penrose pseudo-inverse of a non-square matrix A is denoted by $A^\dagger$. The notation $(A^d \diamond p)(k)$ is used to express the dynamic dependence of the matrix A on p at time k, e.g., A at time k depends on the future d samples of the signal p, i.e., $p(k), \ \ldots, p(k+d-1)$. The dependence on the past samples is denoted in the same way.

5.2 Problem formulation

By referring to Figure 20, we consider the following discrete time LPV data generating system in a linear fractional representation, where the forward LTI part is given by,

$$\left[\begin{array}{c} x(k+1) \\ \hline z(k) \\ y_{\mathrm{o}}(k) \end{array}\right] = \left[\begin{array}{c|cc} A & B_1 & B_2 \\ \hline C_1 & D_{11} & D_{12} \\ C_2 & D_{21} & D_{22} \end{array}\right] \left[\begin{array}{c} x(k) \\ \hline w(k) \\ u(k) \end{array}\right], \tag{5.1a}$$

where $x \in \mathbb{R}^{n_x}$ is the state, $u \in \mathbb{R}^{n_u}$ and $y_{\mathrm{o}} \in \mathbb{R}^{n_y}$ are the measured inputs and noise-free outputs of the system respectively. The output measurements are corrupted by additive Gaussian white noise, i.e., $y(k) = y_{\mathrm{o}}(k) + e(k)$ and $\{A, \ \ldots, \ D_{22}\}$ are unknown constant matrices of appropriate dimensions. The feedback path is represented by

$$w(k) = \Delta(p(k))z(k), \tag{5.1b}$$

where, $\Delta : \mathbb{P} \to \mathbb{R}^{n_p \times n_p}$ is a function of the scheduling parameter p. The variables x, z and w are latent (auxillary) variables whose measurements are not available.

The following assumption are made about the system in (5.1):

A1. the structure of the feedback block Δ is known.

A2. $I - D_{11}\Delta(p(k)) \neq 0$ for all trajectories of scheduling signal $p \in \mathbb{P}$.

We now formally state the identification problem addressed in this chapter.

Problem 1 *Given an N-length training dataset* $\mathcal{D} = \{u(k), y(k), p(k)\}_{k=1}^{N}$, *identify the LPV system represented by the LFR* (5.1), *estimating the unknown matrices* $\{A, \ldots, D_{22}\}$, *to match the input-output behavior of the underlying data generating system.*

5.3 Identification algorithm for LFR

In this section, we describe the proposed method used to identify the LPV-LFR model (5.1). The method consists of two stages. In the first stage, the estimate $\hat{x}(k)$ of the state sequence characterizing the LTI block is obtained using Kernel Canonical Correlation Analysis (KCCA) [81]. In the second stage, the latent variable sequence $z(k)$ and the unknown matrices $\{A, \ldots, D_{22}\}$ are estimated by minimizing a nonlinear least squares cost function using coordinate descent algorithm.

Furthermore, we consider a special case of the LFR model (5.1) with the following assumptions,

- $D_{11} = 0$, which corresponds to the LPV model with affine dependence on $\Delta(p(k))$,
- The feedback block $\Delta(p(k))$ has a diagonal structure.

Under these assumptions, the problem in the second stage reduces to solving an ordinary least squares problem and the unknown model matrices can be obtained by economic singular value decomposition of the least-square solutions.

5.3.1 Regularized KCCA for state estimation

In this section, we use kernel canonical correlation analysis (KCCA) to obtain the state sequence $\{x(k)\}_{k=1}^{N}$ which is compatible with the given dataset $\mathcal{D}$. The concept of canonical correlation analysis is reviewed in Appendix 5.6.

The model (5.1) can be represented in a LPV state-space form. By re-writing the latent variable $z(k)$ in (5.1) as

$$z(k) = (I - D_{11}\Delta(p(k))^{-1} (C_1 x(k) + D_{12} u(k)), \tag{5.2}$$

and $w(k)$ as $\Delta(p(k))z(k)$, the following LPV state space representation is

obtained

$$x(k+1) = \mathcal{A}(p(k))x(k) + \mathcal{B}(p(k))u(k), \tag{5.3a}$$
$$y(k) = \mathcal{C}(p(k))x(k) + \mathcal{D}(p(k))u(k) + e(k), \tag{5.3b}$$

where

$$\begin{aligned}
\mathcal{A}(p(k)) &= A + B_1\Delta(p(k))(I - D_{11}\Delta(p(k)))^{-1}C_1,\\
\mathcal{B}(p(k)) &= B_2 + B_1\Delta(p(k))(I - D_{11}\Delta(p(k)))^{-1}D_{12},\\
\mathcal{C}(p(k)) &= C_2 + D_{21}\Delta(p(k))(I - D_{11}\Delta(p(k)))^{-1}C_1,\\
\mathcal{D}(p(k)) &= D_{22} + D_{21}\Delta(p(k))(I - D_{11}\Delta(p(k)))^{-1}D_{12}.
\end{aligned}$$

The state sequence $x(k)$ of the LPV model (5.3) compatible with the data set $\mathcal{D}$ can be estimated by using KCCA as introduced in [81]. The key idea behind using KCCA approach is that the state is the minimal interface between past and future inputs u, outputs y, and scheduling data samples p. Therefore, maximizing the correlation between past and future data samples yields state estimates which are compatible with the dataset $\mathcal{D}$ ([81; 106]).

For a given horizon length d, let us define past $\bar{p}_k^d \in \mathbb{R}^{dn_p}$ and future scheduling data vector $\bar{p}_{k+d}^d \in \mathbb{R}^{dn_p}$ w.r.t time instance k as

$$\begin{aligned}
\bar{p}_k^d &= [p(k-d)\dots p(k-1)]^\top,\\
\bar{p}_{k+d}^d &= [p(k)\dots p(k+d-1)]^\top.
\end{aligned}$$

The past and future input data $\bar{u}_k^d \in \mathbb{R}^{dn_u}$, $\bar{u}_{k+d}^d \in \mathbb{R}^{dn_u}$ and output data $\bar{y}_k^d \in \mathbb{R}^{dn_y}$, $\bar{y}_{k+d}^d \in \mathbb{R}^{dn_y}$ are defined in a similar manner.

The future output samples of the LPV model (5.3) can be represented in the observability form

$$\bar{y}_{k+d}^d = (\mathcal{O}_f^d \diamond p)(k) \cdot x(k) + (\mathcal{H}_f^d \diamond p)(k) \cdot \bar{u}_{k+d}^d + \bar{e}_{k+d}^d, \tag{5.4}$$

where the d-step forward observability matrix $(\mathcal{O}_f^d \diamond p)(k) \in \mathbb{R}^{dn_y \times n_x}$ and the forward Toeplitz matrix $(\mathcal{H}_f^d \diamond p)(k) \in \mathbb{R}^{dn_y \times dn_u}$, which have dynamic

dependency on p, are explicitly given in (5.5).

$$\begin{bmatrix} y(k) \\ y(k+1) \\ \vdots \\ y(k+d-1) \end{bmatrix} = \underbrace{\begin{bmatrix} \mathcal{C}(p_k) \\ \mathcal{C}(p_{k+1})\mathcal{A}(p_k) \\ \vdots \\ \mathcal{C}(p_{k+d-1}) \prod_{j=2}^{d} \mathcal{A}(p_{k+d-j}) \end{bmatrix}}_{(\mathcal{O}_f^d \diamond p)(k)} x(k) + \underbrace{\begin{bmatrix} \mathcal{D}(p_k) & 0 & \cdots & 0 \\ \mathcal{C}(p_{k+1})\mathcal{B}(p_k) & \mathcal{D}(p_k) & \cdots & \vdots \\ \vdots & \vdots & \ddots & \vdots \\ \mathcal{C}(p_{k+d-1}) \prod_{j=2}^{d-1} \mathcal{A}(p_{k+d-j})\mathcal{B}(p_k) & \mathcal{C}(p_{k+d-1}) \prod_{j=2}^{d-2} \mathcal{A}(p_{k+d-j})\mathcal{B}(p_{k+1}) & \cdots & \mathcal{D}(p_{k+d-1}) \end{bmatrix}}_{(\mathcal{H}_f^d \diamond p)(k)} \times \begin{bmatrix} u(k) \\ u(k+1) \\ \vdots \\ u(k+d-1) \end{bmatrix} + \begin{bmatrix} e(k) \\ e(k+1) \\ \vdots \\ e(k+d-1) \end{bmatrix} \tag{5.5}$$

From (5.4), the state variable $x(k)$ is given as[1]

$$x(k) = \left((\mathcal{O}_f^d \diamond p)(k)\right)^{\dagger} \left(\bar{y}_{k+d}^d - (\mathcal{H}_f^d \diamond p)(k)\bar{u}_{k+d}^d\right), \tag{5.6}$$

which shows that the value of the state $x(k)$ at time k is a function of future input, output and scheduling variables. Here, we assume that $(\mathcal{O}_f^d \diamond p)(k)$ has full column rank for all $k \in \mathbb{Z}$ satisfying the structural observability assumption (see [81, Definition 3.1]).

Similarly, using the state update equation (5.3a), $x(k)$ can be written in terms of past data samples as

$$x(k) = \left(\prod_{i=1}^{d} \mathcal{A}(p_{k-i})\right) x(k-d) + (\mathcal{R}_p^d \diamond p)(k) \cdot \bar{u}_k^d, \tag{5.7}$$

[1] As e is a zero-mean, independent and identically distributed white noise process, the expected value of the term $\left((\mathcal{O}_f^d \diamond p)(k)\right)^{\dagger} \bar{e}_{k+d}^d$ is zero, which gives the unbiased state estimate (5.6) in conditional mean sense.

where $(\mathcal{R}_p^d \diamond p)(k) \in \mathbb{R}^{n_x \times dn_u}$ is the d-step backward reachability matrix which can be easily computed from recursive substitutions of (5.3a).

Shifting (5.6) by d-samples backward in time we obtain $x(k-d)$ which is then substituted in (5.7). More specifically, from (5.6) we have

$$x(k-d) = \left((\mathcal{O}_f^d \diamond p)(k-d)\right)^\dagger \left(\bar{y}_k^d - (\mathcal{H}_f^d \diamond p)(k-d)\bar{u}_k^d\right). \tag{5.8}$$

Then, by substituting (5.8) in (5.7),

$$x(k) = \mathcal{M}_p^d(k)\left(\bar{y}_k^d - (\mathcal{H}_f^d \diamond p)(k-d)\bar{u}_k^d\right) + (\mathcal{R}_p^d \diamond p)(k) \cdot \bar{u}_k^d, \tag{5.9}$$

where $\mathcal{M}_p^d(k) = \prod_{i=1}^d \mathcal{A}(p_{k-i}) \left((\mathcal{O}_f^d \diamond p)(k-d)\right)^\dagger$. Thus, from (5.9) it follows that the state variable $x(k)$ at time k is a function of past inputs, outputs and scheduling variable data samples.

By defining $\bar{z}_k^d = \begin{bmatrix} \bar{u}_k^d \\ \bar{y}_k^d \end{bmatrix}$, $\bar{z}_{k+d}^d = \begin{bmatrix} \bar{u}_{k+d}^d \\ \bar{y}_{k+d}^d \end{bmatrix} \in \mathbb{R}^{d(n_u+n_y)}$, (5.6) and (5.9) can be rewritten as

$$x(k) = \underbrace{\left((\mathcal{O}_f^d \diamond p)(k)\right)^\dagger \left[-(\mathcal{H}_f^d \diamond p)(k) \;\; I\right]}_{\varphi_f(\bar{p}_{k+d}^d)} \bar{z}_{k+d}^d, \tag{5.10}$$

$$x(k) = \underbrace{\left[-\mathcal{M}_p^d(k)(\mathcal{H}_f^d \diamond p)(k-d) + (\mathcal{R}_p^d \diamond p)(k) \;\; \mathcal{M}_p^d(k)\right]}_{\varphi_p(\bar{p}_k^d)} \bar{z}_k^d, \tag{5.11}$$

where $\varphi_p : \mathbb{R}^{dn_p} \to \mathbb{R}^{n_G \times d(n_u+n_y)}$ and $\varphi_f : \mathbb{R}^{dn_p} \to \mathbb{R}^{n_G \times d(n_u+n_y)}$ are (unknown) mappings of past and future scheduling variable samples to an n_G-dimensional feature space. Note that (5.10) and (5.11) both provide the representation of the state $x(k)$. However, (5.10) depends only on future data, while (5.11) depends only on the past data.

According to the KCCA framework, the state sequence can be estimated by maximizing the correlation between appropriate projections of the quantities $\varphi_p(\bar{p}_k^d)\bar{z}_k^d$ and $\varphi_f(\bar{p}_{k+d}^d)\bar{z}_{k+d}^d$. In the following, we present the formulation of the regularized KCCA based on Least-Square Support Vector Machines (LS-SVM) originally introduced in [93] which allows us to estimate the state sequence without explicitly parameterizing the maps $\varphi_p(\bar{p}_k^d)$ and $\varphi_f(\bar{p}_{k+d}^d)$. The regularized version of KCCA formulated with LS-SVM overcomes the problem of *naive kernelization* (see [5]).

To develop the LS-SVM formulation we define the $N \times n_G$-dimensional

matrices

$$\Phi_p = \left[\varphi_p(\bar{p}_1^d)\bar{z}_1^d,\ \ldots,\ \varphi_p(\bar{p}_N^d)\bar{z}_N^d\right]^\top, \tag{5.12}$$

$$\Phi_f = \left[\varphi_f(\bar{p}_{1+d}^d)\bar{z}_{1+d}^d,\ \ldots,\ \varphi_f(\bar{p}_{N+d}^d)\bar{z}_N^{N+d}\right]^\top. \tag{5.13}$$

The primal LS-SVM problem for KCCA (as introduced in [93]) is given by

$$\begin{aligned}
&\max_{v_j, w_j} \sum_{k=1}^{N} \left(\gamma s_k r_k - \gamma_f \frac{1}{2} s_k^2 - \gamma_p \frac{1}{2} r_k^2\right) - \frac{1}{2} v_j^\top v_j - \frac{1}{2} w_j^\top w_j \\
&\text{s.t. } s_k = v_j^\top \varphi_f(\bar{p}_{k+d}^d)\bar{z}_{k+d}^d,\ r_k = w_j^\top \varphi_p(\bar{p}_k^d)\bar{z}_k^d, \quad \forall k = \mathbb{I}_1^N,
\end{aligned} \tag{5.14}$$

where $\gamma, \gamma_p, \gamma_f \in \mathbb{R}^+$ are regularization hyper-parameters. The variable $v_j \in \mathbb{R}^{n_G}$ and $w_j \in \mathbb{R}^{n_G}$ optimizing (5.14) represent the directions in the feature space along which the projections of future and past data (s_k and r_k, respectively) have maximum correlation. In order to solve (5.14) without explicitly specifying the feature maps $\varphi_p(\bar{p}_k^d)$ and $\varphi_f(\bar{p}_{k+d}^d)$, the dual problem is constructed by defining the Lagrangian

$$\begin{aligned}
&\mathcal{L}(v_j, w_j, s, r, \eta_j, \kappa_j) = \\
&\sum_{k=1}^{N} \left(\gamma s_k r_k - \gamma_f \frac{1}{2} s_k^2 - \gamma_p \frac{1}{2} r_k^2\right) - \frac{1}{2} v_j^\top v_j - \frac{1}{2} w_j^\top w_j \\
&- \sum_{k=1}^{N} \eta_j^k \left(s_k - v_j^\top \varphi_f(\bar{p}_{k+d}^d)\bar{z}_{k+d}^d\right) \\
&- \sum_{k=1}^{N} \kappa_j^k \left(r_k - w_j^\top \varphi_p(\bar{p}_k^d)\bar{z}_k^d\right),
\end{aligned} \tag{5.15}$$

where $\eta_j = \left[\eta_j^1 \cdots \eta_j^N\right]^\top \in \mathbb{R}^N$ and $\kappa_j = \left[\kappa_j^1 \cdots \kappa_j^N\right]^\top \in \mathbb{R}^N$ are dual Lagrange multipliers. The dual variables η_j and κ_j can be obtained via Karush-Kuhn-Tucker (KKT) conditions, i.e., by setting the gradients of the Lagrangian w.r.t. primal and dual variables $\frac{\partial \mathcal{L}}{\partial v_j}, \frac{\partial \mathcal{L}}{\partial w_j}, \frac{\partial \mathcal{L}}{\partial s_k}, \frac{\partial \mathcal{L}}{\partial r_k}, \frac{\partial \mathcal{L}}{\partial \eta_j^k}, \frac{\partial \mathcal{L}}{\partial \kappa_j^k}$,

to zero. The KKT optimality conditions are given as follows,

$$\frac{\partial \mathcal{L}}{\partial v_j} = 0 \rightarrow v_j = \sum_{k=1}^{N} \eta_j^k \varphi_f(\bar{p}_{k+d}^d) \bar{z}_{k+d}^d, \tag{5.16a}$$

$$\frac{\partial \mathcal{L}}{\partial w_j} = 0 \rightarrow w_j = \sum_{k=1}^{N} \kappa_j^k \varphi_p(\bar{p}_k^d) \bar{z}_k^d, \tag{5.16b}$$

$$\frac{\partial \mathcal{L}}{\partial s_k} = 0 \rightarrow \gamma r_k = \gamma_f s_k + \eta_j^k, \tag{5.16c}$$

$$\frac{\partial \mathcal{L}}{\partial r_k} = 0 \rightarrow \gamma s_k = \gamma_p r_k + \kappa_j^k, \tag{5.16d}$$

$$\frac{\partial \mathcal{L}}{\partial \eta_j^k} = 0 \rightarrow s_k = v_j^\top \varphi_f(\bar{p}_{k+d}^d) \bar{z}_{k+d}^d, \tag{5.16e}$$

$$\frac{\partial \mathcal{L}}{\partial \kappa_j^k} = 0 \rightarrow r_k = w_j^\top \varphi_p(\bar{p}_k^d) \bar{z}_k^d, \tag{5.16f}$$

Through simple algebraic manipulations, the primal variables can be eliminated based on the KKT conditions. The optimal dual variables satisfy the following generalized eigenvalue problem,

$$\begin{bmatrix} 0 & K_{\text{pp}} \\ K_{\text{ff}} & 0 \end{bmatrix} \begin{bmatrix} \eta_j \\ \kappa_j \end{bmatrix} = \lambda_j \begin{bmatrix} \gamma_f K_{\text{ff}} + I & 0 \\ 0 & \gamma_p K_{\text{pp}} + I \end{bmatrix} \begin{bmatrix} \eta_j \\ \kappa_j \end{bmatrix}, \tag{5.17}$$

where $\lambda_j = 1/\gamma$, and $K_{\text{pp}} = \Phi_p \Phi_p^\top$ and $K_{\text{ff}} = \Phi_f \Phi_f^\top$ are kernel matrices which define the inner product in the feature space. The elements of the kernel matrices are given as

$$[K_{\text{ff}}]_{l,m} = (\bar{z}_{l+d}^d)^\top \underbrace{\varphi_f^\top(\bar{p}_{l+d}^d)\varphi_f(\bar{p}_{m+d}^d)}_{\bar{k}(\bar{p}_{l+d}^d, \bar{p}_{m+d}^d)} \bar{z}_{m+d}^d, \tag{5.18a}$$

$$[K_{\text{pp}}]_{l,m} = (\bar{z}_l^d)^\top \underbrace{\varphi_p^\top(\bar{p}_l^d)\varphi_p(\bar{p}_m^d)}_{\bar{k}(\bar{p}_l^d, \bar{p}_m^d)} \bar{z}_m^d, \tag{5.18b}$$

In (5.18), the function $\bar{k}(\cdot, \cdot)$ is a positive definite kernel defining the inner products $\varphi_f^\top(\bar{p}_{l+d}^d)\varphi_f(\bar{p}_{m+d}^d)$ and $\varphi_p^\top(\bar{p}_l^d)\varphi_p(\bar{p}_m^d)$. Definition of the kernel instead of feature maps $\varphi_p(\bar{p}_k^d)$ and $\varphi_f(\bar{p}_{k+d}^d)$ is called the *kernel trick* and allows one to formulate the generalized eigenvalue problem (5.17). An example of the kernel is radial basis function (RBF) $\bar{k}(p_i, p_j) =$

$c \exp\left(-\frac{\|p_i - p_j\|^2}{\sigma^2}\right)$ where c and σ are hyper-parameters which are usually tuned via cross validation.

By solving the generalized eigenvalue problem (5.17), the dual variables η_j and κ_j are obtained. From the KKT conditions the primal variables are given as $v_j = \Phi_f^\top \eta_j$ and $w_j = \Phi_p^\top \kappa_j$.

The estimate of the state sequence $\hat{x}(k)$ for the LPV model (5.3) compatible with the dataset $\mathcal{D}$ is obtained as follows. We parametrize the j-th component of the estimated state vector $\hat{x}(k)$ as

$$[\hat{x}(k)]_j = v_j^\top \varphi_f(\bar{p}^d_{k+d}) \bar{z}^d_{k+d}$$

Substituting the primal variable $v_j = \Phi_f^\top \eta_j$, and representing the inner product $\varphi_f^\top(\cdot)\varphi_f(\cdot)$ in terms of the kernel function $\bar{k}(\cdot,\cdot)$, we obtain

$$[\hat{x}(k)]_j = \eta_j^\top \begin{bmatrix} \bar{z}^{d\top}_{1+d} \bar{k}(\bar{p}^d_{1+d}, \bar{p}^d_{k+d}) \\ \vdots \\ \bar{z}^{d\top}_{N+d} \bar{k}(\bar{p}^d_{N+d}, \bar{p}^d_{k+d}) \end{bmatrix} \bar{z}^d_{k+d}. \tag{5.19}$$

Similarly, for $w_j = \Phi_p^\top \kappa_j$ the estimate of the j-th component of the state at time k is given by

$$[\hat{x}(k)]_j = \kappa_j^\top \begin{bmatrix} \bar{z}^{d\top}_1 \bar{k}(\bar{p}^d_1, \bar{p}^d_k) \\ \vdots \\ \bar{z}^{d\top}_N \bar{k}(\bar{p}^d_N, \bar{p}^d_k) \end{bmatrix} \bar{z}^d_k. \tag{5.20}$$

Remark: Note that, the eigenvalue problem (5.17) has $2N$ different solutions η_j, κ_j for $j = 1, \ldots, N$. The dual solutions $\eta_j \in \mathbb{R}^N$ and $\kappa_j \in \mathbb{R}^N$ of the generalized eigenvalue problem (5.17) can be computed using following economical *singular value decomposition* (SVD) [81]

$$\left[\begin{array}{c|c} \gamma_f K_{\text{ff}} + I & 0 \\ \hline 0 & \gamma_p K_{\text{pp}} + I \end{array}\right]^{-1} \left[\begin{array}{c|c} 0 & K_{\text{pp}} \\ \hline K_{\text{ff}} & 0 \end{array}\right] = U \Sigma \left[\begin{array}{c} V_1 \\ \hline V_2 \end{array}\right]^\top \tag{5.21}$$

where the dual solutions can be obtained as $\eta_j = [V_1]_j$ and $\kappa_j = [V_2]_j$ with $[\cdot]_j$ denoting the j-th column of the matrix. The dimension of the state $\hat{x}$ can be chosen by considering only those $[\hat{x}(k)]_j$ which correspond to the $\hat{n}_{\text{x}}$ most significant singular values contained in Σ.

We remark that the estimated state sequence $\hat{x}(k)$, compatible with the data $\mathcal{D}$ is estimated up to a state transformation $T : \mathbb{R}^{dn_p} \to \mathbb{R}^{\hat{n}_{\text{x}} \times n_x}$ which can have dynamic dependence on scheduling variable $p(k), \ldots, p(k+d-1)$. Therefore, $\hat{x}(k)$ is estimated in a state-space basis different from the basis of $x(k)$.

5.3.2 Estimation of latent variable $z(k)$ and LTI model parameters

Once the state sequence $\hat{X} = \{\hat{x}(k)\}_{k=1}^{N}$ is estimated, the latent variables $Z = \{z(k)\}_{k=1}^{N}$ and the unknown matrices $\Theta = \{A, B_1, \ldots, D_{22}\}$ of the LTI model (5.1) can be obtained (up to a similarity transformation) by minimizing the following nonlinear least-squares cost

$$\begin{aligned}
&\mathcal{J}(Z, \Theta) = \\
&\sum_{k=1}^{N-1} \|\hat{x}(k+1) - (A\hat{x}(k) + B_1\Delta(p_k)z(k) + B_2u(k))\|_{Q_x}^2 \\
&+ \sum_{k=1}^{N} \|z(k) - (C_1\hat{x}(k) + D_{11}\Delta(p_k)z(k) + D_{12}u(k))\|_{Q_z}^2 \\
&+ \sum_{k=1}^{N} \|y(k) - (C_2\hat{x}(k) + D_{21}\Delta(p_k)z(k) + D_{22}u(k))\|_{Q_y}^2 ,
\end{aligned} \tag{5.22}$$

where $Q_x, Q_z, Q_y \succ 0$ are positive definite weighting matrices. The cost function is minimized w.r.t. $\{Z, \Theta\}$ using the coordinate descent approach described in Algorithm 3 under the assumption that $\Delta(p(k))$ is known.

Note that, the solution at Step 1.1 and 1.2 of the Algorithm 3 can be computed analytically through linear least-squares.

LPV model with affine dependence on $\Delta(p(k))$

We consider a special case of the LFR model (5.1) with the following assumptions:

- $D_{11} = 0$, which corresponds to LPV models with affine dependence on $\Delta(p(k))$,
- $\Delta(p(k)) = \varphi(p(k))I$, i.e., the feedback block Δ has a diagonal structure with known real-valued basis functions $\varphi : \mathbb{P} \to \mathbb{R}$.

As $D_{11} = 0$, we have

$$z(k) = C_1x(k) + D_{12}u(k). \tag{5.23}$$

Algorithm 3 Coordinate descent for the estimation of latent variables z and model parameter matrices Θ

Input: training dataset $\mathcal{D} = \{u(k), y(k), p(k)\}_{k=1}^{N}$; estimated state sequence $\hat{X} = \{\hat{x}(k)\}_{k=1}^{N}$; tolerance ϵ, maximum number of iteration $n_{\max}$; initial guess Z^0

1. **Iterate for** $n = 1, \ldots$
   1.1 $\Theta^n \leftarrow \operatorname{argmin}_{\Theta} \quad \mathcal{J}(Z^{n-1}, \Theta)$
   1.2 $Z^n \leftarrow \operatorname{argmin}_{Z} \quad \mathcal{J}(Z, \Theta^n)$
2. **Until** $\|Z^n - Z^{n-1}\| \leq \epsilon$ or $n = n_{\max}$

Output: Estimated matrices Θ^n

By substitutng (5.23) into (5.1) we obtain the following state space model,

$$\begin{aligned} x(k+1) =& Ax(k) + B_1\Delta(p(k))C_1x(k) \\ &+ B_1\Delta(p(k))D_{12}u(k) + B_2u(k), \qquad (5.24a) \\ y(k) =& C_2x(k) + D_{21}\Delta(p(k))C_1x(k) \\ &+ D_{21}\Delta(p(k))D_{12}u(k) + D_{22}u(k), \qquad (5.24b) \end{aligned}$$

Using the assumption $\Delta(p(k)) = \varphi(p(k))I$, the unknown matrices Θ in (5.24), can be solved in a least-squares sense by minimizing the cost

$$\mathcal{J}(\Theta_1, \Theta_2) = \sum_{k=1}^{N-1} \left\| \begin{bmatrix} \hat{x}(k+1) \\ y(k) \end{bmatrix} - \underbrace{[\Theta_1 \ \Theta_2]}_{\Theta} \begin{bmatrix} 1 \\ \varphi(p(k)) \end{bmatrix} \otimes \begin{bmatrix} \hat{x}(k) \\ u(k) \end{bmatrix} \right\|_2^2, \qquad (5.25)$$

where

$$\Theta_1 = \begin{bmatrix} A & B_2 \\ C_2 & D_{22} \end{bmatrix},$$
$$\Theta_2 = \begin{bmatrix} B_1C_1 & B_1D_{12} \\ D_{21}C_1 & D_{21}D_{12} \end{bmatrix} = \begin{bmatrix} B_1 \\ D_{21} \end{bmatrix} \begin{bmatrix} C_1 & D_{12} \end{bmatrix}.$$

Once the least-squares cost $\mathcal{J}$ in (5.25) is minimized w.r.t. Θ_1 and Θ_2, the matrices $\{A, B_2, C_2, D_{22}\}$ are obtained by appropriate partitioning of Θ_1. The matrices $\{B_1, D_{21}, C_1, D_{12}\}$ are reconstructed by economic singular value decomposition of Θ_2. Specifically, let U, Ξ and V be the matrices obtained from the economic SVD of Θ_2, i.e. $\Theta_2 = U\Xi V^\top$ which gives

$$\begin{bmatrix} B_1 \\ D_{21} \end{bmatrix} = U\Xi^{1/2}, \quad \begin{bmatrix} C_1 & D_{12} \end{bmatrix} = \Xi^{1/2}V^\top. \tag{5.26}$$

The individual matrices are obtained by appropriate partitioning using the estimated state dimension $\hat{n}_\mathrm{x}$ and known dimensions of inputs n_u, outputs n_y and $\Delta(p)$.

5.4 Numerical examples

In this section, the effectiveness of the proposed method is demonstrated via simulation examples. The output samples $y(k)$ used in the training phase are corrupted by an additive zero-mean white noise $e(k)$ with Gaussian distribution. The effect of the noise on the output is quantified in terms of the *Signal-to-Noise Ratio* (SNR) defined as

$$\mathrm{SNR} = 10 \log \frac{\sum_{k=1}^{N} \left(y(k) - e(k)\right)^2}{\sum_{k=1}^{N} \left(e(k)\right)^2}.$$

The quality of the estimated model is assessed on a noise-free validation data of length N_val via *Best Fit Rate* (BFR) and *Variance Accounted For* (VAF) criterion defined as

$$\mathrm{BFR} = \max\left\{1 - \sqrt{\frac{\sum_{k=1}^{N_\mathrm{val}} \left(y(k) - \hat{y}(k)\right)^2}{\sum_{k=1}^{N_\mathrm{val}} \left(y(k) - \bar{y}\right)^2}}, 0\right\} \times 100\%,$$

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

with $\hat{y}$ being the simulated model output and $\bar{y}$ being the sample mean of the output over the validation set. The operator $\mathrm{var}(\cdot)$ denotes the variance of its argument.

All computations are carried out on i5 1.7GHz Intel core processor with 4 GB of RAM, running MATLAB R2016b.

Example 1

The LPV system (5.1) is used for data generation with forward LTI block

$$\left[\begin{array}{c|cc} A & B_1 & B_2 \\ \hline C_1 & D_{11} & D_{12} \\ C_2 & D_{21} & D_{22} \end{array}\right] = \left[\begin{array}{cc|cc} 0 & 1 & 0 & 1.073 \\ -0.1 & 0.7 & 0.816 & 1.075 \\ \hline 0.524 & -0.625 & -0.5 & 0.5 \\ 0.443 & 0.060 & 0.5 & 0.5 \end{array}\right], \tag{5.28}$$

and the feedback path with scalar scheduling variable represented by

$$w(k) = p(k)z(k).$$

Training and validation datasets of length $N = 400$ and $N_{\text{val}} = 400$, respectively, are generated by exciting system (5.1) with input u being a white-noise process with uniform distribution $\mathcal{U}(-1, 1)$. The standard deviation of the noise $e(k)$ corrupting the training output $y(k)$ is 0.05, which corresponds to an SNR equal to 21 dB. The scheduling signal $p(k)$ is given by

$$p(k) = 0.5\sin(k) + \delta(k),$$

where $\delta(k)$ is a random variable with uniform distribution $\mathcal{U}(-0.5, 0.5)$. In the first stage, the KCCA algorithm is run to estimate the state sequence $\hat{x}(k)$, using RBF kernels $\bar{k}(p_i, p_j) = c\exp\left(-\frac{\|p_i - p_j\|^2}{\sigma^2}\right)$ with parameters $c = 2$ and $\sigma = 10.5$ to construct kernel matrices K_{pp} and K_{ff} in (5.17). The LS-SVM regularization parameters are chosen as $\gamma_p = \gamma_f = 500$ and the past and future window length is $d = 3$. These hyper-parameters are chosen through cross-validation.

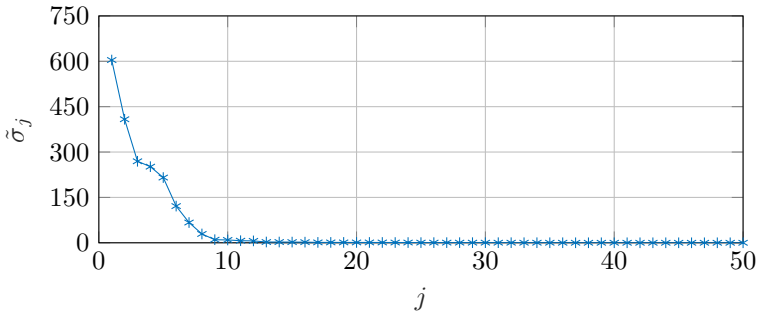

Figure 21: Singular values of the SVD problem (5.21)

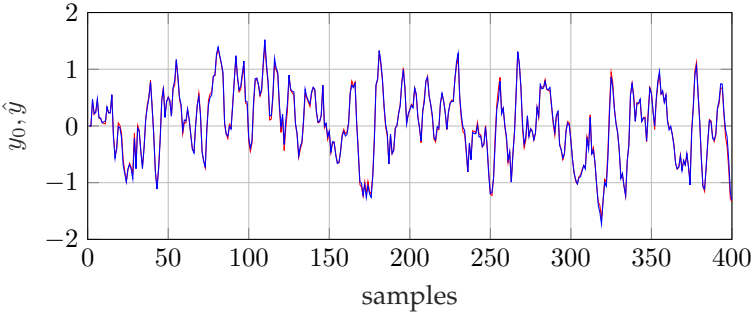

Figure 22: Example 1. True (red) vs estimated (blue) output

Table 16: Example 1: Best Fit Rate (BFR) and Variance Accounted For (VAF) on a noise-free validation data.

BFR	92.71 %
VAF	99.74 %

The dimension of the state is determined by solving the SVD problem (5.21). The first 50 singular values are shown in Figure 21. We observe that there is a significant gap between the first two singular values and rest of them. Based on this observation the SVD is truncated to the first two components. In other words, the selected state dimension is $n_x = 2$. The total computation time to construct the state sequence $\hat{x}(k)$ is 8.7 sec. This includes the time required to solve the generalized eigenvalue problem (5.17) and to obtain $\hat{x}(k)$ based on (5.19).

Using the estimated state $\hat{x}(k)$ the cost function $\mathcal{J}(Z, \theta)$ in (5.22) is optimized using a coordinate descent (Algorithm 3) with $Q_x = 100$, $Q_y = 10$ and $Q_z = 10$. The algorithm is run for $n = 250$ iterations, where each iteration takes around 0.3 sec. The performance of the proposed approach is evaluated using a noise-free validation dataset.

The true and the estimated outputs are shown in Figure 22 and the BFR and VAF criterion are reported in Table 16. The obtained results show a good match between system and model output.

Table 17: Best Fit Rate (BFR) and Variance Accounted For (VAF) on validation data.

BFR	96.05 %
VAF	99.84 %

Example 2

We consider again the data generating system (5.28) but now with $D_{11} = 0$ (instead of $D_{11} = -0.5$), which corresponds to an LPV model with affine dependence on $\Delta(p(k))$.

The state sequence is first estimated using KCCA with the same hyperparameters used in the Example 1. Then, the least-square problem (5.25) is solved followed by singular value decomposition as in (5.26) to compute the estimates of the unknown model matrices. The total computation time to solve the least squares problem followed by SVD is 1.01 sec. The estimation results in terms of BFR and VAF criterion (computed w.r.t. noise free validation data) are reported in Table 17, for an SNR of 21 dB on training data. The results show that an accurate estimate of the output of the true system is obtained with high computational efficiency.

5.5 Conclusions

In this chapter, we have presented a method to identify LPV models in a linear fractional representation. The proposed two stage approach consists of estimation of the state sequence using canonical correlation analysis between past and future data followed by the estimation of latent variable sequence and unknown model parameter matrices by solving a nonlinear least squares problem using coordinate descent algorithm. The proposed method is applicable for MIMO LPV models defined via LFRs with multi-dimensional scheduling signal. Current research activities are focused on the development of alternative algorithms for latent variable estimation and on the application of the presented KCCA based approach to the identification of other model classes such as switched models.

5.6 Appendix

In this appendix, we review the statistical tools of canonical correlations which are used in the chapter for state estimation of LPV models.

Canonical correlation analysis

Canonical Correlation Analysis (CCA) is a statistical method used to determine linear relation between several variables. Given two sets of variables, $u \in \mathbb{R}^{n_u}$ and $y \in \mathbb{R}^{n_y}$ with N samples collected in matrices $U \in \mathbb{R}^{N \times n_u}$ and $Y \in \mathbb{R}^{N \times n_y}$, CCA finds two vectors $v_j \in \mathbb{R}^{n_u}$ and $w_j \in \mathbb{R}^{n_y}$ to maximize the correlation between the so called *variates* Uv_j and Yw_j, which leads to following optimization problem [93; 106],

$$\max_{v_j, w_j} v_j^\top U^\top Y w_j \quad \text{s.t. } v_j^\top U^\top U v_j = 1,\ w_j^\top Y^\top Y w_j = 1,$$

or equivalently,

$$\min_{v_j, w_j} \|Uv_j - Yw_j\|_2 \ \text{ s.t. } v_j^\top U^\top U v_j = 1,\ w_j^\top Y^\top Y w_j = 1.$$

Kernel canonical correlation analysis

Kernel CCA (KCCA) is a nonlinear extension of the classical CCA where data is mapped into high dimensional feature space where classical CCA is applied. In principle, neither the feature maps nor their dimensions are a-priory specified.

Let $\phi_u : \mathbb{R}^{n_u} \to \mathbb{R}^{d_u}$ denote the feature map and let Φ_u be

$$\Phi_u = \begin{bmatrix} \phi_u(u_1) & \phi_u(u_2) & \cdots & \phi_u(u_N) \end{bmatrix}^\top.$$

The matrix Φ_y is defied similarly using feature maps $\phi_y : \mathbb{R}^{n_y} \to \mathbb{R}^{d_y}$ of the variable y. The non-linear extension of the CCA problem is,

$$\begin{aligned} &\min_{v_j, w_j} v_j^\top \Phi_u^\top \Phi_y w_j \\ &\text{s.t. } v_j^\top \Phi_u^\top \Phi_u v_j = 1,\ w_j^\top \Phi_y^\top \Phi_y w_j = 1 \end{aligned} \tag{5.29}$$

The KCCA problem (5.29) can be formulated using least-square support vector machine (LS-SVM) approach [93; 106] and can be solved with

primal-dual optimization. The primal problem in LS-SVM is given by,

$$\begin{aligned}
&\max_{v_j, w_j} \ \mathcal{J}(v_j, w_j, e, r) = \\
&\sum_{i=1}^{N} \left(\gamma e_i r_i - \nu_u \frac{1}{2} e_i^2 - \nu_y \frac{1}{2} r_i^2 \right) - \frac{1}{2} v_j^\top v_j - \frac{1}{2} w_j^\top w_j \\
&\text{s.t. } \ e_i = v_j^\top \phi_u(u_i), \ r_i = w_j^\top \phi_y(y_i), \ \ i = 1, \ldots, N,
\end{aligned} \tag{5.30}$$

and the associated Lagrangian with a and b as dual variables,

$$\begin{aligned}
\mathcal{L}(v_j, w_j, e, r, a, b) = \mathcal{J}(v_j, w_j, e, r) \\
- \sum_{i=1}^{N} a_i \left(e_i - v_j^\top \phi_u(u_i) \right) \\
- \sum_{i=1}^{N} b_i \left(r_i - w_j^\top \phi_y(y_i) \right).
\end{aligned} \tag{5.31}$$

The KKT optimality conditions for the corresponding dual problem leads to the following generalized eigenvalue problem,

$$\begin{aligned}
K_{yy} b &= \frac{1}{\gamma} (\nu_u K_{uu} + I) a, \\
K_{uu} a &= \frac{1}{\gamma} (\nu_y K_{yy} + I) b,
\end{aligned}$$

where, $K_{uu} = \Phi_u^\top \Phi_u$ and $K_{yy} = \Phi_y^\top \Phi_y$ are kernel gram matrices characterizing the inner products (i.e., $\phi_y^\top(\cdot)\phi_y(\cdot)$ and $\phi_u^\top(\cdot)\phi_u(\cdot)$) in the feature space, ν_u, ν_y are regularization hyper-parameters and a and b are the dual vectors associated with the LS-SVM problem.

Chapter 6

Conclusions and future directions

In this thesis, we developed a set of data-driven modeling tools for the identification of *Linear Parameter-Varying* models. By virtue of the scheduling signal, LPV model class provides a framework for describing the nonlinear and time-varying system dynamics accurately but at the same time retaining the simplicity of the LTI models by preserving the linearity between input and output maps. Thus, the LPV paradigm is of significant practical importance for modeling and control of real world systems.

The identification methods developed in this thesis focused on some of the important research questions pertaining to LPV models, for eg., model structure selection, identification with noisy scheduling signals, identification from closed-loop data, identification of LPV models via PieceWise Affine (PWA) regression and identification of LPV models with linear fractional representations. The main contributions and possible future research directions are outlined in this chapter.

6.1 Main contributions

In Chapter 2, two approaches are presented for data-driven model order selection of LPV input-output models. In the first approach, the non-parametric framework of least squares support vector machines (LS-SVM) is extended for model order selection. The proposed regularized

LS-SVM method is computationally efficient which is also applicable to the models having multi-dimensional scheduling parameters. In this framework, the LPV model coefficients are characterized via kernel functions which obviate the need to specify the basis functions a-priory. The model order selection is then performed via convex optimization enforcing sparsity in the over-parameterized model structure.

In the second approach, conventional *LASSO* method is appropriated adapted for model order selection of LPV models when the measurements of the scheduling parameters are noise corrupted. In this method, a bias-corrected cost function is proposed which gives an asymptotic bias-free criterion in order to obtain a consistent parameter estimate along with the correct model order. The bias-corrected cost is also used as a criterion to tune the LASSO hyper-parameters. The results of this chapter are presented in [60; 61].

In Chapter 3, a bias-correction approach is presented for the identification of LPV input-output plant models from closed-loop data. The proposed method provides a consistent estimate of the model parameters by correcting the bias in the least-squares estimates stemming from the correlation between output noise and plant inputs. The bias-correction framework is extended further to deal with the case of noisy measurements of the scheduling signals. In this chapter, the expressions to characterize the bias-eliminating matrices are derived and proofs of consistency of the model parameter estimates are provided. Furthermore, with the presented example of nonlinear two-tank system, it is observed that the proposed method performs well even when the underlying data generating system does not belong to the selected model class. The results of this chapter appeared in the paper [62].

In Chapter 4, as an alternative to the conventional parametric and non-parametric methods, the problem of LPV identification is recast as a *PieceWise Affine* (PWA) regression problem. This framework introduces more flexibility in choosing the model structure and approximating the nonlinear LPV model coefficients by exploiting the universal approximation property of PWA maps. A regularized moving-horizon PWA regression algorithm is proposed based on mixed-integer programming.

The framework of this algorithm is applied to a real world problem of *energy disaggregation* using benchmark dataset. In energy disaggregation problem, the power consumption patterns of individual devices are estimated by using only the measurements of aggregated power. In the

proposed method, the power consumption of each device is described by PWA autoregressive models. The active devices contributing to the instantaneous power are determined by solving a binary quadratic programming problem. It is shown that the proposed disaggregation algorithm using dynamic PWA autoregressive models outperforms the approach relying on static models. The methods presented in this chapter are based on the contributions [58; 59; 64].

In Chapter 5, a two-stage approach is proposed for the identification of LPV models with linear fractional representation (LFR). In the first stage, the kernelized canonical correlation analysis (KCCA) is used to estimate the state sequence of the LPV model. The state of a system is a minimal interface between past and future data samples. Using this fact, the correlation between the projections of past and future data is maximized in order to obtain an estimate of the state sequence. In the second stage, a nonlinear least squares problem is solved to obtain the latent variable sequence charactering the LFR form as well as the unknown model matrices by using the estimated state sequence. A coordinate descent algorithm is employed to solve the nonlinear least squares problem. For an LFR form having affine dependence of model parameters on the feedback block, the unknown matrices are estimated by solving ordinary least-squares followed by singular value decomposition. The results of this chapter are based on the work [63].

6.2 Future research directions

In this section we outline the possible research directions to extend the contributions presented in the thesis.

- **Model order selection**:
  The non-parametric regularized LS-SVM approach presented in Chapter 2 can be further extended for case of noise corrupted scheduling signal measurements by combining it with the parametric bias-correction approach presented in the same chapter. In this case, the advantages of both the methods can be exploited, namely, data-driven estimation of model coefficients without a-prior parameterization in terms of basis functions, and obtaining a consistent estimates in the presence of scheduling signal noise. The main challenge is to explicitly characterize the bias-correcting term for

the nonlinear kernel functions.

- **Closed-loop identification**:
  The closed-loop identification method proposed in Chapter 3 can be extended to models having a more general noise structure, for e.g., Box- Jenkins noise models. In this case, one possible way to handle the colored noise corrupting the output is to introduce instrumental variables in the closed-loop setting. Furthermore, the Gaussian assumption on the scheduling variable noise can be relaxed by using moment generating functions for characterizing the probability distribution of the noise. Moreover, as stated before, the parametric framework can be replaced with non-parametric approaches characterizing LPV model coefficients in terms of kernel functions.

- **Piecewise affine regression for LPV identification**:
  The PWA regression framework based on integer programing for the identification of LPV models presented in Chapter 4 can be further extended to include data-driven model order selection. Moreover, efficient algorithms can be investigated for solving the mixed-integer programming problems which are suitable for real time embedded implementation.

- **Identification of linear fractional representation**:
  The method presented in Chapter 5 assumes that the structure of the feedback block (which is a function of the scheduling variables) in the LFR form is completely known. As a future research direction, the method can be extended to relax this assumption by estimating the feedback block with non-parametric methods. Further investigations can be made to develop an approach which gives consistent estimates in presence of the noise-corrupted scheduling measurements.

- **Modeling of nonlinear system:**
  One of the important research questions which is not investigated in this thesis is the choice of the scheduling variables in order to describe the nonlinear dynamics with LPV models. The choice of scheduling variables is very critical which should be a part of model structure selection step. Often, some of the internal state variables of the nonlinear system are chosen as scheduling parameters which results in a *quasi-LPV* representation. Dedicated methods need to be developed for the identification of quasi-LPV models.

References

[1] H. Abbas, M. Ali, and H. Werner. Linear recurrent neural network for open- and closed-loop consistent identification of LPV models. In *Proc. of the 49th IEEE Conference on Decision and Control*, pages 6851–6856, Atlanta, USA, 2010. 76

[2] H. Abbas and H. Werner. An instrumental variable technique for open-loop and closed-loop identification of input-output LPV models. In *Proc. of the European Control Conference*, pages 2646–2651, Budapest, 2009. 53

[3] A. Ali, M. Ali, and H. Werner. Indirect closed-loop identification of input-output LPV models: A pre-flitering approach. In *Proc. of the 18th IFAC World Congress*, pages 2606–2611, Milano, Italy, 2011. 53

[4] P. Apkarian, P. Gahinet, and G. Becker. Self-scheduled H_∞ control of linear parameter-varying systems: a design example. *Automatica*, 31(9):1251–1261, 1995. 4, 68

[5] F. R. Bach and M. I. Jordan. Kernel independent component analysis. *Journal of Machine Learning Research (JMLR)*, 3:1–49, 2003. 138

[6] A. A. Bachnas, R. Tóth, J. Ludlage, and A. Mesbah. A review on data-driven linear parameter-varying modeling approaches: A high-purity distillation column case study. *Journal of Process Control*, 24(4):272–285, 2014. 4

[7] L. Bako. Identification of switched linear systems via sparse optimization. *Automatica*, 47(4):668–677, 2011. 94

[8] L. Bako, K. Boukharouba, E. Duviella, and S. Lecoeuche. A recursive identification algorithm for switched linear/affine models. *Nonlinear Analysis: Hybrid Systems*, 5(2):242–253, 2011. 94

[9] B. A. Bamieh and L. Giarré. Identification of linear parameter-varying models. *International Journal of Robust Nonlinear Control*, 12(9):841–853, 2002. 12, 132

[10] A. Bemporad, D. Bernardini, and P. Patrinos. A convex feasibility approach to anytime model predictive control. Technical report, IMT Institute for Advanced Studies, Lucca, February 2015. http://arxiv.org/abs/1502.07974. 107

[11] A. Bemporad, G. Ferrari-Trecate, and M. Morari. Observability and controllability of piecewise affine and hybrid systems. *IEEE Transactions on Automatic Control*, 45(10):1864–1876, 2000. 94

[12] A. Bemporad, A. Garulli, S. Paoletti, and A. Vicino. A bounded-error approach to piecewise affine system identification. *IEEE Transactions on Automatic Control*, 50(10):1567–1580, 2005. 3, 94

[13] A. Bemporad and M. Morari. Control of systems integrating logic, dynamics, and constraints. *Automatica*, 35(3):407–427, 1999. 94

[14] K.P. Bennett and O.L. Mangasarian. Multicategory discrimination via linear programming. *Optimization Methods and Software*, 3:27–39, 1994. 106, 107

[15] S. Boonto and H. Werner. Closed-loop system identification of LPV input-output models - application to an arm-driven inverted pendulum. In *Proc. of the 47th IEEE Conference on Decision and Control*, pages 2606–2611, Cancun, 2008. 53

[16] L. Breiman. Hinging hyperplanes for regression, classification, and function approximation. *IEEE Transactions on Information Theory*, 39(3):999–1013, 1993. 94

[17] L. Breiman. Better subset regression using the nonnegative garrote. *Technometrics*, 37(4):373–384, 1995. 13

[18] V. Breschi, A. Bemporad, and D. Piga. Identification of hybrid and linear parameter varying models via recursive piecewise affine regression and discrimination. In *Proc. of the 15th European Control Conference*, pages 2632–2637, 2016. 94, 111

[19] V. Breschi, D. Piga, and A. Bemporad. Piecewise affine regression via recursive multiple least squares and multicategory discrimination. *Automatica*, 73:155–162, 2016. 94, 95, 106, 107, 108, 116

[20] M. Butcher, A. Karimi, and R. Longchamp. On the consistancy of certain identification methods for linear parameter varying systems. In *Proc. of the 17th IFAC World Congress*, pages 4018–4023, Seoul, South Korea, 2008. 15

[21] T. R. Camier, S. Giroux, B. Bouchard, and A. Bouzouane. Designing a NIALM in smart homes for cognitive assistance. *Procedia Computer Science*, 19:524–532, 2013. 118

[22] V. Cerone, D. Piga, and D. Regruto. Set-membership LPV model identification of vehicle lateral dynamics. *Automatica*, 47(8):1794–1799, 2011. 4, 68

[23] V. Cerone, D. Piga, and D. Regruto. A convex relaxation approach to set-membership identification of LPV systems. *Automatica*, 49(9):2853–2859, 2013. 15

[24] Y. Cheng and M. Sznaier. Identification of LPV systems with LFT parametric dependence via convex optimization. In *Proc. of the 54th IEEE Conference on Decision and Control*, pages 1459–1464, Osaka, Japan, 2015. 133

[25] A. Chiuso. The role of vector autoregressive modeling in predictor-based subspace identification. *Automatica*, 43(6):1034–1048, 2007. 53

[26] A. Cominola, M. Giuliani, D. Piga, A Castelletti, and A. E. Rizzoli. Benefits and challenges of using smart meters for advancing residential water demand modeling and management: a review. *Environmental Modelling & Software*, 72:198–214, 2015. 118

[27] P. Van den Hof. Closed-loop issues in system identification. *Annual reviews in control*, 22:173–186, 1998. 52

[28] R. Dong, L. Ratliff, H. Ohlsson, and S. Sastry. A dynamical systems approach to energy disaggregation. In *Proc. of the 52nd IEEE Conference on Decision and Control*, pages 6335–6340, Florence, Italy, 2013. 118

[29] R. Duijkers, R. Tóth, D. Piga, and V. Laurain. Shrinking complexity of scheduling dependencies in LS-SVM based LPV system identification. In *Proc. of the 53rd IEEE Conference on Decision and Control*, pages 2561–2566, Los Angeles, California, USA, 2014. IEEE. 14

[30] L. Farinaccio and R. Zmeureanu. Using a pattern recognition approach to disaggregate the total electricity consumption in a house into the major end-uses. *Energy and Buildings*, 30(3):245–259, 1999. 118

[31] F. Felici, J. W. Van Wingerden, and M. Verhaegen. Subspace identification of MIMO LPV systems using a periodic scheduling sequence. *Automatica*, 43:1684–1697, 2007. 4, 12, 132

[32] G. Ferrari-Trecate, M. Muselli, D. Liberati, and M. Morari. A clustering technique for the identification of piecewise affine systems. *Automatica*, 39(2):205–217, 2003. 94

[33] M. Figueiredo, B. Ribeiro, and A.M. de Almeida. On the regularization parameter selection for sparse code learning in electrical source separation. In *Adaptive and Natural Computing Algorithms*, pages 277–286. Springer, 2013. 118

[34] U. Forssell and L. Ljung. Closed-loop identification revisited. *Automatica*, 35(7):1215–1241, 1999. 52

[35] A. Garulli, S. Paoletti, and A. Vicino. A survey on switched and piecewise affine system identification. In *Proc. of the 16th IFAC Symposium on System Identification*, pages 344–355, Brussels, Belgium, 2012. 94

[36] M. Gilson and P. Van den Hof. On the relation between a bias-eliminated least-squares (BELS) and an IV estimator in closed-loop identification. *Automatica*, 37(10):1593–1600, 2001. 54

[37] M. Grant and S. Boyd. CVX: Matlab software for disciplined convex programming, version 2.1. url: `http://cvxr.com/cvx`, 2014. 25

[38] Gurobi Optimization, Inc. *Gurobi Optimizer Reference Manual*, 2014. 123

[39] G.W. Hart. Nonintrusive appliance load monitoring. *Proceedings of the IEEE*, 80(12):1870–1891, 1992. 117

[40] W.P.M.H. Heemels, B. De Schutter, and A. Bemporad. Equivalence of hybrid dynamical models. *Automatica*, 37(7):1085–1091, 2001. 94

[41] M. Hong, T. Söderström, and W. X. Zheng. Accuracy analysis of bias-eliminating least squares estimates for errors-in-variables systems. *Automatica*, 43:1590–1596, 2007. 54

[42] K. Hsu, K. Poolla, and T. Vincent. Identification of structured nonlinear systems. *IEEE Transactions on Automatic Control*, 53(11):2497–2513, 2008. 3

[43] K. Hsu, T. L. Vincent, and K. Poolla. Nonparametric methods for the identification of linear parameter varying systems. In *Proc. of the Int. Symposium on Computer-Aided Control System Design*, pages 846–851, San Antonio, Texas, USA, Sept. 2008. 14

[44] M.J. Johnson and A.S. Willsky. Bayesian nonparametric hidden semi-Markov models. *The Journal of Machine Learning Research*, 14(1):673–701, 2013. 118

[45] A. L. Juloski, S. Weiland, and W.P.M.H. Heemels. A Bayesian approach to identification of hybrid systems. *IEEE Transactions on Automatic Control*, 50(10):1520–1533, 2005. 94

[46] J.Z. Kolter and T. Jaakkola. Approximate inference in additive factorial HMMs with application to energy disaggregation. In *Int. Conference on Artificial Intelligence and Statistics*, pages 1472–1482, 2012. 118

[47] I. D. Landau and A. Karimi. An output error recursive algorithm for unbiased identification in closed loop. *Automatica*, 33(5):933–938, 1997. 53

[48] I. D. Landau and A. Karimi. Recursive algorithms for identification in closed-loop: A unified approach and evaluation. *Automatica*, 33(8):1499–1523, 1997. 53

[49] J. Lataire, E. Louarroudi, R. Pintelon, and Y. Rolain. Benchmark data on a linear time-and parameter-varying system. In *Proc. of the 17th IFAC Symposium on System Identification*, pages 1477–1482, Beijing, China, 2015. 4, 68

[50] F. Lauer. On the complexity of piecewise affine system identification. *Automatica*, 62:148–153, 2015. 94

[51] V. Laurain, M. Gilson, R. Tóth, and H. Garnier. Refined instrumental variable methods for identification of LPV Box–Jenkins models. *Automatica*, 46(6):959–967, 2010. 12, 34, 35, 132

[52] L. H. Lee and K. Poolla. Identification of linear parameter-varying systems via LFTs. In *Proc. of the 35th IEEE Conference on Decision and Control*, pages 1545–1550, Kobe, Japan, 1996. 133

[53] L. H. Lee and K. Poolla. Identification of linear parameter-varying systems using nonlinear programming. *Journal of dynamic systems, measurement and Control*, 121(1):71–78, 1999. 133

[54] I. J. Leonaritis and S. A. Billings. Input-output parametric models for nonlinear systems. *International Journal of Control*, 41:303–344, 1985. 3

[55] L. Ljung. *System identification: theory for the user*. Prentice-Hall Englewood Cliffs, NJ, 1999. 3, 52

[56] S. Makonin, F. Popowich, L. Bartram, B. Gill, and I.V. Bajic. AMPds: a public dataset for load disaggregation and eco-feedback research. In *Electrical Power and Energy Conference (EPEC)*, pages 1–6, 2013. 122

[57] A. Marcos and G.J. Balas. Development of linear-parameter-varying models for aircraft. *Journal of Guidance Control and Dynamics*, 27(2):218–228, 2004. 4

[58] M. Mejari, V. Naik, D. Piga, and A. Bemporad. Regularized moving-horizon PWA regression for LPV system identification. In *Submitted, 18th IFAC Symposium on System Identification*, 2017. 10, 152

[59] M. Mejari, V. Naik, D. Piga, and A. Bemporad. Energy disaggregation using piecewise affine regression and binary quadratic programming. In *Submitted, 57th IEEE Conference on Decision and Control*, 2018. 10, 152

[60] M. Mejari, D. Piga, and A. Bemporad. Regularized least square support vector machines for order and structure selection of LPV-ARX models. In *Proc. of the 15th European Control Conference*, pages 1649–1654, Aalborg, Denmark, 2016. 9, 132, 151

[61] M. Mejari, D. Piga, and A. Bemporad. LPV model order selection from noise-corrupted output and scheduling signal measurements. In *Proc. of the 20th IFAC World Congress*, pages 8355 – 8360, 2017. 9, 151

[62] M. Mejari, D. Piga, and A. Bemporad. A bias-correction method for closed-loop identification of Linear Parameter-Varying systems. *Automatica*, 87:128 – 141, 2018. 9, 132, 151

[63] M. Mejari, D. Piga, Tóth, and A. Bemporad. Identification of linear parameter varying models in linear fractional representations. *Submitted, 57th IEEE Conference on Decision and Control*, 2018. 11, 152

[64] V. Naik, M. Mejari, D. Piga, and A. Bemporad. Regularized moving-horizon piecewise affine regression using mixed-integer quadratic programming. In *Proc. 25th Mediterranean Conference on Control and Automation*, pages 1349–1354, Valletta, Malta, 2017. 10, 152

[65] V. V. Naik and A. Bemporad. Embedded mixed-integer quadratic optimization using accelerated dual gradient projection. In *Proc. of the 20th IFAC World Congress*, pages 10723–10728, Toulouse, France, 2017. 103, 110

[66] H. Nakada, K. Takaba, and T. Katayama. Identification of piecewise affine systems based on statistical clustering technique. *Automatica*, 41(5):905–913, 2005. 94

[67] M. Nemani, R. Ravikanth, and B. A. Bamieh. Identification of linear parametrically varying systems. In *Proc. of the 34th IEEE Conference on Decision and Control*, pages 2990–2995, New Orleans, LA, 1995. 133

[68] B. Ninness. Strong laws of large numbers under weak assumptions with application. *IEEE Transactions on Automatic Control*, 45(11):2117–2122, 2000. 37, 66, 72, 92

[69] C. Novara, F. Ruiz, and M. Milanese. Direct identification of optimal SM-LPV filters and application to vehicle yaw rate estimation. *IEEE Transactions on Control Systems Technology*, 19(1):5–17, 2011. 4

[70] H. Ohlsson and L. Ljung. Identification of switched linear regression models using sum-of-norms regularization. *Automatica*, 49(4):1045–1050, 2013. 94

[71] N. Ozay, C. Lagoa, and M. Sznaier. Set membership identification of switched linear systems with known number of subsystems. *Automatica*, 51:180–191, 2015. 94

[72] A. Packard. Gain scheduling via linear fractional transformations. *Systems & Control Letters*, 22(2):79–92, 1994. 132

[73] S. Paoletti, A. L. Juloski, G. Ferrari-Trecate, and R. Vidal. Identification of hybrid systems a tutorial. *European journal of control*, 13(2):242–260, 2007. 94

[74] O. Parson, S. Ghosh, M. Weal, and A. Rogers. An unsupervised training method for non-intrusive appliance load monitoring. *Artificial Intelligence*, 217:1–19, 2014. 118

[75] P. Patrinos and A. Bemporad. An accelerated dual gradient-projection algorithm for embedded linear model predictive control. *IEEE Transactions on Automatic Control*, 59(1):18–33, 2014. 103

[76] R. K. Pearson. Nonlinear input-output modeling. *Jornal of Process Control*, 5(4):197–211, 1995. 3

[77] D. Piga, A. Cominola, M. Giuliani, A. Castelletti, and A. E. Rizzoli. Sparse optimization for automated energy end use disaggregation. *IEEE Transactions on Control Systems Technology*, 24(3):1044–1051, 2016. 118, 119, 122

[78] D. Piga, P. Cox, R. Tóth, and V. Laurain. LPV system identification under noise corrupted scheduling and output signal observations. *Automatica*, 53:329–338, 2015. 12, 15, 16, 34, 36, 37, 38, 41, 43, 46, 47, 54, 61, 68, 71, 72

[79] D. Piga and R. Tóth. LPV model order selection in an LS-SVM setting. In *Proc. of the 52nd IEEE Conference on Decision and Control*, pages 4128–4133, Florence, Italy, 2013. 13, 14, 15, 25, 27, 29

[80] D. Piga and R. Tóth. A bias-corrected estimator for nonlinear systems with output-error type model structures. *Automatica*, 50(9):2373–2380, 2014. 3, 36, 54

[81] S. Rizvi, J. Mohammadpour, F. Abbasi, and R. Tóth. State-space LPV model identification using kernelized machine learning. *Automatica*, 88:38–47, 2018. 133, 135, 136, 137, 141

[82] C.R. Rojas and H. Hjalmarsson. Sparse estimation based on a validation criterion. In *Proc. of the 50th IEEE Conference on Decision and Control*, pages 2825–2830, Orlando, Florida (USA), 2011. 13

[83] J. Roll, A. Bemporad, and L. Ljung. Identification of piecewise affine systems via mixed-integer programming. *Automatica*, 40(1):37–50, 2004. 94, 95, 116

[84] W. Rugh. *Nonlinear System Theory: The Volterra/Wiener Approach*. Johns Hopkins Univ. Press, Baltimore, 1981. 3

[85] A. G. Ruzzelli, C. Nicolas, A. Schoofs, and G.M.P. O'Hare. Real-time recognition and profiling of appliances through a single electricity sensor. In *IEEE Communications Society Conference on Sensor Mesh and Ad Hoc Communications and Networks (SECON)*, pages 1–9, 2010. 118

[86] C. W. Scherer. Mixed $\mathcal{H}_2/\mathcal{H}_\infty$ control for time-varying and linear parametrically-varying systems. *International Journal of Robust and Nonlinear Control*, 6(9-10):929–952, 1996. 132

[87] J. S. Shamma and M. Athans. Analysis of gain scheduled control for nonlinear plants. *IEEE Transactions on Automatic Control*, 35(8):898–907, 1990. 3

[88] J. Sjöberg, Q. Zhang, L. Ljung, A. Benveniste, B. Delyon, P. Glorennec, H. Hjalmarsson, and A. Juditsky. Nonlinear black-box modeling in system identification: a unified overview. *Automatica*, 31(12):1691–1724, 1995. 3

[89] R. S. Smith and J. Doyle. The two tank experiment: A benchmark control problem. In *Proc. of the American Control Conference*, pages 2026–2031, June 1988. 84

[90] T. Söderström. Errors-in-variables methods in system identification. *Automatica*, 43(6):939–958, 2007. 34

[91] T. Söderström and P. Stoica. *System Identification*. Prentice Hall, 1989. 3, 59

[92] D. Srinivasan, W.S. Ng, and A.C. Liew. Neural-network-based signature recognition for harmonic source identification. *IEEE Transactions on Power Delivery*, 21(1):398–405, 2006. 118

[93] J. Suykens, T. Van Gestel, J. De Brabanter, B. De Moor, and J. Vandewalle. *Least Squares Support Vector Machines*. World Scientific Publishing, Singapore, 2002. 8, 14, 138, 139, 148

[94] K. Suzuki, S. Inagaki, T. Suzuki, H. Nakamura, and K. Ito. Nonintrusive appliance load monitoring based on integer programming. In *SICE Annual Conference, 2008*, pages 2742–2747, 2008. 118

[95] M. Sznaier and C. Mazzarro. An LMI approach to control-oriented identification and model (in)validation of LPV systems. *IEEE Transactions on Automatic Control*, 48(9):1619–1624, 2003. 12, 132

[96] M. Tanelli, D. Ardagna, and Lovera M. LPV model identification in virtualized service center environments. In *15th IFAC Symposium on System Identification*, pages 862 – 867, France, 2009. 4

[97] R. Tibshirani. Regression shrinkage and selection via the lasso. *Journal of the Royal Statistical Society. Series B*, 58(1):267–288, 1996. 13

[98] R. Tóth. *Modeling and identification of linear parameter-varying systems*. Lecture Notes in Control and Information Sciences, Vol. 403, Springer, Heidelberg, 2010. 3, 52, 132

[99] R. Tóth, H. Hjalmarsson, and C. R. Rojas. Order and structural dependence selection of LPV-ARX models revisited. In *Proc. of the 51st IEEE Conference on Decision and Control*, pages 6271–6276, Maui, Hawaii (USA), 2012. 13

[100] R. Tóth, V. Laurain, M. Gilson, and H. Garnier. Instrumental variable scheme for closed-loop LPV model identification. *Automatica*, 48(9):2314 –2320, 2012. 53

[101] R. Tóth, V. Laurain, W. X. Zheng, and K. Poolla. Model structure learning: A support vector machine approach for LPV linear-regression models. In *Proc. of the 50th IEEE Conference on Decision and Control*, pages 3192–3197, Orlando, Florida (USA), 2011. 14, 15, 17, 18

[102] R. Tóth, C. Lyzell, M. Enqvist, P.S.C. Heuberger, and P.M.J. Van den Hof. Order and structural dependence selection of LPV-ARX models using a nonnegative garrote approach. In *Proc. of the 48th IEEE Conference on Decision and Control*, pages 7406–7411, Shanghai, China, 2009. 13

[103] J. W. van Wingerden and M. Verhaegen. Subspace identification of bilinear and LPV systems for open-and closed-loop data. *Automatica*, 45(2):372–381, 2009. 12, 53, 132

[104] V. Vapnik. *Statistical Learning Theory*. Wiley-Interscience, 1998. 21

[105] V. Verdult, M. Lovera, and M. Verhaegen. Identification of linear parameter-varying state-space models with application to helicopter rotor dynamics. *International Journal of Control*, 77(13):1149–1159, 2004. 4

[106] V. Verdult, J. A. K. Suykens, J. Boets, I. Goethals, and B. De Moor. Least squares support vector machines for kernel CCA in nonlinear state-space identification. In *Proc. of the 16th Intl. symp. math. theory of networks and syst*, Leuven, Belgium, 2004. 136, 148

[107] V. Verdult and M. Verhaegen. Kernel methods for subspace identification of multivariable LPV and bilinear systems. *Automatica*, 41:1557–1565, 2005. 12, 132

[108] M. Zeifman and K. Roth. Nonintrusive appliance load monitoring: Review and outlook. *IEEE Transactions on Consumer Electronics*, 57(1):76–84, 2011. 118, 119

[109] W. X. Zheng. A bias correction method for identification of linear dynamic errors-in-variables models. *IEEE Transactions on Automatic Control*, 47(7):1142–1147, 2002. 54

[110] W. X. Zheng and C. Feng. A bias-correction method for indirect identification of closed-loop systems. *Automatica*, 33(8):1499–1523, 1997. 54

[111] K. Zhou and J. C. Doyle. *Essentials of Robust Control*. Prentice-Hall, 1998. 5, 132

[112] K. Zhou, J. C. Doyle, and K. Glover. *Robust and optimal control*. Prentice-Hall, 1996. 5

[113] A. Zoha, A. Gluhak, M.A. Imran, and S. Rajasegarar. Non-intrusive load monitoring approaches for disaggregated energy sensing: A survey. *Sensors*, 12(12):16838–16866, 2012. 118, 119