



**HAL**  
open science

# Realization theory of cyber-physical systems and its application to system identification and model reduction

Petreczky Mihaly

► **To cite this version:**

Petreczky Mihaly. Realization theory of cyber-physical systems and its application to system identification and model reduction. Optimization and Control [math.OC]. Université de Lille, 2023. tel-04144797

**HAL Id: tel-04144797**

**<https://hal.science/tel-04144797v1>**

Submitted on 28 Jun 2023

**HAL** is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

**Université de Lille**  
Ecole Doctorale Mathématiques-Sciences du numérique et  
de leurs interactions  
Unité de recherche CRISAL

Mémoire de recherche  
soumis en vue de l'obtention de  
**l'Habilitation à diriger des Recherches**

Specialité: Automatique, Génie Informatique et  
Traitement du Signal et des Images

par  
MIHÁLY PETRECZKY

Titre:

**Realization theory of cyber-physical systems and its  
application to system identification and model  
reduction**

**(Théorie de la réalisation des systèmes cyber-physiques  
et son application à l'identification des systèmes et à la  
réduction des modèles)**

**Soutenue publiquement le 26 janvier 2023 devant le jury composé de**

Garant:	Laurentiu Hetel	Directeur de recherche CNRS au CRISAL
Rapporteurs:	Jamal Daafouz Antoine Girard Bart De Moor	Professeur à l'Université de Lorraine Directeur de recherche CNRS au L2S Professeur au Katholieke Universiteit Leuven
Examineur:	Michel Verhaegen	Professeur au Technische Universiteit Delft
Président du jury:	Lotfi Belkoura	Professeur à l'Université de Lille



# Contents

<b>1</b>	<b>General introduction</b>	<b>7</b>
1.1	Global research objectives: reliable models for control of complex cyber-physical systems . . . .	7
1.2	Principal research topics: realization theory and its applications . . . . .	10
1.2.1	Realization theory . . . . .	10
1.2.2	Application of realization theory . . . . .	12
1.3	Other research directions not related to realization theory . . . . .	13
1.4	Outline of the manuscript . . . . .	15
<b>I</b>	<b>Selected topics on realization theory: realization theory of LPV and bilinear systems</b>	<b>17</b>
<b>2</b>	<b>Realization theory of LPV systems</b>	<b>23</b>
2.1	Introduction . . . . .	23
2.2	Notation used in the technical part of the manuscript . . . . .	24
2.3	Preliminaries . . . . .	25
2.4	LPV-SSAs and linear switched systems . . . . .	26
2.5	Reachability, observability, minimality . . . . .	27
2.6	Existence of a realization, sub-Markov parameters and Hankel-matrix . . . . .	29
2.7	Ho-Kalman realization algorithms . . . . .	32
2.8	Conclusions . . . . .	35
<b>3</b>	<b>Realization theory of stochastic bilinear systems</b>	<b>37</b>
3.1	Introduction . . . . .	37
3.2	Informal definition of GBSs, problem formulation, preliminaries . . . . .	38
3.3	Informal description of the solution: relationship between GBSs and linear switched systems. . .	41
3.4	Mathematical framework . . . . .	44
3.5	Formal description of the relationship with realization theory of LSS . . . . .	46
3.6	Main results on realization theory of GBSs . . . . .	49
3.6.1	Existence of a realization . . . . .	50
3.6.2	Minimality of GBSs . . . . .	50
3.6.3	Realization and minimization algorithms . . . . .	51
3.6.4	Applications to JMLS . . . . .	52
3.7	Consequences for system identification: weak realizations, GBSs as optimal filters, realization of input-output relation vs. output process . . . . .	53
3.7.1	Interpretations of GBS and their role in system identification . . . . .	53
3.7.2	The stochastic realization problem versus realization of an input-output relation . . . . .	55
3.8	Conclusions and further work . . . . .	57

<b>II</b>	<b>Selected topics on application of realization theory</b>	<b>59</b>
<b>4</b>	<b>Model reduction of linear switched systems</b>	<b>65</b>
4.1	Introduction . . . . .	65
4.2	Definition and notation . . . . .	67
4.3	Model reduction . . . . .	67
4.4	Model reduction by balanced truncation . . . . .	68
4.5	Moment matching . . . . .	70
4.6	Conclusions . . . . .	73
<b>5</b>	<b>Identification of LPV systems</b>	<b>75</b>
5.1	System identification of deterministic LPV-SS systems . . . . .	76
5.1.1	Deterministic subspace identification algorithm LPV-SSA . . . . .	79
5.1.2	Identifiability of LPV-SSA . . . . .	82
5.2	Stochastic subspace identification algorithm . . . . .	85
5.2.1	Stochastic subspace identification of GBSs . . . . .	86
5.2.2	Identification of stochastic LPV-SSA with inputs and its relationship with identification of GBSs and of deterministic LPV-SSA . . . . .	90
5.2.3	Role of realization theory . . . . .	96
5.3	Conclusions and further work . . . . .	96
<b>6</b>	<b>Reverse engineering of network structure for stochastic linear and bilinear systems</b>	<b>97</b>
6.1	Introduction . . . . .	97
6.2	Granger causality: linear case . . . . .	99
6.2.1	Technical preliminaries: linear stochastic realization theory . . . . .	100
6.2.2	Classical Granger causality between two processes . . . . .	103
6.2.3	Conditional Granger causality and sLTI-SS in coordinated form . . . . .	104
6.2.4	Directed acyclic graphs induced by Granger causality relations . . . . .	106
6.2.5	Reverse engineering of the network graph . . . . .	109
6.3	Granger causality: bilinear case . . . . .	110
6.4	Role of realization theory . . . . .	112
6.5	Conclusions and future work . . . . .	112
<b>III</b>	<b>Future research</b>	<b>115</b>
<b>7</b>	<b>Future research directions</b>	<b>117</b>
7.1	Axis A: reliable models for control of cyber-physical systems . . . . .	118
7.1.1	Axis A1: Fundamental research: further research on realization theory and structure theory	120
7.1.2	Axis A2: Methodological research: control-oriented and theoretically sound system identification and model reduction algorithms . . . . .	127
7.1.3	Interplay between fundamental and methodological research . . . . .	133
7.2	Emerging research directions:axes B and C . . . . .	133
7.2.1	Axis B: Reliable AI for cyber-physical systems: using control theory for machine learning	134
7.2.2	Axis C: Reverse engineering of network topology of interconnected systems . . . . .	136
7.2.3	Interplay between research axes A,B,C . . . . .	138
7.3	Conclusions . . . . .	138

<b>IV</b>	<b>General conclusions</b>	<b>141</b>
<b>8</b>	<b>Summary and general conclusions</b>	<b>143</b>
<b>V</b>	<b>CV</b>	<b>147</b>
<b>A</b>	<b>CV</b>	<b>149</b>
A.1	Professional experience . . . . .	149
A.2	Education . . . . .	149
A.3	Supervision of postdocs and PhD students . . . . .	150
A.3.1	Supervision of PhD students . . . . .	150
A.3.2	Supervision of postdocs . . . . .	151
A.4	Scientific collaboration . . . . .	152
A.5	Funded projects . . . . .	153
A.6	Organizational activities and service to the scientific community . . . . .	153
A.7	Teaching experience . . . . .	154
A.8	List of publications which appeared after completing my PhD . . . . .	156
<b>B</b>	<b>Review of realization theory of linear switched systems</b>	<b>165</b>
B.1	Introduction . . . . .	165
B.2	Linear switched systems: basic definition . . . . .	165
B.3	Characterization of minimality by reachability and observability . . . . .	166
B.4	Algebraic and geometric characterization of reachability and observability, minimization algorithm	167
B.5	Existence of a realization, Kalman-Ho algorithm . . . . .	169
B.6	Stability and minimality . . . . .	174



# Chapter 1

## General introduction

In this report I will present some selected results of my research activities. I will start with the description of my global research objectives in Section 1.1. In Section 1.2, I will discuss my research activities related to my main research topic, which is realization theory and its applications to system identification and model reduction. In Section 1.3, I will present my research activities on those topics which fall outside realization theory and its applications. Finally, in Section 1.4 I will describe the structure of the report

### 1.1 Global research objectives: reliable models for control of complex cyber-physical systems

The focus of my research has been the development of reliable models for control, fault detection and prediction of complex cyber-physical systems. Therefore, the topic of my research falls within control theory. *Cyber-physical systems* (abbreviated as *CPS*) [8, 269, 157] are systems combining computer software with physical hardware and they are an integral part of modern technology (e.g., automotive industry, aerospace, smart buildings, smart grids, manufacturing systems, mechatronics, networked control systems). Control, fault detection and prediction of cyber-physical systems have been subject of intensive research. However, much remains to be done when it comes to finding models for such systems.

When describing my research philosophy, for the simplicity of presentation I will talk about modeling for controller synthesis. However the same ideas apply when building models for other purposes, such as fault detection and prediction. To begin with, notice that in order to build provenly correct controllers, mathematical models of the to be controlled plant are used. The model is usually a dynamical system in the mathematical sense, and so is the controller. It is then shown by formal proof that the interconnection of the controller and the model, which is a dynamical system itself with known equations, has the desired properties. As a rule, it can also be shown that these properties will still hold if the plant's model is slightly changed.

The problem with this approach is that it guarantees the correctness of the controller for the model, not for the real plant. Hence, after designing the controller, extensive testing is still required. However, this approach has the advantage that if the designed controller does not work well on the physical plant, then there is a clear way to improve it. Namely, one should use a more accurate model of the plant or improve the controller by making it less sensitive to the modeling error.

That is, the ideal workflow proposed by control theory/engineering methods is as follows:

- **Step 1 (Modeling for control)** Construct a model of the plant.
- **Step 2 (Model-based controller synthesis)** Use the model to compute a controller meeting the control objectives.



- **Step 3 (Testing)** Test the controller on the physical plant or on , if the control objectives are not met, then go back to **Step 1** and construct a more realistic model or a more robust controller.

The workflow suggests the importance of being able to construct good models. Indeed, from a practical point of view, the theoretical guarantees of **Step 2** are meaningful, only if they help to reduce the number and cost of each iteration. In order to achieve this, methods for carrying out the modeling step of **Step 1** are required.

In a nutshell, my research has been centered around the theoretical questions of modeling and its interplay with controller synthesis, i.e., the interplay of **Step 1.** and **Step 2.** The practical relevance of control theory is to a large extent determined by the presence of suitable modeling methods <sup>1</sup>.

For a modeling framework to be suitable for controller synthesis, it should provide a class of mathematical models, algorithms and theory to accomplish the following goals:

- O1: Learning from data: black-box and gray-box system identification.** There should be algorithms for estimating models from measurement data. These methods should allow the use of prior knowledge for constructing models. The problem for learning models is called *system identification* in control theory community.
- O2: Model reduction: simplifying models.** There should be algorithms for simplifying existing models, i.e. replacing them by a simpler model observable behavior of which is close to the original one. The problem of simplifying models is called *model reduction* in control theory community.
- O3: Compatibility with controller synthesis: models suitable for controller, theoretical guarantees, invariance w.r.t. modeling methods.** First of all, both system identification and model reduction methods should result in models which can be used for controller synthesis. Second, there should be theoretical guarantees that a controller calculated based on a model obtained via system identification or model reduction will achieve the desired control objectives for the underlying system. More precisely, for system identification, we would like to have theoretical guarantees that if sufficient amount of measurements are available, then any controller synthesized for the model learned from data will also achieve the control objectives when applied to the true physical system. In other words, for a sufficiently large number of measurements the model learned from data is good enough for controller synthesis. Likewise, for a simplified model obtained from model reduction, we would like to have theoretical guarantees that the controller synthesized based on this model will work for the original model as well. Furthermore, if two system identification or model reduction methods result in two different models with the same observable behavior, we would like to have guarantees that controller synthesized for one of them will work for the other model as well, i.e., for observationally equivalent models it should not matter which one is used for controller synthesis.

The motivation for these requirements is quite intuitive.

Clearly, in the absence of methods for learning models from data or for simplifying overly complex models, modeling becomes too difficult and there is little to be gained from using model based methods instead of just tuning controllers in a ad-hoc fashion. If the models generated by system identification or model reduction are not in a form which is suitable for controller synthesis, then there is little use in building those models.

In the absence of theoretical guarantees for system identification and model reduction, the modeling step becomes completely ad-hoc: if the controller synthesized based on the current model fails, it is not clear if the model class has to be changed, the amount of data used to find the model has to be increased or another estimation method has to be used. In this way, whatever is gained due to theoretically solid control synthesis methods will be lost when trying to find good models. That is, instead of ad-hoc tuning of the controller, the user will tune the model in a ad-hoc manner. The usefulness of such an approach is dubious.

---

<sup>1</sup>Note that the modeling and controller synthesis step can be merged, leading to adaptive control [250]. For the sake of simplicity I stick to the workflow above. Note that the scientific challenges which arise in modeling for control do not disappear when using adaptive control, in fact, they tend to become even greater.

## 1.1. GLOBAL RESEARCH OBJECTIVES: RELIABLE MODELS FOR CONTROL OF COMPLEX CYBER-PHYSICAL SYSTEMS

---

Finally, if two models obtained via systems identification or model reduction are observationally equivalent, then there is no way of deciding which one is the better, so either model should lead to a correct controller.

Unfortunately, a coherent modeling framework as discussed above is available only for a very small class of systems. Namely, such a framework exists for classical *Linear Time-Invariant (LTI)* model. Even for that case there are gaps in the existing theory. However, linear time-invariant models are not suitable for representing the behavior of more complex cyber-physical systems.

The high-level objective of my research was to accomplish the goals above (**O1,O2,O3**) for various model classes which are more general than linear time-invariant models and which are suitable for representing larger classes of cyber-physical systems. That is, the objective is to come up with a *methodology containing system identification and model reduction algorithms along with theoretical guarantees that the models produced by those algorithms are useful for controller synthesis for cyber-physical systems*. So far, I have been focusing on the following model classes: *Hybrid Systems, Stochastic Bilinear Systems, and Linear Parameter-Varying (LPV) Systems*. In order to achieve the stated goals for these model classes, I used *realization theory* and developed it further when it was necessary.

The goal of realization theory is to characterize the relationship between the observed behavior and the internal structure of the models. More precisely, realization theory attempts to answer the following questions for a given class of dynamical systems:

- **Existence of a realization, realization algorithm** When is it possible to construct a dynamical system of the given class which generates the specified input/output behavior ? Can we provide an algorithm for constructing such a dynamical system from the input-output behavior ?
- **Minimality** How to characterize *minimal dynamical systems* which generate the specified input/output behavior ? By minimal we mean minimal dimensional, i.e. with the least number of state variables. Are minimal dynamical systems unique in some sense ? Do minimal dynamical systems have structural properties which are useful for control, i.e., can they be driven to a desired state by using control inputs (*controllability*) and can their internal state be estimated from the observed behavior (*observability*). Can we propose algorithms for checking minimality ?

The standard answers to this are usually as follows. One has to find a class of dynamical systems and a class of algebraic structures together with a suitable notion of dimension and basis elements such that the following holds. The algebraic structure (*Hankel-structure*) generated by the input-output data has a finite dimension, if and only if the input-output behavior can be generated by a dynamical system from the designated class. The dynamical system then can be constructed from a basis of the Hankel-structure. Moreover, minimality is usually equivalent to observability and controllability. In addition, minimal dynamical systems realizing the same behavior are isomorphic. Note that for each class of systems the notions of minimality, isomorphism, controllability, observability, Hankel-structure have to be defined separately, and the above results have to be proven separately.

This means that for control, it is enough to work with minimal dynamical systems, as such systems can indeed be controlled, and their states can be observed, hence controllers can be built. Moreover, due to the fact that minimal representations are isomorphic, the resulting controller will not depend on the choice of the dynamical system, as long as it represents the same input-output behavior. Furthermore, the parameters of such models could be recovered from input-output measurements, hence such models could be learned from data. Moreover, system identification and model reduction algorithms tend to result in minimal systems. That is, minimal models are likely candidates for models for which a package of system identification and model reduction algorithms can be constructed, such that the models produced by these algorithms are useful for control. Furthermore, algorithms for constructing models from input-output data actually give rise to system identification algorithms, and algorithms for transforming models to minimal ones give rise to model reduction algorithm. Realization theory is also used for proving theoretical correctness of system identification and model reduction algorithms.

That is, *realization theory provides the theoretical foundations for designing the desired package of theoretically sound system identification and model reduction algorithms which are useful for control*.

In the next section, Section 1.2, I will present my research activities on realization theory and its application to system identification and model reduction.

## 1.2 Principal research topics: realization theory and its applications

In the past decade the focus of my research was realization theory and its applications to model reduction and system identification. Within realization theory, I focused on stochastic bilinear and linear parameter-varying (LPV) systems, and to a lesser extent on hybrid systems. Within application of realization theory, I focused on using my results on realization theory of hybrid, LPV and stochastic bilinear systems to model reduction and system identification. In addition, I made some excursions to realization theory of semi-algebraic systems, i.e., systems which are defined by polynomial equations and inequalities, and later, using the latter results, I also worked on realization theory of recurrent neural networks. The reason for focusing on LPV, bilinear, hybrid and semi-algebraic systems is that they can be used to model a wide class of cyber-physical systems, including those which exhibit nonlinear behavior, while at the same time they can be treated with numerical algorithms. This is especially true for bilinear and LPV systems, which encode complex behavior by a finite collection of matrices. Moreover, bilinear and LPV systems overlap with the class of the hybrid systems, the overlap corresponds to the class of *linear switched systems* and *jump-Markov linear systems*. More precisely, linear switched systems/jump-Markov linear systems can be viewed as LPV systems whose scheduling signals take values from a finite set. Furthermore, deterministic bilinear systems can be viewed as LPV systems without control inputs. In particular, bilinear systems contain deterministic autonomous switched systems as a special case. Likewise, stochastic bilinear systems can be viewed as stochastic LPV systems without control inputs, and they contain autonomous jump-Markov linear systems as a special case. Finally, all these system classes can be viewed as subclasses of semi-algebraic systems. These relationships are illustrated on Figure 1.1. Note that on Figure 1.1 the deterministic bilinear and switched systems are not viewed as special cases of their stochastic counterparts. This is motivated by the fact that from the point of view of realization theory and system identification the deterministic case is not really a special case of the stochastic one: in stochastic realization theory the external behavior is always a process or pair of processes, while in the deterministic case it is an input-output relation.

Below I will describe my research activities in the past dozen of years in more detail.

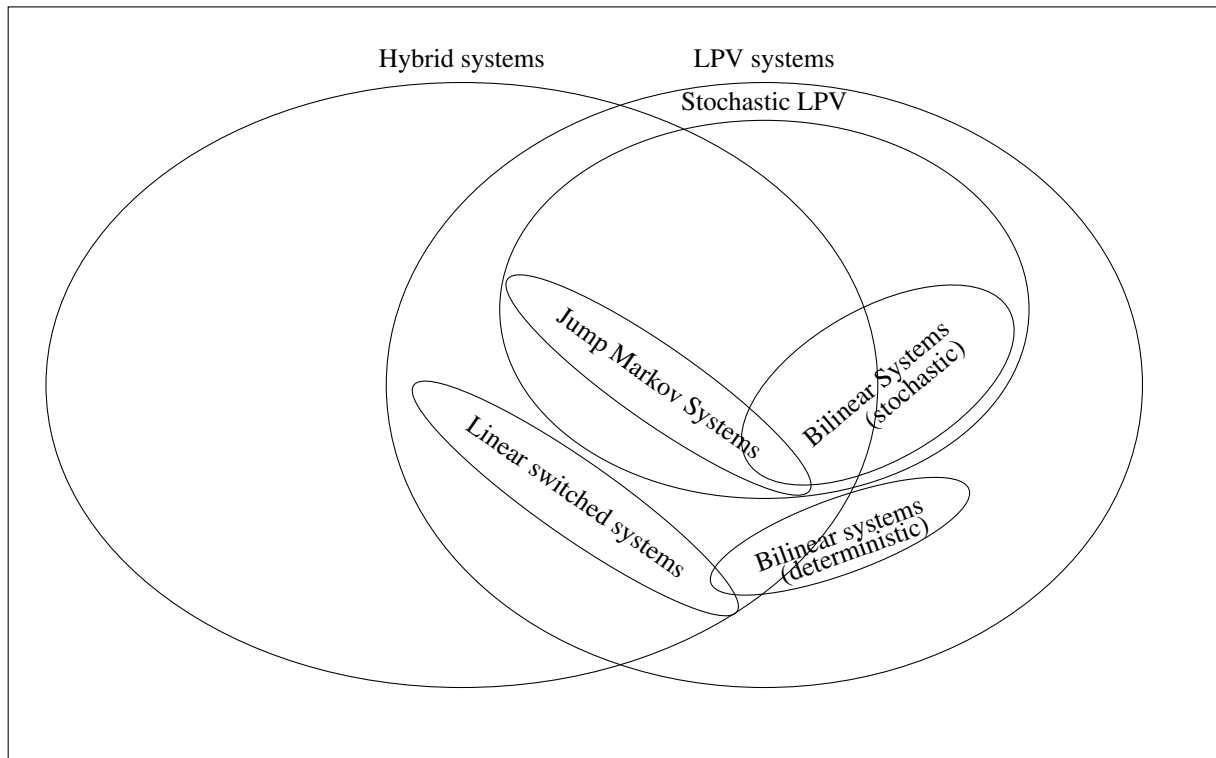
### 1.2.1 Realization theory

I worked on realization theory of following system classes:

**Stochastic bilinear systems and jump-Markov systems.** I worked on realization theory of stochastic bilinear systems [223, 226] with René Vidal (Johns Hopkins University), resulting in a fairly complete theory. These results turned out to be useful for realization theory and identification of stochastic LPV systems and of stochastic jump-Markov systems. In particular, existence of a realization by stochastic bilinear systems was characterized, along with minimality of such realizations. In addition, a formal definition of a stochastic bilinear realization in forward innovation form was proposed, and it was shown that minimal stochastic bilinear systems in forward innovation realizing the same output are isomorphic, and any stochastic bilinear system can be converted to a minimal one in forward innovation form while preserving its output process. Moreover, a covariance realization algorithm was formulated, which calculates a minimal stochastic bilinear realization in forward innovation form for a given process. These results are relevant for theoretical analysis of system identification algorithms, especially for subspace methods, as it was demonstrated in [173, 172].

**LPV systems.** I worked on realization theory of deterministic LPV state-space models with affine dependence [216], in collaboration Roland Tóth (Eindhoven University of Technology) and Guillaume Mercère (Université de Poitiers). I also worked on realization theory of stochastic LPV state-space models with affine dependence [173] in collaboration with Manas Mejari (Université de Lille). Manas Mejari was a postdoc

Figure 1.1: Relationship between various system classes.  
Semi-algebraic systems



under my supervision and this work was funded by the project CPER Data 'Machine learning meets control'. Furthermore, I worked on the question of preserving minimality while converting LPV models to LFR models [5]. This is a joint work with my PhD student Ziad Alkhoury (Université de Poitiers/École des Mines de Douai (IMT-Lille-Douai) and his co-advisor Guillaume Mercère (Université de Poitiers). I also worked on embedding non-linear models to LPV state-space models [1] together with Abbas Hosam (Electrical Engineering Department, Faculty of Engineering, Assiut University, Egypt), Roland Tóth (Eindhoven University of Technology), Nader Meskin (Department of Electrical Engineering, College of Engineering, Qatar University, Qatar), Javad Mohammadpour Velni (School of Electrical and Computer Engineering, The University of Georgia, GA, USA).

**Realization theory of neural networks.** I explored realization theory of recurrent neural networks by embedding them into rational systems (systems whose right-hand side are fractions of polynomials) and using realization theory of rational systems. This work was done in collaboration with Thibault Defourneau, who had a postdoctoral appointment under my supervision, and resulted in [72]. This work was funded by the regional project CPER Data 'Machine learning meets control'. The significance of these results is that they represent an alternative approach towards realization theory of RNNs and that they open up the possibility of studying RNNs through system classes which are perhaps easier to handle theoretically. In particular, the obtained results allowed us to formulate sufficient conditions for existence of a RNN representing a certain behavior and minimality of RNNs. In turn, these results give an insight to what can be expected for RNNs.

**Semi-algebraic systems, rational and Nash systems.** I worked on realization theory of rational and Nash systems [183, 184, 185, 182, 181, 187], leading to necessary conditions for existence of a realization and conditions for minimality, and an observability reduction algorithm. Nash systems are a subclass of semi-algebraic systems, where the right-hand sides of the equations are semi-algebraic and smooth functions. This work was done in collaboration Jana Nemcova (Institute of Chemical Technology, Prague) and Jan van Schuppen (Delft University of Technology). This line of research has not resulted in a complete theory. In [224] a non-trivial necessary condition for existence of a realization is proposed for discrete-time semi-algebraic systems defined by polynomial equations and inequalities. This work was done in collaboration with René Vidal (Johns Hopkins University).

**Spaces of nonlinear and hybrid systems.** In [222, 214], I worked on definition distance and topology for a large class of hybrid systems. This work was carried out in collaboration with René Vidal (Johns Hopkins University) and Ralf Peeters (Maastricht University). We showed that the space of systems forms a Nash manifold. The topology of this manifold is compatible with a distance, for which an explicit formula exists. The coordinate charts of the manifold correspond to local canonical forms.

**Hybrid systems.** I continued working on realization theory of hybrid systems, leading to several results which were not included into my PhD thesis: [209, 217, 224, 218]. Among these publications, [209, 217] dealt with extending the results of my PhD thesis to discrete-time systems with external switching, and the results represent a fairly complete realization theory for the systems in question. This work was carried out in collaboration with Laurent Bako (Laboratoire Ampère, École Centrale de Lyon) and Jan H. van Schuppen (Delft University of Technology). In contrast, [299, 224, 218] dealt with hybrid systems with autonomous switching, and the results are partial.

In [218] an observability reduction method is proposed for piecewise-affine hybrid systems in continuous-time. All this work was carried out in collaboration with Ralf Peeters, Ronald Westra (Maastricht University), and Jan H. van Schuppen (Delft University of Technology) respectively. Some other publications which appeared after my PhD present significantly revised versions of the results included into my PhD thesis: [203, 204, 220, 219, 221, 205].

## 1.2.2 Application of realization theory

In addition to developing realization theory for bilinear and LPV systems, I also worked on its application to model reduction and system identification. In terms of model reduction, I focused more on linear switched systems, which can be viewed as a subclass of LPV systems. However, first results on LPV systems were derived too. In system identification, I focused more on LPV systems.

Below I will describe by research on these topics in more detail.

**System identification.** I used the previously derived results on realization theory to derive new results on system identification of hybrid and LPV systems:

**Identification of hybrid systems.** In [210, 207, 208] I developed a computationally effective characterization of identifiable parameterization of linear switched systems in collaboration with Laurent Bako (IMT-Lille-Douai, École Centrale Lyon), Stéphane Lecoecue, Koffi Motchon (IMT-Lille-Douai) and Jan H. van Schuppen (Delft University of Technology). In [206] I also developed a characterization of persistence of excitation of switching signals in collaboration with Laurent Bako. The analysis relies on realization theory. System identification of hybrid systems was applied to fault detection of rails in [113], the latter was a joint work with Stéphane Lecoecue and Blaise Guepie (IMT-Lille-Douai).

**Identification of LPV systems.** I worked on system identification of LPV systems, more precisely, on identifiability of LPV systems. This work was part of the PhD thesis of Ziad Alkhoury, whom I co-advised with Guillaume Mercère (Université de Poitiers). The results were published in [7, 6]. Furthermore, I

### 1.3. OTHER RESEARCH DIRECTIONS NOT RELATED TO REALIZATION THEORY

---

collaborated with Pepijn Cox, Roland Tóth (Eindhoven University of Technology) on system identification algorithms for LPV systems [66]. This work was extended to stochastic system in [172, 173] in collaboration with Manas Mejari (Université de Lille). Manas Mejari was a postdoc under my supervision and this work was funded by the project CPER Data 'Machine learning meets control'.

**Identification of polynomial systems.** In addition, I worked on system identification of polynomial and rational systems in collaboration with Jana Nemcova (Institute of Chemical Technology, Prague) and Jan H. van Schuppen (Delft University of Technology) the results were published [186].

#### Model reduction

**Model reduction of hybrid systems.** I worked on model reduction of linear switched systems using moment matching in collaboration with John Leth, Rafael Wisniewski (Aalborg University) [227], Mert Bastug (Aalborg University/École des Mines de Douai (IMT-Lille-Douai)/École Centrale Lille) [30, 31]. The papers [30, 31] were part of the PhD thesis of Mert Bastug, who was co-advised by me, Rafael Wisniewski and John Leth. Furthermore, I worked on model reduction of hybrid systems in collaboration with Ion Victor Gosea (Max Planck Institute for Dynamics of Complex Technical Systems, Magdeburg, Germany) and Athanasios C. Antoulas (Max Planck Institute for Dynamics of Complex Technical Systems, Magdeburg, Germany, Department of Electrical and Computer Engineering, Rice University, Houston) [105] using moment matching and Löwner matrices. In addition, with Ion Victor Gosea, Athanasios C. Antoulas and with my colleague, Christophe Fiter from CRISAL [106] we derived new error bounds for balanced truncation of switched systems with reset maps. Furthermore, with Laurentiu Hetel (CRISAL, CNRS), and Mert Bastug (postdoc at CRISAL, CNRS) we worked on model reduction of networked control systems, using results from model reduction of hybrid systems [27, 26, 25]. I also worked on balanced truncation of hybrid systems with reset maps and state-dependent switching together with John Leth, Rafael Wisniewski, Rasmus Pedersen (Aalborg University) [212] and Ion Victor Gosea and Athanasios C. Antoulas [108].

**Model reduction of LPV and bilinear systems.** In collaboration with John Leth, Rafael Wisniewski (Aalborg University) [227], Mert Bastug (Aalborg University/École des Mines de Douai (IMT-Lille-Douai)/École Centrale Lille) Roland Tóth (Eindhoven University of Technology) and Denis Efimov (CRISAL) I worked on model reduction of LPV systems [28] using moment matching. The paper [28] was part of the PhD thesis of Mert Bastug. With Rafael Wisniewski and John Leth I also worked on model reduction of bilinear systems using moment matching, leading to [228]. With Ion Victor Gosea and Athanasios C. Antoulas I worked on Loewner framework for LPV systems [107].

All the results mentioned above used results from realization theory of switched, hybrid and LPV systems, which were previously developed by my collaborators and me.

**Reverse engineering of networks of linear and stochastic bilinear systems.** I worked on realization theory of the interconnection structure for stochastic linear and bilinear systems. The goal of this work is to relate the interconnection graph of state-space representations with the statistical properties of their outputs. This relationship is useful for learning network structure from data and it has applications in systems biology and neuroscience. This is a joint work with my PhD student Mónica Józsa (University of Groningen/IMT-Lille-Douai) and her co-advisor Kanat Camlibel (University of Groningen). The results of this work were published in [137, 136, 134, 135].

### 1.3 Other research directions not related to realization theory

In addition to realization theory and its application to system identification and model reduction, I made a number of excursions to other domains. This was motivated by applications I came across and by the desire to explore the application of realization theory to control design and estimation.

**Optimal control and minimax observers for linear descriptor systems.** I worked on this topic, in collaboration with Serhiy Zhuk (IBM Research Dublin) resulting in the following publications [229, 317]. This work used several results from realization theory of linear descriptor systems.

In addition to being a non-trivial application of realization theory, this work is a first step towards addressing realization theory of hybrid/LPV descriptor models with a bounded noise. The latter is expected to be useful for extending realization theory for hybrid/LPV systems to systems with bounded noise and algebraic constraints. Indeed, in many applications the assumption that the noise is stochastic is not realistic and the model may contain algebraic equations in addition to difference/differential equations. Minimax observers are the deterministic counterpart of Kalman filters. Since Kalman filters were crucial in developing realization theory for stochastic linear systems and stochastic bilinear/LPV/jump-Markov systems [167], it is expected that minimax observer will play a similarly crucial role in realization theory of deterministic systems with bounded noise.

**Stability analysis.** I worked on stability analysis of sampled-data systems in collaboration with Laurentiu Hetel, Jean-Pierre Richard (CRISAL), Hassan Omran (École Centrale de Lyon) and Françoise Lamnabhi-Lagarigue (SUPELEC). This work was published in [193]. This work was relevant for me as it allowed me to understand better the relationship between stability and dissipativity conditions of sampled-data systems with their structural properties, such as observability, controllability, minimality. In turn, this is necessary in order to apply realization theory to control, as planned in the original work plan. Note that sampled-data control systems are an important class of hybrid/LPV systems with many additional properties, hence they are a logical choice for starting to explore the relationship between stability/dissipativity and structural properties (observability, controllability, minimality) of hybrid/LPV systems.

**Observer design.** With my colleagues from CRISAL, Laurentiu Hetel, Denis Efimov and Luciene Etienne (now at IMT-Lille-Douai) I worked on observer design for sampled-data control systems, resulting in the following publications [84, 83]. I worked on observer design of rational systems with Jana Nemcova (Institute of Chemical Technology, Prague) and Jan van Schuppen (Delft University of Technology) [188]. With Mohamed Djemai (Université of Valenciennes) and his co-worker I also worked on observer design for Petri-nets using theory of singular linear systems [13]. My motivation for participating in this work was to better understand the relationship between observability, as defined in realization theory, and observer design for networked and rational control systems respectively. As it was mentioned above, sampled-data control systems are an important special class of hybrid/LPV system, so it was a logical to start studying the relationship between observer design and observability with sampled-data systems.

**Hybrid and discrete-event control of mechatronics systems.** In [272, 211, 215, 38] we investigated control problems arising in certain mechatronics applications (printers, MRI scanners, etc.). In these applications, the underlying plant is a hybrid or discrete-event system and the control objectives are discrete (can be expressed by logical formulae). This work was done in collaboration with Rolf J. M. Theunissen Dirk A. van Beek, Jacobus E. Rooda, Ramon R. Schiffelers, Esmée Bertens (Eindhoven University of Technology), Jan H. van Schuppen (Delft University of Technology), and Roland Faber (Océ).

**Application of control theory to water management.** In [121, 122, 120] I investigated the application of model predictive control for controlling the locks of navigable canals. The control objective was to maintain the water level high enough to allow navigation. This work was done in collaboration with Eric Duviella, Lala Rajaoarisoa, Klaudia Horvath (IMT-Lille-Douai), Karine Chuquet (VNF), and Joaquim Blesa (Universitat Politècnica de Catalunya).

## 1.4 Outline of the manuscript

In this report I will present some selected topics I worked on in the past decade. These topics deal with realization theory and its applications to system identification and model reduction. The report is divided into three parts.

In the first part, I present two major results on realization theory which were derived in the past decade:

Chapter 2 (realization theory of LPV systems):

In this chapter I present an overview of realization theory of LPV systems with affine dependence on the scheduling parameters. This chapter is based on [213, 216]. These results were developed in collaboration with Guillaume Mercère (Université de Poitiers, France) and Roland Tóth (Eindhoven University of Technology, The Netherlands).

Chapter 3 (realization theory of generalized stochastic bilinear systems):

I present realization theory of stochastic bilinear systems. Note that the latter class contains linear jump-Markov systems and stochastic LPV systems as a special case. This chapter is based on [223, 226]. This work was carried out in collaboration with René Vidal (Johns Hopkins University, USA).

In the second part, I present some of my selected results on application of realization theory to system identification and model reduction:

Chapter 4 (model reduction of hybrid systems):

In this chapter I present an overview of application of realization theory to model reduction of switched systems. More precisely:

- I discuss results on moment matching using Markov-parameters, which were developed in the PhD thesis of Mert Bastug [24] whom I co-advised with John Leth and Rafael Wisniewski (Aalborg University, Denmark). The presentation is based on the thesis [24] and the related publications [31, 30].
- I discuss results on balanced truncation. This work was done in collaboration with John Leth, Rafael Wisniewski, Ion Victor Gosea (Max Planck Institute, Germany), Athansios C. Antoulas (Max Planck Institute, Germany and Rice University, USA) and Christophe Fiter (CRISAL, France). The presentation is based on [227, 106].
- I also discuss moment matching in frequency domain, which was developed in collaboration with Ion Victor Gosea (Max Planck Institute, Germany) and Athansios C. Antoulas (Max Planck Institute, Germany and Rice University, USA). The presentation is based on [105].

Chapter 5 (identifiability and subspace identification of LPV systems):

In this chapter I present selected results on system identification of LPV and hybrid systems:

- I discuss identifiability of LPV systems. This work is based on the PhD thesis of Ziad Alkhoury (Université de Poitiers and IMT-Lille-Douai, France), whom I co-advised with Guillaume Mercère. The presentation is based on [7, 4].
- I present a subspace identification algorithm for deterministic LPV systems. These results were developed in collaboration with Roland Tóth and Pepijn Cox (Eindhoven University of Technology, The Netherlands). The presentation is based on [66].
- Finally, I discuss subspace identification algorithms for stochastic LPV. These results were developed in collaboration with Manas Mejari (CRISAL, France), of whom I was the advisor during his postdoctoral appointment. The presentation is based on [172, 173].

Chapter 6 (reverse engineering of networks):

In this chapter I present selected results on reverse engineering of network graphs of dynamical systems. The results are based on the PhD thesis of Mónika Józsa (Groningen University, The Netherlands, IMT-Lille-Douai, France), whom I co-advised with Kanat Camlibel (Groningen University, The Netherlands). The presentation is based on [133, 137, 136].



In the third part (Chapter 7), I will present my plans for future research.

The manuscript ends by presenting a short summary and conclusions in Chapter 8.

Finally, in Appendix A, I present my CV with a complete list of my publications and a detailed overview of my service to scientific community (editorial activities, organization of workshops and invited sessions), of my participation in funded projects, and my role in advising PhD students and postdoctoral researchers.

## **Acknowledgements**

I am extremely grateful to Bart De Moor, Jamal Daafouz and Antoine Girard for accepting to be reviewers ('rapporteur') of my HDR and to serve on my defense committee. I would like to warmly thank Michel Verhaegen, Xavier Bombois and Lotfi Belkoura for agreeing serve on my defense committee. I would like to thank Laurentiu Hetel for agreeing to be my advisor for HDR and for guiding me through the process and providing a lots of valuable suggestions and comments. I would also thank my colleagues for their support.

## **Part I**

# **Selected topics on realization theory: realization theory of LPV and bilinear systems**



# Introduction

First, I will briefly recall some important milestones in the development of realization theory, on which my own contribution was based on. Second, I will explain the relationship between my own contribution and the existing literature.

## A brief history of realization theory

As it was mentioned before, realization theory was originally championed by the founder of modern control theory, R.E. Kalman, [143, 142]. Below, I will review those results in realization theory which were relevant for my own research on the subject.

### Realization theory of linear systems

**Deterministic linear systems.** Realization theory was first worked out for *deterministic linear time-invariant systems (LTI systems for short)* resulting in a large body of literature. We will not present a historical overview of the various papers which appeared in the literature, such an overview can be found for example in [140, 144, 71] and the references therein. In a nutshell, the main results of realization theory of LTI systems are as follows. For LTI systems input-output behavior is traditionally formalized as input-output maps. Later on, a more general formal framework was proposed, the so called behavioral approach [306, 303, 304], however, for the sake of simplicity we will stick to the traditional framework. The input-output maps which can potentially be represented by LTI systems turned out to be the ones which correspond to *proper rational transfer functions*, if interpreted in the frequency domain. It turns out that the existence of a LTI realization is equivalent to the finite rank property of the classical infinite *Hankel-matrix* [140, 144]. The entries of the Hankel-matrix, the so called *Markov-parameters*, they can be viewed as outcomes of input-output experiments. If the rank of the Hankel-matrix is finite, then a minimal linear system can be computed from a finite sub-matrix of the Hankel-matrix. The corresponding algorithm is known as *Kalman-Ho realization algorithm* [140, 144]. Moreover, all minimal linear state-space representations of a certain input-output map are controllable and observable. Conversely, every controllable and observable state-space representation is minimal. Furthermore, all such minimal state-space representations are isomorphic and one can compute them directly from input-output data. For a detailed discussion see for example [140, 144].

**Stochastic linear systems.** Later on, realization theory was extended to stochastic linear time-invariant systems (stochastic LTI) without inputs, see the monographs [57, 163] and the references therein. The main idea behind this extension was to show that the covariances of the output process of a stochastic LTI are the Markov-parameters of a deterministic LTI system, and conversely, that any minimal deterministic LTI system whose Markov-parameters are the covariances give rise to a stochastic LTI system whose output is the given output process. Incidentally, the latter relationship is closely related to dissipativity and solution of LMIs. Moreover, it turned out that minimal deterministic LTI systems whose Markov-parameters are the covariances of the output process actually give rise to stochastic LTI systems in the so called *forward innovation form*, i.e., stochastic LTI systems whose noise process is the innovation process of their output process. Recall that the innovation process

---

is the difference between the current output and the best possible (linear) prediction of the current output. Furthermore, it was shown that minimal stochastic LTI systems in forward innovation form are isomorphic, if they represent the same output process. In addition, the matrices of a minimal stochastic LTI system can be calculated from the output covariances, using the so called *covariance realization algorithm*, which is the combination of Ho-Kalman realization algorithm applied to covariances and solving an algebraic Riccati equation or LMI. These results were later extended to stochastic LTI systems with inputs, see [163, 147] and the references therein.

**Role of realization theory in control and system identification** Realization theory of LTI systems (both deterministic and stochastic) played a significant role in control, system identification and model reduction of these systems. For control design, the fact that any LTI system could be brought to a minimal one, while preserving input-output behavior means that it is enough to work with minimal LTI systems, if the only information about the underlying plant is its input-output behavior. Indeed, in the latter case, there is no a-priori reason for not using minimal LTI systems as models for the plant, as minimal LTI systems are indistinguishable from any other class of LTI systems based solely on the observed input-output behavior. The fact that such systems are controllable and observable means that controller synthesis for such systems is easy, as existence of a feedback controller guaranteed by controllability, and existence of an observer is guaranteed by observability. Moreover, due to the fact that minimal representations are isomorphic, the resulting controller will not depend on the choice of the state-space representation. Moreover, minimality turned out to be a useful technical property for relating various properties of input-output behavior to properties of state-space representations. For example, for minimal LTI systems dissipativity with respect to a supply rate, which is a property of the input-output behavior, is equivalent to the existence of a quadratic storage function, and the latter is equivalent to the existence of a solution of an LMI [305]. In turn, such a correspondence is useful for characterizing conservativeness of LMI based approaches, and leads to such important results as Kalman-Yakubovich-Popov Lemma and Real Bounded Lemma [305, 85]. Note that it is exactly the implication that dissipativity implies existence of storage functions which is the most challenging one, and it is precisely realization theory which allows the proof of that direction.

Furthermore, linear realization theory turned out to be of fundamental importance for system identification see [282, 57, 147] and the references therein. In particular, the Kalman-Ho realization algorithm gave rise for subspace identification algorithms e.g., [282, 147], identifiability analysis, e.g., [101, 114, 279], theory of identifiable local canonical forms and recursive parametric system identification algorithms, e.g. [114, 199, 171]. Minimality and minimization algorithms are closely related to balanced truncation for linear systems, and the notion of Markov-parameters is essential to model reduction by moment matching, [12]. In fact, system identification and model reduction are the main application domains of realization theory.

### **Extending realization theory beyond LTI systems**

Motivated by the success of realization theory for LTI systems, there were several attempts to extend realization theory to other classes of systems. When extending realization theory to other classes of systems, the aim was to show, to the extent it was possible, the same results as for LTI systems. That is, minimality was shown to be equivalent to counterparts of controllability and observability, minimal realizations were shown to be isomorphic in a certain sense. Likewise, existence of a realization was shown to be equivalent to finiteness of the dimension of a suitably defined algebraic structure which is generated by the input-output behavior. Finally, constructive procedures were formulated for converting a system to minimal ones and for constructing them from input-output behaviors.

**Time varying linear systems, linear systems over rings.** One direction was to extend LTI realization theory to other classes of linear systems. Since realization theory of linear systems is algebraic in nature, it can be extended to systems matrices of which are defined over any field [143]. The latter is relevant for systems describing codes [243], and for time-varying and LPV systems [145, 274]. There are extensions to systems defined on rings [258]. There are also extensions of realization theory to infinite-dimensional systems [128], the latter class of systems

---

included partial-differential equations and systems with delays. It is also possible to reformulate realization theory in terms of modules over ring of polynomials [143]. These extensions did not give rise to such deep applications to control, system identification and model reduction as realization theory of LTI systems.

**Deterministic and stochastic bilinear systems.** Realization theory was extended to deterministic bilinear systems, leading to a theory which follows closely realization theory of LTI systems, see [257, 256, 125, 126, 127] and the references therein. In particular, for bilinear systems minimality is equivalent to span-reachability (the linear span of reachable states is the full state space) and observability, minimal realizations of the same input-output map are related by a linear isomorphism, existence of a bilinear realization is equivalent to the rank of a suitably defined Hankel-matrix. Furthermore, any bilinear system can be converted to minimal one while preserving input-output behavior, and a minimal bilinear realization can be computed from the Hankel-matrix. The latter matrix is constructed using input-output data, more precisely, from counterparts of Markov-parameters, which, in turn, can be obtained from suitable input-output experiments. More precisely, they represent the high-order derivatives of the output with respect to the time (continuous-time case) and/or input values. Partial results on extending realization theory of bilinear systems to the stochastic case appeared in [73, 92, 235] and the references therein.

**Analytic and smooth systems.** Realization theory was also extended to analytic and smooth non-linear systems [60, 131, 130, 132, 90, 261, 91, 126, 238, 58, 59, 96] and the references therein. More precisely, the characterization of minimality in terms of controllability and observability conditions and isomorphism of minimal realization still carries over to analytic and smooth non-linear systems even though there are several technical subtleties [126]. Generalization of Hankel-rank conditions are also available [60, 131, 130, 132]. In fact, for analytic input affine non-linear systems the theoretical framework of *generating (Fliess) series* [90, 91, 126] gives rise to counterparts of Markov-parameters, which allows to define Hankel-matrices and their Lie-rank [132, 90, 91, 126]. However, the characterization of existence of a realization has only partially been solved and the problem of finding a realization algorithms has not been completely solved either, see [129, 194, 195] and the references therein. Note that the geometric characterization of controllability and observability for nonlinear systems turned out to be useful not only for realization theory, but also for control, giving rise to the field of geometric control theory [126, 281]. Realization theory of nonlinear systems with polynomial/rational systems was also addressed, see [22, 256, 257, 23, 189, 190, 296] and the references therein, leading to conditions on minimality and existence of a realization. However, the computational aspects have not been worked out in detail, and a lot of subtle technical issues remain open.

**Hybrid and LPV systems.** The realization problem for hybrid systems was first formulated in [111], but no theory was developed. Realization theory for linear and bilinear switched systems with linear reset maps was developed in [202, 203, 204, 220, 219, 221]. The latter work was partially based on my PhD thesis. Realization theory for LPV was investigated in [276] using a behavioral setting, and for LFR systems in [33, 32, 34]. Partial results on realization theory of discrete-time LPV systems with affine dependence were developed in [275].

**Application of realization theory for non-LTI systems.** Realization theory for bilinear and analytic systems was used for identifiability analysis, see [294, 277, 200]. Moreover, realization algorithms for bilinear systems were used for subspace identification algorithms [286, 86, 76] and the references therein. The results on realization theory of LFR and LPV systems [33, 32, 275] were applied to model reduction of these systems [34, 275].

## My research on realization theory

**Novelty, open questions.** Despite these developments several open questions remained at the moment when I started my independent research career. In particular, the discrete-time version of realization theory of linear switched systems was not developed. There were no results on realization theory of jump-Markov linear

---

switched systems with stochastic switching and stochastic additive noise. Furthermore, there were no results on hybrid systems with autonomous, state-dependent, switching, especially in continuous-time. Realization theory of stochastic bilinear systems was only partially developed: results concerned only white noise inputs, no characterization of minimality or existence was available, published proofs of the correctness of the realization algorithm contained gaps. Realization of LPV systems with affine dependence was only partially developed: while a realization algorithm was formulated in [275], the rest of the theory was not developed.

My work on realization theory was aimed at filling these gaps.

**Brief overview of my recent research on realization theory.** I worked on realization theory of hybrid systems [209, 217, 224, 218], extending previous results to the discrete-time case and to systems with state-dependent switching. These results were derived in collaboration with several co-authors: Jan H. van Schuppen, Laurent Bako, René Vidal. I developed for linear switched systems to develop realization theory for LPV systems with affine dependence on parameters [216] and for stochastic bilinear and linear jump-Markov systems [226]. The work on LPV realization theory was done in collaboration with Guillaume Mercère, Roland Tóth, Pepijn Cox. The work on stochastic bilinear systems and jump-Markov systems was carried out in collaboration with René Vidal. A more elaborate discussion on the novelty of my work on realization theory of LPV systems and stochastic bilinear systems will be presented in Chapter 2 and Chapter 3 respectively.

Note that realization theory of linear/bilinear switched and hybrid systems with external switching relies on the same algebraic tool as realization theory of bilinear systems, namely, on the theory of recognizable formal power series [37, 152, 256]. Note that the latter theory is also present in computer science, and it is related to weighted and probabilistic automata [152]. In addition, the same tools are also used for noncommutative  $nD$  systems which arise in robust control [16, 33, 32, 141] and the references therein. As a result, the resulting theory is quite complete and parallels closely that of LTI and bilinear systems. In particular, minimality is shown to be equivalent to span-reachability and observability, minimal realizations of the same input-output behavior are isomorphic, and existence of a realization by a linear/bilinear switched/hybrid system is equivalent to a finite rank condition of a suitably defined Hankel-matrix. Note that the Hankel-matrix is defined via counterparts of Markov-parameters. For linear switched/hybrid systems they are impulse responses (discrete-time) or high-order derivatives of impulse responses along various switching signals. Moreover, algorithms were formulated for transforming a system to a minimal one and for computing a minimal realization from the Hankel-matrix.

## Contents of Part I

In order to give a glimpse of my research on realization theory after completing my PhD, I will present two topics, which I think are representative of my work in that direction. The first topic (Chapter 2) is realization theory of linear parameter-varying systems with affine dependence on parameters, the second one (Chapter 3) is realization theory of stochastic bilinear systems. The reason for this choice is twofold. First, I believe that these two results are the most significant ones among the various results I have obtained in realization theory. Second, these results have been used in system identification and model reduction. Hence, it is possible to demonstrate the interaction between realization theory and system identification on the example of these results. Finally, these results rely heavily on my previous results on realization theory of linear switched systems [202, 203, 219, 221, 205].

## Chapter 2

# Realization theory of LPV systems

### 2.1 Introduction

In this chapter we present a realization theory for LPV state-space representations with affine static dependence of coefficients, abbreviated as *LPV-SSA*. *Linear parameter-varying* (LPV) systems are linear systems where the coefficients are functions of a time-varying signal, the so-called *scheduling variable*. That is, LPV systems are a class of mathematical models having a certain structure (linear and time-varying). Control design and system identification of LPV systems is a popular topic [244, 178, 293, 274, 275, 17, 283, 75, 265, 288, 47]. LPV-SSAs are linear state-space representations, matrices of which are affine functions of the scheduling variables. LPV-SSAs are popular models for control synthesis, model reduction and system identification. This popularity is due to the existence of efficient control synthesis algorithms for LPV-SSAs [244, 178].

In this chapter we will consider both the *discrete-time* (DT) and the *continuous-time* (CT) cases. Realization theory of LPV-SSAs is fairly complete. In particular, there exist necessary and sufficient condition for an input-output map to admit a realization by an LPV-SSA. This condition involves the rank of a suitably defined Hankel-matrix. In addition, a minimal LPV-SSA realization of an input-output map can be calculated from the Hankel-matrix using a Ho-Kalman-like realization algorithm. Moreover, minimality is equivalent to observability and span-reachability and that all minimal LPV-SSA realizations of the same input-output map are isomorphic. The latter isomorphism is linear and does not depend on the scheduling variable.

Realization theory of LPV-SSA models can be solved using realization theory of linear switched systems [202, 203, 219, 209, 205]. Conversely, linear switched systems are a subclass of LPV-SSAs, and when the results of realization theory of LPV-SSAs are applied to linear switched systems, one arrives at the same results as in [202, 203, 219, 209, 205].

Many aspects of realization theory of LPV systems were addressed before. However, a complete realization theory of LPV systems with affine dependence was lacking. In [274, 276], realization theory was developed for LPV state-space representations where the system matrices depend on the parameters in a *meromorphic and dynamic way*, *i.e.*, the matrices are meromorphic functions of the scheduling variables and their derivatives (in continuous-time), or of the current and future values of the scheduling variables (discrete-time). The system theoretic transformations (passing from an input-output to a state-space representation, transforming a state-space representation to a minimal one, etc.) of [274, 276] introduce LPV models with a dynamic and nonlinear dependence on the parameters. In [35], using [63], realizability of LPV input-output model by LPV state-space representations with a *nonlinear* (hence not necessarily affine) and static dependence is studied. In contrast, we deal with the realizability of input-output maps and not of input-output equations, and we are interested in LPV state-space representations with affine and static dependence on the parameter. That is, [274, 276, 35] do not solve the realization problem for LPV-SSAs, in contrast to the results presented in this chapter. Some elements of realization theory of LPV-SSAs, such as Hankel-matrices and Markov-coefficients of LPV-SSAs, appeared in



[283, 288, 275]. However, in contrast to [283, 288, 275], in the current chapter, these concepts are defined directly for input-output maps, and they are used to characterize existence of an LPV-SSA realization of an input-output map. In fact, the finite rank of the Hankel-matrix represents the necessary and sufficient condition for the existence of an LPV-SSA realization. In addition, we discuss the conditions for the correctness of the realization algorithm in more details. The Kalman-Ho realization algorithm was discussed in [275], but it was formulated with the assumption that an LPV-SSA realization exists. Moreover, the conditions under which the algorithm yields a true realization of the input-output function were not discussed in detail. Extended observability and reachability matrices were also presented in [274, 245], but their system-theoretic interpretation and their relationship with minimality were not explored. This chapter is based on [216].

In Section 2.2 we present the notation used in this and subsequent chapters. In Section 2.3 we present the definition of LPV-SSAs, input-output maps, equivalence and minimality. In Section 2.4 we present the relationship between LPV-SSA and linear switched systems. In Section 2.5 we present the main results on minimality and in Section 2.6 the main results on existence of a realization. Finally, in Section 2.7 realization algorithms similar to the Ho-Kalman algorithm will be discussed and their correctness analyzed.

## 2.2 Notation used in the technical part of the manuscript

In this chapter, and later on, throughout the manuscript, the following notation will be used.

**Spaces of functions** We denote by  $AC(\mathbb{R}_+, \mathbb{R}^n)$  the set of all absolutely continuous functions of the form  $f: \mathbb{R}_+ \rightarrow \mathbb{R}^n$ . We denote by  $PC(\mathbb{R}_+, \mathbb{R}^m)$  the set of piecewise-continuous functions of the form  $f: \mathbb{R}_+ \rightarrow \mathbb{R}^m$ , i.e. functions  $f: \mathbb{R}_+ \rightarrow \mathbb{R}^m$  which have finitely many points of discontinuity on each finite interval and at each point of discontinuity, the left- and right-hand side limits exists and are finite.

**Standard basis vectors** For each  $j = 1, \dots, m$ ,  $e_j$  is the  $j$ th unit vector of  $\mathbb{R}^m$ , i.e.  $e_j = (\delta_{1,j}, \dots, \delta_{n,j})$ ,  $\delta_{i,j}$  is the Kronecker symbol.

**Time axis,  $\xi$  operator** Let  $\mathbb{T} = \mathbb{R}_0^+ = [0, +\infty)$  be the time axis in the *continuous-time* (CT) case and  $\mathbb{T} = \mathbb{N}$  in the *discrete-time* (DT) case. Note that in both cases we exclude negative time instances.

Denote by  $\xi$  the differentiation operator  $\frac{d}{dt}$  (in CT) and the forward time-shift operator  $q$  (in DT), i.e., if  $z: \mathbb{T} \rightarrow \mathbb{R}^n$ , then  $(\xi z)(t) = \frac{d}{dt}z(t)$ , if  $\mathbb{T} = \mathbb{R}_0^+$ , and  $(\xi z)(t) = z(t+1)$ , if  $\mathbb{T} = \mathbb{N}$ . Denote by  $\xi^k$  the  $k$ -fold application of  $\xi$ , i.e., for any  $z: \mathbb{T} \rightarrow \mathbb{R}^n$ ,  $\xi^0 z = z$ , and  $\xi^{k+1} z = \xi(\xi^k z)$  for all  $k \in \mathbb{N}$

**Set of infinite sequences** Denote by  $X^{\mathbb{N}}$  the set of all functions of the form  $f: \mathbb{N} \rightarrow X$ .

**Notation from automata theory: words** The notation described below is standard in automata theory, see [97, 80].

**Set of strings (words) over an alphabet.** Consider a set  $X$  which will be called the *alphabet*. Denote by  $X^+$  the set of finite sequences of elements of  $X$ , i.e. each element  $w \in X^+$  is of the form  $w = a_1 a_2 \dots a_k$  for some  $a_1, a_2, \dots, a_k \in X$  and  $k \in \mathbb{N}, k > 0$ . Let  $\varepsilon \notin X^+$  be a symbol, which we will call the *empty sequence or empty word*. Denote by  $X^*$  the set  $X^+ \cup \{\varepsilon\}$ . The elements of  $X^*$  will be referred to as *strings or words* over  $X$ .

**Length of a string.** If  $w \in X^+$ ,  $w = a_1 a_2 \dots a_k$  for some  $a_1, a_2, \dots, a_k \in X$ , then  $a_i$  is called the *ith letter* of  $w$ , for  $i = 1, \dots, k$  and  $k$  is called the *length*  $w$ . By convention, the length of  $\varepsilon$  is defined to be zero. The length of a word  $w \in X^*$  is denoted by  $|w|$ .

**Concatenation of words.** For any two words  $w, v \in X^*$ , we define the *concatenation*  $wv \in X^*$  of  $w$  and  $v$  as follows. If  $w, v \in X^+$  are of the form  $v = v_1 v_2 \dots v_k, k > 0$  and  $w = w_1 w_2 \dots w_m, m > 0, v_1, v_2, \dots, v_k, w_1, w_2, \dots, w_m \in X$ , then define  $wv = v_1 v_2 \dots v_k w_1 w_2 \dots w_m$ . If  $v = \varepsilon$  and  $w \in X^*$ , then define  $wv = w$ . Similarly, if  $w = \varepsilon$  and  $v \in X^*$ ,

### 2.3. PRELIMINARIES

then define  $vw = v$ . With the operation of concatenation,  $X^*$  forms a semi-group whose unit element is  $\varepsilon$ . For  $a \in X$  and  $k \in \mathbb{N}$ ,  $k > 0$ , we denote by  $a^k$  the sequence  $\overbrace{aa \cdots a}^{k\text{-times}}$ ; by convention  $a^0 = \varepsilon$ .

**Definition 2.1** (Lexicographic ordering). *Assume that  $X$  be a finite subset of integers. We define a lexicographic ordering  $\prec$  on  $X^*$  as follows. For any  $v, s \in X^*$ ,  $v \prec s$ , if either  $|v| < |s|$  or  $0 < |v| = |s|$ ,  $v \neq s$  and for some  $l \leq |s|$ ,  $v_l < s_l$  with the usual ordering of integers and  $v_i = s_i$  for  $i = 1, \dots, l-1$ . Here  $v_i$  and  $s_i$  denote the  $i$ th letter of  $v$  and  $s$  respectively. Note that  $\prec$  is a complete ordering and  $X^* = \{v_1, v_2, \dots\}$  with  $v_1 \prec v_2 \prec \dots$ . Note that  $v_1 = \varepsilon$  and for all  $0 < i \in \mathbb{N}$ ,  $q \in X$ ,  $v_i \prec v_i q$ .*

*Denote by  $\mathbf{N}(M)$  the number of sequences from  $X^*$  of length at most  $M$ . It then follows that  $|v_i| \leq M$  if and only if  $i \leq \mathbf{N}(M)$ .*

#### Products of matrices indexed by words

**Notation 2.1.** *Consider a collection of matrices  $\{A_q \in \mathbb{R}^{n \times n}\}_{q \in X}$ . For any  $v \in X^*$ , define the matrix  $A_v \in \mathbb{R}^{n \times n}$  as follows. If  $v = \varepsilon$ , then  $A_\varepsilon = I_n$  is the identity matrix, and if  $v = q_1 \cdots q_k$  with  $q_1, \dots, q_k \in X$ ,  $k > 0$ , then*

$$A_v = A_{q_k} A_{q_{k-1}} \cdots A_{q_1}.$$

## 2.3 Preliminaries

An LPV *state-space* (SS) representation with *affine* linear dependence on the *scheduling variable* (abbreviated as LPV-SSA) is a continuous-time (CT) or discrete-time (DT) state-space representation of the form

$$\Sigma \begin{cases} (\xi x)(t) &= A(p(t))x(t) + B(p(t))u(t), \\ y(t) &= C(p(t))x(t) \end{cases} \quad (2.1)$$

where  $x(t) \in \mathbb{X} = \mathbb{R}^{n_x}$  is the state,  $y(t) \in \mathbb{Y} = \mathbb{R}^{n_y}$  is the output,  $u(t) \in \mathbb{U} = \mathbb{R}^{n_u}$  is the input, and  $p(t) \in \mathbb{P} \subseteq \mathbb{R}^{n_p}$  is the value of the *scheduling variable* at time  $t$ , and  $A, B, C$  are matrix valued functions on  $\mathbb{P}$  defined as

$$A(\mathbf{p}) = A_0 + \sum_{i=1}^D A_i \mathbf{p}_i, \quad B(\mathbf{p}) = B_0 + \sum_{i=1}^D B_i \mathbf{p}_i, \quad C(\mathbf{p}) = C_0 + \sum_{i=1}^D C_i \mathbf{p}_i, \quad (2.2)$$

for every  $\mathbf{p} = [\mathbf{p}_1 \quad \mathbf{p}_2 \quad \cdots \quad \mathbf{p}_{n_p}]^\top \in \mathbb{P}$ , with constant matrices  $A_i \in \mathbb{R}^{n_x \times n_x}$ ,  $B_i \in \mathbb{R}^{n_x \times n_u}$ ,  $C_i \in \mathbb{R}^{n_y \times n_x}$  for all  $i \in \{0, \dots, n_p\}$ . Recall from Section 2.2 that  $(\xi x)(t) = \frac{d}{dt}x(t)$  in CT, and  $(\xi x)(t) = x(t+1)$  in DT. Intuitively,  $p$  corresponds to varying-operating conditions, nonlinear/time-varying dynamical aspects and /or external effects influencing the plant behavior and it is allowed to vary in the set  $\mathbb{P}$ , see [274].

In the sequel, we use the tuple

$$\Sigma = (\mathbb{P}, \{A_i, B_i, C_i, D_i\}_{i=0}^{n_p})$$

to denote an LPV-SSA of the form (2.1) and use  $\dim(\Sigma) = n_x$  to denote its state dimension.

In order to formulate the main result, we have to define what we mean by input-output function of an LPV-SSA. To this end, define the following spaces. Denote by  $\mathcal{X}$  the set of state-trajectories,  $\mathcal{U}$  the set of input trajectories,  $\mathcal{Y}$  the set of output trajectories and  $\mathcal{P}$  the set of scheduling signals.

For the continuous-time case,  $\mathcal{X} = AC(\mathbb{R}_0^+, \mathbb{X})$  is the set of absolutely continuous functions on the real time axis  $\mathbb{R}_+$  with values in the state-space  $\mathbb{X}$ , and  $\mathcal{Y} = PC(\mathbb{R}_0^+, \mathbb{Y})$ ,  $\mathcal{U} = PC(\mathbb{R}_0^+, \mathbb{U})$ ,  $\mathcal{P} = PC(\mathbb{R}_0^+, \mathbb{P})$  are the sets of piecewise continuous functions with values in the set of outputs  $\mathbb{Y}$ , inputs  $\mathbb{U}$  and scheduling variables  $\mathbb{P}$  respectively.

In the discrete-time case  $\mathcal{X} = \mathbb{X}^{\mathbb{N}}$ ,  $\mathcal{Y} = \mathbb{Y}^{\mathbb{N}}$ ,  $\mathcal{U} = \mathbb{U}^{\mathbb{N}}$ ,  $\mathcal{P} = \mathbb{P}^{\mathbb{N}}$ , i.e., they are the sets of functions (sequences) defined on natural numbers  $\mathbb{N}$  with values in  $\mathbb{X}$ ,  $\mathbb{Y}$ ,  $\mathbb{U}$  and  $\mathbb{P}$  respectively.

By a solution of  $\Sigma$  we mean a tuple of trajectories  $(x, y, u, p) \in (\mathcal{X}, \mathcal{Y}, \mathcal{U}, \mathcal{P})$  such that (2.1) holds for all  $t \in \mathbb{T}$ . For an initial state  $x_0 \in \mathbb{X}$  define the *input-to-state map*  $\mathfrak{X}_{\Sigma, x_0}$  and the *input-output map*  $\mathfrak{Y}_{\Sigma, x_0}$  of  $\Sigma$  induced by  $x_0$  as

$$\mathfrak{X}_{\Sigma, x_0} : \mathcal{U} \times \mathcal{P} \rightarrow \mathcal{X}, \quad \mathfrak{Y}_{\Sigma, x_0} : \mathcal{U} \times \mathcal{P} \rightarrow \mathcal{Y}, \quad (2.3)$$

such that for any  $(x, y, u, p) \in \mathcal{X} \times \mathcal{Y} \times \mathcal{U} \times \mathcal{P}$ ,  $x = \mathfrak{X}_{\Sigma, x_0}(u, p)$  and  $y = \mathfrak{Y}_{\Sigma, x_0}(u, p)$  holds if and only if  $(x, y, u, p)$  is a solution of (2.1) and  $x(0) = x_0$ .

We say that  $\Sigma$  is *span-reachable* from the zero initial state (*span-reachable* for short), if

$$\text{Span}\{\mathfrak{X}_{\Sigma, 0}(u, p)(t) \mid (u, p) \in \mathcal{U} \times \mathcal{P}, t \in \mathbb{T}\} = \mathbb{X}.$$

That is, span-reachability means that the linear span of all the states reachable from the zero initial with some input and some scheduling generates the whole state-space.

We say that  $\Sigma$  is *observable*, if for any two initial states  $\bar{x}_0, \hat{x}_0 \in \mathbb{R}^{n_x}$ ,  $\mathfrak{Y}_{\Sigma, \hat{x}_0} = \mathfrak{Y}_{\Sigma, \bar{x}_0}$  implies  $\hat{x}_0 = \bar{x}_0$ . That is, observability means that for any two distinct states of the system, the resulting outputs will be different for some input and scheduling signals.

Let  $\Sigma$  of the form (2.1) and  $\Sigma' = (\mathbb{P}, \{A'_i, B'_i, C'_i\}_{i=0}^{n_p})$  with  $\dim(\Sigma) = \dim(\Sigma') = n_x$ . A nonsingular matrix  $T \in \mathbb{R}^{n_x \times n_x}$  is said to be an *isomorphism* from  $\Sigma$  to  $\Sigma'$ , if

$$\forall i \in \{0, \dots, n_p\} : A'_i T = T A_i, \quad B'_i = T B_i, \quad C'_i T = C_i,$$

We formalize the input-output behavior of LPV-SSAs as maps of the form

$$\mathfrak{F} : \mathcal{U} \times \mathcal{P} \rightarrow \mathcal{Y}. \quad (2.4)$$

While any input-output map of an LPV-SSA induced by some initial state is of the above form, the converse is not true. Moreover, for the sake of simplicity, we will concentrate on input-output maps which can be induced by an LPV-SSA from the *zero initial state*. That is, the LPV-SSA  $\Sigma$  is a *realization* of an input-output map  $\mathfrak{F}$  of the form (2.4), if  $\mathfrak{F} = \mathfrak{Y}_{\Sigma, 0}$ .

An LPV-SSA  $\Sigma$  is a *minimal realization* of  $\mathfrak{F}$ , if  $\Sigma$  is a realization of  $\mathfrak{F}$ , and for every LPV-SSA  $\Sigma'$  which is a realization of  $\mathfrak{F}$ ,  $\dim(\Sigma) \leq \dim(\Sigma')$ .

To make our terminology less cumbersome, we will refer to the input-output map of an LPV-SSA  $\Sigma$  induced by the zero initial state as the *input-output map* of  $\Sigma$ . That is, the input-output map of  $\Sigma$  is the input-output map  $\mathfrak{Y}_{\Sigma, 0}$  induced by the zero initial state, and it is *denoted* by  $\mathfrak{Y}_{\Sigma}$ . We say that an LPV-SSA  $\Sigma$  is *minimal*, if it is a minimal LPV-SSA realization of its own input-output map  $\mathfrak{F} = \mathfrak{Y}_{\Sigma, 0}$ . We say that two LPV-SSAs  $\Sigma_1, \Sigma_2$  are *input-output equivalent*, if their input-output maps are equal, i.e.,  $\mathfrak{Y}_{\Sigma_1, 0} = \mathfrak{Y}_{\Sigma_2, 0}$ .

All the notions introduced above can be extended to include non-zero initial states, see [216].

## 2.4 LPV-SSAs and linear switched systems

In this section we will present the relationship between LPV-SSA and linear switched systems (abbreviated by *LSS* in the sequel). We will use the terminology of [202, 203, 219, 209, 205], when discussing LSSs, see Appendix B for a brief overview.

If  $\mathbb{P}$  is finite, then an LPV-SSA can naturally be interpreted as a LSS, and vice versa, and LSS can easily be viewed as an LPV-SSA scheduling space of which is finite, and the definitions above coincide with the corresponding definitions for LSSs. Hence, realization theory of LPV-SSA clearly implies realization theory of LSSs. In fact, we can identify an LPV-SSA of the form (2.1) with the LSS

$$\mathfrak{G}(\Sigma) : \begin{cases} \xi z(t) = A_{\sigma(t)} z(t) + B_{\sigma(t)} v(t) \\ v(t) = C_{\sigma(t)} z(t) \end{cases} \quad (2.5)$$

## 2.5. REACHABILITY, OBSERVABILITY, MINIMALITY

where  $\sigma(t) \in Q$ ,  $Q = \{0, \dots, n_p\}$  is the set of discrete mode, and for each discrete mode  $i \in Q$ , the linear system belonging to this discrete mode is determined by the matrices  $(A_i, B_i, C_i)$  of the LPV-SSA. The  $\mathfrak{S}(\Sigma)$  above can also be viewed as an LPV-SSA

$$(\mathbb{P}_{sw}, \{\hat{A}_i, \hat{B}_i, \hat{C}_i\}_{i=0}^{n_p}), \mathbb{P}_{sw} = \{0, e_1, \dots, e_{n_p}\},$$

$e_i$  is the  $i$ th standard unit vector of  $\mathbb{R}^{n_p}$ , and

$$(\hat{A}_0, \hat{B}_0, \hat{C}_0) = (A_0, B_0, C_0), \quad (\hat{A}_i, \hat{B}_i, \hat{C}_i) = (A_i - A_0, B_i - B_0, C_i - C_0), \quad i = 1, \dots, n_p.$$

This latter point of view was used in [216]. We call  $\mathfrak{S}(\Sigma)$  the *LSS associated with the LPV-SSA*  $\Sigma$ .

In particular, if  $A_0 = 0, B_0 = 0, C_0 = 0$  and  $\mathbb{P}$  contains 0 and the standard basis unit vectors  $\{e_1, \dots, e_{n_p}\}$ , then the LSS associated with  $\Sigma$  is just the restriction of  $\Sigma$  to scheduling signals which take as values the unit vectors.

Conversely, with any LSS

$$H \begin{cases} (\xi x)(t) = A_{\sigma(t)}x(t) + B_{\sigma(t)}u(t) \\ y(t) = C_{\sigma(t)}x(t) \end{cases} \quad (2.6)$$

defined over the set of discrete modes  $Q = \{0, \dots, n_p\}$ , with  $A_q \in \mathbb{R}^{n \times n}$ ,  $B_q \in \mathbb{R}^{n \times m}$ ,  $C_q \in \mathbb{R}^{p \times n}$ ,  $q \in Q$ , we can associate the LPV-SSA  $\mathfrak{L}(H)$  of the form (2.1) such that the matrix functions  $A, B, C$  satisfy (2.2). We call  $\mathfrak{L}(H)$  the *LPV-SSA associated with the LSS*  $H$ . Note that  $H$  can also be viewed as an LPV-SSA

$$(\mathbb{P}_{sw}, \{\hat{A}_i, \hat{B}_i, \hat{C}_i\}_{i=0}^{n_p}), \mathbb{P}_{sw} = \{0, e_1, \dots, e_{n_p}\} \\ (\hat{A}_0, \hat{B}_0, \hat{C}_0) = (A_0, B_0, C_0), \quad (\hat{A}_i, \hat{B}_i, \hat{C}_i) = (A_i - A_0, B_i - B_0, C_i - C_0), \quad i = 1, \dots, n_p.$$

Moreover, the LSS associated with the LPV-SSA  $\mathfrak{L}(H)$  is the LSS  $H$  itself. In the same manner, if we consider an LPV-SSA  $\Sigma$  and the LSS  $\mathfrak{S}(\Sigma)$  associated with  $\mathfrak{S}(\Sigma)$ , then the LPV-SSA associated  $\mathfrak{S}(\Sigma)$  is  $\Sigma$  itself.

The correspondence between LPV-SSAs and LSSs described above make it possible to develop realization theory of LPV-SSA based on that of LSSs. In particular, the conditions for minimality, observability and span-reachability for LPV-SSA can be derived from the corresponding results for LSSs. There is a one-to-one correspondence between input-output maps of LPV-SSAs and their associated LSSs, hence conditions for existence of a realization by an LPV-SSA can be derived from the corresponding conditions for LSSs.

Below we will elaborate on this results in more details.

## 2.5 Reachability, observability, minimality

The results on (span-)reachability, observability and minimality for LPV-SSAs rely on those of for LSSs presented in [202, 203, 219, 209, 205] (see also Theorem B.2 of Appendix B) and on the following relationship

**Theorem 2.1** ([216]). *Let  $\Sigma$  be an LPV-SSA.*

- $\Sigma$  is span-reachable if and only if the associated LSS  $\mathfrak{S}(\Sigma)$  is span-reachable.
- $\Sigma$  is minimal if and only if the associated LSS  $\mathfrak{S}(\Sigma)$  is minimal.
- Two LPV-SSAs  $\Sigma_1$  and  $\Sigma_2$  are input-output equivalent, if and only if the associated LSSs  $\mathfrak{S}(\Sigma_1)$  and  $\mathfrak{S}(\Sigma_2)$  are input-output equivalent.
- Two LPV-SSAs  $\Sigma_1$  and  $\Sigma_2$  are isomorphic if and only if the LSSs  $\mathfrak{S}(\Sigma_1)$  is isomorphic to  $\mathfrak{S}(\Sigma_2)$ .

Using Theorem 2.1, the results on minimality of LSSs from [202, 203, 219, 209, 205] (see Theorem B.1 of Appendix B) the following characterization of minimal LPV-SSA realizations can be derived.

**Theorem 2.2** (Minimal realizations, [216]). *Assume that  $\mathfrak{F}$  is an input-output map of the form (2.4). Assume that the LPV-SSA  $\Sigma$  is a realization of  $\mathfrak{F}$ . Then  $\Sigma$  is a minimal realization of  $\mathfrak{F}$ , if and only if  $\Sigma$  is observable and span-reachable. Any two minimal LPV-SSA realizations of  $\mathfrak{F}$  are isomorphic.*

Note that Theorem 2.2 does not exclude the possibility that two LPV state-space representations of the same input-output map are related by a non-constant isomorphism, if these state-space representations are not minimal or they are not LPV-SSAs, see [153].

In fact, since observability respectively span-reachability of an LPV-SSA are equivalent with observability respectively span-reachability of the associated LSS, we can use the algebraic rank conditions and geometric conditions from [202, 203, 219, 209, 205] (Theorem B.2 of Appendix B) to formulate necessary and sufficient conditions for span-reachability and observability of LPV-SSAs. As a particular example, we present below a set of conditions which rely on the rank of extended observability and reachability matrices as defined in [275].

**Definition 2.2** (Ext. reachability & observability matrices). *The  $n$ -step extended reachability matrices  $\mathcal{R}_n$  of  $\Sigma$ ,  $n \in \mathbb{N}$ , are defined recursively as follows*

$$\mathcal{R}_0 = [ B_0 \quad \dots \quad B_{n_p} ], \quad \mathcal{R}_{n+1} = [ \mathcal{R}_0 \quad A_0 \mathcal{R}_n \quad \dots \quad A_{n_p} \mathcal{R}_n ]. \quad (2.7a)$$

*The extended  $n$ -step observability matrices  $\mathcal{O}_n$  of  $\Sigma$ ,  $n \in \mathbb{N}$ , are defined recursively as follows*

$$\mathcal{O}_0 = [ C_0^\top \quad \dots \quad C_{n_p}^\top ]^\top, \quad \mathcal{O}_{n+1} = [ \mathcal{O}_0^\top \quad A_0^\top \mathcal{O}_n^\top \quad \dots \quad A_{n_p}^\top \mathcal{O}_n^\top ]^\top. \quad (2.8)$$

We can now present the following characterization of span-reachability and observability of LPV-SSAs.

**Theorem 2.3** (Rank conditions, [216]). *The LPV-SSA  $\Sigma$  is span-reachable, if and only if  $\text{rank}\{\mathcal{R}_{n_x-1}\} = n_x$ , and  $\Sigma$  is observable, if and only if  $\text{rank}\{\mathcal{O}_{n_x-1}\} = n_x$ .*

Theorem 2.1 indicates that we can use the minimization algorithms for LSSs described in [202, 203, 219, 209, 205]. To this end, it is sufficient to calculate the LSS associated with the given LPV-SSA, apply to this latter LSS the minimization algorithm from [202, 203, 219, 209, 205], and then calculate the LPV-SSA which is associated with the LSS produced by the minimization algorithm in the previous step. The resulting LPV-SSA will be a minimal dimensional realization of the input-output function of the original LPV-SSA.

There are several versions of the minimization algorithm for LSSs. One of them uses Kalman-decomposition [202, 260]. This latter minimization algorithm can be reformulated in terms of extended reachability and observability matrices. For the sake of completeness, and in order to illustrate the discussion above, below we present that particular algorithm, formulated directly in terms of LPV-SSAs and the  $n$ -step extended reachability and observability matrices.

Choose a basis  $\{b_i\}_{i=1}^{n_x} \subset \mathbb{R}^{n_x}$  such that  $\text{Span}\{b_1, \dots, b_r\} = \text{Im}\{\mathcal{R}_{n_x-1}\}$  and  $\text{Span}\{b_{r_m+1}, \dots, b_r\} = (\text{Im}\{\mathcal{R}_{n_x-1}\} \cap \ker\{\mathcal{O}_{n_x-1}\})$  for some  $r, r_m \geq 0$ . Define  $T = [b_1 \quad b_2 \quad \dots \quad b_{n_x}]^{-1}$ , and let

$$\hat{A}_i = TA_i T^{-1}, \quad \hat{B}_i = TB_i, \quad \hat{C}_i = C_i T^{-1}, \quad i = 0, 1, \dots, n_p$$

Then it is easy to see that

$$\hat{A}_i = \begin{bmatrix} A_i^m & 0 & A_i'' \\ A_i' & \hat{A}' & A_i''' \\ 0 & 0 & A_i'''' \end{bmatrix}, \quad \hat{B}_i = \begin{bmatrix} B_i^m \\ B_i' \\ 0 \end{bmatrix}, \quad \hat{C}_i = \begin{bmatrix} (C_i^m)^\top \\ 0 \\ (C_i')^\top \end{bmatrix}^\top, \quad (2.9)$$

where  $A_i^m \in \mathbb{R}^{r_m \times r_m}$ ,  $B_i^m \in \mathbb{R}^{r_m \times n_u}$ ,  $C_i^m \in \mathbb{R}^{n_y \times r_m}$ ,  $A_i'' \in \mathbb{R}^{(n-r) \times (n-r)}$ ,  $B_i' \in \mathbb{R}^{(r-r_m) \times n_u}$ ,  $C_i' \in \mathbb{R}^{n_y \times (n-r)}$ . Clearly,  $\hat{\Sigma} = (\mathbb{P}, \{\hat{A}_i, \hat{B}_i, \hat{C}_i\}_{i=0}^{n_p})$  is isomorphic to  $\Sigma$  and can be viewed as the Kalman-decomposition of  $\Sigma$ . The following corollary explains how to use the transformation above for minimizing LPV-SSAs.

**Corollary 2.1** (Minimization). *With the notation of (2.9), the LPV-SSA  $\Sigma^m = (\mathbb{P}, \{A_i^m, B_i^m, C_i^m\}_{i=0}^{n_p})$  is a minimal realization of  $\mathfrak{F} = \mathfrak{Y}_\Sigma$ .*

In order to illustrate the procedure above, we present the following example.

**Example 2.1** (Illustrative example). *Consider an LPV-SSA  $\Sigma$  as in (2.1), with  $\mathbb{P} = \mathbb{R}$ ,*

$$A_0 = \begin{bmatrix} 0 & 1 & -1 \\ -1 & 0 & 1 \\ -1 & 1 & 0 \end{bmatrix}, B_0 = \begin{bmatrix} 1 \\ 0 \\ 0 \end{bmatrix}, C_0 = \begin{bmatrix} 1 \\ -1 \\ -1 \end{bmatrix}^\top,$$

$$A_1 = \begin{bmatrix} 1 & -1 & -1 \\ -1 & 2 & 0 \\ -1 & 0 & 2 \end{bmatrix}, B_1 = \begin{bmatrix} 0 \\ 1 \\ 1 \end{bmatrix}, C_1 = \begin{bmatrix} 2 \\ -2 \\ -2 \end{bmatrix}^\top.$$

*It is easy to see that  $\text{rank}\{\mathcal{R}_2\} = 2$  and  $\text{rank}\{\mathcal{O}_2\} = 1$ . If we set,  $b_1 = [1 \ 0 \ 0]^\top$ ,  $b_2 = [2 \ 1 \ 1]^\top$ ,  $b_3 = [0 \ 0 \ 1]^\top$ , then  $\{b_1, b_2\}$  span  $\text{Im}\{\mathcal{R}_2\}$  and  $b_2$  spans  $\text{Im}\{\mathcal{R}_2\} \cap \ker\{\mathcal{O}_2\}$ . If we apply the basis transformation  $T = [b_1 \ b_2 \ b_3]^{-1}$ , then we obtain the matrices  $\hat{A}_i = TA_iT^{-1}$ ,  $\hat{B}_i = TB_i$ ,  $\hat{C}_i = C_iT^{-1}$ ,  $i = 0, 1$  which are of the form (2.9), with*

$$A_0^m = 2, A_1^m = 3, B_0^m = 1, B_1^m = -2, C_0^m = 1, C_1^m = 2$$

*By Corollary 2.1,  $\Sigma^m = (\mathbb{P}, \{A_i^m, B_i^m, C_i^m\}_{i=0}^{n_p})$  is a minimal realization of the input-output function  $\mathfrak{Y}_{\Sigma,0} = \mathfrak{F}$  from the initial state zero.*

The matrix  $T$  and hence  $\Sigma^m$  can easily be computed from  $\Sigma$ . Note that for computing  $\Sigma^m$ , or checking the rank conditions of Theorem 2.3, it is not necessary to compute the matrices  $\mathcal{R}_{n_x-1}$  and  $\mathcal{O}_{n_x-1}$ , instead it is sufficient to compute a basis of  $\text{Im}\{\mathcal{R}_{n_x-1}\}$  and  $\ker\{\mathcal{O}_{n_x-1}\}$ , which can be done in polynomial time w.r.t.  $n_p$  and  $n_x$  [28, Algorithm 1 – Algorithm 2].

## 2.6 Existence of a realization, sub-Markov parameters and Hankel-matrix

In order to formulate conditions of existence of an LPV-SSA realization, we have to define the class of input-output maps which could potentially arise from LPV-SSAs. This class of input-output maps will be the one which has *impulse response representation (IIR)*.

To this end, we will use the following notation. In accordance with the notation defined in Section 2.2, we will denote by  $\{0, 1, \dots, n_p\}^*$  the set of all finite sequences of elements of  $\{0, 1, \dots, n_p\}$  (i.e. words over the finite alphabet  $\mathbb{I}_{0,n_p}$ ). We will use the notation and terminology of Section 2.2 regarding words, the empty word, the length and concatenation of two words, etc. Let  $\mathfrak{F}$  be of the form (2.4). For the purposes of this chapter it is sufficient to know that if  $\mathfrak{F}$  has *impulse response representation (IIR)*, then

$$\mathfrak{F}(u, p)(t) = \begin{cases} \int_0^t (h_{\mathfrak{F}} \diamond p)(\delta, t) u(\delta) d\delta & \text{continuous-time} \\ \sum_{\delta=0}^{t-1} (h_{\mathfrak{F}} \diamond p)(\delta, t) u(\delta) & \text{discrete-time} \end{cases} \quad (2.10)$$

for a suitable parameter-dependent function  $(h_{\mathfrak{F}} \diamond p)(\delta, t)$  which will be discussed below. Intuitively, in discrete-time (DT) the latter function is a linear combination of products of the entries of  $p(t), \dots, p(\delta)$ . In continuous-time, this function is a linear combination of iterated integrals of past values of  $p$ . More precisely, we assume that there exists functions, called the *sub-Markov parameters* of  $\mathfrak{F}$ ,

$$\theta_{i,j,\mathfrak{F}} : \{0, 1, \dots, n_p\}^* \rightarrow \mathbb{R}^{n_y \times n_u}, \quad i, j \in \{0, 1, \dots, n_p\}, \quad (2.11)$$

such that the following holds. In discrete-time

$$(h_{\mathfrak{F}} \diamond p)(\delta, t) = \sum_{i,j=0}^{n_p} p_i(t) p_j(\delta) \left[ \theta_{i,j,\mathfrak{F}}(\varepsilon) + \sum_{s_1, \dots, s_{t-\delta-1}=0}^{n_p} \theta_{i,j,\mathfrak{F}}(s_1 \cdots s_{t-\delta-1}) \prod_{i=1}^k p_{s_i}(t + \delta) \right], \quad (2.12)$$

and in continuous-time,

$$(h_{\mathfrak{F}} \diamond p)(\delta, t) = \sum_{i,j=0}^{n_p} p_i(t)p_j(\delta) \left[ \theta_{i,j,\mathfrak{F}}(\varepsilon) + \sum_{k=1}^{\infty} \sum_{s_1, \dots, s_k=0}^{n_p} \theta_{i,j,\mathfrak{F}}(s_1 \cdots s_k) \int_{\delta}^t p_{s_k}(\tau_1) \int_{\delta}^{\tau_1} p_{s_{k-1}}(\tau_2) \cdots \int_{\delta}^{\tau_{k-1}} p_{s_k}(\tau_k) d\tau_1 \cdots d\tau_k \right] \quad (2.13)$$

Note that in discrete-time the sum in (2.12) are actually finite, while for continuous-time the sum in (2.13) is infinite. For the convergence of the latter we need to impose some mild constraints on  $\theta_{i,j}$ , see [216] for more details. These conditions hold automatically for input-output functions which arise from LPV-SSAs.

That is,  $(h \diamond p)(\delta, t)$  is sum of products (DT) or iterated integrals (CT) of scheduling signals, and the sub-Markov parameters

**Example 2.2** (Illustrative example  $w_s \diamond p$ ). *In order to illustrate the notation above, consider the case when  $n_p = 1$ . In this case, for discrete-time,  $(h_{\mathfrak{F}} \diamond p)(2, 5)$  is of the form*

$$\theta_{0,0,\mathfrak{F}}(00) + p(4)\theta_{0,0,\mathfrak{F}}(01) + \cdots + p(2)p(5)p(3)p(4)\theta_{1,1,\mathfrak{F}}(11).$$

*For continuous-time,  $(h_{\mathfrak{F}} \diamond p)(2, 5)$  is of the form*

$$\theta_{0,0,\mathfrak{F}}(\varepsilon) + \cdots + p(5)p(2)\theta_{1,1,\mathfrak{F}}(\varepsilon) + \cdots + p(2)p(5)\theta_{1,1,\mathfrak{F}}(101) \int_2^5 p(s_1) \int_2^{s_1} \int_2^{s_2} p(s_3) ds_3 ds_2 ds_1 + \cdots$$

*That is, in discrete-time,  $(h_{\mathfrak{F}} \diamond p)(2, 5)$  is a polynomial of  $p(2), p(3), p(4), p(5)$ , while in continuous-time, it is an infinite sum of iterated integrals.*

**Example 2.3** (Illustrative example: IIR representation). *Next, we illustrate how the sub-Markov parameters and the maps  $(h_{\mathfrak{F}} \diamond p)$ ,  $(g_{\mathfrak{F}} \diamond p)$  relate to  $\mathfrak{F}$ . Let  $n_u = n_y = 1$  and let  $\mathfrak{F}$  be an input-output map of the form (2.4) in continuous-time defined as follows:*

$$\mathfrak{F}(u, p)(t) = \int_0^t (1 + 2p(t)) e^{2(t-\tau) + 2 \int_{\tau}^t p(s) ds} (1 - 2p(\tau)) u(\tau) d\tau$$

*Then  $\mathfrak{F}$  admits an IIR with*

$$(h_{\mathfrak{F}} \diamond p)(\tau, t) = (1 + 2p(t)) e^{2(t-\tau) + 3 \int_{\tau}^t p(s) ds} (1 - 2p(\tau)),$$

*and for every  $s \in \mathbb{I}_{0,n_p}^*$  which contains  $k$  0's and  $l$  1's,  $\theta_{\mathfrak{F}}(s) = 2^k 3^l \begin{bmatrix} 1 & -2 \\ 2 & -4 \end{bmatrix}$ . The LPV-SSA  $\Sigma$  from Example 2.1 is a realization of  $\mathfrak{F}$*

We can show that there is a one-to-one correspondence between input-output maps admitting an IIR and sub-Markov parameters. It turns out that any input-output map which is realizable by an LPV-SSA admits an IIR, and the sub-Markov parameters can be expressed via the matrices of this LPV-SSA realization. More precisely, if an LPV-SSA of the form (2.1) is a realization of the input-output map  $\mathfrak{F}$ , then  $\mathfrak{F}$  has an IIR such that for any  $p \in \mathcal{P}$ ,

$$(h_{\mathfrak{F}} \diamond p)(s, t) = C_{p(t)} \Phi_p(t, s) B_{p(s)}$$

where  $\Phi_p : \{(t, \tau) \in \mathbb{T} \times \mathbb{T} \mid \tau \leq t\} \rightarrow \mathbb{R}^{n_x \times n_x}$  is defined as follows. In discrete-time,

$$\Phi_p(t, s) = A(p(t-1))A(p(t-2)) \cdots A(p(s+1)).$$

In continuous-time,

$$X_{p,s} : [s, \infty) \ni t \mapsto \Phi_p(t, s)$$

is the unique absolutely continuous matrix-valued function which satisfies

$$\frac{d}{dt} X_{p,s}(t) = A(p(t))X_{p,s}(t), \quad X_{p,s}(s) = I_n.$$

Note that sub-Markov parameters are closely related to generating Fliess-series used in non-linear systems theory [126, 91], more specifically, the sub-Markov parameters play the role similar to generating series in the Fliess-series expansion of the input-output function  $\mathfrak{F}$ . If say  $\theta_{i,j,\mathfrak{F}} = 0$  for  $i \neq 0$  or  $j \neq 0$ , and we assume single input, i.e.,  $n_u = 1$ , then existence of IIR means that the function which maps the restriction of  $p$  to  $(\delta, t)$  to  $(h_{\mathfrak{F}} \diamond p)(\delta, t)$  has a Fliess-series expansion with  $\theta_{0,0,\mathfrak{F}}$  being the corresponding generating series. In general case,  $\theta_{i,j,\mathfrak{F}} p_i(t) p_j(\delta)$  can be viewed as the generating series of the Fliess-series expansion of the  $(i, j)$ th entry of  $(h_{\mathfrak{F}} \diamond p)(\delta, t)$ . Just like in the case of generating series for Fliess-series expansion, it turns out that the sub-Markov parameters uniquely determine the corresponding input-output function. In fact, the existence of an LPV-SSA realizing an input-output function can be shown to be equivalent to the sub-Markov parameters being of a certain form, as explained below.

**Lemma 2.1** (Existence of the IIR, [216]). *The LPV-SSA  $\Sigma$  of the form (2.1) is a realization of an input-output map  $\mathfrak{F}$ , if and only if  $\mathfrak{F}$  has an IIR and for all  $i, j \in \{0, 1, 2, \dots, n_p\}$ ,  $s \in \{0, 1, 2, n_p\}^*$ ,*

$$\theta_{i,j,\mathfrak{F}}(s) = C_i A_s B_j, \quad (2.14)$$

where  $A_s$  is defined as in Notation 2.1, Section 2.2, i.e. for  $s = \varepsilon$ ,  $A_s$  is the identity matrix, and for  $s = s_1 s_2 \cdots s_n$ ,  $s_1, \dots, s_n \in \{0, 1, \dots, n_p\}$ ,  $n > 0$ ,  $A_s = A_{s_n} A_{s_{n-1}} \cdots A_{s_1}$ .

The lemma above allows us to formulate a link between sub-Markov parameters of LPV-SSAs and generalized Markov parameters of linear switched systems. Let us consider the LSS  $\mathfrak{S}(\Sigma)$  associated with the LPV-SSA  $\Sigma$  and let us recall from [202, 203, 219, 209, 205] or Appendix B the notion of generalized Markov parameters of the input-output function of  $\mathfrak{S}(\Sigma)$  induced by the initial state zero. In fact, (2.14) means that sub-Markov parameters of the input-output map of an LPV-SSA  $\Sigma$  correspond to the Markov parameters of the input-output map of the associated LSS  $\mathfrak{S}(\Sigma)$ . This allows to relate the problem of existence of a LPV-SSA realization with the problem of existence of a LSS realization of a certain associated input-output function which acts on switching signals instead of scheduling signals. Moreover, using the correspondence between generalized Markov parameters of input-output functions of LSSs and sub-Markov parameters we can formulate Ho-Kalman-like realization algorithms similar to those described in [202, 203, 219, 209, 205].

More precisely, we can define the *switched input-output function*  $\mathfrak{S}(\mathfrak{F})$  associated with  $\mathfrak{F}$  as follows. Let  $Q = \{0, 1, \dots, n_p\}$  be the set of discrete modes, let  $\sigma : \mathbb{T} \rightarrow Q$  be a switching signal and let  $u \in \mathcal{U}$  be a control input. For continuous-time, assume that for any  $q \in Q$ ,  $\sigma^{-1}(q)$  is either an empty set or an interval of the form  $[s, \tau)$ ,  $s < \tau$ ,  $s, \tau \in \mathbb{T}$ . The define  $\mathfrak{S}(\mathfrak{F})(u, p) = y$  as follows:

$$y(t) = \begin{cases} \sum_{j=0}^{t-1} \theta_{\sigma(t), \sigma(j), \mathfrak{F}}(\sigma_{j+1} \cdots \sigma_{t-1}) u(j) & \text{discrete-time} \\ \sum_{i=0}^k \int_0^{T_i} G_{q_k \cdots q_i}(T_i - T_{i-1} - s, T_{i+1} - T_i, \dots, T_k - T_{k-1}) u(s + T_i) & \text{continuous-time} \end{cases} \quad (2.15)$$

where in continuous-time,  $\sigma(s) = q_i$  for  $s \in [T_{i-1}, T_i)$ ,  $0 = T_0 < T_1 < \cdots < T_{k-1} < T_k = t$ , and

$$G_{q_k \cdots q_i}(\tau_i, \dots, \tau_k) = \sum_{j_i, \dots, j_k=0}^{\infty} \theta_{q_k, q_i, \mathfrak{F}}(q_i^{j_i} \cdots q_k^{j_k}) \prod_{r=i}^k \frac{\tau_r^{j_r}}{j_r!} \quad (2.16)$$

with  $q^l$  stands for the sequences consisting of  $l$  copies of  $q$ , and  $q^0$  is the empty sequence. Note that the additional requirements [216] on IIR means that the infinite sum in (2.16) is absolutely convergent. Using the terminology



of [202, 203, 219, 209, 205],  $\mathfrak{S}(\mathfrak{F})$  admits a generalized kernel representation and its Markov-parameters satisfy  $S_{i,j}^{\mathfrak{S}(\mathfrak{F})} = \theta_{i,j,\mathfrak{F}}$  for all  $i, j \in \{0, 1, \dots, n_p\} = Q$ . Note that since Markov parameters determine an input-output map admitting a generalized kernel representation uniquely,  $\mathfrak{S}(\mathfrak{F})$ , if exists, is unique. Finally, the existence of  $\mathfrak{S}(\mathfrak{F})$  for discrete-time is obvious, and for continuous-time is proven in [216]. The LPV-SSA realizations of  $\mathfrak{F}$  and the LSS realizations of  $\mathfrak{S}(\mathfrak{F})$  are related as follows.

**Theorem 2.4** ([216]). *An LPV-SSA  $\Sigma$  is a realization of  $\mathfrak{F}$ , if and only if the associated LSS  $\mathfrak{S}(\Sigma)$  is a realization of  $\mathfrak{S}(\mathfrak{F})$ . An LSS  $H$  is a realization of  $\mathfrak{S}(\mathfrak{F})$  if and only if the associated LPV-SSA  $\mathfrak{L}(H)$  is a realization of  $\mathfrak{F}$ .*

That is, existence of an LPV-SSA realization of the input-output map  $\mathfrak{F}$  is equivalent to existence of an LSS realization of the associated switched input-output map  $\mathfrak{S}(\mathfrak{F})$ . This means that we can re-use the results of [202, 203, 219, 209, 205] on existence of an LSS realization and to calculate an LPV-SSA realization.

For the sake of completeness, we state the results below. Define the *Hankel-matrix* of  $\mathfrak{F}$  as

$$H_{\mathfrak{F}} = \begin{bmatrix} \theta_{\mathfrak{F}}(v_1 v_1) & \theta_{\mathfrak{F}}(v_2 v_1) & \cdots & \theta_{\mathfrak{F}}(v_k v_1) & \cdots \\ \theta_{\mathfrak{F}}(v_1 v_2) & \theta_{\mathfrak{F}}(v_2 v_2) & \cdots & \theta_{\mathfrak{F}}(v_k v_2) & \cdots \\ \theta_{\mathfrak{F}}(v_1 v_3) & \theta_{\mathfrak{F}}(v_2 v_3) & \cdots & \theta_{\mathfrak{F}}(v_k v_3) & \cdots \\ \vdots & \vdots & \cdots & \vdots & \cdots \end{bmatrix},$$

where  $v_1 \prec v_2 \prec \cdots$  is the lexicographic ordering of  $\{0, 1, \dots, n_p\}^*$ , as defined in Definition 2.1 for  $X = \{0, 1, \dots, n_p\}$ , and  $\theta_{\mathfrak{F}} : \{0, 1, \dots, n_p\}^* \rightarrow \mathbb{R}^{n_y \times n_u}$  such that for all  $s \in \{0, 1, \dots, n_p\}^*$ ,

$$\theta_{\mathfrak{F}}(s) = \begin{bmatrix} \theta_{0,0,\mathfrak{F}}(s) & \cdots & \theta_{0,n_p,\mathfrak{F}}(s) \\ \theta_{1,0,\mathfrak{F}}(s) & \cdots & \theta_{1,n_p,\mathfrak{F}}(s) \\ \vdots & \cdots & \vdots \\ \theta_{n_p,0,\mathfrak{F}}(s) & \cdots & \theta_{n_p,n_p,\mathfrak{F}}(s) \end{bmatrix}. \quad (2.17)$$

Note that  $\theta_{\mathfrak{F}}(s)$  coincides with the matrix which is obtained from the Markov-parameter  $M^{\mathfrak{S}(\mathfrak{F})}(s)$  of  $\mathfrak{S}(\mathfrak{F})$  as defined in [205, eq. (3.14)] by removing the first column of  $M^{\mathfrak{S}(\mathfrak{F})}(s)$ , if we identify  $q \in Q$  with the integer  $q + 1$ , for all  $q \in Q$ . Alternatively,  $\theta_{\mathfrak{F}}$  coincides with the matrix  $M^{\mathfrak{S}(\mathfrak{F})}(s)$  defined in (B.9) of Appendix B. Note that the latter column has all zero elements, as it corresponds to the effect of the initial state, and the latter is zero in out setting. Hence, the Hankel-matrix  $H_{\mathfrak{F}}$  equals the Hankel-matrix  $H_{\mathfrak{S}(\mathfrak{F})}$  of the associated switched input-output map  $\mathfrak{S}(\mathfrak{F})$  as defined in [205, Definition 26], after removing the all zero columns which arise from the first column of  $M^{\mathfrak{S}(\mathfrak{F})}$ . Alternatively,  $H_{\mathfrak{F}}$  coincides with the the matrix  $H^{\mathfrak{S}(\mathfrak{F})}$  as defined in Definition B.2 of Appendix B. Then using [205, Theorem 3] (Theorem B.3 of Appendix B), and the one-to-one correspondence between LSS realizations of  $\mathfrak{S}(\mathfrak{F})$  we can state the following characterization of existence of a LPV-SSA realization.

**Theorem 2.5** (Existence of a realization [216]). *An input-output map  $\mathfrak{F}$  has a LPV-SSA realization, if and only if  $\mathfrak{F}$  has an IIR and  $\text{rank } \mathcal{H}_{\mathfrak{F}} = n_{\mathfrak{F}} < \infty$ . Any minimal LPV-SSA realization of  $\mathfrak{F}$  has a state dimension which equals  $n_{\mathfrak{F}}$ .*

In addition to existence results, the correspondence between LPV-SSAs and LSSs allows us to use realization algorithms for LSSs for LPV-SSAs. This will be discussed in the next section.

## 2.7 Ho-Kalman realization algorithms

The correspondence between LSSs and LPV-SSA also allows us to formulate Ho-Kalman-like algorithm for computing a minimal LPV-SSA realization of an input-output map. in particular, [205, Algorithm 1] (see also Algorithm 12 in Appendix B). More precisely, by reinterpreting the sub-Markov parameters of a potential input-output function of a LPV-SSA as generalized Markov-parameters of the associated switched input-output function, we can compute a minimal LSS from a finite Hankel-like matrix constructed from sub-Markov parameters,

## 2.7. HO-KALMAN REALIZATION ALGORITHMS

as described in [205, Algorithm 1] or Algorithm 12 in Appendix B. We can then compute the LPV-SSA associated with the thus computed LSS. Moreover, we are able to analyze the correctness of the resulting realization algorithm.

For the sake of completeness, and because we will use it in Chapter 5 we will present the realization algorithm in details. More precisely, we can define the following finite sub-matrices  $H_{\mathfrak{F},L,M}$  of the Hankel-matrix

$$H_{\mathfrak{F},L,M} = \begin{bmatrix} \theta_{\mathfrak{F}}(v_1 v_1) & \theta_{\mathfrak{F}}(v_2 v_1) & \cdots & \theta_{\mathfrak{F}}(v_{\mathbf{N}(M)} v_1) \\ \theta_{\mathfrak{F}}(v_1 v_2) & \theta_{\mathfrak{F}}(v_2 v_2) & \cdots & \theta_{\mathfrak{F}}(v_{\mathbf{N}(M)} v_2) \\ \vdots & \vdots & \cdots & \vdots \\ \theta_{\mathfrak{F}}(v_1 v_{\mathbf{N}(L)}) & \theta_{\mathfrak{F}}(v_2 v_{\mathbf{N}(L)}) & \cdots & \theta_{\mathfrak{F}}(v_{\mathbf{N}(M)} v_{\mathbf{N}(L)}) \end{bmatrix}, \quad (2.18)$$

where  $\theta_{\mathfrak{F}}$  is the matrix of sub-Markov parameters defined in (2.17). Recall from Definition 2.1, Section 2.2 that  $\mathbf{N}(K)$  is the unique integer such that  $\{v_1, \dots, v_{\mathbf{N}(K)}\}$  is the set of all element of  $Q^*$  of length at most  $K$ . Intuitively,  $H_{\mathfrak{F},L,M}$  is the sub-matrix of  $H_{\mathfrak{F}}$ , obtained by keeping the columns of  $H_{\mathfrak{F}}$  indexed by words of length at most  $M$  and keeping the rows indexed by words of length at most  $L$ . The matrix  $H_{\mathfrak{F},L,M}$  corresponds to the matrix  $H_{\mathfrak{G}(\mathfrak{F}),L,M}$  defined in (B.12) in Appendix B, or to the matrix defined in [205, eq. (3.17)] for  $f = \mathfrak{G}(\mathfrak{F})$ .

We are now ready to state the realization algorithm Algorithm 1 based on [205, Algorithm 1], or, alternatively, Algorithm 12 of Appendix B. One way to compute the factorization  $H_{\mathfrak{F},N,N+1} = \mathbf{OR}$  in Algorithm 1 is as follows.

---

### Algorithm 1 Realization algorithm

**Inputs:** an integer  $N > 0$  and the Hankel-matrix  $H_{\mathfrak{F},N,N+1}$ .

**Output:** LPV-SSA  $\Sigma_N$

---

- 1: Compute a decomposition  $H_{\mathfrak{F},N,N+1} = \mathbf{OR}$ , where  $\mathbf{O} \in \mathbb{R}^{J_N \times n}$  and  $\mathbf{R} \in \mathbb{R}^{n \times J_{N+1}}$  and  $\text{rank } \mathbf{R} = \text{rank } \mathbf{O} = n$ ,  $I_N = \mathbf{N}(N)n_y(n_p + 1)$  and  $J_{N+1} = \mathbf{N}(N+1)n_u(n_p + 1)$ .
- 2: Consider the decomposition

$$\mathbf{R} = \left[ \mathbf{C}_{v_1}, \quad \dots, \quad \mathbf{C}_{v_{\mathbf{N}(N+1)}} \right],$$

such that  $\mathbf{C}_{v_i} \in \mathbb{R}^{n \times ((n_p+1)n_u)}$ ,  $i = 1, 2, \dots, \mathbf{N}(N+1)$ , i.e.  $\mathbf{C}_{v_i} \in \mathbb{R}^{n \times ((n_p+1)n_u)}$ ,  $i = 1, 2, \dots, \mathbf{N}(N+1)$  are the block columns of  $\mathbf{R}$ . Define  $\bar{\mathbf{R}}, \mathbf{R}_q \in \mathbb{R}^{n \times J_N}$ ,  $J_N = \mathbf{N}(N)n_u(n_p + 1)$ ,  $q \in Q$  as follows

$$\bar{\mathbf{R}} = \left[ \mathbf{C}_{v_1}, \quad \dots, \quad \mathbf{C}_{v_{\mathbf{N}(N)}} \right], \quad \mathbf{R}_q = \left[ \mathbf{C}_{v_1 q}, \quad \dots, \quad \mathbf{C}_{v_{\mathbf{N}(N)} q} \right].$$

Note that for any  $i \in \{1, \dots, \mathbf{N}(N)\}$  there exists  $j = j(i, q) \in \{2, \dots, \mathbf{N}(N+1)\}$  such that  $v_i q = v_j$ , hence  $\mathbf{R}_q$  is well defined.

- 3: Return  $\Sigma_N = (\mathbb{P}, \{A_i, B_i, C_i\}_{i=0}^{n_p})$  such that

- $[B_0, \dots, B_{n_p}]$  is formed by the first  $n_u(n_p + 1)$  columns of  $\mathbf{R}$
  - $[C_0^T, C_1^T, \dots, C_{n_p}^T]^T$  is the the first  $n_y(n_p + 1)$  rows of  $\mathbf{O}$
  - $A_q = \mathbf{R}_q \bar{\mathbf{R}}^+$ ,  $q = 0, 1, \dots, n_p$ , where  $\bar{\mathbf{R}}^+$  is the Moore-Penrose pseudo-inverse of  $\bar{\mathbf{R}}$ .
- 

If  $\text{rank} H_{\mathfrak{F},N,N+1} = n$  and  $H_{\mathfrak{F},N,N+1} = USV$  is the SVD decomposition of  $H_{\mathfrak{F},N,N+1}$  with  $S$  being the  $n \times n$  diagonal matrix, then define  $\mathbf{O} = US^{1/2}$  and  $\mathbf{R} = S^{1/2}V$ .

The following theorem gives conditions under which the state-space representation returned by Algorithm 1 is a realization of the map  $\mathfrak{F}$ .

**Theorem 2.6** ([216], correctness of Ho-Kalman algorithm Algorithm 1). *If  $\text{rank} H_{\mathfrak{F},N,N} = \text{rank} H_{\mathfrak{F}}$ , then Algorithm 1 returns a minimal realization of  $\mathfrak{F}$ . The condition  $\text{rank} H_{\mathfrak{F},N,N} = \text{rank} H_{\mathfrak{F}}$  holds for a given  $N$ , if there exists a LPV-SSA realization  $\Sigma$  of  $\mathfrak{F}$  such that  $\dim \Sigma \leq N + 1$ .*

The procedure outlined above is a variation of the Ho-Kalman-like realization algorithm described in [275], and hence the discussion above gives sufficient conditions for the correctness of the realization algorithm in [275]. A detailed presentation of the realization procedure described above and the corresponding correctness analysis can be found in [216].

Note that sub-Markov parameters can be computed from the responses of  $\mathfrak{F}$ , in fact, from input-output data, see [66] and Chapter 5. This then suggests that the realization algorithm Algorithm 1 could be used for system identification, by applying it to sub-Markov parameters based on data. This approach will be discussed in Chapter 5 and it was explored in [66]. One of the challenges with this approach is the choice of the integer  $N$ : Theorem 2.6 suggests that in Algorithm 1 the integer  $N$  should be chosen to be larger than or equal to the dimension of a LPV-SSA realization of  $\mathfrak{F}$ . However, this latter choice results in an algorithm which scales badly with the size of systems,  $H_{\mathfrak{F},N,N+1}$  grows exponentially with  $N$ .

In order to address the problem above, and because it is interesting on its own right and it has other applications, below we present a modification of Algorithm 1 which uses the notion of selections. Selections play an important role in the application of realization theory to model reduction and system identification, but also in the study of geometry and topology of spaces of systems [214]. We will present only the SISO case ( $n_u = n_y = 1$ ), the full MIMO case was discussed in [214, 30] for LSSs and in [66] for LPV-SSAs.

We will call any subset  $\alpha \subseteq Q^* \times Q$ , where  $Q = \{0, 1, \dots, n_p\}$ , a *selection*. Finite selections will be used to define Hankel-like matrices, entries of which are Markov parameters.

Assume that  $\alpha$  and  $\beta$  are selection respectively and assume that  $\alpha$  and  $\beta$  are both finite sets of cardinality  $n$  and  $l$  respectively. Fix an enumeration of the elements of  $\alpha$  and  $\beta$  as follows.

$$\alpha = \{(u_i, q_i)\}_{i=1}^n, \quad \beta = \{(v_j, \sigma_j)\}_{j=1}^l, \quad (2.19)$$

Let us now define the matrix  $H_{\mathfrak{F},\alpha,\beta} \in \mathbb{R}^{n \times l}$  as follows:

$$H_{\mathfrak{F},\alpha,\beta} = \begin{bmatrix} \theta_{q_1, \sigma_1, \mathfrak{F}}(v_1 u_1) & \theta_{q_1, \sigma_2, \mathfrak{F}}(v_2 u_1) & \cdots & \theta_{q_1, \sigma_l, \mathfrak{F}}(v_l u_1) \\ \theta_{q_2, \sigma_1, \mathfrak{F}}(v_1 u_2) & \theta_{q_2, \sigma_2, \mathfrak{F}}(v_2 u_2) & \cdots & \theta_{q_2, \sigma_l, \mathfrak{F}}(v_l u_2) \\ \vdots & \vdots & \cdots & \vdots \\ \theta_{q_n, \sigma_1, \mathfrak{F}}(v_1 u_n) & \theta_{q_n, \sigma_2, \mathfrak{F}}(v_2 u_n) & \cdots & \theta_{q_n, \sigma_l, \mathfrak{F}}(v_l u_n) \end{bmatrix} \quad (2.20)$$

If  $\alpha, \beta$  has the same cardinality, then the matrix  $H_{\mathfrak{F},\alpha,\beta}$  is a square one. Intuitively, the rows of  $H_{\mathfrak{F},\alpha,\beta}$  are indexed the elements of  $\alpha$ , and the columns by the elements of  $\beta$ .

In order to present the algorithm, we define the matrices  $H_{q,\mathfrak{F},\alpha,\beta} \in \mathbb{R}^{n \times l}$ ,  $H_{\mathfrak{F},\alpha,q} \in \mathbb{R}^{n \times 1}$  and  $H_{q,\mathfrak{F},\beta} \in \mathbb{R}^{1 \times n}$ :

$$H_{q,\mathfrak{F},\alpha,\beta} = \begin{bmatrix} \theta_{q_1, \sigma_1, \mathfrak{F}}(v_1 q u_1) & \theta_{q_1, \sigma_2, \mathfrak{F}}(v_2 q u_1) & \cdots & \theta_{q_1, \sigma_l, \mathfrak{F}}(v_l q u_1) \\ \theta_{q_2, \sigma_1, \mathfrak{F}}(v_1 q u_2) & \theta_{q_2, \sigma_2, \mathfrak{F}}(v_2 q u_2) & \cdots & \theta_{q_2, \sigma_l, \mathfrak{F}}(v_l q u_2) \\ \vdots & \vdots & \cdots & \vdots \\ \theta_{q_n, \sigma_1, \mathfrak{F}}(v_1 q u_n) & \theta_{q_n, \sigma_2, \mathfrak{F}}(v_2 q u_n) & \cdots & \theta_{q_n, \sigma_l, \mathfrak{F}}(v_l q u_n) \end{bmatrix} \quad (2.21)$$

$$H_{\mathfrak{F},\alpha,q} = [\theta_{q_1, q, \mathfrak{F}}(u_1) \quad \theta_{q_2, q, \mathfrak{F}}(u_2) \quad \cdots \quad \theta_{q_n, q, \mathfrak{F}}(u_n)]^T \quad (2.22)$$

$$H_{q,\mathfrak{F},\beta} = [\theta_{q, \sigma_1, \mathfrak{F}}(v_1) \quad \theta_{q, \sigma_2, \mathfrak{F}}(v_2) \quad \cdots \quad \theta_{q, \sigma_l, \mathfrak{F}}(v_l)] \quad (2.23)$$

---

**Algorithm 2** Realization algorithm for LPV-SSA with selections
 

---

- 1: Consider the factorization  $H_{\mathfrak{F},\alpha,\beta} = O_{n_m} R_{n_m}$  such that  $O_{n_m}$  is full column rank,  $R_{n_m}$  is full row rank and  $\text{rank } O_n = \text{rank } R_n = n_m$ .
  - 2: Define  $\hat{A}_q = O_{n_m}^+ H_{q,\mathfrak{F},\alpha,\beta} R_{n_m}^+$ ,  $\hat{B}_q = O_{n_m}^+ H_{\mathfrak{F},\alpha,q}$ ,  $\hat{C}_q = H_{q,\mathfrak{F},\beta} R_{n_m}^+$ , where  $O_{n_m}^+, R_{n_m}^+$  is the Moore-Penrose inverse of  $O_n$  and  $R_n$  respectively.
  - 3: Return  $\hat{\Sigma} = (\mathbb{P}, \{(\hat{A}_i, \hat{B}_i, \hat{C}_i)\}_{i=0}^{n_p})$ .
-

## 2.8. CONCLUSIONS

---

The algorithm above can be shown to be correct, i.e., to result in an LPV-SSA realization of  $\mathfrak{F}$ , under suitable hypotheses.

**Lemma 2.2** (Adapted from [66], correctness of the realization algorithm with selections from Algorithm 2). *If  $n_m$  is the dimension of a minimal LPV-SSA realization of  $\mathfrak{F}$ , then the LPV-SSA  $\hat{\Sigma}$  defined in Algorithm 2 is a minimal realization of  $\mathfrak{F}$ . Moreover, if  $n_m$  is the dimension of a minimal  $\mathfrak{F}$  realization of  $f$ , then there exists a pair of selections  $\alpha, \beta \subseteq Q^* \times Q$  such that the cardinality of the sets  $\alpha, \beta$  is  $n_m$  and  $\text{rank } H_{\mathfrak{F}, \alpha, \beta} = n_m$ .*

From [219] and the correspondence between LPV-SSAs and LSSs it follows that we can choose  $\alpha_N = \beta_N = \{(v, q) \mid v \in Q^*, |v| \leq N, q \in Q\}$ , where  $N$  is any integer not smaller than the dimension of a minimal LPV-SSA realization of  $\mathfrak{F}$ . Note that the number of elements of  $\alpha$  and  $\beta$  can be chosen as the dimension of the supposed LPV-SSA realization of the input-output function. This is the main advantage of Algorithm 2 in comparison with Algorithm 1. More precisely, the number of sub-Markov parameters which are necessary for Algorithm 1 grows exponentially with the integer  $N$ , and the latter integer is chosen based on the supposed dimension of a LPV-SSA realization of the input-output function at hand. In contrast, Algorithm 2 requires  $n$  sub-Markov parameters, where  $n$  is the supposed dimension of a LPV-SSA realization of the input-output function. This means that Algorithm 2 scales well with the size of the underlying system. Moreover, it is more practical for system identification algorithm, as it requires the estimation of a smaller number of sub-Markov parameters. These issues were discussed in [66] and will briefly be reviewed in Chapter 5.

Note that Algorithm 2 can also be adapted to LSSs, by using the relationship between LSSs and LPV-SSAs. This adaptation is presented in Algorithm 13 in Appendix B.

## 2.8 Conclusions

I have presented a brief overview of my contribution to realization theory of LPV systems. More precisely, I have presented a complete realization theory for LPV-SSAs, which mirrors the results for LTI, bilinear [126] and switched linear systems [219, 203]. The results of this chapter turned out to be useful for model reduction and system identification [5, 7, 65, 66, 28], presented in Chapter 4 – 5. Future work will be directed towards extending the results to the behavioral setting [304] and to LFT/LFR systems [5, 33, 32].



## Chapter 3

# Realization theory of stochastic bilinear systems

### 3.1 Introduction

In this chapter, we discuss realization theory for a class of stochastic bilinear systems. This class includes subclasses of jump-linear system with stochastic switching, autonomous stochastic LPV systems and stochastic bilinear systems with white noise input. In contrast to previous chapters, where we were looking for representations of input-output maps, in this chapter we are interested in representations of discrete-time stochastic processes.

More precisely, we will introduce the class of *generalized bilinear systems* (GBS for short). Roughly speaking, a GBS is a discrete-time system bilinear system equipped with an additive noise term. The motivation for GBSs is that they include stochastic bilinear, LPV and jump-Markov systems as special cases.

The results of this chapter are similar in spirit to those of stochastic realization theory for linear systems and they appeared in [226]. More specifically, we provide a necessary and sufficient condition for processes to have a realization by a GBS in terms a finite rank condition for the Hankel-matrix defined for this process. We also give a characterization of minimal GBSs in terms of rank conditions for their reachability and observability matrices, and we show that any two minimal realizations of the same output and inputs are related by a linear isomorphism, if both these realizations are in the so called *forward innovation form*. We show that any GBS realization of a given process can be converted to a minimal one in forward innovation form. We present a realization algorithm for computing a minimal GBS realization from the covariances of the output process.

The results of this chapter use realization theory of LSSs, or which is equivalent, from realization theory of recognizable formal power series/bilinear systems. The latter was used [226], but for the sake of consistency, we prefer to use realization theory of LSSs, which, in our opinion, is also more intuitive.

Realization theory for stochastic bilinear systems was addressed [73, 92, 270, 271, 235]. However, [73] addresses only the weak realization problem, it presents only sufficient conditions for existence of a weak realization, and it considers only bilinear systems whose inputs are white noise stochastically independent of the noise process of the system. Moreover, [73] skips a lot of mathematical details, including the proof of correctness of the realization algorithm. Hence, the contribution of this chapter is new even for the class of systems from [73]. As to [92, 235], they concentrate on the weak realization problem for infinite dimensional bilinear systems, but they do not address the finite-dimensional case, the strong realization problem or minimality. In [270, 271], the strong realization problem was addressed for a special class for bilinear systems with known noise.

There exist a number of papers on identification of bilinear systems, see e.g., [62, 86, 287, 288, 76]. However, all these papers study the identification problem and not the realization problem. Our results will be useful for analyzing consistency of subspace identification algorithms for bilinear and LPV systems. For example, our result on isomorphism of minimal realizations provides sufficient conditions for the well-posedness of the identification

problem studied in the cited papers. Indeed, note that all the cited papers use try to estimate the state of a time-varying Kalman filter of the bilinear system. In this chapter we show that any GBS can be transformed to a minimal GBS in forward innovation form, whose stationary Kalman-filter is the system itself. This provides a theoretical justification for assuming that it is sufficient to estimate the state of the Kalman-filter. The results on minimality help us to establish that it is sufficient to work with minimal GBSs, and that all minimal GBS realization of the same output and input processes are isomorphic in some sense, i.e. it is sufficient to find a GBS system matching the output and inputs up to a similarity transformation. This also opens up the possibility to characterize the geometry and topology of the space of minimal GBSs, like it was done in [199, 115]. This is important when trying to show consistency of the algorithms [62, 86, 287, 288, 76].

**Outline of the chapter** The chapter is organized as follows. In Section 3.2 we present an informal description of the class of GBSs and we define the realization problem for GBSs. We also explain the relationship between GBSs and stochastic bilinear systems, jump-Markov linear systems and LPV systems. In Section 3.3 we present an informal description of the solution to the stochastic realization problem. In Section 3.4 we present the formal problem formulation. In Section 3.5 we present our solution to the GBS realization problem as well as applications to several classes of systems. In Section 3.7 we discuss some consequences of the results of Section 3.5 for system identification. In particular, we address the existence of a realization in forward innovation form and its relationship with filtering. In addition, we show that that under mild assumptions the solution to the strong realization problem (Problem 3.1) implies finding a state-space representation of the input-output relation  $\{\mu_q\}_{q \in Q} \mapsto \mathbf{y}$ .

## 3.2 Informal definition of GBSs, problem formulation, preliminaries

We start by introducing informally the class of systems considered. Let  $Q$  be a finite set. Consider a collection  $\{\mu_q\}_{q \in Q}$  of  $\mathbb{R}$ -valued stochastic processes, i.e. for each  $q \in Q$ ,  $\mu_q$  is a  $\mathbb{R}$  valued stochastic process. We will call the stochastic processes  $\mu_q$ ,  $q \in Q$  *input processes*. A *generalized bilinear system* (GBS) with respect to  $\{\mu_q\}_{q \in Q}$  is a system of the form

$$\mathcal{G} \begin{cases} \mathbf{x}(t+1) = \sum_{q \in Q} (A_q \mathbf{x}(t) + K_q \mathbf{v}(t)) \mu_q(t) \\ \tilde{\mathbf{y}}(t) = C \mathbf{x}(t) + D \mathbf{v}(t). \end{cases} \quad (3.1)$$

where  $A_q \in \mathbb{R}^{n \times n}$ ,  $K_q \in \mathbb{R}^{n \times m}$ ,  $C \in \mathbb{R}^{p \times n}$  and  $D \in \mathbb{R}^{p \times m}$  are the system matrices,  $\tilde{\mathbf{y}}$ ,  $\mathbf{x}$ ,  $\mathbf{v}$  are stochastic processes taking values in  $\mathbb{R}^p$ ,  $\mathbb{R}^n$  and  $\mathbb{R}^m$  respectively, We call  $\mathbf{x}$  the *state process*,  $\tilde{\mathbf{y}}$  the *output process*,  $\mathbf{v}$  the *noise process*, and  $\{\mu_q\}_{q \in Q}$  the collection *input processes* of  $\mathcal{G}$ . We will identify  $\mathcal{G}$  with the tuple

$$\mathcal{G} = (n, p, m, Q, \mathbf{x}, \mathbf{v}, \{\mu_q\}_{q \in Q}, \tilde{\mathbf{y}}, C, D, \{A_q, K_q\}_{q \in Q}) \quad (3.2)$$

The integer  $n$  is called the *dimension* of  $\mathcal{G}$ .

### Subclasses of GBSs: stochastic bilinear systems, jump-Markov linear systems, stochastic LPV systems

As it was pointed out above, the motivation for studying the realization problem for GBSs is that they include stochastic bilinear, LPV and jump-Markov systems.

In particular, if  $Q = \{0, 1, \dots, d\}$ ,  $\mu_0 = 1$ , then GBSs are stochastic bilinear systems from [73, 92, 270, 271, 62, 86, 287, 288, 76] or autonomous LPV systems, if we identify  $\mu_q$ ,  $q \in Q$  with the scheduling parameter. In this case, (3.1) becomes

$$\begin{aligned} \mathbf{x}(t+1) &= A(\mu(t))\mathbf{x}(t) + K(\mu(t))\mathbf{v}(t) \\ \tilde{\mathbf{y}}(t) &= C\mathbf{x}(t) + D\mathbf{v}(t). \end{aligned} \quad (3.3)$$

### 3.2. INFORMAL DEFINITION OF GBSS, PROBLEM FORMULATION, PRELIMINARIES

where  $\boldsymbol{\mu}(t) = (\boldsymbol{\mu}_1(t), \dots, \boldsymbol{\mu}_d(t))^T$  and  $A(\boldsymbol{\mu}(t)) = A_0 + \sum_{i=1}^d A_i \boldsymbol{\mu}_i(t)$ ,  $K(\boldsymbol{\mu}(t)) = K_0 + \sum_{i=1}^d K_i \boldsymbol{\mu}_i(t)$ . In this case, it is usually assumed that  $\{\boldsymbol{\mu}_i\}_{i=1}^d$  are zero mean white noise processes which are mutually uncorrelated.

Jump-Markov linear systems (switched systems with stochastic switching) can be encoded by GBSSs as follows. Consider a jump-Markov linear system without input [64] of the form

$$\begin{aligned} \mathbf{x}(t+1) &= M_{\boldsymbol{\theta}(t)} \mathbf{x}(t) + B_{\boldsymbol{\theta}(t)} \mathbf{v}(t) \\ \tilde{\mathbf{y}}(t) &= C \mathbf{x}(t) + D \mathbf{v}(t) \end{aligned} \quad (3.4)$$

where  $\mathbf{x}$  is the state process,  $\tilde{\mathbf{y}}$  is the output process,  $\mathbf{v}(t)$  is the noise process, and  $\boldsymbol{\theta}$  is the switching processes taking values in a finite set  $\Theta$ , and  $M_q, B_q, q \in \Theta, C, D$  are matrices of appropriate dimension. Assume  $\boldsymbol{\theta}$  is taking its values in a finite set  $\Theta$ . Two cases should be distinguished. The first one is when  $\boldsymbol{\theta}(t)$  are independent and identically distributed (i.i.d for short). In this case, we can take  $Q = \Theta$  and  $\boldsymbol{\mu}_q(t) = \chi(\boldsymbol{\theta}(t) = q)$ , where  $\chi$  is the indicator function, i.e.,  $\chi(A)$  equals 1 when  $A$  holds, and it is zero otherwise, and  $A_q = M_q, K_q = B_q$ . With this identification, (3.4) becomes (3.1).

The second case is when  $\boldsymbol{\theta}$  is a stationary and ergodic Markov process whose state space is the finite set  $\Theta$ , and with transition probabilities  $P(\boldsymbol{\theta}(t) = q_2 \mid \boldsymbol{\theta}(t-1) = q_1) = p_{(q_2, q_1)} > 0, q_1, q_2 \in \Theta$ . Let us take  $Q = \Theta \times \Theta$ ,  $\boldsymbol{\mu}_{(q_2, q_1)}(t) = \chi(\boldsymbol{\theta}(t+1) = q_2, \boldsymbol{\theta}(t) = q_1)$  for all  $q_1, q_2 \in \Theta, t \in \mathbb{Z}$ . In this case, let us define  $A_{(q_2, q_1)} = M_{q_1}, K_{(q_2, q_1)} = B_{q_1}$  and then (3.4) becomes (3.1). Note that in this case the GBS which is derived from the jump-Markov linear system differs from the original one in the sense that its system matrices in the state update equation could depend not only on the current discrete state, but also on the next one. This is not the case for the GBS obtained directly from the jump-Markov linear system, but it might be the case when the latter GBS is transformed to say a minimal one. This technical issue does not effect the realization problem for jump-Markov linear systems, see Subsection 3.6.4 later in the chapter.

The framework of GBSS also allows to combine stochastic bilinear systems with jump-Markov systems, and deal with stochastic switched bilinear/LPV systems with Markovian switching [226].

To sum up, GBSSs contain as special cases many widely used system classes. Note that the examples discussed above involved mostly systems without control inputs. This is especially the case for LPV and jump-Markov linear systems. However, in many cases, a stochastic systems with control input can be represented as a parallel interconnection of a deterministic and an autonomous stochastic system. The deterministic part depends only on the control input but not the noise, and the stochastic part depends only on the noise but not on the control input. The realization problem then can be solved by solving the realization problem for the deterministic and stochastic parts separately. For stochastic LPV systems this decomposition was worked out in [173]. We believe that a similar approach will work for other relevant system classes as well. In particular, this hypothesis means that the main bottleneck towards developing realization theory for stochastic systems is solving the realization problem for systems without control inputs. The general case then follows by combining the solution for stochastic autonomous systems with that of for deterministic systems.

**The realization problem for GBSSs.** In this chapter we will be interested in realization theory of GBSSs, hence we should define what we mean by a realization by a GBSS.

**Definition 3.1** (Realization by GBSS). *A GBSS  $\mathcal{G}$  w.r.t.  $\{\boldsymbol{\mu}_q\}_{q \in Q}$  is said to be a realization of the stochastic processes  $\mathbf{y}$  if output process of  $\mathcal{G}$  equals  $\mathbf{y}$ . That is, if  $\mathcal{G}$  is of the form (3.2), then  $\mathcal{G}$  is a realization of  $\mathbf{y}$ , if  $\tilde{\mathbf{y}} = \mathbf{y}$ .*

In this chapter we will be interested in existence of a GBSS realizations of a process and characterization of minimal dimensional GBSS realizations.

Some remarks are in order. Note that the choice of the input processes may play a role in existence of a GBSS realization. More precisely, let  $\{\hat{\boldsymbol{\mu}}_q\}_{q \in Q}$  another collection of input processes. In principle, it may happen that a process  $\mathbf{y}$  has a realization by a GBSS w.r.t.  $\{\hat{\boldsymbol{\mu}}_q\}_{q \in Q}$ , but there exists no GBSS w.r.t.  $\{\boldsymbol{\mu}_q\}_{q \in Q}$ . It may also happen, that a GBSS w.r.t.  $\{\hat{\boldsymbol{\mu}}_q\}_{q \in Q}$  is minimal dimensional among all those GBSSs w.r.t.  $\{\hat{\boldsymbol{\mu}}_q\}_{q \in Q}$  which are realizations of  $\mathbf{y}$ , but there exist a GBSS w.r.t.  $\{\boldsymbol{\mu}_q\}_{q \in Q}$  which is a realization of  $\mathbf{y}$  and which has a smaller dimension. That



is, strictly speaking, we whenever we speak of a GBS realization of  $\mathbf{y}$ , we should indicate the input processes of that GBS. However, such an approach would make the presentation cumbersome. For the ease of discussion, in the sequel, unless stated otherwise, *the input process  $\{\mu_q\}_{q \in Q}$  will be fixed, and by a GBS we will always mean a GBS with respect to  $\{\mu_q\}_{q \in Q}$ . In particular, by a GBS realization of  $\mathbf{y}$  we will always mean a GBS w.r.t.  $\{\mu_q\}_{q \in Q}$  which is a realization of  $\mathbf{y}$ . Moreover, by a minimal dimensional (or simply minimal) GBS realization of  $\mathbf{y}$  we mean a GBS w.r.t.  $\{\mu_q\}_{q \in Q}$  which is a realization of  $\mathbf{y}$  and dimension of which is not larger than the dimension of any other GBS w.r.t.  $\{\mu_q\}_{q \in Q}$  which is a realization of  $\mathbf{y}$ .* In this chapter, we consider the following realization problem.

**Problem 3.1** (Strong realization problem for GBSs). *Given a process  $\mathbf{y}$  find conditions for existence of a GBS  $\mathcal{G}$  which is a realization of  $\mathbf{y}$ , as well as conditions for a GBS to be a minimal dimensional realization of  $\mathbf{y}$ . Formulate algorithms for computing a minimal dimensional GBS realization of  $\mathbf{y}$ .*

The proposed problem is a direct counterpart of the well known *strong realization* problem for linear stochastic systems [147, 163, 162]. Similarly to linear systems [147, 163, 162], we can also formulate the concept of a weak realization for GBS. Informally, a GBS  $\mathcal{G}$  is a weak realization of  $\mathbf{y}$ , if certain covariances of the output of  $\mathcal{G}$  equal the corresponding covariances of  $\mathbf{y}$ . The *weak realization problem* is to find conditions for existence and minimality of a weak realization of a process  $\mathbf{y}$  and formulate algorithms for computing a minimal dimensional weak realization of  $\mathbf{y}$ . Clearly, if  $\mathcal{G}$  is a realization of  $\mathbf{y}$ , then it is also a weak realization of  $\mathbf{y}$ . Hence, a solution of Problem 3.1 also solves the weak realization problem. We refer the reader to [226] for details on the weak realization problem for GBSs.

Note that the realization problem of Problem 3.1 is different from the realization problem for deterministic systems which have been discussed so far. In Problem 3.1, we fix the input processes  $\{\mu_q\}_{q \in Q}$ , and we are interested in GBSs which are driven by these fixed inputs, and whose output is  $\mathbf{y}$ . This is consistent with the traditional formulation of the realization problem for *stochastic systems* with inputs [147, 163], but it is different from *deterministic realization problem*, where a state-space representation of an input-output relation  $\{\mu_q\}_Q \mapsto \mathbf{y}$  is sought after. Note that from the point of view of system identification, the problem formulation with a fixed input is the most natural one. Indeed, in system identification, only the response of the system to one particular input is known. Therefore, in system identification, constructing a state-space representation and a noise process which is consistent with an input and an output process is the best we can do. The general relationship between these two problems is still open even for linear systems, and for this reason we will not attempt to explore it for GBSs. However, for certain classes of GBSs, they are equivalent. More precisely, since minimal GBSs in forward innovation form realizing the same process will be shown to be isomorphic, so this system class the two formulations are closely related: if two such systems yield the same output process for some input (satisfying certain properties), they will yield the same output process for any other input. In particular, the assumption that the input is stochastic and it satisfies some technical properties is analogous to persistence of excitation: in order to be able to reconstruct a state-space representation of a system, we need the knowledge of the input-output behavior for sufficiently rich input signals. Once the state-space representation is found, it remains valid for any other input signal. We will elaborate on it in Section 3.7.

**Notation and terminology** Throughout the chapter, we use the standard terminology and notation of probability theory, see [43]. Throughout the chapter, we fix a probability space  $(\Omega, \mathcal{F}, P)$  ( $\mathcal{F}$  is a  $\sigma$ -algebra over  $\Omega$ , and  $P$  is a probability measure on  $\mathcal{F}$ ). All the random variables and stochastic processes should be understood with respect to this probability space. Moreover, all the stochastic processes considered in this chapter are discrete-time processes with the time axis  $\mathbb{Z}$ , where  $\mathbb{Z}$  is the set of integers. We denote the expectation of a random variable  $\mathbf{z}$  by  $E[\mathbf{z}]$ . We denote the conditional expectation of  $\mathbf{z}$  w.r.t. to a  $q$ -algebra  $\mathcal{G}$  by  $E[\mathbf{z} | \mathcal{G}]$ .

### 3.3 Informal description of the solution: relationship between GBSSs and linear switched systems.

The basic idea is to relate the realization problem for GBSSs with the realization problem for linear switched systems (abbreviated by *LSS* in the sequel). We will use the terminology of [202, 203, 219, 209, 205] when discussing LSSs, see Appendix B for an overview.

More precisely, recall from [57, 163, 147] that there is a correspondence between stochastic LTI state-space representations of  $\mathbf{y}$  and deterministic LTI state-space representations of deterministic input-output maps  $f^{\mathbf{y}}$ . This correspondence is such that the Markov parameters of  $f^{\mathbf{y}}$  are covariances  $E[\mathbf{y}(t+k)\mathbf{y}^T(t)]$  of  $\mathbf{y}$ . Moreover, this correspondence preserves minimality. We extend this idea to GBSSs by establishing a correspondence between GBSSs realizations of  $\mathbf{y}$  and LSS realizations of a deterministic input-output map  $f^{\mathbf{y}}$ . Similarly to the linear case, the Markov-parameters of  $f^{\mathbf{y}}$  will be equal to certain covariances of  $\mathbf{y}$  and  $\{\boldsymbol{\mu}_q\}_{q \in Q}$ . Below we present the basic steps, without discussing the assumptions which are necessary to make those steps mathematically well posed. That latter will be done in subsequent sections.

As the first step, we will argue that if  $\mathbf{y}$  has a realization by an GBSSs  $\mathcal{G}$  of the form (3.2), then the covariance matrices  $\Lambda_{q_1 \dots q_k}^{\mathbf{y}}$ ,  $w = q_1 \dots q_k \in Q$ ,  $k \geq 1$ , defined below

$$\Lambda_{q_1 \dots q_k}^{\mathbf{y}} = \frac{1}{\sqrt{\mathbf{p}(\boldsymbol{\mu}_{q_1}) \dots \mathbf{p}(\boldsymbol{\mu}_{q_k})}} E[\mathbf{y}(t)\mathbf{y}^T(t-k)\boldsymbol{\mu}_{q_1}(t-k) \dots \boldsymbol{\mu}_{q_k}(t-1)], \quad (3.5)$$

correspond to the Markov parameters of a LSS

$$\Sigma_{\mathcal{G}} \begin{cases} z(t+1) = \left( \sqrt{\mathbf{p}(\boldsymbol{\mu}_q)} A_{q(t)} \right) z(t) + B_q(t)w(t) \\ y(t) = Cz(t) \end{cases} \quad (3.6)$$

which we will call the *LSS associated with  $\mathcal{G}$* . Here  $q(t)$  takes values in  $Q$ . The number  $\mathbf{p}(\boldsymbol{\mu}_q)$  from (3.5) are suitable numbers defined uniquely by the process  $\{\boldsymbol{\mu}_q\}_{q \in Q}$ . For example,  $\{\boldsymbol{\mu}_q\}_{q \in Q}$  are i.i.d. processes, then typically  $\mathbf{p}(\boldsymbol{\mu}_q) = E[\boldsymbol{\mu}_q^2(t)]$ . If  $Q = \Theta \times \Theta$ ,  $\boldsymbol{\mu}_{(q_2, q_1)}(t) = \chi(\boldsymbol{\theta}(t+1) = q_2, \boldsymbol{\theta}(t) = q_1)$  for all  $q_1, q_2 \in \Theta, t \in \mathbb{Z}$ , where  $\boldsymbol{\theta}(t)$  is an ergodic Markov process taking values in a finite set  $\Theta$ , then  $\mathbf{p}(\boldsymbol{\mu}_{(q_1, q_2)})$  is the transition probability of  $\Theta$ .

The matrices  $B_q$  of (3.6) are defined as follows:

$$\forall q \in Q: B_q = \frac{1}{\sqrt{\mathbf{p}(\boldsymbol{\mu}_q)}} (A_q P_q C^T + K_q Q_q D^T) \quad (3.7)$$

where  $\{P_q = E[\mathbf{x}(t)\mathbf{x}^T(t)\boldsymbol{\mu}_q^2(t)]\}_{q \in Q}$  is the unique collection of positive semi-definite matrices  $\{P_q\}_{q \in Q}$  which satisfy the Lyapunov-like equation

$$P_q = \mathbf{p}(\boldsymbol{\mu}_q) \sum_{q_1 \in Q, q_1 q \in L} (A_{q_1} P_{q_1} A_{q_1}^T + K_{q_1} Q_{q_1} K_{q_1}^T), \quad (3.8)$$

where  $Q_{q_1} = E[\mathbf{v}(t)\mathbf{v}^T(t)\boldsymbol{\mu}_{q_1}^2(t)]$ ,  $q_1 \in Q$ . In this case,

$$\Lambda_{q_w}^{\mathbf{y}} = CA_w B_q \sqrt{\mathbf{p}(\boldsymbol{\mu}_w)}, \quad (3.9)$$

$$E[\mathbf{y}(t)\mathbf{y}^T(t)\boldsymbol{\mu}_q^2(t)] = (CP_q C^T + CQ_q D^T)$$

for all  $q \in Q$ ,  $w \in Q^*$ , where  $\mathbf{p}(\boldsymbol{\mu}_\varepsilon) = 1$  and  $\mathbf{p}(\boldsymbol{\mu}_{q_1 \dots q_k}) = \mathbf{p}(\boldsymbol{\mu}_{q_1})\mathbf{p}(\boldsymbol{\mu}_{q_2}) \dots \mathbf{p}(\boldsymbol{\mu}_{q_k})$  for all  $q_1, \dots, q_k \in Q$ .

We can also relate potential input-output maps of deterministic LSS with covariances of  $\mathbf{y}$ . More precisely, let us define the input-output map  $f^{\mathbf{y}}: \mathcal{U} \times \mathcal{Q} \rightarrow \mathcal{Y}$ , where  $\mathcal{U} = (\mathbb{R}^p)^{\mathbb{N}}$ ,  $\mathcal{Q} = Q^{\mathbb{N}}$ ,  $(\mathcal{Y} = \mathbb{R}^p)^{\mathbb{N}}$ , i.e.,  $\mathcal{U}, \mathcal{Q}, \mathcal{Y}$

are the sets of all functions from the set of natural numbers  $\mathbb{N}$  to the sets  $\mathbb{R}^p$ ,  $Q$  and  $\mathbb{R}^p$  respectively. Note that  $\mathcal{U}$  and  $\mathcal{Y}$  coincide in our case, however, we prefer to denote them by different symbols as they represent the space of continuous input and output spaces of the input-output map  $f^y$ . The input-output map  $f^y$  is defined as follows: for all  $u \in \mathcal{U}$ ,  $\sigma \in \mathcal{Q}$ :

$$f^y(u, \sigma(t)) = \sum_{i=0}^{t-1} \Lambda_{\sigma(i)\dots\sigma(t-1)}^y u(i). \quad (3.10)$$

We will call  $f^y$  the *input-output map associated with  $\mathbf{y}$* . Let us recall from [202, 203, 219, 209, 205] or Appendix B the notion of generalized Markov parameters of the input-output function. From the definition of Markov-parameter  $M^{f^y}$  and  $S_{q_1, q_2}^{f^y}$  [209, Definition 11, eq. (7)] or Appendix B.5<sup>1</sup> of input-output maps it follows that

$$S_{q_1, q_2}^{f^y}(v) = \Lambda_{q_2 v}^y$$

That is, a GBS is a realization of  $\mathbf{y}$ , if the associated LSS is a realization of the input-output map  $f^y$  associated with  $\mathbf{y}$ .

The attentive reader might have noticed that the definitions above are well-posed only if suitable technical assumptions are made  $\mathcal{G}$ . In particular, we have to make sure that the right-hand side of (3.5) does not depend on  $t$ , that  $E[\mathbf{y}(t)\mathbf{y}^T(t)\boldsymbol{\mu}_q^2(t)]$ ,  $Q_q = E[\mathbf{v}(t)\mathbf{v}^T(t)\boldsymbol{\mu}_q^2(t)]$ , and  $P_q = E[\mathbf{x}(t)\mathbf{x}^T(t)\boldsymbol{\mu}_q^2(t)]$ , do not depend on  $t$ , and that (3.8) has a unique solution which equals  $\{P_q\}_{q \in Q}$ . Later on in the chapter we will introduce these technical assumptions. The input processes  $\{\boldsymbol{\mu}_q\}_{q \in Q}$  which satisfy these assumption will be called *admissible*, the processes  $\mathbf{y}, \mathbf{x}, \mathbf{v}$  will be called *ZMWSSI with respect to  $\{\boldsymbol{\mu}_q\}_{q \in Q}$* , and the GBSs with input processes  $\{\boldsymbol{\mu}_q\}_{q \in Q}$  satisfying these technical assumptions will be called *stationary GBSs*.

Next, we show that we can construct a GBS realization of  $\mathbf{y}$  from a LSS realization of the associated input-output map  $f^y$ . By looking at the equations (3.9) it is clear that if we find a LSS realization  $\Sigma$  of  $f^y$ , then the matrices  $C$  and  $A_q$  of the GBS will be the same as those of  $\Sigma$ . The difficulty is to define the noise gain  $K_q$  and the noise process  $\mathbf{v}$  of the GBS. Since we have only  $\mathbf{y}$  at our disposal, we have to define  $\mathbf{v}$  as some function of  $\mathbf{y}$ . We will define  $\mathbf{v}$  as the *innovation process  $\mathbf{e}$  of  $\mathbf{y}$  with respect to  $\{\boldsymbol{\mu}_q\}_{q \in Q}$* :  $\mathbf{e}(t) = \mathbf{y}(t) - \hat{\mathbf{y}}(t)$ , where  $\hat{\mathbf{y}}(t)$  is the element of the the closure of linear space generated by the random variables of the form  $\mathbf{y}(t-k)\boldsymbol{\mu}_{q_1}(t-k)\dots\boldsymbol{\mu}_{q_k}(t-1)$ ,  $q_1, \dots, q_k \in Q$ ,  $k \geq 1$  such that  $E[\|\mathbf{y}(t) - \hat{\mathbf{y}}(t)\|_2^2]$  is minimal. By closure we mean closure in the topology of mean-square integrable random variables. We will present the precise definition later on. It suffices to note that  $\mathbf{e}(t)$  is well defined if  $\mathbf{y}$  is a ZMWSSI process with respect to  $\{\boldsymbol{\mu}_q\}_{q \in Q}$  (the notion of ZMWSSI processes will be defined later). The definition of  $\mathbf{e}(t)$  is similar to the classical definition of the innovation process for linear systems: however, instead of looking at the difference of  $\mathbf{y}(t)$  and its best linear prediction with respect to past outputs, we define  $\mathbf{e}(t)$  as the difference between  $\mathbf{y}(t)$  and the best linear prediction of  $\mathbf{y}(t)$  based on products of past inputs and outputs. We will say that a stationary GBS  $\mathcal{G}$  of the form (3.2) is in *forward innovation form*, if its noise process  $\mathbf{v}$  is the innovation process  $\mathbf{e}$  of its output  $\hat{\mathbf{y}}$  with respect to its input processes  $\{\boldsymbol{\mu}_q\}_{q \in Q}$  and  $D$  is the identity matrix.

Let  $\Sigma$  be a minimal LSS realization of  $f^y$  of the form

$$\Sigma \begin{cases} x(t+1) = \bar{A}_{\sigma(t)} x(t) + \bar{B}_{\sigma(t)} u(t) \\ y(t) = \bar{C}_{\sigma(t)} x(t) \end{cases}, \quad (3.11)$$

where  $\sigma(t) \in Q$  is the switching signal,  $x(t), u(t), y(t)$  are the state, input and output signals of the LSS  $\Sigma$  respectively.

By renaming the elements of  $Q$ , without loss of generality, we can assume that  $Q = \{1, \dots, D\}$ . Recall from [205, Definition 23] the definition of the observability matrix  $\mathcal{O}(\Sigma)$  of  $\Sigma$ ,

$$\mathcal{O}(\Sigma) = \left[ (\bar{C}\bar{A}_{v_1})^T, (\bar{C}\bar{A}_{v_2})^T, \dots, (\bar{C}\bar{A}_{v_{N(n-1)}})^T \right]^T \in \mathbb{R}^{pDN(n-1) \times n}. \quad (3.12)$$

<sup>1</sup>In the notation of [209],  $S_{q_1, q_2}^{f^y}(v)$  corresponds to  $S^{y^f}(q_2 v q_1)$

### 3.3. INFORMAL DESCRIPTION OF THE SOLUTION: RELATIONSHIP BETWEEN GBSS AND LINEAR SWITCHED SYSTEMS.

with  $\tilde{C} = [\tilde{C}_1, \tilde{C}_2^T, \dots, \tilde{C}_D^T]^T \in \mathbb{R}^{pD \times n}$ . We used the ordered enumeration  $v_1, v_2, \dots$ , defined in Definition 2.1, and the integer  $\mathbf{N}(n-1)$  is defined after Definition 2.1. Since  $\Sigma$  is minimal, it is span-reachable from the zero initial state and observable. It can be shown that necessarily  $\tilde{C}_q = \tilde{C}$  due to span-reachability of  $\Sigma$ . Let us consider the following sub-matrix of the observability matrix  $\hat{\mathcal{O}}(\Sigma)$ :

$$\hat{\mathcal{O}}(\Sigma) = \left[ (\tilde{C}\bar{A}_{v_1})^T, (\tilde{C}\bar{A}_{v_2})^T, \dots, (\tilde{C}\bar{A}_{v_{\mathbf{N}(n-1)}})^T \right]^T \in \mathbb{R}^{pD\mathbf{N}(n-1) \times n}. \quad (3.13)$$

Note that the rank of  $\hat{\mathcal{O}}(\Sigma)$  is the same as the rank of  $\mathcal{O}(\Sigma)$ , since their row-spaces are the same. Since  $\Sigma$  is observable and its observability matrix is full column rank, hence  $\hat{\mathcal{O}}(\Sigma)$  is also full column rank. Thus  $\hat{\mathcal{O}}(\Sigma)$  has a left inverse. Let  $Y_n(t)$  be formed by the products of future outputs and inputs:

$$Y_n(t) = \left[ (\mathbf{z}_{v_1}^{\mathbf{y}^+}(t))^T, (\mathbf{z}_{v_2}^{\mathbf{y}^+}(t))^T \dots (\mathbf{z}_{v_{\mathbf{N}(n-1)}}^{\mathbf{y}^+}(t))^T \right]^T, \quad (3.14)$$

with  $\mathbf{z}_{v_1}^{\mathbf{y}^+}(t) = \mathbf{z}_e^{\mathbf{y}^+}(t) = \mathbf{y}^T(t)$  and  $\mathbf{z}_{v_i}^{\mathbf{y}^+}$ ,  $i > 1$  is as defined in (3.15). We used the ordered enumeration  $v_1, v_2, \dots$ , and integer  $\mathbf{N}(n-1)$  defined after Definition 2.1, where

$$\mathbf{z}_e^{\mathbf{y}^+}(t) = \mathbf{y}(t), \mathbf{z}_{q_1 \dots q_k}^{\mathbf{y}^+}(t) = \mathbf{y}(t+k) \prod_{i=1}^k \frac{\mu_{q_i}(t+k-i)}{\sqrt{\mathbf{p}(\mu_{q_i})}}. \quad (3.15)$$

Intuitively, the random  $\mathbf{z}_w^{\mathbf{y}^+}(t)$  represents the product of future outputs with past inputs.

Let  $\mathbf{x}(t)$  be the element of the closure of the linear space generated by  $\mathbf{y}(t-k)\mu_{q_1}(t-k) \dots \mu_{q_k}(t-1)$ ,  $q_1, \dots, q_k \in \mathcal{Q}$ ,  $k \geq 1$  such that  $E[\|\hat{\mathcal{O}}^{-1}(\Sigma)Y_n(t) - \mathbf{x}(t)\|_2^2]$  is minimal, i.e.,  $\mathbf{x}(t)$  is the orthogonal projection (in the sense to be defined later) of the product of future outputs and inputs to the linear space generated by the products of past outputs and inputs.

For each  $q \in \mathcal{Q}$ , define a matrix  $K_q \in \mathbb{R}^{n \times p}$  such that

$$K_q = (B_q \sqrt{\mathbf{p}(\mu_q)} - \frac{1}{\sqrt{\mathbf{p}(\mu_q)}} A_q P_q C^T) Q_q^{-1}, \quad (3.16)$$

$P_q = E[\mathbf{x}(t)\mathbf{x}^T(t)\mu_q^2(t)]$ , and

$$Q_q = (E[\mathbf{y}(t)\mathbf{y}^T(t)\mu_q^2(t)] - CP_q C^T) = E[\mathbf{e}(t)\mathbf{e}(t)\mu_q^2(t)]. \quad (3.17)$$

Define the GBS

$$\mathcal{G}_\Sigma = (n, p, p, \mathcal{Q}, \mathbf{x}, \mathbf{e}, \{\mu_q\}_{q \in \mathcal{Q}}, \mathbf{y}, C, I_p, \left\{ \frac{1}{\sqrt{\mathbf{p}(\mu_q)}} A_q, K_q \right\}_{q \in \mathcal{Q}}). \quad (3.18)$$

The GBS  $\mathcal{G}_\Sigma$  is called the *GBS associated with the LSS  $\Sigma$* . Note that  $\mathcal{G}_\Sigma$  is in forward innovation form, and its state is the projection of products of future outputs with inputs on the products of past outputs with inputs.

Note that the transition from GBSs to LSS and back preserves dimension and the state-space. Hence, minimality of GBSs is equivalent to minimality of the corresponding LSSs. Hence, we can use [209, Theorem 1] (alternatively Theorem B.1 of Appendix B) to characterize minimality of GBSs. In general, minimal GBSs realizations of the same output and input processes  $\mathbf{y}$  will not be isomorphic, although their LSSs are. The reason behind it is that different GBSs realizations with the same matrices  $C, A_q$  but different noises and noise gain matrices may yield the same associated LSS. However, if we fix the noise process to be the innovation process of  $\mathbf{y}$ , then this source of freedom in the choice of GBS realizations disappears. In fact, it can be shown that all minimal GBS realizations of  $\mathbf{y}$  in forward innovation form are isomorphic. This situation parallels that of for stochastic linear systems described in [57, 163, 147].

In addition, existence of a GBS realization of  $\mathbf{y}$  is equivalent to existence of a LSS realization of the associated input-output map  $f^{\mathbf{y}}$ . In turn, the latter can be characterized by a finite rank condition of the Hankel-matrix of  $f^{\mathbf{y}}$ ,

see [209, Theorem 2], Theorem B.3 of Appendix B. Note that this Hankel-matrix can be constructed from the covariances  $\Lambda_w^y$ . In fact, by using [209, Algorithm 1] or the adaptation of the reduced-basis realization algorithm from [66], we can compute a minimal LSS realization  $\Sigma$  of  $f^y$  from finitely many covariances  $\Lambda_w^y$ . In turn, we can compute from  $\Sigma$  the associated GBS realization  $\mathcal{G}_\Sigma$  of  $\mathbf{y}$ , which will be minimal and in forward innovation form. In this way we obtain a counterpart of the covariance realization algorithm for linear systems, see [57, 163, 147]. That is, a minimal GBS realization of  $\mathbf{y}$  in forward innovation form can be computed from a finite number of covariances of  $\mathbf{y}$  and  $\{\mu_q\}_{q \in Q}$ .

Note that in the discussion above we could have used LPV-SSAs instead of LSSs, as these two system classes are closely related, see Section 2.4. Indeed, assume that  $Q = \{1, \dots, D\}$ . We can then identify the LSS (3.6) with the LPV-SSA  $(\mathbb{R}^D, \{\sqrt{\mathbf{p}(\mu_q)}A_i, B_i, C_i\}_{i=0}^D)$ , where  $A_0 = B_0 = 0$ ,  $C_0 = 0$ ,  $C_i = C$ ,  $i \in Q$ . Conversely, we can use the LPV-SSA  $(\mathbb{R}^D, \{\bar{A}_i, \bar{B}_i, \bar{C}_i\}_{i=0}^D)$ , with  $A_0 = 0$ ,  $\bar{B}_0 = 0$ ,  $\bar{C}_0 = 0$ ,  $\bar{C}_i = C$ ,  $i \in Q$  instead of (3.11) and adapt the discussion accordingly. Which formalism is used, LSSs or LPV-SSA is largely matter of taste. In this chapter we prefer to stick to LSSs, as the latter formalism seems more natural to us.

### 3.4 Mathematical framework

In order for the realization problem to be meaningful, a number of assumptions have to be made on the input and output processes and GBSs. In this section we will present these assumptions.

We denote by  $\mathcal{H}_1$  the Hilbert-space of square integrable real valued random variables with the scalar product  $\langle \mathbf{z}, \mathbf{x} \rangle = E[\mathbf{z}\mathbf{x}]$ .

**Definition 3.2 (ZMSI).** *A stochastic process  $\mathbf{z}$  is called zero mean and square integrable (abbreviated by ZMSI), if  $E[\mathbf{z}(t)] = 0$  and  $E[\mathbf{z}^T(t)\mathbf{z}(t)] < +\infty$ .*

Recall that a process  $\mathbf{z}$  is *wide sense stationary*, if for every  $s, t, k \in \mathbb{Z}$ , the expectations  $E[\mathbf{z}(t+k)\mathbf{z}^T(s+k)]$ ,  $E[\mathbf{z}(t+k)]$  exist and their value is independent of  $k$ . In the sequel, we use the notation and terminology regarding finite sequences of elements of  $Q$  introduced in Section 2.2.

In the sequel, we will be interested in GBSs input processes of which are *admissible* [226]. For the formal definition, we refer to [226]. For the purposes of this chapter, it is sufficient to know that if a collection  $\{\mu_q\}_{q \in Q}$  is *admissible*, then the following holds. For each word  $w \in Q^+$  of the form  $w = q_1 q_2 \dots q_k \in Q^+$ ,  $k \geq 1$ ,  $q_1, \dots, q_k \in Q$ , define the process  $\mu_w$  as

$$\forall t \in \mathbb{Z} : \mu_w(t) = \mu_{q_1}(t-k+1)\mu_{q_2}(t-k+2) \dots \mu_{q_k}(t). \quad (3.19)$$

- there exists a set  $L \subseteq Q^+$ , such that  $\mu_w = 0$  for all  $w \notin L$ ,
- for each  $w, v \in Q^+$ , the process  $[\mu_w, \mu_v]^T$  is wide-sense stationary,
- the linear span of  $\{\mu_q\}_{q \in Q}$  contains the constant 1 random variable, and
- there exists a set  $L \subseteq Q^+$ , such that the conditional expectation of  $\mu_{vq}(t)\mu_{wq'}(t)$  with respect to past values  $\{\mu_q(s)\}_{s < t}$  is either zero if  $q \neq q'$  or it is  $\mathbf{p}(\mu_q)\mu_v(t-1)\mu_w(t-1)$  for some positive real numbers  $\mathbf{p}(\mu_q) > 0$ .

That is, the random variable  $\{\mu_w(t)\}_{w \in Q^+}$  are mutually orthogonal for each fixed time instance  $t$ , and they have a Markov-like property: the conditional expectation of  $\mu_{vq}(t)\mu_{wq}(t)$  with respect to the past values of  $\{\mu_q\}$  depends only on the past values  $\mu_v(t-1)\mu_w(t-1)$  and a number  $p_q$ . Intuitively,  $p_q$  is such that if  $z(t)$  is a random process which is conditionally independent of future inputs with respect to the past outputs, then  $E[(z(t)\mu_{wq}(t))(z(t)\mu_{vq}(t))] = p_q E[(z(t)\mu_w(t-1))(z(t)\mu_v^T(t-1))]$ , i.e.,  $p_q$  describes a recursion for the covariances of the random variables  $\mathbf{z}(t)\mu_w(t)$  and the recursion is on the length of  $w$ .

Below we present some examples of admissible input processes.

### 3.4. MATHEMATICAL FRAMEWORK

**Example 3.1** (Zero mean i.i.d. input). Let  $Q = \{0, 1, \dots, d\}$ ,  $\mu_0 = 1$ , and assume that for each  $i = 1, \dots, d$ ,  $\mu_i$  is a zero mean and it is an i.i.d. process. and for each  $t \in \mathbb{Z}$ ,  $\mu(t)$  is square integrable and  $E[\mu_q^2(t)] = \mathbf{p}(\mu_q)$ . then  $\{\mu_q\}_{q \in Q}$  is a collection of admissible input processes.

**Example 3.2** (Discrete valued i.i.d process). Assume there exists an i.i.d process  $\theta$  which takes its values from a finite set  $Q$ . Let  $\mu_q(t) = \chi(\theta(t) = q)$  for all  $q \in Q$ ,  $t \in \mathbb{Z}$ . Let  $S = Q \times Q$ ,  $L = Q^+$  and  $\mathbf{p}(\mu_q) = P(\theta(t) = q)$ ,  $\alpha_q = 1$  for all  $q \in Q$ . Then  $\{\mu_q\}_{q \in Q}$  is a collection of admissible input processes.

**Example 3.3** (Markov chain). Assume that  $\theta$  is a stationary and ergodic Markov process whose state space is the finite set  $\Theta$ . Assume  $P(\theta(t) = q_2 | \theta(t-1) = q_1) = p_{(q_2, q_1)} > 0$ ,  $q_1, q_2 \in \Theta$ . Let us take  $Q = \Theta \times \Theta$ ,  $\mu_{(q_2, q_1)}(t) = \chi(\theta(t+1) = q_2, \theta(t) = q_1)$  for all  $q_1, q_2 \in \Theta, t \in \mathbb{Z}$ . Then  $\{\mu_q\}_{q \in Q}$  with  $\mathbf{p}(\mu_q) = p_{(q_2, q_1)}$ ,  $q = (q_2, q_1)$  and  $\alpha_q = 1$  for all  $q \in Q$  is a collection of admissible input processes.

For the rest of the chapter, unless stated otherwise,  $\{\mu_q\}_{q \in Q}$  is assumed to be admissible.

In order to understand why admissible input processes are necessary, we will introduce the class of ZMWSSI processes with respect to  $\{\mu_q\}_{q \in Q}$  (ZMWSSI w.r.t.  $\{\mu_q\}_{q \in Q}$  for short, or ZMWSSI if  $\{\mu_q\}_{q \in Q}$  is clear from the context), see [226] for a complete definition. As it was mentioned above,  $\{\mu_q\}_{q \in Q}$  is assumed to be admissible, and the notion of a ZMWSSI process w.r.t.  $\{\mu_q\}_{q \in Q}$  is defined only for admissible input processes  $\{\mu_q\}_{q \in Q}$ . For the purposes of this chapter, it is sufficient to remember the following. Let us define the processes

$$\forall t \in \mathbb{Z} : \mathbf{z}_w^r(t) = \mathbf{r}(t - |w|) \mu_w(t-1) \frac{1}{\sqrt{\mathbf{p}(\mu_w)}}. \quad (3.20)$$

where  $\mathbf{p}(\mu_\varepsilon) = 1$  and if  $w = vq$  for some  $v \in Q^*$  and  $q \in Q$ , then  $\mathbf{p}(\mu_w) = \mathbf{p}(\mu_v) \mathbf{p}(\mu_q)$ , and the covariances

$$\begin{aligned} \forall w \in Q^+ : \Lambda_w^r &= E[\mathbf{r}(t) (\mathbf{z}_w^r(t))^T], \\ \forall w, v \in Q^+ : T_{v, w}^r &= E[\mathbf{z}_v^r(t) (\mathbf{z}_w^r(t))^T]. \end{aligned} \quad (3.21)$$

Note that if  $\mathbf{y}$  is ZMWSSI, then the notation of (3.21) applied to  $\mathbf{y}$  is consistent with the notation of (3.5). Then  $\mathbf{r}$  is a ZMWSSI process with respect to (w.r.t.)  $\{\mu_q\}_{q \in Q}$ , if

- the right-hand side of (3.21) does not depend on  $t$ ,
- the future values of  $\{\mu_q(s)\}_{s \geq t}$  of  $\{\mu_q\}_{q \in Q}$  do not depend on the past values  $\{\mathbf{r}(s)\}_{s \leq t}$  of  $\mathbf{r}$ ; more precisely, the  $\sigma$ -algebras generated by  $\{\mathbf{r}(s)\}_{s \leq t}$  and  $\{\mu_q(s)\}_{q \in Q, s \geq t}$  respectively are conditionally independent with respect to the  $\sigma$  algebra generated by  $\{\mu_q(s)\}_{s < t}$ ;

If  $\mathbf{r}$  is ZMWSSI w.r.t.  $\{\mu_q\}_{q \in Q}$ , then the following recursion holds:

$$T_{q, q'}^r = 0, \quad \text{if } q \neq q' \quad (3.22)$$

$$T_{wq, vq'}^r = \begin{cases} T_{w, v}^r & \text{if } q = q' \text{ and } wq, vq \in L \\ 0 & \text{otherwise,} \end{cases} \quad (3.23)$$

$$T_{wq, q'}^r = \begin{cases} (\Lambda_w^r)^T & \text{if } q = q' \\ 0 & \text{if } q \neq q'. \end{cases} \quad (3.24)$$

That is, if we interpret  $Q^+$  as an additional time axis, then  $\mathbf{z}_w^r$  are wide-sense stationary w.r.t. to both  $t$  and  $w$ .

In order to develop realization theory, we have restrict attention to admissible inputs and to GBSs for which the state and noise processes are ZMWSSI w.r.t. the admissible inputs. We will call such GBSs stationary GBSs. More precisely, we will say that a GBS of the form (3.2) is stationary w.r.t.  $\{\mu_q\}_{q \in Q}$  (stationary, if  $\{\mu_q\}_{q \in Q}$  clear from the context, if the following conditions hold:

1.  $\{\boldsymbol{\mu}_q\}_{q \in Q}$  are admissible inputs,
2. The joint state and noise process  $[\mathbf{x}^T, \mathbf{v}^T]^T$  is ZMWSSI with respect to  $\{\boldsymbol{\mu}_q\}_{q \in Q}$ .
3.  $\forall w \in Q^+, t \in \mathbb{Z} : E[\mathbf{v}(t - |w|)\boldsymbol{\mu}_w(t - 1)\mathbf{v}^T(t)] = 0$ .
4.  $\forall w \in Q^+, t \in \mathbb{Z} : E[\mathbf{x}(t - |w|)\boldsymbol{\mu}_w(t - 1)\mathbf{v}^T(t)] = 0$  and  $\forall q \in Q, t \in \mathbb{Z} : E[\mathbf{x}(t)\boldsymbol{\mu}_q^2(t)\mathbf{v}^T(t)] = 0$ .
5. The matrix  $\sum_{q \in Q} \mathbf{p}(\boldsymbol{\mu}_q)A_q \otimes A_q$  is stable, i.e., all its eigenvalues are inside the open unit disk.

Intuitively, Part 2 requires that the input and noise processes are wide sense stationary and conditionally independent from the future inputs given the past inputs. Part 3 – 4 require the noise process to be uncorrelated and the past state and future and present noises to be uncorrelated. Part 5 is a stability condition, which is necessary for the state process to be wide sense stationary. It could be tempting to simplify the definition of stationarity by requiring the noise process to be completely independent of the inputs and of the past states. Unfortunately, this would be too restrictive for realization theory, since the realization algorithm yields a system whose noise process is uncorrelated with but not necessarily independent of the past inputs. More precisely, in realization theory we would like to construct a system from  $\mathbf{y}(t)$  which satisfies our chosen set of assumptions. However, in a system constructed from  $\mathbf{y}(t)$ , the noise process  $\mathbf{v}(t)$  and the state  $\mathbf{x}(t)$  must be functions of past outputs and inputs, as we have no other information at our disposal. Hence, the noise process will then depend on the past inputs, and if the inputs are not independent, for example, they are a Markov process, it will depend on future inputs too. Similarly, as both the noises and states depend on past inputs and outputs, in general, they will not be independent. The best we can hope for is to replace independence by zero correlation. Since our processes are not Gaussian, the latter does not translate into independence. The reason why we cannot assume Gaussian distribution of the state and noise, is the presence of products of inputs and states (noises) in (3.1): products of Gaussian are not Gaussian.

Note that the state of a stationary GBS is uniquely determined by the system matrices, and past inputs and noises. To state this relationship, let us recall from Notation 2.1, Section 2.2 the notation for products of matrices indexed by sequences, applied to  $\{A_q\}_{q \in Q}$  and  $X = Q$ . If a GBS  $\mathcal{G}$  of the form (3.1) is stationary, then

$$\begin{aligned} \mathbf{x}(t) &= \sum_{N=0}^{\infty} \sum_{w \in Q^*, |w|=N, q \in Q} \sqrt{\mathbf{p}(\boldsymbol{\mu}_{qw})} A_w K_q \mathbf{z}_{qw}^{\mathbf{v}}(t). \\ \tilde{\mathbf{y}}(t) &= \sum_{N=0}^{\infty} \sum_{w \in Q^*, |w|=N, q \in Q} \sqrt{\mathbf{p}(\boldsymbol{\mu}_{qw})} A_w K_q \mathbf{z}_{qw}^{\mathbf{v}}(t). \end{aligned} \quad (3.25)$$

for all  $t \in \mathbb{Z}$ , where the infinite sum on the right-hand converges in the mean-square sense. In fact, in [226, Lemma 3] it is shown that in order to define a stationary GBS, it is sufficient to define the system matrices and a noise process in such a way that they satisfy the corresponding conditions of a stationary GBS then this choice uniquely determines the state process, which will satisfy (3.25).

### 3.5 Formal description of the relationship with realization theory of LSS

In this section we present the formal description of the relationship between the realization problem for GBSs and that of for LSSs.

Recall from Section 3.3 the informal definition of a LSS associated with a stationary GBS. The formal definition is as follows.

**Definition 3.3** (LSS associated with a GBS). *Let  $\mathcal{G}$  be a stationary GBS realization of  $\mathbf{y}$  of the form (3.1). Define the LSS  $\Sigma_{\mathcal{G}}$  associated with  $\mathcal{G}$  as  $\Sigma_{\mathcal{G}} = (n, \{(\sqrt{\mathbf{p}(\boldsymbol{\mu}_q)}A_q, B_q, C_q) \mid q \in Q\})$ , where  $C_q = C$  for all  $q \in Q$ , and the*

### 3.5. FORMAL DESCRIPTION OF THE RELATIONSHIP WITH REALIZATION THEORY OF LSS

matrices  $B_q$ ,  $q \in Q$  are defined as follows:

$$B_q = \frac{1}{\sqrt{\mathbf{p}(\boldsymbol{\mu}_q)}} (A_q P_q C^T + K_q Q_q D^T) \quad (3.26)$$

where  $\{P_q = E[\mathbf{x}(t)\mathbf{x}^T(t)\boldsymbol{\mu}_q^2(t)]\}_{q \in Q}$ .

**Remark 3.1** (Computation of  $\Sigma_{\mathcal{G}}$ ). Note that in order to compute the matrices  $\{B_q\}_{q \in Q}$  from Definition 3.3, we can use the fact from [226] that  $\{P_q = E[\mathbf{x}(t)\mathbf{x}^T(t)\boldsymbol{\mu}_q^2(t)]\}_{q \in Q}$  is the unique collection of positive semi-definite matrices  $\{P_q\}_{q \in Q}$  which satisfy the Lyapunov-like equation

$$P_q = \mathbf{p}(\boldsymbol{\mu}_q) \sum_{q_1 \in Q, q_1 q \in L} (A_{q_1} P_{q_1} A_{q_1}^T + K_{q_1} Q_{q_1} K_{q_1}^T), \quad (3.27)$$

where  $Q_{q_1} = E[\mathbf{v}(t)\mathbf{v}^T(t)\boldsymbol{\mu}_{q_1}^2(t)]$ ,  $q_1 \in Q$ . In turn, we can use [226, Lemma 5] to note that  $P_q = \lim_{N \rightarrow \infty} P_q^N$ , where the sequence  $\{P_q^N\}_{q \in Q, 0 \leq N \in \mathbb{Z}}$  is defined by  $P_q^0 = 0$  and

$$P_q^{N+1} = \mathbf{p}(\boldsymbol{\mu}_q) \sum_{q_1 \in Q, q_1 q \in L} (A_{q_1} P_{q_1}^N A_{q_1}^T + K_{q_1} Q_{q_1} K_{q_1}^T) \quad (3.28)$$

First, we show that the LSS associated with a GBS  $\mathcal{G}$  are realizations of the input-output map  $f_{\mathbf{y}}$  associated with the output  $\mathbf{y}$  of  $\mathcal{G}$ . To this end, we need to recall from [227] or Appendix B.6 that a LSS  $(n, \{A_q, B_q, C_q\}_{q \in Q})$  is called *strongly stable*, if the eigenvalues of the matrix  $\sum_{q \in Q} A_q \otimes A_q$  are all inside the unit circle. Strongly stable LSS are closed under minimization and have a number of useful properties [227].

More precisely, using [226, Lemma 4], [227, Lemma 11] [209, Lemma 1] we can derive the following.

**Lemma 3.1** (From GBS to LSS). *If  $\mathcal{G}$  stationary GBS which is a realization of  $\mathbf{y}$ , then  $\mathbf{y}$  is ZMWSSI w.r.t.  $\{\boldsymbol{\mu}_q\}_{q \in Q}$  and the LSS  $\Sigma_{\mathcal{G}}$  is a realization of the input-output map  $f_{\mathbf{y}}$ . Moreover,  $\Sigma_{\mathcal{G}}$  is strongly stable.*

The converse of Lemma 3.1 also holds. That is, we can show that LSS realizations of the input-output map  $f_{\mathbf{y}}$  give rise to GBS realizations of the process  $\mathbf{y}$ . In order to state this result formally, we will need a mild technical assumption on  $\mathbf{y}$ .

**Definition 3.4** (SII process). *A process  $\mathbf{y}$  is said to be square integrable (SII) with respect to  $\{\boldsymbol{\mu}_q\}_{q \in Q}$  (SII w.r.t.  $\{\boldsymbol{\mu}_q\}_{q \in Q}$  for short, or SII if  $\{\boldsymbol{\mu}_q\}_{q \in Q}$  is clear from the context), if for all  $w \in \mathcal{Q}^+$ , the process  $\mathbf{z}_w^{\mathbf{y}^+}$  defined in (3.15), i.e.,*

$$\mathbf{z}_w^{\mathbf{y}^+}(t) = \mathbf{y}(t), \mathbf{z}_{q_1 \dots q_k}^{\mathbf{y}^+}(t) = \mathbf{y}(t+k) \prod_{i=1}^k \frac{\boldsymbol{\mu}_{q_i}(t+k-i)}{\sqrt{\mathbf{p}(\boldsymbol{\mu}_{q_i})}}, q_1, \dots, q_k \in Q, k > 0$$

is square integrable.

**Remark 3.2** (Conditions for  $\mathbf{y}$  being SII). *One may wonder when  $\mathbf{y}$  is SII. Below we present a class of process which are SII.*

*One case is when  $\mathbf{y}$  zero mean wide-sense stationary and square integrable, and  $\boldsymbol{\mu}_q$  is essentially bounded for all  $q \in Q$ , i.e., there exists a constant  $K > 0$  such that  $|\boldsymbol{\mu}_q(t)| \leq K$  almost everywhere for all  $q \in Q, t \in \mathbb{Z}$ , then  $\mathbf{y}$  is SII. This is the case when for example  $\boldsymbol{\mu}_q$  arises from a discrete value. In this case, for any  $w \in \mathcal{Q}^+$  we have  $E[(\mathbf{z}_w^{\mathbf{y}^+}(t))^T \mathbf{z}_w^{\mathbf{y}^+}(t)] \leq E[\mathbf{y}^T(t+|w|)\mathbf{y}(t+|w|)] K^2 \frac{1}{p_w} < +\infty$ .*

*Another case is when  $\mathbf{y}(t)$  is zero mean wide-sense stationary and square integrable, and  $\boldsymbol{\mu}_w(t)$  have finite fourth order moments. In this case, by Hölders inequality, we have  $E[(\mathbf{z}_w^{\mathbf{y}^+}(t))^T \mathbf{z}_w^{\mathbf{y}^+}(t)] \leq (E[(\mathbf{y}^T(t+k)\mathbf{y}(t+k))^2] E[\boldsymbol{\mu}_w^4(t+k)])^{1/2} < +\infty$ , where The right-hand side of the inequality above is finite by assumption. This assumption was made in [73]. In particular, if the inputs are bounded, and  $\mathbf{y}$  is ZMWSSI, then  $\mathbf{y}$  is SII.*

Now we are ready to state the result relating LSS realizations of  $f_{\mathbf{y}}$  with GBS realizations of  $\mathbf{y}$ .



**Lemma 3.2** (From LSS to GBS). *If  $\mathbf{y}$  is ZMWSSI and SII w.r.t.  $\{\boldsymbol{\mu}_q\}_{q \in Q}$ ,  $f^{\mathbf{y}}$  has a realization by a strongly stable LSS then  $\mathbf{y}$  has a realization by stationary GBS.*

The proof of Lemma 3.2 is constructive, as it relies on the notion of a GBS realization  $\mathcal{G}_\Sigma$  associated with a minimal LSS realization of  $f^{\mathbf{y}}$ . The latter was already mentioned in Section 3.3. Below we will review the definition of the GBS associated with a LSS and discuss how to compute it. The construction of the latter GBS is instrumental for developing realization algorithms for GBSs. Moreover, the construction of the GBS also introduces the notion of forward innovation form, which is interesting on its own right.

In order to continue the discussion, we need the notion of forward innovation process and GBS in forward innovation form. To this end, we introduce the following notation.

**Notation 3.1** (Orthogonal projection  $E_l$ ). *Let  $Z$  be a subset of square integrable random variables in  $\mathbb{R}^p$  and let  $M$  be the Hilbert space generated by the coordinates of the elements of  $Z$ . Let  $\mathbf{z}$  be a square integrable random variable with values in  $\mathbb{R}^k$ . We denote by  $E_l[\mathbf{z} | Z]$  the orthogonal projection of  $\mathbf{z}$  onto the subspace  $M$ . By the orthogonal projection of  $\mathbf{z}$  onto  $M$  we mean the vector-valued random variable  $\mathbf{z}^* = (\mathbf{z}_1^*, \dots, \mathbf{z}_k^*)$  such that  $\mathbf{z}_i^* \in M$  is the orthogonal projection of the  $i$ th coordinate  $\mathbf{z}_i$  of  $\mathbf{z}$  onto  $M$ , as it is usually defined for Hilbert spaces.*

**Definition 3.5** (Forward innovation process). *The forward innovation process of  $\mathbf{y}$  w.r.t.  $\{\boldsymbol{\mu}_q\}_{q \in Q}$  (or simply the innovation process is  $\{\boldsymbol{\mu}_q\}_{q \in Q}$  is clear from the context) is defined as the process  $\mathbf{e}$ :*

$$\forall t \in \mathbb{Z} : \mathbf{e}(t) = \mathbf{y}(t) - E_l[\mathbf{y}(t) | \{\mathbf{z}_w^{\mathbf{y}}(t)\}_{w \in Q^+}]. \quad (3.29)$$

We say that  $\mathbf{y}$  is *full rank*, if for each  $q \in Q$  the covariance matrix  $E[\mathbf{e}(t)\mathbf{e}^T(t)\boldsymbol{\mu}_q^2(t)]$  is invertible.

Now we are ready to define the notion of GBS in forward innovation form.

**Definition 3.6** (GBS in forward innovation form). *Let  $\mathcal{G}$  be a stationary GBS of the form (3.1) which is a realization of  $\mathbf{y}$ . We say that  $\mathcal{G}$  is in forward innovation form, if  $\mathbf{v} = \mathbf{e}$  and  $D = I_p$ , i.e. the noise process of  $\mathcal{G}$  is the forward innovation process of  $\mathbf{y}$ .*

Intuitively, the innovation process  $\mathbf{e}$  is the prediction error of the best linear predictor of  $\mathbf{y}$ , if the prediction is done based on the products of the past values of  $\mathbf{y}$  with the past inputs. Hence GBS realization of  $\mathbf{y}$  in forward innovation can be seen as a *Kalman-filter* for predicting the current value of  $\mathbf{y}$ .

Now we are in position to review the notion of the GBS associated with a minimal LSS realization of  $f^{\mathbf{y}}$ . To this end assume that  $\mathbf{y}$  is ZMWSSI and SII, and let  $\Sigma = (n, \{(A_q, B_q, C_q) | q \in Q\})$  be a strongly stable LSS realization of  $f^{\mathbf{y}}$ . Since minimization preserves strong stability, see [227, Theorem 3] or Theorem B.5, Appendix B.6, we can assume that  $\Sigma$  is minimal. In this case,  $C_{q_1} = C_{q_2}$  for all  $q_1, q_2 \in Q$ . Indeed, for all  $w \in Q^+$ ,  $C_{q_1}A_wB_q = S_{q_1, q}^{f^{\mathbf{y}}}(w) = \Lambda_{q_1}^{\mathbf{y}} = S_{q_2, q}^{f^{\mathbf{y}}}(w) = C_{q_2}A_wB_q$ , i.e.,  $C_{q_1}x = C_{q_2}x$  for any  $x \in \text{Im}\mathcal{R}(\mathcal{S})$ . Since  $\Sigma$  is minimal and thus span-reachable,  $\mathcal{R}(\Sigma) = \mathbb{R}^n$  and hence  $C_{q_1} = C_{q_2}$ . In the sequel, we denote by  $C$  the matrix  $C_q$ ,  $q \in Q$ .

Using the discussion above, we can define the GBS associated with  $\Sigma$  as follows.

**Definition 3.7** (GBS associated with  $\Sigma$ ). *Let  $\Sigma = (n, \{(A_q, B_q, C_q) | q \in Q\})$  be a minimal strongly stable LSS realization of  $f^{\mathbf{y}}$ . The GBS associated with the  $\Sigma$  is defined as*

$$\mathcal{G}_\Sigma = (n, p, p, Q, \mathbf{x}, \mathbf{e}, \{\boldsymbol{\mu}_q\}_{q \in Q}, \mathbf{y}, C, I_p, \left\{ \frac{1}{\sqrt{\mathbf{p}(\boldsymbol{\mu}_q)}} A_q, K_q \right\}_{q \in Q}), \quad (3.30)$$

where  $C_q = C$  for all  $q \in Q$ , and

$$\forall t \in \mathbb{Z} : \mathbf{x}(t) = E_l[\tilde{\mathcal{E}}^{-1}(\Sigma)(Y_n(t)) | \{\mathbf{z}_w^{\mathbf{y}}(t)\}_{w \in Q^+}], \quad (3.31)$$

where  $Y_n(t)$  is as in (3.14), i.e.,

$$Y_n(t) = \left[ (\mathbf{z}_{v_1}^{\mathbf{y}^+}(t))^T, (\mathbf{z}_{v_2}^{\mathbf{y}^+}(t))^T \quad \dots \quad (\mathbf{z}_{v_{N(n-1)}}^{\mathbf{y}^+}(t))^T \right]^T$$

### 3.6. MAIN RESULTS ON REALIZATION THEORY OF GBSS

where  $\mathbf{z}_v^+(t)$ ,  $v \in Q^*$  is as in (3.15), and  $\tilde{\mathcal{O}}^{-1}(\Sigma)$  is a left inverse of  $\tilde{\mathcal{O}}(\Sigma)$ , and  $\tilde{\mathcal{O}}(\Sigma)$  is defined in the same way as the observability matrix  $\mathcal{O}(\Sigma)$  of  $\Sigma$  (see (B.5) in Appendix B), but with  $C_1$  instead of  $\tilde{C}$ <sup>2 3 4</sup>, i.e.,

$$\tilde{\mathcal{O}}(\Sigma) = \left[ (C_1 A_{v_1})^T, (C_1 A_{v_2})^T, \dots, (C_1 A_{v_{\mathbf{N}(n-1)}})^T \right]^T \in \mathbb{R}^{p\mathbf{N}(n-1) \times n} \quad (3.32)$$

and  $K_q$  satisfies (3.16), i.e.

$$K_q = (B_q \sqrt{\mathbf{p}(\mu_q)} - \frac{1}{\sqrt{\mathbf{p}(\mu_q)}} A_q P_q C^T) (E[\mathbf{e}(t)\mathbf{e}(t)\mu_q^2(t)])^{-1}, \quad P_q = E[\mathbf{x}(t)\mathbf{x}^T(t)\mu_q^2(t)].$$

**Remark 3.3** (Computing  $\mathcal{G}_\Sigma$ ). *In order to compute the matrices of  $\mathcal{G}_\Sigma$  we can proceed as follows: compute the sequence  $\{P_q^N, K_q^N\}_{0 \leq N \in \mathbb{Z}, q \in Q}$  as follows: set  $\hat{P}_q^0 = 0$  and for all  $i = 0, \dots, N$*

$$\begin{aligned} P_q^{i+1} &= \sum_{q_1 \in Q, q_1 q \in L} \mathbf{p}(\mu_q) (A_{q_1} P_{q_1}^i A_{q_1}^T + K_{q_1}^i Q_{q_1}^i (K_{q_1}^i)^T) \\ Q_q^i &= (\mathbf{p}(\mu_q) T_{q,q}^y - C P_q^i C^T) \\ K_q^i &= (B_q \sqrt{\mathbf{p}(\mu_q)} - A_q P_q^i C^T) (Q_q^i)^{-1}. \end{aligned}$$

Then from [226, Theorem 3] it follows  $K_q = \lim_{N \rightarrow \infty} K_q^N$ ,  $P_q = \lim_{N \rightarrow \infty} P_q^N$ ,  $q \in Q$ .

Based on [226, Theorem 4] we can state the following relationship between  $\Sigma$  and the GBS associated with  $\Sigma$ .

**Lemma 3.3.** *The GBS  $\mathcal{G}_\Sigma$  is stationary and it is a realization of  $\mathbf{y}$  in forward innovation form.*

Note that [226, Theorem 4] states the result in terms of representations of recognizable formal power series instead of discrete-time LSSs. However, there is a one-to-one correspondence between the two, see [209], so the statement of Lemma 3.3 and [226, Theorem 4] are equivalent.

Lemma 3.2 is an easy consequence of Lemma 3.3. Indeed, assume that there exists a strongly stable realization of  $f^y$ . Then by [227, Theorem 3] or Theorem B.5 from Appendix B.6, any minimal LSS realization of  $f^y$  is also strongly stable. Let then  $\Sigma$  be a minimal realization of  $f^y$ . Then  $\Sigma$  is strongly stable and by Lemma 3.3  $\mathcal{G}_\Sigma$  is a stationary GBS realization of  $\mathbf{y}$ .

Notice that the LSS associated with a GBS  $\mathcal{G}$  has the same state-space dimension as  $\mathcal{G}$ . Conversely, the GBS  $\mathcal{G}_\Sigma$  has the same state-space dimension as the LSS  $\Sigma$ . Moreover, the LSS associated with  $\mathcal{G}_\Sigma$  is  $\Sigma$  itself. This means that we can state the following result relating minimal LSS realizations of  $f^y$  and minimal stationary GBS realizations of  $\mathbf{y}$ .

**Lemma 3.4.** *A stationary GBS  $\mathcal{G}$  is a minimal realization of  $\mathbf{y}$ , if and only if the associated LSS  $\Sigma_{\mathcal{G}}$  is minimal. In particular, the GBS  $\mathcal{G}_\Sigma$  from Lemma 3.3 is a minimal realization of  $\mathbf{y}$ .*

Lemma 3.4 follows from [226, Appendix.D, Proof of Theorem 2], by using the correspondence between representations of recognizable formal power series and discrete-time LSSs [209].

The correspondence between GBSs and LSS described above allow us to derive a fairly complete realization theory of GBSs using realization theory of LSSs. These results will be reviewed briefly in the next section.

## 3.6 Main results on realization theory of GBSs

Below we will discuss conditions for existence and minimality of GBSs and minimal realization algorithms for GBSs. The results to be presented below are based on the correspondence between GBSs and LSSs.

<sup>2</sup>Here, without loss of generality, we assume that  $Q = \{1, \dots, D\}$  (otherwise, the elements of  $Q$  can be suitably renamed)

<sup>3</sup>Alternatively,  $\tilde{\mathcal{O}}(\Sigma)$  is obtained from the observability matrix  $\mathcal{O}(\Sigma)$  of  $\Sigma$  by taking the first  $p$  rows of the block  $\tilde{C}A_{v_i}$ ,  $i = 1, \dots, \mathbf{N}(n-1)$  and deleting the other rows of the same block. Note that the deleted rows are equals to one of the first  $p$  rows of the block  $\tilde{C}A_{v_i}$ .

<sup>4</sup>Since  $\Sigma$  is minimal, it is observable and its observability matrix is full column rank, hence it has a left inverse.

### 3.6.1 Existence of a realization

In order to formulate conditions for existence of GBSs, we need to define the notion of a Hankel-matrix defined from covariances of  $\mathbf{y}$ . To this end, without loss of generality we assume that  $Q = \{1, \dots, D\}$  and define the map  $\Psi_{\mathbf{y}} : Q^* \rightarrow \mathbb{R}^{p \times pD}$  as

$$\forall w \in Q^* : \Psi_{\mathbf{y}}(w) = [\Lambda_{1w}^{\mathbf{y}}, \Lambda_{2w}^{\mathbf{y}}, \dots, \Lambda_{Dw}^{\mathbf{y}}],$$

and using the lexicographic ordering of Definition 2.1 define the Hankel-like matrix  $H_{\mathbf{y},i,j}$  of  $\mathbf{y}$  as

$$H_{\mathbf{y},i,j} = \begin{bmatrix} \Psi_{\mathbf{y}}(v_1 v_1), & \Psi_{\mathbf{y}}(v_2 v_1), & \dots & \Psi_{\mathbf{y}}(v_{N(j)} v_1) \\ \Psi_{\mathbf{y}}(v_1 v_2), & \Psi_{\mathbf{y}}(v_2 v_2), & \dots & \Psi_{\mathbf{y}}(v_{N(j)} v_2) \\ \vdots & \vdots & \dots & \vdots \\ \Psi_{\mathbf{y}}(v_1 v_{M(i)}), & \Psi_{\mathbf{y}}(v_2 v_{M(i)}), & \dots & \Psi_{\mathbf{y}}(v_{N(j)} v_{N(i)}) \end{bmatrix} \quad (3.33)$$

We say that the covariance sequence  $\Psi_{\mathbf{y}}$  is *recognizable*, if  $\sup_{0 \leq i, j \in \mathbb{Z}, i, j \geq 1} \text{rank} H_{\mathbf{y},i,j} = n < +\infty$  for some positive integer  $n$ .  $\Psi_{\mathbf{y}}$  is *square summable*, if

$$\|\Psi_{\mathbf{y}}(\varepsilon)\|_F^2 + \sum_{k=1}^{\infty} \sum_{q_1 \in Q} \dots \sum_{q_k \in Q} \|\Psi_{\mathbf{y}}(q_1 q_2 \dots q_k)\|_F^2 \quad (3.34)$$

is convergent, where  $\|\cdot\|_F$  is the Frobenius norm.

We can then state the following necessary and sufficient conditions for existence of a stationary GBS realization.

**Theorem 3.1** (Necessary and sufficient conditions for existence of a GBS realization). *Assume  $\mathbf{y}$  is SII w.r.t.  $\{\mu_q\}_{q \in Q}$ . Then  $\mathbf{y}$  has a realization by a stationary GBS if and only if  $\mathbf{y}$  is ZMWSSI and the covariance sequence  $\Psi_{\mathbf{y}}$  is recognizable and square summable.*

Theorem 3.1 is a reformulation of [226, Theorem 1], and it is a direct consequence of Lemma 3.1, Lemma 3.2 and [209, Theorem 2] (Theorem B.3 of Appendix B). Indeed, notice that  $\text{rank} H_{\mathbf{y},i,j} = \text{rank} H_{f^{\mathbf{y}},i,j}$ , where  $H_{f^{\mathbf{y}},i,j}$  is the Hankel-matrix defined in [209, Definition 13] ((B.12) of Appendix B). In fact,  $H_{\mathbf{y},i,j}$  is obtained from  $H_{f^{\mathbf{y}},i,j}$  by keeping the first  $p$  rows of each block row  $[M^f(v_1 v_r) \quad M^f(v_2 v_r) \quad \dots \quad M^f(v_{N(M)} v_r)]$ . It is easy to see that  $\text{rank} H_{f^{\mathbf{y}},i,j} = \sup_{i,j} \text{rank} H_{f^{\mathbf{y}},i,j}$ , and hence  $\Psi_{\mathbf{y}}$  being recognizable is equivalent to  $f^{\mathbf{y}}$  being realizable by an LSS. Note that  $\Psi_{\mathbf{y}}$  is square summable if and only if  $\sum_{w \in Q^*} \|M^f(w)\|_F^2$  is convergent, which [222] means that  $f^{\mathbf{y}}$  is realizable by a strongly stable discrete-time LSS, if it is realizable by an LSS at all.

### 3.6.2 Minimality of GBSs

Having established conditions for existence of a GBS realization, we turn to the question of minimality. We will call a stationary GBS  $\mathcal{G}$  a *minimal realization* of  $\mathbf{y}$  if it realizes  $\mathbf{y}$  and it has the minimal dimension among all possible stationary GBS realizations of  $\mathbf{y}$ .

Let  $\mathcal{G}$  be a stationary GBS of the form (3.1). For each  $k = 0, \dots, n-1$  define the  $k$ -step *observability matrix*  $\mathcal{O}_k(\mathcal{G})$  and the  $k$ -step *reachability matrix*  $\mathcal{R}_k(\mathcal{G})$  of  $\mathcal{G}$  recursively as follows:  $\mathcal{O}_0(\mathcal{G}) = C$ , and  $\mathcal{R}_0(\mathcal{G}) = [B_1 \quad \dots \quad B_D]$ , where  $B_q$  is as in (3.7), and for  $k > 0$

$$\begin{aligned} \mathcal{O}_k(\mathcal{G}) &= [C^T, (\mathcal{O}_{k-1}(\mathcal{G})A_1)^T, (\mathcal{O}_{k-1}(\mathcal{G})A_2)^T, \dots, (\mathcal{O}_{k-1}(\mathcal{G})A_D)^T]^T, \\ \mathcal{R}_k(\mathcal{G}) &= [\mathcal{R}_0(\mathcal{G}), A_1 \mathcal{R}_{k-1}(\mathcal{G}), A_2 \mathcal{R}_{k-1}(\mathcal{G}), \dots, A_D \mathcal{R}_{k-1}(\mathcal{G})]^T. \end{aligned}$$

We say that  $\mathcal{G}$  satisfies the *observability rank condition*, if  $\text{rank} \mathcal{O}_{n-1}(\mathcal{G}) = n$ , and we say that  $\mathcal{G}$  satisfies the *reachability rank condition* if  $\text{rank} \mathcal{R}_{n-1}(\mathcal{G}) = n$ .

### 3.6. MAIN RESULTS ON REALIZATION THEORY OF GBSS

Consider a stationary GBSs  $\mathcal{G}$  of the form (3.1), and let  $\hat{\mathcal{G}} = (n, p, \hat{m}, Q, \hat{\mathbf{x}}, \hat{\mathbf{v}}, \hat{\mathbf{y}}, \hat{C}, \hat{D}, \{\hat{A}_q, K_q\}_{q \in Q})$  another stationary GBS. An *isomorphism from  $\mathcal{G}$  to  $\hat{\mathcal{G}}$*  is a non-singular matrix  $T \in \mathbb{R}^{n \times n}$  such that  $\hat{\mathbf{x}} = T\mathbf{x}$ ,  $D\mathbf{v} = \hat{D}\hat{\mathbf{v}}$ , and

$$\begin{aligned} CT^{-1} &= \hat{C}, \quad TA_qT^{-1} = \hat{A}_q, \quad \forall q \in Q, \\ TK_q\mathbf{v}(t)\boldsymbol{\mu}_q(t) &= \hat{K}_q\hat{\mathbf{v}}(t)\boldsymbol{\mu}_q(t), \quad \forall q \in Q, t \in \mathbb{Z}. \end{aligned}$$

If, in addition,  $m = \hat{m}$ ,  $\mathbf{v} = \hat{\mathbf{v}}$  and  $D = \hat{D}$  and  $TK_q = \hat{K}_q$ ,  $q \in Q$ , then we call  $T$  a *strong isomorphism* from  $\mathcal{G}$  to  $\hat{\mathcal{G}}$ .

With the above notions of reachability and observability of GBSs, we can state the following characterization of minimality from [226, Theorem 2].

**Theorem 3.2** (Minimality of GBSs). *Assume  $\mathbf{y}$  is ZMSII w.r.t.  $\{\boldsymbol{\mu}_q\}_{q \in Q}$ .*

- (i) *A stationary GBS realization of  $\mathbf{y}$  is a minimal realization of  $\mathbf{y}$  if and only if it satisfies the observability and the controllability rank conditions.*
- (ii) *If  $\mathcal{G}$  and  $\hat{\mathcal{G}}$  are minimal GBS realizations of  $\mathbf{y}$  in forward innovation form, then there exists an isomorphism from  $\mathcal{G}$  to  $\hat{\mathcal{G}}$ . If  $\mathbf{y}$  is in addition full rank, then this isomorphism is a strong one.*
- (iii) *If  $\mathbf{y}$  has a realization by a stationary GBS, then it has a minimal GBS realization in forward innovation form.*

Part (i) of Theorem 3.2 is a direct consequence of Lemma 3.4 and that minimality of LSSs is equivalent to their span-reachability and observability [209, Theorem 1] (Theorem B.1 of Appendix B). Note that  $\mathcal{O}_k(\mathcal{G})$  and  $\mathcal{R}_k(\mathcal{G})$  correspond to the spaces  $W_k$  and  $V_k$  of the associated LSS  $\Sigma_{\mathcal{G}}$  defined in [205, Remark 1] (Remark B.1 of Appendix B), and the observability/reachability rank condition for GBS is equivalent to  $\mathcal{W}_* = \{0\}$ ,  $\mathcal{V}_* = \mathbb{R}^n$ , which in turn are equivalent to observability/span-reachability of  $\Sigma_{\mathcal{G}}$ . Part (ii) is a consequence of isomorphism of minimal LSSs, the argument is not entirely trivial, see [226] for the proof. Part (iii) is a direct consequence of the second statement of Lemma 3.4.

**Remark 3.4.** *Note that we define isomorphism between GBSs as a linear transformation which is constant, i.e. it does not depend on the inputs. Theorem 3.2 implies that such state-space transformations are sufficient to relate minimal GBSs in forward innovation form which realize the same output.*

For the linear case, i.e.  $Q = \{0\}$ ,  $\boldsymbol{\mu}_0 = 1$ , Theorem 3.2 boils down to well known results [163, Theorem 6.5.2, Theorem 6.5.4, Theorem 6.6.1, Theorem 8.7.4].

#### 3.6.3 Realization and minimization algorithms

Below we present algorithms for checking minimality of GBSs and for transforming a stationary GBS to a minimal one with the same output. Moreover, we present a Ho-Kalman-like algorithm for computing a minimal stationary GBS realization of  $\mathbf{y}$  from covariances  $\{\Lambda_w^{\mathbf{y}}\}_{w \in Q^+, |w| \leq N}$ ,  $\{T_{\sigma, \sigma}^{\mathbf{y}}\}_{\sigma \in Q}$  of  $\mathbf{y}$ .

##### Checking minimality

In order to check minimality of a GBS  $\mathcal{G}$ , the LSS  $\Sigma_{\mathcal{G}}$  associated with GBS  $\mathcal{G}$  has to be computed and its minimality checked as indicated in [205, Theorem 1 and Remark 1] (Remark B.1 and Theorem B.2 of Appendix B). In order to compute  $\Sigma_{\mathcal{G}}$  we can proceed as explained in Remark 3.1. That is, in order to compute  $\Sigma_{\mathcal{G}}$ , we have to calculate  $B_q$ ,  $q \in Q$  from (3.7), and for that we have to compute the state-covariances  $\{P_q\}_{q \in Q}$  from (3.8). By [226, Lemma 5] this can be done as follows.  $P_q = \lim_{N \rightarrow \infty} P_q^N$ , where the sequence  $\{P_q^N\}_{q \in Q, 0 \leq N \in \mathbb{Z}}$  is defined by  $P_q^0 = 0$  and

$$P_q^{N+1} = \mathbf{p}(\boldsymbol{\mu}_q) \sum_{q_1 \in Q, q_1 q \in L} (A_{q_1} P_{q_1}^N A_{q_1}^T + K_{q_1} Q_{q_1} K_{q_1}^T) \quad (3.35)$$

### Minimization algorithm

In order to transform a GBS  $\mathcal{G}$  to a minimal one, we can proceed as follows:

---

#### Algorithm 3 Minimization algorithm

**Input:** GBS  $\mathcal{G}$

**Output:** Minimal GBS  $\mathcal{G}_m$

---

- 1: Compute the associated LSS  $\Sigma_{\mathcal{G}}$ .
  - 2: Transform it to a minimal LSS  $\Sigma_m$  as explained in [209, Procedure 3] (see Procedure B.3 of Appendix B).
  - 3: Let  $\mathcal{G}_m = \mathcal{G}_{\Sigma_m}$ , where  $\mathcal{G}_{\Sigma_m}$  is the GBS associated with  $\Sigma_m$ , and  $\mathcal{G}_{\Sigma_m}$  is computed as explained in Remark 3.3. In order to compute the covariances  $\{T_{q,q}^y\}_{q \in Q}$  which are necessary to apply Remark 3.3, use the formula (3.35) for  $\mathcal{G}$  and set  $P_q = \lim_{N \rightarrow \infty} P_q^N$  and then use (3.9).
  - 4: **return**  $\mathcal{G}_m$
- 

### Covariance realization algorithm

Using a finite number of covariances  $\{\Lambda_w^y\}_{w \in Q^+, |w| \leq N+2}$  and  $\{T_{q,q}^y\}_{q \in Q}$  we can compute a GBS  $\mathcal{G}$  realization of  $\mathbf{y}$  as follows.

---

#### Algorithm 4 Covariance realization algorithm

**Input:** Integer  $N > 0$  and covariances  $\{\Lambda_w^y\}_{w \in Q^+, |w| \leq N+2}$  and  $\{T_{q,q}^y\}_{q \in Q}$

**Output:** A GBS  $\mathcal{G}_N$ .

---

- 1: Construct the Hankel-matrix  $H_{f_y, N, N+1}$  from  $\{\Lambda_w^y\}_{w \in Q^+, |w| \leq N+2}$ .
  - 2: Compute a minimal LSS  $\Sigma_N$  from  $H_{f_y, N, N+1}$  using [209, Algorithm 1] (Algorithm 12 of Appendix B) or an adaptation of Algorithm 2 to LSSs (Algorithm 13 of Appendix B).
  - 3: Compute the GBS  $\mathcal{G}_{\Sigma_N}$  associated with  $\Sigma_N$  as explained in Remark 3.3.
  - 4: **return**  $\mathcal{G}_N = \mathcal{G}_{\Sigma_N}$ .
- 

If  $N$  is larger than the dimension of a GBS realization of  $\mathbf{y}$ , then  $\mathcal{G}_N$  will be a minimal realization of  $\mathbf{y}$ .

## 3.6.4 Applications to JMLS

To illustrate the usefulness of our results, we apply them to jump-Markov linear systems.

To start with, consider the special case when  $\theta$  is an i.i.d process which takes its values from a finite set  $Q$ , and  $\mu_q(t) = \chi(\theta(t) = q)$  for all  $q \in Q$ ,  $t \in \mathbb{Z}$ , see Example 3.2. Then  $\{\mu_q\}_{q \in Q}$  is a collection of admissible input processes. With this choice of the input processes, stationary GBSs correspond to a subclass of *jump-Markov linear systems with i.i.d switching*, abbreviated by *JMLSIIID*. Since  $\mu_q(t)$  takes only values 0 and 1, from Remark 3.2 it follows that if  $\mathbf{y}$  is ZMWSSI, then  $\mathbf{y}$  is SII. Hence, in Theorem 3.1 – 3.2 the assumption that  $\mathbf{y}$  is SII can be dropped and the expression ‘stationary GBS’ can be replaced by JMLSIIID. The algorithms from Section 3.6.3 remain valid for JMLSIIIDs too.

Assume that  $\theta$  is a stationary and ergodic Markov process taking its values in a finite set  $\Theta = \{1, \dots, d\}$ . Define  $Q$  and  $\{\mu_q\}_{q \in Q}$  as in Example 3.3. Then  $\{\mu_q\}_{q \in Q}$  is an admissible collection of input processes. A stationary GBSs of the form (3.1) can then be written as  $\mathbf{x}(t+1) = A_{(\theta(t+1), \theta(t))} \mathbf{x}(t) + K_{(\theta(t+1), \theta(t))} \mathbf{v}(t)$ ,  $\hat{\mathbf{y}}(t) = C \mathbf{x}(t) + D \mathbf{v}(t)$ , and hence it can be regarded as a jump-Markov linear system. We refer to this subclass simply as *jump-Markov linear systems* or *JMLS* for short. Note that according to the standard definition [64], the state transition matrices of a jump-Markov linear system depend only on the current value of  $\theta$ . More precisely, [64] considers systems of the form  $\mathbf{z}(t+1) = F_{\theta(t)} \mathbf{z}(t) + G_{\theta(t)} \mathbf{w}(t)$ ,  $\hat{\mathbf{y}}(t) = H_{\theta(t)} \mathbf{z}(t) + L_{\theta(t)} \mathbf{w}(t)$ , where  $F_q, L_q, H_q, G_q$ ,  $q \in \Theta$  are matrices of suitable dimensions.

### 3.7. CONSEQUENCES FOR SYSTEM IDENTIFICATION: WEAK REALIZATIONS, GBSS AS OPTIMAL FILTERS, REALIZATION OF INPUT-OUTPUT RELATION VS. OUTPUT PROCESS

Such a system can be transformed a JMLS of the form (3.1) with the same output process  $\hat{\mathbf{y}}$ , see [225, Remark 9]. Conversely, a JMLS of the form (3.1) can be viewed as a jump-Markov system  $\mathbf{x}(t+1) = A_{\tilde{\theta}(t)}\mathbf{x}(t) + K_{\tilde{\theta}(t)}\mathbf{v}(t)$ ,  $\hat{\mathbf{y}}(t) = C\mathbf{x}(t) + D\mathbf{v}(t)$  in the sense of [64], driven by the Markov process  $\tilde{\theta}(t) = (\theta(t+1), \theta(t))$ . Alternatively, a JMLS can be converted to a system of the type [64] driven by the same Markov process  $\theta$  and producing the same output, see [225, Remark 10]. That is, JMLS and the systems from [64] represent the same set of output processes. In fact, [225, 223] uses realization theory of JMLS to solve the realization problem for a class of systems which contains the systems considered in [64]. From boundedness of  $\{\mu_\sigma\}_{\sigma \in \Sigma}$  and Remark 3.2 it follows that in Theorem 3.1 – Theorem 3.2 the assumption that  $\mathbf{y}$  is SII can be dropped and the expression ‘stationary GBS’ can be replaced by ‘JMLS’. The algorithms discussed in Section 3.6.3 remain valid for JMLSs too. That is, the results of this chapter yield a complete realization theory for JMLSs.

## 3.7 Consequences for system identification: weak realizations, GBSs as optimal filters, realization of input-output relation vs. output process

In this section we would like to discuss some implications of the results of this chapter which we believe are of fundamental importance for system identification. In particular, in Subsection 3.7.1 we show that GBSs can be interpreted not only as devices generating an output process, but also as partial parameterizations of probability distributions (high-order moments) and as optimal filters which predict current outputs based on past outputs and scheduling signals. In particular, in Subsection 3.7.1 we introduce the notion of a weak realization. In Subsection 3.7.2 we discuss the relationship between the realization problem formulated in Problem 3.1 and the problem of realizing the input-output relation  $\{\mu_q\}_{q \in Q} \mapsto \hat{\mathbf{y}}$  induced by a GBS.

### 3.7.1 Interpretations of GBS and their role in system identification

Recall from [57, 163, 147] that there are three views of linear stochastic systems:

- Stochastic systems as generators of an output process.
- Stochastic systems as parameterizations of probability distributions of the output process.
- Stochastic system as optimal filter, for predicting the current value of an output process based on its past values.

For stochastic LTI systems all three views are equivalent. However, for other classes of systems the equivalence is far from obvious. Below we will discuss the relationship between these three interpretations for GBSs.

Recall that if  $\mathcal{G}$  is a stationary GBS realization of  $\mathbf{y}$ , then the moments  $\{\Lambda_w^{\mathbf{y}}\}_{w \in Q^+}$ ,  $\{T_{q,q}^{\mathbf{y}}\}_{q \in Q}$  satisfy (3.9). That is, the matrices and noise variances of a stationary  $\mathcal{G}$  realization of  $\mathbf{y}$  determine some higher order order moments of  $\mathbf{y}$ . Note that these higher order moments do not determine the probability distribution of  $\mathbf{y}$  uniquely. This is in contrast with outputs of Gaussian stochastic linear time-invariant systems. However, the matrices of  $\mathcal{G}$  together with the noise variance can be viewed as finite encodings of the infinite number of moments  $\{\Lambda_w^{\mathbf{y}}\}_{w \in Q^+}$ ,  $\{T_{q,q}^{\mathbf{y}}\}_{q \in Q}$  of  $\mathbf{y}$ , and hence as a partial parameterization of the probability distribution of  $\mathbf{y}$ .

Conversely, consider a tuple  $\mathcal{S} = (\{A_q, K_q, Q_q\}_{q \in Q}, C, D)$ , such that  $\sum_{q \in Q} \mathbf{p}(\mu_q) A_q \otimes A_q$  is stable and  $Q_q = Q_q^T > 0$  for all  $q \in Q$ . It then follows from [226, Lemma2] that (3.8) has a unique solution and hence  $\{B_q\}_{q \in Q}$  from (3.7) are well defined. We will say that  $\mathcal{S}$  is a *weak realization of  $\mathbf{y}$* , if  $\{\Lambda_w^{\mathbf{y}}\}_{w \in Q^+}$ ,  $\{T_{q,q}^{\mathbf{y}}\}_{q \in Q}$  satisfy (3.9). The intuition behind this definition is as follows. Assume that  $\mathcal{S} = (\{A_q, K_q, Q_q\}_{q \in Q}, C, D)$  is a weak realization of  $\mathbf{y}$ , and consider the GBS  $\mathcal{G}_{\mathcal{S}} = (n, p, m, \Sigma, \mathbf{x}, \mathbf{v}, \{\mu_q\}_{q \in Q}, \tilde{\mathbf{y}}, C, D, \{A_q, K_q\}_{q \in Q})$  such that  $\mathbf{v}$  is white noise (zero mean, i.i.d.) and the  $\sigma$ -algebras generated by  $\{\mu_q(t)\}_{q \in Q, t \in \mathbb{Z}}$  and  $\{\mathbf{v}(t)\}_{t \in \mathbb{Z}}$  are independent and  $E[\mathbf{v}(t)\mathbf{v}^T(t)\mu_q^2(t)] = Q_q$ . It then follows  $\mathbf{v}$  is ZMWSSI and that  $\mathcal{G}_{\mathcal{S}}$  is well defined and the processes  $\mathbf{x}$  and  $\tilde{\mathbf{y}}$  are uniquely determined by the elements of the tuple  $\Sigma$  and the process  $\mathbf{v}$ , see [226, Lemma 3]. Moreover, certain

moments of the output  $\tilde{\mathbf{y}}$  of  $\mathcal{G}$  coincides with those of  $\mathbf{y}$ :  $\Lambda_w^{\tilde{\mathbf{y}}} = \Lambda_w^{\mathbf{y}}$  and  $T_{q,q}^{\tilde{\mathbf{y}}} = T_{q,q}^{\mathbf{y}}$ ,  $w \in Q^+$ ,  $q \in Q$ . That is, a weak realization of  $\mathbf{y}$  is a finite encoding of the infinite number of moments  $\{\Lambda_w^{\mathbf{y}}\}_{w \in Q^+}$ ,  $\{T_{q,q}^{\mathbf{y}}\}_{q \in Q}$  of  $\mathbf{y}$ .

Note that if  $\mathcal{S} = (\{A_q, K_q, Q_q\}_{q \in Q}, C, D)$  is a weak realization of  $\mathbf{y}$ , then it determines uniquely the covariances  $\{T_{q,q}^{\mathbf{y}}\}_{q \in Q}$  and the LSS  $\Sigma_{\mathcal{S}} = (n, \{\sqrt{\mathbf{p}(\mathbf{u}_q)}A_q, B_q, C_q = C\}_{q \in Q})$  where  $\{B_q\}_{q \in Q}$  is defined by (3.7), (3.8), and  $\Sigma_{\mathcal{S}}$  is a realization of  $f_{\mathbf{y}}$ . In turn,  $\Sigma_{\mathcal{S}}$  and  $\{T_{q,q}^{\mathbf{y}}\}_{q \in Q}$  can be used to calculate the GBS  $\mathcal{G}_{\Sigma_{\mathcal{S}}}$  associated with the LSS  $\Sigma_{\mathcal{S}}$ , as defined in (3.18), and  $\mathcal{G}_{\Sigma_{\mathcal{S}}}$  is a realization of  $\mathbf{y}$  in forward innovation form. That is, any weak realization of  $\mathbf{y}$  yields a GBS realization of  $\mathbf{y}$  in forward innovation form.

That is, any GBS realization of  $\mathbf{y}$  can be mapped to a weak realization of  $\mathbf{y}$ , and the latter is a finite encoding of the infinite number of moments  $\{\Lambda_w^{\mathbf{y}}\}_{w \in Q^+}$ ,  $\{T_{q,q}^{\mathbf{y}}\}_{q \in Q}$ . Conversely, any weak realization of  $\mathbf{y}$  can be mapped to a realization of  $\mathbf{y}$ .

In other words, stationary GBSs can be viewed both as generators of output processes but also as (partial) parameterizations probability distributions.

Next, we will argue that GBS realizations of  $\mathbf{y}$  can be viewed as predictors. To this end, let  $\mathcal{G}$  be a stationary GBS realization of  $\mathbf{y}$  in forward innovation form and assume that  $\mathcal{G}$  is of the form (3.1). First note that that  $\mathcal{G}$  can be interpreted as a linear filter which predicts the value  $\mathbf{y}(t)$  based on  $\{\mathbf{z}_w^{\mathbf{y}}(t)\}_{w \in Q^+}$ , i.e., based on products of past values of  $\{\mathbf{y}(s)\}_{s < t}$  with past values of the input process  $\{\boldsymbol{\mu}_q(s)\}_{q \in Q, \mathbb{Z} \ni s < t}$ . Indeed, we can formally rewrite the equations (3.1) with  $D = I$  and  $\mathbf{v} = \mathbf{e}$  as follows:  $\mathbf{e}(t) = \mathbf{y}(t) - C\mathbf{x}(t)$  and hence

$$\mathbf{x}(t+1) = \sum_{q \in Q} (A_q - K_q C)\mathbf{x}(t) + K_q \mathbf{y}(t) \boldsymbol{\mu}_q(t), \quad \hat{\mathbf{y}}(t) = C\mathbf{x}(t), \quad (3.36)$$

where  $\hat{\mathbf{y}}(t) = E_t[\mathbf{y}(t) | \{\mathbf{z}_w^{\mathbf{y}}(t)\}_{w \in Q^+}]$  is the orthogonal projection of  $\mathbf{y}(t)$  onto  $\{\mathbf{z}_w^{\mathbf{y}}(t)\}_{w \in (\mathcal{Q})^+}$ . Set  $\tilde{A}_q = A_q - K_q C$ . Note that if  $\sum_{q \in Q} \mathbf{p}(\boldsymbol{\mu}_q) \tilde{A}_q \otimes \tilde{A}_q$  is stable, then by repeating the steps of the proof of [226, Lemma1] it can be shown that

$$\mathbf{x}(t) = \sum_{N=0}^{\infty} \sum_{w \in Q^*, q \in Q, |w|=N} \sqrt{\mathbf{p}(\boldsymbol{\mu}_{qw})} \tilde{A}_w K_q \mathbf{z}_{qw}^{\mathbf{y}}(t), \quad \tilde{\mathbf{y}}(t) = \sum_{N=0}^{\infty} \sum_{w \in Q^*, q \in Q, |w|=N} \sqrt{\mathbf{p}(\boldsymbol{\mu}_{qw})} C \tilde{A}_w K_q \mathbf{z}_{qw}^{\mathbf{y}}(t), \quad (3.37)$$

and view  $\mathcal{G}$  as a predictor which maps past values  $\{\mathbf{y}(s), \boldsymbol{\mu}(s)\}_{s=-\infty}^{t-1}$  to the estimate of  $\hat{\mathbf{y}}(t)$  such that the prediction error variance  $E[\|\hat{\mathbf{y}}(t) - \mathbf{y}(t)\|^2]$  is minimal.

In fact, more is true, as  $\mathbf{z}_v^{\mathbf{y}^+}(t) = E_t[\mathbf{z}_v^{\mathbf{y}^+}(t) | \{\mathbf{z}_w^{\mathbf{y}}(t)\}_{w \in (\mathcal{Q})^+}]$  for any  $v \in Q^+$ , and hence  $\mathcal{G}$  can be viewed as the best linear predictor from the products of past values of  $\mathbf{y}$  and  $\{\boldsymbol{\mu}_q\}_{q \in Q}$  to the products of future values of  $\mathbf{y}$  and  $\{\boldsymbol{\mu}_q\}_{q \in Q}$ .

Similarly to the linear case,  $\mathcal{G}$  can also be interpreted as a filter using only finite past starting from  $t_f$  to predict the current value of  $\mathbf{y}$ , i.e., it predicts  $\mathbf{y}(t)$  based on products of past values of  $\{\mathbf{y}(s)\}_{s=t_f}^{t-1}$  with past values of the input signal  $\{\boldsymbol{\mu}(s)\}_{s=t_f}^{t-1}$ . More precisely, let us define  $\mathbf{x}_{\mathcal{G}}(t | t_f)$  and  $\mathbf{y}_{\mathcal{G}}(t | t_f)$  recursively as follows:  $\mathbf{x}_{\mathcal{G}}(t_f | t_f) = 0$  and for all  $k \geq t_f$

$$\mathbf{x}_{\mathcal{G}}(t+1 | t_f) = \sum_{q \in Q} (A_q - K_q C)\mathbf{x}_{\mathcal{G}}(t | t_f) + K_q \mathbf{y}(t) \boldsymbol{\mu}_q(t), \quad \mathbf{y}_{\mathcal{G}}(t | t_f) = C\mathbf{x}_{\mathcal{G}}(t | t_f). \quad (3.38)$$

If  $\sum_{q \in Q} \mathbf{p}(\boldsymbol{\mu}_q) (A_q - K_q C) \otimes (A_q - K_q C)$  is stable, then from (3.37) it follows that

$$\lim_{t-t_f \rightarrow \infty} \mathbf{x}_{\mathcal{G}}(t | t_f) - \mathbf{x}(t) = \lim_{t-t_f \rightarrow \infty} \sum_{w \in Q^*, q \in Q, |w| \geq t-t_f} C \tilde{A}_w K_q \mathbf{z}_{qw}^{\mathbf{y}}(t) = 0,$$

and hence

$$\begin{aligned} \lim_{t-t_f \rightarrow +\infty} \mathbf{y}_{\mathcal{G}}(t | t_f) - \hat{\mathbf{y}}(t) &= 0, \\ E[\|\mathbf{y}(t) - \mathbf{y}_{\mathcal{G}}(t | t_f)\|^2] &\geq E[\|\mathbf{y}(t) - \hat{\mathbf{y}}(t)\|^2], \quad \lim_{t-t_f \rightarrow +\infty} E[\|\mathbf{y}(t) - \mathbf{y}_{\mathcal{G}}(t | t_f)\|^2] = E[\|\mathbf{y}(t) - \hat{\mathbf{y}}(t)\|^2]. \end{aligned} \quad (3.39)$$

### 3.7. CONSEQUENCES FOR SYSTEM IDENTIFICATION: WEAK REALIZATIONS, GBSS AS OPTIMAL FILTERS, REALIZATION OF INPUT-OUTPUT RELATION VS. OUTPUT PROCESS

In particular, for any  $t_f$ , for large enough  $t$ ,  $\mathbf{y}_{\mathcal{G}}(t | t_f)$  can be made arbitrarily close to the best linear prediction of  $\mathbf{y}(t)$  based on  $\{\mathbf{z}_w^{\mathbf{y}}(t)\}_{w \in Q^+, |w| \leq t_f}$ . In other words, a GBS realization in forward innovation form is asymptotically the best linear predictor in  $\{\mathbf{z}_w^{\mathbf{y}}(t)\}_{w \in Q^+, |w| \leq t_f}$  for  $\mathbf{y}(t)$  as  $t - t_f \rightarrow +\infty$ .

#### 3.7.2 The stochastic realization problem versus realization of an input-output relation

Finally, we remark that the existence and uniqueness of GBS realizations in forward innovation form provides the following justification for the realization problem formulated in Problem 3.1, which in turn, will also serve as a justification for the identification problem for stochastic LPV.

Intuitively, we would like to be able to say that if  $\mathbf{y}$  is the output response of an unknown stationary GBS  $\mathcal{G}_0$  to the admissible input processes  $\{\mathbf{u}_q\}_{q \in Q}$ , then any minimal GBS realization  $\mathcal{G}$  of  $\mathbf{y}$  will have the property that  $\mathcal{G}_0$  and  $\mathcal{G}$  produce the same output response to *any* inputs, and not only to  $\{\mathbf{u}_q\}_{q \in Q}$ . In general, we cannot prove this. However, if we restrict attention to minimal realizations in forward innovation form, then this intuition holds true.

Note that restricting attention to minimal realizations in forward innovation form can be justified as follows. If the only thing we can observe of  $\mathcal{G}_0$  is its response  $\mathbf{y}$  to the admissible input process  $\{\mathbf{u}_q\}_{q \in Q}$ , then based on this data we cannot falsify the hypothesis that  $\mathcal{G}_0$  is a minimal GBS in forward innovation form. For even if it is not, it can always be brought to such a form without changing the observations. That is, based on our observations we have to remain agnostic about this assumption.

In order to show that two minimal GBS realizations of  $\mathbf{y}$  in forward innovation form will generate the same output for all choices of input processes, we have to define formally the input-output behavior of a GBS. The difficulty lies in the fact that the equations (3.2) are defined for negative time instances too and existence of a solution to (3.2) cannot be guaranteed for all input processes. In order to address this problem, we will associate with every GBS an LPV-SSA. This LPV-SSA corresponds to interpreting (3.2) for positive times. Then the state process  $\mathbf{x}(t)$  of the GBS corresponds to the solution of this LPV-SSA, obtained by feeding to the LPV-SSA the noise  $\mathbf{v}$  and interpreting  $\{\mu_q\}_{q \in Q}$  as scheduling signal, waiting  $t + t_f$  time instances and letting  $t_f \rightarrow +\infty$ . In other words, the state of GBS is the steady-state behavior of the corresponding LPV-SSA. Then the input-output map of this LPV-SSA can be viewed as a formalization of the input-output relation  $\{\mu_q\}_{q \in Q} \mapsto \tilde{\mathbf{y}}$  induced by the GBS. We then argue that the LPV-SSAs associated with any two minimal GBS realizations of  $\mathbf{y}$  in forward innovation form are input-output equivalent.

The details are as follows. Assume without loss of generality that  $Q = \{1, \dots, D\}$  and let  $\mathcal{G}$  be a stationary GBS of the form (3.2). Set  $\mathbb{P} = \mathbb{R}^D$ ,  $A_0 = 0$ ,  $K_0 = 0$ ,  $C_0 = C$ ,  $D_0 = I$ ,  $C_i = 0$ ,  $D_i = 0$ ,  $i \in Q$ , and consider the LPV-SSA  $\mathcal{L}_{\mathcal{G}} = (\mathbb{P}, \{A_i, K_i, C_i, D_i\}_{i=0}^D)$ . For a fixed  $t_f \in \mathbb{N}$  consider the stochastic processes  $\sigma^{t_f} \mathbf{v} : \mathbb{N} \ni t \mapsto \mathbf{v}(t - t_f)$ , and  $\sigma^{t_f} \boldsymbol{\mu} : \mathbb{N} \ni t \mapsto (\mu_1(t - t_f), \dots, \mu_D(t - t_f))$  defined on  $\mathbb{N}$ . Let  $\sigma^{t_f} \tilde{\mathbf{y}}$  be the output response of the LPV-SSA  $\mathcal{L}_{\mathcal{G}}$  to  $\sigma^{t_f} \mathbf{v}$  and  $\sigma^{t_f} \boldsymbol{\mu}$ , i.e.,  $\sigma^{t_f} \tilde{\mathbf{y}} = \mathfrak{Y}_{\mathcal{L}_{\mathcal{G}}}(\sigma^{t_f} \mathbf{v}, \sigma^{t_f} \boldsymbol{\mu})$ . From (3.25) it follows that  $\lim_{t_f \rightarrow +\infty} \sigma^{t_f} \tilde{\mathbf{y}}(t + t_f) = \mathbf{y}(t)$  in the mean square sense for all  $t$ . That is,  $\mathbf{y}(t)$  can be viewed as an approximation of the output of the LPV-SSA  $\mathcal{L}_{\mathcal{G}}$  obtained from the following experiment: we feed in the control input  $\sigma^{t_f} \mathbf{v}$  and the scheduling signal  $\sigma^{t_f} \boldsymbol{\mu}$ , we wait  $t_f$  time instants for the system to get close to a steady regime and then we measure the output at time  $t + t_f$ . In other words,  $\mathbf{y}(t)$  represents the steady state response of a LPV-SSA excited by a white noise control input and scheduling signal  $\boldsymbol{\mu}$ . If  $\mathcal{G}$  is in forward innovation form, and  $\tilde{\mathcal{G}}$  is any other minimal realization of  $\mathbf{y}$  in forward innovation form, then by Theorem 3.2,  $\tilde{\mathcal{G}}$  and  $\mathcal{G}$  are isomorphic, and hence so are the associated LPV-SSAs  $\mathcal{L}_{\tilde{\mathcal{G}}}$  and  $\mathcal{L}_{\mathcal{G}}$ . In particular,  $\mathcal{L}_{\tilde{\mathcal{G}}}$  and  $\mathcal{L}_{\mathcal{G}}$  are input-output equivalent.

This suggests that we can view outputs of a stationary GBS as the steady-state behavior of a LPV-SSA driven by a white noise input and a suitable scheduling signal. If the corresponding noise process is the innovation process of  $\mathbf{y}$  and the GBS is minimal, then the matrices of any minimal GBS realization of  $\mathbf{y}$  in forward innovation form will determine an LPV-SSA, such that this LPV-SSA has the same input-output map as the original one which was used to generate  $\mathbf{y}$ . That is, we use admissible stochastic input processes only to find an LPV-SSA which has the same input-output behavior as the original one. This will turn out to be important for system identification of LPV systems, since when they are used for control, the scheduling signal is deterministic. The



discussion above indicates that as long as the observed scheduling signal can be viewed as a sample path of an admissible input process in the sense defined in this chapter, we can use the corresponding output to identify the response of the underlying LPV system to *any scheduling signal, including those which are deterministic*. That is, stochasticity of the scheduling signal is not a restriction of the type of system we consider, but it is a persistence of excitation condition for the identification experiment.

The discussion above also brings forward an additional argument in favor of the hypothesis that the data generating system is a GBS in forward innovation form. Namely, we can formulate the following two technical results.

**Lemma 3.5** (Invertible GBSs are in forward innovation form). *Assume that  $\mathcal{G}$  is of the form (3.2),  $D = I$ ,  $\mathcal{G}$  is stationary and it is a realization of  $\mathbf{y}$ , and assume that the matrix  $\sum_{q \in Q} \mathbf{p}(\mu_q)(A_q - B_q C) \otimes (A_q - B_q C)$  is stable (all its eigenvalues are inside the complex unit disk). Then  $\mathbf{v}$  is the innovation process of  $\mathbf{y}$  w.r.t.  $\{\mu_q\}_{q \in Q}$  and  $\mathcal{G}$  is a realization of  $\mathbf{y}$  in forward innovation form.*

*Proof of Lemma 3.5.* Note that we can write  $\mathbf{v}(t) = \mathbf{y}(t) - C\mathbf{x}(t)$  and hence the first equation of (3.36) holds, i.e.,  $\mathbf{x}(t+1) = \sum_{q \in Q} (A_q - K_q C)\mathbf{x}(t) + K_q \mathbf{y}(t) \mu_q(t)$ . Since  $\sum_{q \in Q} \mathbf{p}(\mu_q) \tilde{A}_q \otimes \tilde{A}_q$ ,  $\tilde{A}_q = A_q - K_q C$ , is stable, then by repeating the steps of the proof of [226, Lemma 1] it can be shown that  $\mathbf{x}(t) = \sum_{N=0}^{\infty} \sum_{w \in Q^*, q \in Q, |w|=N} \sqrt{\mathbf{p}(\mu_{qw})} \tilde{A}_w K_q \mathbf{z}_{qw}^{\mathbf{y}}(t)$ , and hence the elements of  $\mathbf{x}(t)$  belong to the Hilbert-space generated by  $\{\mathbf{z}_w^{\mathbf{y}}(t)\}_{w \in Q^+}$ . Note that  $E[\mathbf{v}(t) | \{\mathbf{z}_w^{\mathbf{y}}(t)\}_{w \in Q^+}] = 0$ , see the proof of [226, eq. (37), proof of Theorem 4], hence,  $E[\mathbf{y}(t) | \{\mathbf{z}_w^{\mathbf{y}}(t)\}_{w \in Q^+}] = C\mathbf{x}(t)$  and therefore  $\mathbf{e}(t) = \mathbf{v}(t)$ .  $\square$

**Lemma 3.6** (Minimality conditions for GBS in innovation form). *Assume that  $\mathcal{G}$  is of the form (3.2) is stationary realization of  $\mathbf{y}$  in forward innovation form, and that  $\mathbf{y}$  is full rank. Then  $\mathcal{G}$  is a minimal realization of  $\mathbf{y}$ , if and only if the associated LPV-SSA  $\mathcal{L}_g = (\mathbb{R}^{D+1}, \{A_i, K_i, C_i\}_{i=0}^D)$ ,  $A_0 = 0, K_0 = 0, C_0 = C, D_0 = I, C_i = 0, D_i = 0, i = 1, \dots, D$  is minimal.*

*Proof of Lemma 3.6.* Note that  $\mathcal{G}$  is minimal if and only if the observability and reachability matrices satisfy the following rank conditions  $\text{rank } \mathcal{O}_{n-1}(\mathcal{G}) = n$  and  $\text{rank } \mathcal{R}_{n-1}(\mathcal{G}) = n$ . Note that the rows of the extended observability matrix  $\mathcal{O}_{n-1}$  of the associated LPV-SSA  $\mathcal{L}_g$  are either zero or they coincide with the rows of the observability matrix  $\mathcal{O}_{n-1}(\mathcal{G})$ , i.e.,  $\text{rank } \mathcal{O}_{n-1}(\mathcal{G}) = \text{rank } \mathcal{O}_{n-1}$ . That is, by Theorem 2.3,  $\mathcal{G}$  satisfies the observability rank condition if and only if the LPV-SSA  $\mathcal{L}_g$  is observable. We will show that  $\text{Im } \mathcal{R}_{n-1}(\mathcal{G}) = \text{Im } \mathcal{R}_{n-1}$ , where  $\mathcal{R}_{n-1}$  is the extended controllability matrix of the LPV-SSA  $\mathcal{L}_g$ . From this, using Theorem 2.3 it follows that  $\mathcal{G}$  satisfies the reachability rank condition if and only if  $\mathcal{L}_g$  is span-reachable. Then the statement of the lemma follows from Theorem 3.2 and Theorem 2.2.

We show that  $\text{Im } \mathcal{R}_{n-1}(\mathcal{G}) = \text{Im } \mathcal{R}_{n-1}$ . To this end, from (3.25) it follows that  $\mathbf{x}(t)$  belongs to the linear space generated by the columns of  $A_w K_q$ ,  $w \in Q^*, q \in Q$ . Since  $B_q = E[\mathbf{x}(t)(\mathbf{z}_q^{\mathbf{y}}(t))^T]$ , it then follows that the columns of  $B_q$  also belong to the linear space generated by the columns of  $A_w K_q$ ,  $w \in Q^*, q \in Q$ . Therefore, the columns of  $A_v B_q$ ,  $v \in Q^*, q \in Q$  also belong to the linear space generated by the columns of  $A_w K_q$ ,  $w \in Q^*, q \in Q$ . In turn, using the correspondence between LPV-SSAs and LSSs described in Chapter 2 and [205, Theorem 2 and Remark 1], it is easy to see that latter subspace equals  $\text{Im } \mathcal{R}_{n-1}$ . That is,  $\text{Im } A_v B_q$  is a subspace of  $\text{Im } \mathcal{R}_{n-1}$ , and therefore  $\text{Im } \mathcal{R}_{n-1}(\mathcal{G}) \subseteq \text{Im } \mathcal{R}_{n-1}$ . Conversely, from [226, eq. (37), proof of Theorem 4] it follows that  $E[\mathbf{x}(t)\mathbf{z}_{qv}^{\mathbf{y}}(t)] = \sqrt{\mathbf{p}(\mu_w)} A_v B_q$ , i.e., for every  $w \in Q^+$ , the columns  $E[\mathbf{x}(t)(\mathbf{z}_w^{\mathbf{y}}(t))^T]$  belong to the space generated by  $A_v B_q$ ,  $q \in Q, v \in Q^*$ . Notice that by [205, Theorem 2 and Remark 1] applied to the LSS  $\Sigma_g$ , the latter space equals  $\text{Im } \mathcal{R}_{n-1}(\mathcal{G})$ . Since the elements of  $\mathbf{z}_q^{\mathbf{e}}(t)$  are limits of finite linear combinations of the rows of  $\{\mathbf{z}_w^{\mathbf{y}}(t)\}_{w \in Q^+}$ , it then follows that the columns of  $E[\mathbf{x}(t)(\mathbf{z}_q^{\mathbf{e}}(t))^T]$  are the limits of finite linear combinations of columns of  $E[\mathbf{x}(t)(\mathbf{z}_w^{\mathbf{y}}(t))^T]$ ,  $w \in Q^+$ , and hence the columns of  $E[\mathbf{x}(t)(\mathbf{z}_w^{\mathbf{y}}(t))^T]$ ,  $w \in Q^+$  also belong to  $\text{Im } \mathcal{R}_{n-1}(\mathcal{G})$ . From [226, Proof of Theorem 4] it follows that  $K_q Q_q = E[\mathbf{x}(t)(\mathbf{z}_q^{\mathbf{e}}(t))^T]$ , where  $Q_q = E[\mathbf{e}(t)\mathbf{e}^T(t)\mu_q^2(t)]$ , and hence the columns of  $K_q Q_q$  belong to  $\text{Im } \mathcal{R}_{n-1}(\mathcal{G})$ . Since  $Q_q$  is non-singular, it then follows that the columns of  $K_q$  belong to  $\text{Im } \mathcal{R}_{n-1}(\mathcal{G})$ . Since  $\text{Im } \mathcal{R}_{n-1}(\mathcal{G})$  is  $A_q$ -invariant for all  $q \in Q$  and  $A_0 = 0$ , it then follows that  $\text{Im } A_v K_q \subseteq \text{Im } \mathcal{R}_{n-1}$  for all  $v \in \mathbb{I}_{0, n_p}^*$ ,  $q \in \mathbb{I}_{0, n_p}$ , and thus  $\text{Im } \mathcal{R}_{n-1} \subseteq \text{Im } \mathcal{R}_{n-1}(\mathcal{G})$ .  $\square$

### 3.8. CONCLUSIONS AND FURTHER WORK

---

The results of Lemma 3.5 and Lemma 3.6 suggests the following sufficient conditions that the system generating  $\mathbf{y}$  is a minimal one in forward innovation form. Assume that the observed output  $\mathbf{y}$  is generated by a minimal LPV-SSA  $\Sigma = (\mathbb{R}^{D+1}, \{A_i, K_i, C_i, D_i\}_{i=0}^D)$  driven by white noise  $\mathbf{v}$ , such that  $C_i = 0, D_i = 0, i = 1, \dots, D, D_0 = I, A_0 = 0, K_0 = 0$  and  $\sum_{q=1}^D \mathbf{p}(\mu_q)(A_q - K_q C_0) \otimes (A_q - K_q C_0)$  and  $\sum_{q \in Q} \mathbf{p}(\mu_q) A_q \otimes A_q$  are stable. It then follows that the GBS  $\mathcal{G} = (n, p, p, \mathbf{x}, \mathbf{v}, \mathbf{y}, \{A_i, K_i\}_{i=1}^D, C, I)$  is a minimal GBS realization of  $\mathbf{y}$  in forward innovation form, and the LPV-SSA associated with any minimal GBS realization of  $\mathbf{y}$  in forward innovation form is isomorphic with  $\Sigma$ . In particular, any minimal GBS realization of  $\mathbf{y}$  in forward innovation form has the same input-output behavior as the system which generated  $\mathbf{y}$ . Note that the assumptions on the LPV-SSA system which generates  $\mathbf{y}$  are very mild. Indeed, we assume minimality, some stability conditions on the matrices, and a few other conditions on the matrices, i.e., the assumptions are purely assumptions on the matrices. That is, the assumption that the observations are generated by minimal GBSs in forward innovation form are reasonable not only because it cannot be falsified based on observations, but also because it can be replaced by a sufficient condition which represents some mild assumptions on the matrices of the underlying system. In fact, these assumptions are mainly stability assumptions, as explained in the remark below.

**Remark 3.5.** *In fact, we could drop the assumption that  $\Sigma$  is minimal: if it is not, we can replace it by a minimal LPV-SSA  $\Sigma_m = (\mathbb{R}^{D+1}, \{A_i^m, K_i^m, C_i^m, D_i^m\}_{i=0}^D)$  which is input-output equivalent. We can easily adapt the proof of [226, Theorem 6] to show that if  $C_i = 0, D_i = 0, i = 1, \dots, D, D_0 = I, A_0 = 0, K_0 = 0$  and  $\sum_{q=1}^D \mathbf{p}(\mu_q)(A_q - K_q C) \otimes (A_q - K_q C)$  and  $\sum_{q \in Q} \mathbf{p}(\mu_q) A_q \otimes A_q$  are stable, then  $C_i^m = 0, A_i^m = 0, K_i^m = 0, D_i^m = 0, i = 1, \dots, D, D_0^m = I$  and  $\sum_{q=1}^D \mathbf{p}(\mu_q)(A_q^m - K_q^m C_0^m) \otimes (A_q^m - K_q^m C_0^m)$  and  $\sum_{q \in Q} \mathbf{p}(\mu_q) A_q^m \otimes A_q^m$  are stable.*

## 3.8 Conclusions and further work

I have presented selected results from my work on realization theory of a class of stochastic bilinear systems, which includes autonomous stochastic LPV and jump-Markov systems. These results are based on applying realization theory of deterministic linear switched systems. In [172] the results of this chapter were used to propose a statistically consistent system identification algorithm. In [173] these results were extended to classes of stochastic LPV systems with control input, and a statistically consistent system identification algorithm was proposed. The application of the results of this chapter to jump-Markov systems with control inputs remains an open problem. We conjecture that the results of this chapter can also be extended to continuous-time case, but that extension remains a topic of future research.



## **Part II**

# **Selected topics on application of realization theory**



---

## Introduction

In the second part of the manuscript I present some of my contributions to applications of realization theory of linear switched, LPV and stochastic bilinear systems. More precisely, I will discuss my work on model reduction of linear switched systems (Chapter 4), on system identification of linear parameter-varying systems (Chapter 5), and on reverse engineering of network structures (Chapter 6). Below I will briefly describe the contents of each Chapters 4 – 6 and the role of realization theory in each of them.

### Chapter 4: model reduction

In Chapter 4, I present an overview of my work on model reduction of linear switched systems.

**Novelty** At the moment when I started working on model reduction of switched systems there were few results on this topic, see for example [94, 312, 313, 316, 44, 45, 46, 198, 179]. Most of the papers proposed various modifications of balanced truncation algorithms. However, the available results did not provide any error bounds on the difference between the input-output behavior of the original and the reduced models. Moreover, there were no algorithms based on moment matching using Markov parameters or multivariate Laplace transforms. My work on model reduction was aimed at filling these gaps.

**Contents of the chapter** The methods presented in Chapter 4 can be divided into the following categories:

- Moment matching based on Markov parameters.
- Balanced truncation.
- Moment matching in frequency domain, based on multivariate Laplace transform.

The results on moment matching based on Markov-parameters described in Chapter 4 originate from the PhD thesis of Mert Bastug [24], whom I co-advised together with John Leth and Rafael Wisniewski. The other results were derived in collaboration with John Leth, Rafael Wisniewski, Ion Victor Gosea, Athanasios Antoulas, Christophe Fiter.

**Role of realization theory** The results of Chapter 4 rely heavily on realization theory. Indeed, the methods based on matching Markov parameter rely on the notion of Markov parameters for linear switched systems and the latter notion originates from realization theory. Moreover, the model reduction algorithm itself is closely related to partial realization algorithm. The methods based on balanced truncation use the relationship between observability/reachability and minimality, and they can be viewed as a numerical implementation of the minimization algorithm. In turn, characterization of minimality is an integral part of realization theory. Furthermore, realization theory is necessary to show that the notion of singular values of grammians is independent of the choice of state-space representation. In turn, singular values of grammians play a central role in analytic error bounds for the approximation error in balanced truncation. Hence, in order to show that the performance of balanced truncation does not depend on the choice of the state-space representation, i.e., balanced truncation gives the same result for any state-space representation of the given input-output behavior, realization theory is necessary.

### Chapter 5: system identification

In Chapter 5 I present some selected results on system identification of LPV systems which are based on my work on realization theory.

---

**Novelty** At the moment when I started working on system identification of LPV systems, it was already a mature topic with a rich literature, see [18, 155, 175, 231, 88, 288, 268, 284, 289, 49, 99, 150, 273] and the references therein. In particular, there is a significant literature on subspace identification of LPV systems, see [284, 88, 288, 289, 67] and the references therein. However, when I started research on LPV system identification, the existing literature did not address the problem of identifiability of LPV state-space representations, nor there were results on statistical consistency of subspace identification algorithms. In fact, for the case of LTI systems both topics required results from realization theory. Moreover, subspace identification methods suffered from curse of dimensionality and required strong assumptions on observability to work. My research in this topics was aimed at filling these gaps.

**Contents of the chapter** In Chapter 5 I will discuss

- identifiability analysis of deterministic LPV systems
- subspace identification of deterministic and stochastic LPV systems using reduced basis of the Hankel-matrix.

The result on identifiability of LPV systems originate from the PhD thesis of Ziad Alkhoury, whom I co-advised with Guillaume Mercère. By identifiability of a parameterizations we mean that there exist no two parameter values for which the corresponding models have the same input-output behavior. In Chapter 5 we present constructive conditions for identifiability of LPV state-space parameterizations.

The results on subspace identification algorithm were derived in collaboration with Pepijn Cox, Roland Tóth and Manas Mehari. For deterministic LPV systems one uses modifications of the realization algorithm from Chapter 2. For purely stochastic LPV systems without inputs one uses the realization algorithm from Chapter 3 applied to empirical covariances. Finally, stochastic LPV systems with inputs can be decomposed into a stochastic and a deterministic part. The stochastic part does not depend on the control input and its dynamics is driven by a stochastic noise process. The deterministic part is driven by the control input and has no additive noise terms. This decomposition also translates to the output of the system, as the latter is the sum of a stochastic process which does not depend on the control input and a process which depends only on the control input. The former is the output of the stochastic subsystem, the latter is the output of the deterministic one. As a result, in order to identify a stochastic LPV state-space representation it is sufficient to identify an autonomous stochastic LPV state-space representation of the stochastic component of the output, and a deterministic LPV state-space representation of the deterministic component of the output. We then propose subspace identification algorithms for identifying both components. The thus derived system identification algorithm can be shown to be statistically consistent.

**Role of realization theory** Identifiability of LPV systems relies on realization theory of LPV systems presented in Chapter 2. More precisely, the characterization of identifiable parameterizations is based on characterizing parameterizations of minimal LPV state-space representations and formulating conditions for excluding isomorphic copies of the same state-space representation within the parameterization. From realization theory of LPV systems presented in Chapter 2 it follows that two minimal state-space representations have the same input-output behavior if and only if they are related by an isomorphism which does not depend on the state. As it was noted above, subspace identification for LPV systems relies on the realization algorithm for deterministic LPV state-space representations from Chapter 2 and on stochastic realization theory of generalized bilinear systems presented in Chapter 3. Moreover, the results of Chapter 3 lead to a proof of existence of a state-space representation in forward innovation form. The latter is often taken as a starting point for subspace identification of LPV systems, but its existence is usually not proven formally.

## Chapter 6: Granger-causality for network identification

In Chapter 6 the application of stochastic realization theory to reverse engineering of network structure of dynamical systems is discussed.

---

**Novelty** The problem of network identification, i.e. identifying a dynamical system with a (partially) known network graph is a topic which has gained increased popularity, without claiming completeness we mention [297, 68, 298, 280, 191, 138, 146, 278, 104, 123, 310]. Some of the paper assume that the underlying system consists of several subsystems, each with a separate transfer function, the transfer functions can exchange measurable signals, the presence/absence of such signals is determined by the network graph of the system, and the goal is to identify the transfer function of each node [297, 68, 298, 280, 50]. The latter problem is well-posed and practically relevant. However, in many applications the goal is to identify the network graph of the underlying state-space representation [191, 138, 146, 278].

In contrast to the problem of identifying transfer functions at a node of a network, the problem of identifying the network graph of state-space representations is ill-posed in general, as state-space representations with different network graphs could have the same observed behavior. Motivated by this observation, several author tried to define the network graph of a system independently of the underlying state-space representation [104, 123, 310], but the resulting definitions did not solve the problem of finding the network graph of the underlying state-space representation.

Another approach is to abandon the idea of finding the network graph of the state-space representation, and focus instead on the problem of deciding if a state-space representation with a certain network graph could exist for the observed behavior. In other words, instead of trying to estimate the network graph from the observed data, we try to decide if a certain network graph is consistent with the data. This is useful in systems biology and neuroscience, as it allows to estimate from the observed data the likelihood of the presence/absence of an interaction between subsystems. In turn, this likelihood can be used to guide experiment design. More precisely, since biology is an empirical science, in order to prove the presence or absence of an interaction, experiments have to be carried out anyway. It is then reasonable to focus on experiments which can prove/refute the presence of most likely interactions.

However, at the moment when I started working on this topic, there was no formal correspondence between the observed behavior and the existence of a state-space representation with a certain network graph. My goal was to close this gap. This was done by establishing a formal correspondence between Granger-causality of observed processes and the existence of stochastic linear and bilinear state-space representation with a given network graph. There is a rich literature on characterizing Granger-causality in terms of transfer functions [110, 54, 53, 98, 21, 255, 74] or VAR models [77, 78, 79]. However, the problem of relating Granger-causality with stochastic linear and bilinear state-space representations has not been addressed in the literature before. The results discussed in Chapter 6 fill this gap.

**Contents of the chapter** In Chapter 6 two types of systems were considered: stochastic linear time-invariant state-space representations (sLTI for short), and generalized stochastic bilinear systems. In both cases it was shown that existence of Granger-causality relations between components is equivalent to existence of a state-space representation with a certain network graph. By network graph we mean the directed graph, nodes of which are subsystems, and there is an edge from one node to another, if the state of the sub-system corresponding to the source nodes is present in the update equation of the subsystem which corresponds to the second node. For sLTI systems the classical definition of Granger-causality was used, for stochastic bilinear systems a new definition was proposed. Note that the classical definition of Granger-causality is equivalent to absence of feedback, and it plays an important role in proving consistency of open-loop identification algorithms. The same is expected for Granger-causality for stochastic bilinear systems. The results of Chapter 6 originate from the PhD thesis [133] of Mónika Józsa, whom I co-advised with Kanat M. Camlibel.

**Role of realization theory** The results of Chapter 6 rely on classical stochastic realization theory for linear systems and on the results of Chapter 3 on stochastic bilinear systems.



---

## Chapter 4

# Model reduction of linear switched systems

### 4.1 Introduction

The goal of this chapter is to give an overview of my work on application of realization theory to model reduction of *linear switched systems (LSS for short)*. Linear switched systems represent the simplest subclass of hybrid systems. Recall that a hybrid system is a finite collection of continuous-state dynamical systems, indexed by a set of so called *discrete modes (or states)*. The state of each dynamical system is governed by a set of differential or difference equations. The operating discrete mode in any time instant can be chosen arbitrarily, or it may depend on the value of the continuous state or other constraints, which are referred to as *guards*. The transitions between the discrete states may result in a jump in the state of the underlying continuous dynamical system. This jump is defined by the application of the so called *reset maps*. LSSs [161, 260] are the simplest and most widely studied subclass of hybrid systems, for which the continuous subsystems are linear systems, and the change of the discrete state is externally generated.

Model reduction is the field of replacing complex model by simpler ones such that the resulting simpler model can be used for control engineering. As a rule, the simpler model is a dynamical system with less state variables (of smaller order) than the original model, and it is referred to as the reduced model. Model reduction is a mature field with an extensive literature [11]. Most of the literature on model reduction deals with linear systems. For linear systems, model reduction techniques can be grouped in two large categories: model reduction with balanced truncation and model reduction based on moment matching [11].

Roughly speaking, methods based on balanced truncation remove states which are difficult to observe or to control. As a rule, balanced truncation methods for linear system come with analytic bounds on the approximation error of the original model by the reduced order one. Usually, one uses the standard  $H_\infty, H_2$  norms to measure the difference of output responses.

Moment matching methods rely on removing the states which do not contribute to the filtered output response for certain inputs. For example, one may wish to restrict attention to output responses for inputs with certain frequency, or components of the outputs lying in a certain frequency range, or the first couple of time instants of the impulse response. In contrast to balanced truncation, for moment matching methods there are no analytical error bounds for the difference of output responses of the reduced and the original model. Intuitively, the reasons for this is quite clear: in balanced truncation we remove states which do not contribute to the input-output behavior either because they have little influence on the outputs or because they are not sensitive to control inputs, hence we can claim that for any input, the outputs of the original and reduced model should be similar. In moment matching we want to make sure that the (filtered) response of the reduced model to certain inputs is similar to that of the original model, we do not aim at making the (unfiltered) response of the reduced model close to that of the

original one for all inputs.

These methods can be extended LSSs. When I started working on this topic there was already a fair amount of literature on model reduction of LSSs. The model reduction methods for LSSs available at that time could be grouped into the following categories.

**LMI-based methods.** These methods compute the matrices of the reduced order model by solving a set of LMI. The advantage of this approach that error bounds are available. The disadvantage is that the proposed conditions are only sufficient, and the trade-off between the dimension of the reduced model and the error bound is not clear. Moreover, the computational complexity of solving those LMIs might be too high. Without claiming completeness, we mention the following papers [94, 312, 313, 316].

**Methods based on local grammians.** These algorithms are based on finding observability/controllability grammians for each linear subsystem. For these methods often there are no error bounds and the reduced order model need not be well-posed. Examples of such papers include [179, 198, 44, 45, 46, 109]. When I started working on model reduction of LSSs, there were no analytic error bounds available for these type of methods when applied to deterministic LSSs.

**Methods based on common grammians.** These methods rely on finding the same observability/controllability grammian for each linear subsystem. These grammians are derived as solutions of a suitable LMI. Such algorithms were described in [251, 252, 234]. These algorithms apply only to LSSs which have a global quadratic Lyapunov function. At the moment when I started working on model reduction, there were no analytic error bounds available for these methods.

Finally, it is worth mentioning that model reduction is related to the notion of approximate bisimulation introduced in [100, 233, 267, 266]. While the general goal of model reduction and of finding approximate simulation relations is the same (both aim at replacing a complex model by a simpler one), the details are very different. In particular, in model reduction, the aim is to replace a model with a model of the same type but with less states. For example, a switched system is replaced by another switched system with a smaller number of states. In contrast, the goal in using approximate simulation relations is to replace a hybrid/nonlinear system by a finite-state transition system, which is approximately similar to the original system.

In my research on model reduction I focused on the following topics:

**Analytic error bounds and new algorithms for balanced truncation** In [227] error bounds were provided for balanced truncation based on common grammians for LSSs. In [106, 108] balanced truncation methods based on local grammians were proposed together with analytic error bounds for LSSs with linear reset maps. In [212] this was extended to piecewise-linear systems with autonomous switching.

**Moment matching.** The idea behind these algorithms is to find a reduced order linear switched system such that certain coefficients of the series expansions of the input-output maps of the original and the reduced order system coincide. The series expansion can be the Taylor series with respect to switching times, in which case the so-called Markov parameters are matched. Alternatively, the series expansion can be a Laurent-series expansion of a multivariate Laplace transform of the input-output map around a certain frequency. The former approach was pursued in [31, 30, 24], the latter in [105]. While those methods do not allow for analytical error bounds, under suitable assumption it can be guaranteed that the reduced model will have the same input-output behavior for certain switching signals [31, 30, 24].

My main contribution to model reduction was the development of analytic error bounds for balanced truncation of LSSs and the development of moment matching algorithms for LSSs.

In this chapter I will discuss selected topics on model reduction I worked on. For the sake of simplicity, I will discuss in detail only LSSs in continuous time and without reset maps. Most of the presented results are true for the discrete-time case too, and some can be extended to include LSSs with reset maps which are not identity. I will cite the relevant literature on these extensions.

Realization theory is relevant for model reduction in many ways. First, minimization and realization algorithms can be viewed as simple model reduction algorithms. Moreover, the relationship between span-reachability, observability and minimality is closely related to the existence of grammians which are used in balanced truncation. Realization theory is even more critical for moment matching, as the latter can be viewed as partial

realization algorithm. We will elaborate on the relationship between realization theory and model reduction when presenting the particular methods.

This chapter is based on [230].

In Section 4.3 we discuss the basic principle of model reduction. In Section 4.4 we present balanced truncation and in Section 4.5 we discuss moment matching for LSSs.

Throughout the chapter we will use notation and terminology of [203, 219, 209, 202] for LSSs., see Appendix B for an overview. For the convenience of the reader, in order to keep the chapter self-contained, in Section 4.2 we briefly recall the basic terminology and notation for LSSs.

## 4.2 Definition and notation

We follow the presentation of [203, 227] and Appendix B. A linear switched system (LSS) is a control system of the form

$$\Sigma \begin{cases} (\xi x)(t) = A_{\sigma(t)}x(t) + B_{\sigma(t)}u(t) \\ y(t) = C_{\sigma(t)}x(t) \end{cases} \quad (4.1)$$

where  $x(t) \in \mathbb{R}^n$  is the state at time  $t$ ,  $\sigma(t) \in \mathcal{Q}$  is the discrete mode at time  $t$ ,  $y(t) \in \mathbb{R}^p$  is the output at time  $t$ , and  $u(t) \in \mathbb{R}^m$  is the continuous-valued input at time  $t$ , and  $(\xi x)(t) = \frac{d}{dt}x(t)$  in the continuous-time case, and  $(\xi x)(t) = x(t+1)$  in the discrete-time. The set  $\mathcal{Q}$  is a finite one, and it is referred to as the set of discrete modes. Moreover,  $A_q \in \mathbb{R}^{n \times n}$ ,  $B_q \in \mathbb{R}^{n \times m}$ ,  $C_q \in \mathbb{R}^{p \times n}$  are the matrices of the linear system in the discrete state  $q \in \mathcal{Q}$ . The following notation

$$\Sigma = (n, \{(A_q, B_q, C_q) \mid q \in \mathcal{Q}\})$$

is used as a short-hand representation for LSSs of the form (4.1). The number  $n$  is called the *dimension (order)* of  $\Sigma$  and will be denoted by  $\dim \Sigma$ .

The solution of a LSS is defined as follows. Recall the notation of Section 2.2 and let us introduce the following notation. We write  $\mathcal{U}$ ,  $\mathcal{Q}$ ,  $\mathcal{X}$  and  $\mathcal{Y}$  to denote either  $PC(\mathbb{R}_+, \mathbb{R}^m)$ ,  $PC(\mathbb{R}_+, \mathcal{Q})$ ,  $AC(\mathbb{R}_+, \mathbb{R}^n)$  and  $PC(\mathbb{R}_+, \mathbb{R}^p)$ . Then a solution of (4.1) is a tuple  $(x, u, \sigma, y) \in \mathcal{X} \times \mathcal{U} \times \mathcal{Q} \times \mathcal{Y}$ , such that  $(x, y, \sigma, y)$  satisfy (4.1). Note that for every initial state  $x_0 \in \mathbb{R}^n$ , input  $u \in \mathcal{U}$  and switching signal  $\sigma \in \mathcal{Q}$  there is a unique state trajectory  $x$  and output trajectory  $y$ , such that  $(x, u, \sigma, y)$  is a solution. We can define the *input-output map* of an LSS induced by the initial state  $x_0$  as the map  $Y_{\Sigma, x_0} : \mathcal{U} \times \mathcal{Q} \rightarrow \mathcal{Y}$  such that  $Y_{\Sigma, x_0}(u, \sigma) = y$ , where  $(x, u, \sigma, y)$  is the unique solution of the LSS (4.1) such that  $x(0) = x_0$ . We formalize the input-output behavior of  $\Sigma$  as the input-output map of  $\Sigma$  induced by the initial state  $x_0 = 0$ . This is done for the sake of simplicity, see Appendix B.2 for more comments on the subject. We call the input-output map  $Y_{\Sigma, 0}$  induced by the initial state 0 of  $\Sigma$  the *input-output map* of  $\Sigma$ , and we denote  $Y_{\Sigma, 0}$  by  $Y_{\Sigma}$ . That is, a potential input-output map of a LSS is a function of the form

$$f : \mathcal{U} \times \mathcal{Q} \rightarrow \mathcal{Y}. \quad (4.2)$$

The LSS  $\Sigma$  is a *realization of an input-output map  $f$  of the form (4.2)*, if  $Y_{\Sigma} = f$ , i.e. if the input-output map of  $\Sigma$  coincides with  $f$ . If  $\Sigma$  is a realization of  $f$ , then  $\Sigma$  is a *minimal realization of  $f$* , if for any LSS realization  $\tilde{\Sigma}$  of  $f$ ,  $\dim \Sigma \leq \dim \tilde{\Sigma}$ . Two LSSs  $\Sigma_1, \Sigma_2$  are said to be *input-output equivalent*, if their input-output maps are equal, i.e.  $Y_{\Sigma_1} = Y_{\Sigma_2}$ . A LSS  $\Sigma$  is said to be *minimal*, if it is a minimal realization of its own input-output map  $f = Y_{\Sigma}$ .

## 4.3 Model reduction

In model reduction, we would like to find LSSs of *smaller* dimension, input-output maps of which are *close* (but not necessarily equal) to that of the original LSS. This is in contrast to realization theory, where we were interested in finding a minimal LSS with *exactly* the same input-output map as the original one. The latter is a special case of the former. Model reduction algorithms follow the following general pattern.

**Algorithm 5** Mode reduction/minimization algorithm

**Inputs:**  $\Sigma = (n, \{A_q, B_q, C_q\}_{q \in Q})$ , matrices  $V \in \mathbb{R}^{n \times r_1}$ ,  $W \in \mathbb{R}^{r_2 \times n}$ .

- 1: Let  $r = \text{rank } WV$  and let  $S \in \mathbb{R}^{r \times r_2}$ ,  $T \in \mathbb{R}^{r_1 \times r}$ ,  $SWVT = I_r$ .
- 2:  $\bar{A}_q = SWA_qVT$ ,  $\bar{C}_q = C_qVT$ ,  $\bar{B}_q = SWB_q$ .
- 3: **return**  $\bar{\Sigma} = (r, \{\bar{A}_q, \bar{B}_q, \bar{C}_q\}_{q \in Q})$ .

Intuitively, Algorithm 5 restricts the system to the set  $\text{Im}V$  and then merges those of its states  $x_1, x_2$  for which  $x_1 - x_2 \in \ker W$ .

Algorithm 5 can be viewed as a generalization of minimization algorithm [205, Procedure 3], [209, Procedure 3], Procedure B.3 in Appendix B. More precisely, recall from [205, Definition 21], Definition B.1, Appendix B the definition of the subspaces  $\mathcal{W}^*$ ,  $\mathcal{V}^*$  for the LSS  $\Sigma$ . Intuitively,  $\mathcal{V}^*$  is the linear subspace generated by the states of  $\Sigma$  which are reachable from the zero initial state, i.e.,  $\mathcal{V}^*$  are generated by states of the form  $x(t)$ , where  $(x, u, \sigma, y)$  is a solution of  $\Sigma$ ,  $x(0) = 0$ , and  $t$  is a time instance. Similarly,  $\mathcal{W}^*$  is the subspace of all unobservable states of  $\Sigma$ , i.e., a state  $x_0$  belongs to  $\mathcal{W}^*$ , if and only if the input-output map  $Y_{\Sigma, x_0}$  generated by  $x_0$  is zero.

If  $\ker W = \mathcal{W}^*$  and  $\text{Im}V = \mathcal{V}^*$ , with  $\mathcal{W}^*$ ,  $\mathcal{V}^*$  from Definition B.1, then the LSS  $\bar{\Sigma}$  returned by Algorithm 5 is a minimal LSS which is input-output equivalent to  $\Sigma$ , i.e., Algorithm 5 is just an implementation of Procedure B.3. Algorithms for computing such matrices  $W, V$  such that  $\ker W = \mathcal{W}^*$  and  $\text{Im}V = \mathcal{V}^*$  are described in [205, Remark 1].

In case of model reduction, Algorithm 5 can again be used. However, instead of applying it with matrices  $W$  and  $V$  such that  $\ker W = \mathcal{W}^*$  and  $\text{Im}V = \mathcal{V}^*$ , we use matrices  $W, V$  such that  $\ker W \subseteq \mathcal{W}^*$  and  $\text{Im}V \subseteq \mathcal{V}^*$ , i.e., we restrict the system to a subset of the set of reachable states, or we merge states which do not produce the same input-output behavior. The resulting LSS model will no longer be a realization of  $f$ , but its input-output map will approximate  $f$  in a suitable sense. Depending on the method we use, we will either be able to provide a global error bound on the difference between the input-output maps of the original model and the reduced one, or state that for certain switching sequences the two input-output maps coincide. We will elaborate on various methods below.

## 4.4 Model reduction by balanced truncation

Let  $\Sigma$  be a LSS of the form (2.6), and assume that  $\Sigma$  is *quadratically stable*, i.e., there exists a matrix  $P > 0$  such that

$$\forall q \in Q: A_q^T P + P A_q < 0.$$

In this case  $\Sigma$  is globally uniformly asymptotically (exponentially) stable [161] with the Lyapunov function  $V(x) = x^T P x$ . A matrix  $\mathcal{Q}$  will be called an *observability grammian*, if

$$\forall q \in Q: A_q^T \mathcal{Q} + \mathcal{Q} A_q + C_q^T C_q \leq 0, \mathcal{Q} > 0. \quad (4.3)$$

Likewise, a matrix  $\mathcal{P}$  will be called *controllability grammians*, if

$$\forall q \in Q: A_q \mathcal{P} + \mathcal{P} A_q^T + B_q B_q^T \leq 0, \mathcal{P} > 0. \quad (4.4)$$

Note that in contrast to the linear case, controllability/observability grammians for LSSs are not unique, since they are solutions of LMIs and not of Lyapunov equation.

The procedure for balanced truncation is as follows. We apply Algorithm 5 with the following choice of  $W$  and  $V$ . Find  $U$  such that  $\mathcal{P} = U U^T$  and find an orthogonal  $L$  such that  $U^T \mathcal{Q} U = L \Lambda^2 L^T$ , where  $\Lambda = \text{diag}(\sigma_1, \dots, \sigma_n)$  and  $\sigma_1 \geq \dots \geq \sigma_n \geq 0$ . Pick  $r \leq n$ . Define

$$W = [I_r \quad 0] \Lambda^{1/2} L^T U^{-1}, \quad V = U L \Lambda^{-1/2} \begin{bmatrix} I_r \\ 0 \end{bmatrix}.$$

#### 4.4. MODEL REDUCTION BY BALANCED TRUNCATION

Then  $\text{rank } W = \text{rank } V = r$ ,  $\text{rank } WV = r$ ,  $S = T = I_r$ .

The intuition behind the procedure above is similar to that of balanced truncation for linear systems: By applying the transformation  $\mathcal{S} = \Lambda^{1/2} L^T U^{-1}$  to  $\Sigma$ , we obtain a LSS  $\Sigma_{bal} = (n, \{\mathcal{S}A_q\mathcal{S}^{-1}, \mathcal{S}B_q, C\mathcal{S}^{-1}\}_{q \in Q})$ , such that  $\Lambda = \mathcal{S}^{-T} \mathcal{Q} \mathcal{S}^{-1} = \mathcal{S} \mathcal{P} \mathcal{S}^T$  is both an observability and a reachability grammian. We obtain  $\tilde{\Sigma}$  from  $\Sigma_{bal}$  by taking the upper-left  $r \times r$ ,  $r \times m$ ,  $p \times r$ , blocks of  $A_q$ ,  $B_q$ ,  $C_q$ ,  $q \in Q$  respectively. That is, we discard those states which correspond to small values of the diagonals of  $\Lambda$ . The intuition behind it is that the discarded states are either difficult to reach (it requires high energy input to reach them) or difficult to observe (their contribution to the energy of the output is small). More precisely, let us fix an integer  $r > 0$  which represents the desired state dimension of the reduced order model. Let  $(\tilde{x}, u, \sigma, \tilde{y})$  be a solution of  $\Sigma_{bal}$  such that  $\tilde{x}(0) = 0$ , assume that for all  $t > \tau_0$ ,  $u(t) = 0$ . It then can be shown [227] that

$$\begin{aligned} \sum_{i=1}^r \tilde{x}_i^2(\tau_0) \frac{1}{\sigma_i} + \sum_{i=r+1}^n \frac{1}{\sigma_i} \tilde{x}_i^2(\tau_0) &\leq \int_0^{\tau_0} \|u(s)\|^2 ds \\ \sum_{i=1}^r \tilde{x}_i^2(\tau_0) \sigma_i + \sum_{i=r+1}^n \sigma_i \tilde{x}_i^2(\tau_0) &\geq \int_{\tau_0}^{\infty} \|\tilde{y}(s)\|^2 ds \end{aligned} \quad (4.5)$$

and  $\tilde{x}_i(\tau_0)$  denotes the  $i$ th component of  $\tilde{x}(\tau_0)$ . The numbers  $\sigma_1, \dots, \sigma_n$  are called *singular values* of the pair  $(\mathcal{P}, \mathcal{Q})$  and they are the square roots of the eigenvalues of the product  $\mathcal{P} \mathcal{Q}$ .

That is, if  $\sigma_{r+1}, \dots, \sigma_n$  are small, and the energy of  $u$  is small, i.e.,  $\int_0^{\tau_0} \|u(s)\|^2 ds$  is small, then the values  $\tilde{x}_{r+1}(\tau_0), \dots, \tilde{x}_n(\tau_0)$  have to be small due to the first inequality, and they contribute little to the output starting from the time instance  $T_0$  due to the second inequality. That is, only the first  $r$  components of  $\tilde{x}$  are relevant for the input-output behavior. Hence, by discarding the last  $n - r$  state components we can expect to get a LSS, input-output behavior of which is close to the original one.

This intuition can be formalized as the error bound below.

**Theorem 4.1** (Error bound for balanced truncation [227]). *For any  $\sigma \in \mathcal{Q}$ ,  $u \in \mathcal{U}$  such that  $\int_0^{\infty} \|u(s)\|_2^2 ds < +\infty$ ,*

$$\int_0^{\infty} \|Y_{\Sigma}(u, \sigma)(s) - Y_{\tilde{\Sigma}}(u, \sigma)(s)\|^2 ds \leq (2 \sum_{k=r+1}^n \sigma_k)^2 \int_0^{\infty} \|u(s)\|^2 ds$$

**Further extensions** The results discussed above also hold for discrete-time LSSs [227]. The assumption that  $\mathcal{P}, \mathcal{Q}$  do not depend on  $q \in Q$  implies quadratic stability, which is a quite restrictive assumption. In [106] this assumption was replaced by local stability of the linear subsystems. Moreover, an error bound similar to Theorem 4.1 was derived in [106], but it holds only for switching signals with a sufficiently large dwell time. Note that [106] allows for LSSs with non-trivial linear reset maps. Extension of [106] to arbitrary switching was developed in [108]. Finally, balanced truncation was extended to linear switched systems with autonomous switching [212].

**Relationship with realization theory** First, the existence of positive definite grammians  $\mathcal{P}, \mathcal{Q}$  is a necessary (but not sufficient) condition for minimality of quadratically stable LSSs [227]: the kernels of positive semi-definite observability (resp. controllability) grammians are contained in the set of unobservable states (resp. states which are not in the span of reachable states). Intuitively, when the grammians are positive definite, we can bring them to a balanced form and then identify the states corresponding to small singular values with unobservable/unreachable states. Then balanced truncation can be thought of as a numerical implementation of the minimization procedure Procedure B.3. Theorem 4.1 provides means to evaluate the effect of the discarded ‘small’ singular values on the approximation error.

Realization theory is also necessary to show that balanced truncation well posed. More precisely, the application of balanced truncation relies on the availability of a quadratically stable LSS and by Theorem 4.1, the quality of the reduced model relies on the singular values of the observability/controllability grammians. If the original model is not quadratically stable, then one may wonder if there exist input-output equivalent quadratically stable models. Using realization theory it is shown in [227] that in order to decide if balanced truncation can be applied, it is sufficient to transform the original model to a minimal one (which also makes sense for the purposes of

model reduction), and then to check if the minimal model is quadratically stable. Moreover, any minimal model can be used for balanced truncation without introducing more conservativity. Indeed, in [227] it is shown that the singular values of any pair of controllability/observability grammians for any LSS are not smaller than the singular values of some pair of grammians of a minimal input-output equivalent LSS. Moreover, due to isomorphism, all controllability/observability grammians of minimal input-output equivalent LSSs are related by a similarity transform and have the same singular values.

Finally, realization theory and the notion of Hankel-matrix can be used to relate singular values of grammians to norms of a Hankel-operator [227].

## 4.5 Moment matching

Consider a LSS  $\Sigma$  of the form (2.6), and let us denote its input-output map by  $f$ . For the sake of simplicity, we assume that  $p = m = 1$ , i.e., we deal only with the SISO case, and we consider systems only in continuous-time. Recall from that since  $f$  is realizable by a LSS then  $f$  has to have a generalized kernel representation, see [205, Definition 24], Appendix B.5, i.e. there must exist a sequence of analytic functions  $\{G_v^f\}_{v \in Q^+}$ , such that for any switching signal  $\sigma$  and input  $u$ ,

$$f(u, \sigma)(t) = \int_0^t G^f(\sigma, s, t) u(s) ds$$

and if  $\sigma(\tau) = q_i$  for  $\tau \in [T_{i-1}, T_i]$ ,  $T_0 = 0 < T_1 < \dots < T_k$ ,  $i = 1, 2, \dots$ , then

$$G^f(\sigma, s, t) = G_{q_i \dots q_k}^f(T_i - s, T_{i+1} - T_i, T_{i+2} - T_{i+1}, \dots, t - T_{k-1})$$

if  $s \in [T_{i-1}, T_i]$  and  $t \in (T_{k-1}, T_k]$  for some integers  $0 < i < k$ . That is,  $G_{q_k}^f, G_{q_{k-1}q_k}^f, G_{q_1 \dots q_k}^f$  determine the response of the input-output function  $f$  to the switching signal which switches through the discrete states  $q_1, q_2, \dots, q_k$ .

The idea of moment matching is to find a reduced order LSS  $\bar{\Sigma}$  such that for certain sequences  $v \in Q^+$ ,  $G_v^f$  is close to  $G_v^{\bar{\Sigma}}$ . Intuitively, this means that the input-output map of  $\bar{\Sigma}$  will be close to that  $f$ .

In Subsection 4.5 we present the Markov-parameter based approach [24, 30, 31], where we look for a reduced order model  $\bar{\Sigma}$  such that certain Taylor-series coefficients of  $G_v^{\bar{\Sigma}}$  (Markov parameters of  $Y_{\bar{\Sigma}}$ ) and  $G_v^f$  (Markov parameters of  $f$ ) coincide. In Subsection 4.5 we present the frequency-domain based approach [105], where we consider multi-variate Laplace transforms  $H_v^f$  of  $G_v^f$  and we look for reduced order models  $\bar{\Sigma}$  such that the Laplace transform  $H_v^{\bar{\Sigma}}$  of the function  $G_v^{\bar{\Sigma}}$  coincides with  $H_v^f$  for some complex values.

### Matching Markov parameters

The basic idea of model reduction algorithms based matching Markov-parameters is to reduce the dimension of the original model while preserving certain high-order derivatives of the elements of the generalized kernel representation  $\{G_v^f\}_{v \in Q^+}$  of its input-output map  $f$ .

More precisely, recall from Appendix B.5 or [205, Definition 25] the notion of a Markov parameter  $S_{q, q_0}^f(v)$ ,  $v \in Q^*$ :

$$S_{q, q_0}^f(\varepsilon) = G_{q_0 q}^f(0, 0), \quad S^f(q, q_0)(q_1 q_2 \dots q_k) = \frac{d}{dt_1} \dots \frac{d}{dt_k} G_{q_0 q_1 \dots q_k}^f(0, t_1, \dots, t_k, 0)|_{t_1 = \dots = t_k = 0}$$

for all  $q, q_0, q_1, \dots, q_k \in Q$ ,  $k \geq 1$ . In fact,  $S_{q, q_0}^f(v)$ ,  $v \in Q^*$  are the Taylor series expansion of  $G_{q_1, \dots, q_k}^f$ , i.e., for all  $q_1, \dots, q_k \in Q$ ,  $k \geq 1$ ,  $\alpha_1, \dots, \alpha_k \in \mathbb{N}$ ,

$$\frac{d^{\alpha_1}}{dt_1^{\alpha_1}} \dots \frac{d^{\alpha_k}}{dt_k^{\alpha_k}} G_{q_1 \dots q_k}^f(t_1, \dots, t_k)|_{t_1 = t_2 = \dots = t_k = 0} = S_{q_1, q_k}(q_1^{\alpha_1} \dots q_k^{\alpha_k})$$

#### 4.5. MOMENT MATCHING

Furthermore, recall from [30] or from Appendix B.5 the notion of selection. More precisely, for the particular case of SISO systems, we call any subset  $\alpha \subseteq Q^* \times Q$  a *selection*. Let  $\alpha$  and  $\beta$  be two selections. Consider a LSS  $\bar{\Sigma}$ . Then  $\bar{\Sigma}$  is called a  $(\alpha, \beta)$ -*partial realization of  $f$* , if for every  $(v, q_0) \in \beta$ ,  $(u, q) \in \alpha$ ,

$$S_{q, q_0}^{Y_{\bar{\Sigma}}}(vu) = S_{q, q_0}^f(vu) \quad (4.6)$$

That is,  $\bar{\Sigma}$  is an  $(\alpha, \beta)$ -partial realization of  $f$ , if those Markov-parameters of  $f$  and of the input-output map  $Y_{\bar{\Sigma}}$  of  $\bar{\Sigma}$  which are indexed by  $(\alpha, \beta)$  coincide. This means that certain high-order derivatives of  $\{G_v^f\}_{v \in Q^+}$  and of  $\{G_v^{Y_{\bar{\Sigma}}}\}_{v \in Q^+}$  are the same. That is, a  $(\alpha, \beta)$ -partial realization of  $f$  can be viewed as an LSS input-output map of which approximates  $f$ . If  $\alpha = Q^* \times Q$  or  $\beta = Q^* \times Q$ , then any  $(\alpha, \beta)$ -partial realization of  $f$  is a realization of  $f$ .

The idea behind moment matching is then to replace a LSS  $\Sigma$  by a reduced order LSS  $\bar{\Sigma}$  such that  $\bar{\Sigma}$  is a  $(\alpha, \beta)$ -partial realization of the input-output map  $f = Y_{\Sigma}$  of  $\Sigma$ . The various algorithms differ in the way the selections  $\alpha$ ,  $\beta$  are chosen. The moment matching algorithms which produce  $(\alpha, \beta)$ -partial realizations arise from Algorithm 5 by a suitable choice of the matrices  $W$  and  $V$ . In order to explain these choices in more detail, we introduce the following definitions. Define the subspaces

$$\begin{aligned} \mathcal{O}_{\alpha}(\Sigma) &= \bigcap_{(v, q) \in \alpha} \ker C_q A_v, \\ \mathcal{R}_{\beta}(\Sigma) &= \text{Span}\{A_w B_{q_0} \mid (w, q_0) \in \beta\}. \end{aligned}$$

The matrices  $W$  and  $V$  can be chosen as follows.

- (A)  $\ker W = \mathcal{O}_{\alpha}(\Sigma)$  and  $V = I_n$ . Then Algorithm 5 returns a  $(\alpha, \{\varepsilon\} \times Q)$ -partial realization of  $f$  [30, Theorem 3].
- (B)  $\text{Im} V = \mathcal{R}_{\beta}(\Sigma)$  and  $W = I_n$ . Then Algorithm 5 returns a  $(\{\varepsilon\} \times Q, \beta)$ -partial realization of  $f$ , [30, Theorem 2].
- (C)  $\ker W = \mathcal{O}_{\alpha}(\Sigma)$ ,  $\text{Im} V = \mathcal{R}_{\beta}(\Sigma)$ ,  $\text{rank } W = \text{rank } V = \text{rank } WV$ . Then Algorithm 5 returns  $(\alpha, \beta)$ -partial realization of  $f$ , [30, Theorem 4].

There are two strategies for choosing  $\alpha$ ,  $\beta$ .

The first one is to choose  $\alpha$  (resp.  $\beta$ ) to be finite such that  $\dim \mathcal{O}_{\alpha}(\Sigma) = n - r$  (resp.  $\dim \mathcal{R}_{\beta}(\Sigma) = r$ ), and in this case the reduced order model will have dimension  $r$ . In this case,  $\mathcal{O}_{\alpha}(\Sigma) = \ker O_{\alpha}$  (resp.  $\mathcal{R}_{\beta}(\Sigma) = R_{\beta}$ ), where

$$\begin{aligned} O_{\alpha} &= [C_{s_1} A_{v_1}^T \quad \dots \quad C_{s_r} A_{v_r}^T]^T, \\ R_{\beta} &= [A_{w_1} B_{q_1} \quad \dots \quad A_{w_r} B_{q_r}] \end{aligned}$$

where  $\alpha = \{(v_i, s_i)\}_{i=1}^r$ ,  $\beta = \{(w_i, q_i)\}_{i=1}^r$ . Using these matrix representations the matrices  $W$  and  $V$  described above can easily be computed.

The second option for choosing selections is to choose  $(\alpha, \beta)$  to be consistent with a certain set of switching signals. In this latter case, the dimension of the reduced order model cannot be fixed in advance, but it is known that the reduced order model will have the same input-output behavior along those switching sequences which belong to this designated set. More precisely, assume that the switching signal  $\sigma \in \mathcal{Q}$  has the property that  $\sigma(s) = q_i$ ,  $s \in [T_{i-1}, T_i]$ ,  $T_0 = 0$ ,  $T_i = \sum_{r=1}^i t_r$  for some  $q_i \in Q$ ,  $0 < t_i \in \mathbb{R}_+$ ,  $0 < i \in \mathbb{N}$ . We will say that a selection  $(\alpha, \beta)$  is *consistent with  $\sigma$* , if for every  $i > 0$ , for every  $\omega_1, \dots, \omega_k \in \mathbb{N}$ ,

$$((q_i)^{\omega_i} (q_{i+1})^{\omega_{i+1}} \dots (q_k)^{\omega_k}, q_i) \in \beta, ((q_1)^{\omega_1} (q_2)^{\omega_2} \dots (q_i)^{\omega_i}, q_i) \in \alpha.$$

It then turns out that if the selection  $(\alpha, \beta)$  is consistent with a switching signal  $\sigma$ , then all  $(\alpha, \beta)$ -partial realizations have the same input-output behavior along the switching signal  $\sigma$ . The latter observation then characterizes the effect of moment matching algorithms which use selections. The formal theorem is as follows.



**Theorem 4.2** (Properties of moment matching using selections [30]). *Assume that  $(\alpha, \beta)$  is consistent with  $\sigma$  and  $\bar{\Sigma}$  is  $(\alpha, \beta)$ -partial realization of  $f$ . Then  $Y_{\bar{\Sigma}}(u, \sigma) = f(u, \sigma)$ , for all  $u \in \mathcal{U}$ .*

Note that for a pair of selections to be consistent with a switching signal (or a set of switching signals), the selections involved have to be infinite sets. If the prefixes of the sequences of discrete modes of the desired switching signals form a regular language, then there exist algorithms to compute matrix representations of  $\mathcal{O}_\alpha(\Sigma)$ ,  $\mathcal{R}_\beta(\Sigma)$ , see [30, 31].

**Further extensions.** The model reduction method described above was extended to LPV models [28] and bilinear systems [228]. In addition, the method above was applied to LSSs arising from asynchronous sampling of linear time-invariant systems [25].

**Relationship with realization theory.** To begin with, the whole idea of matching Markov parameters relies on the notion of Markov-parameters and partial-realization, which are integral parts of realization theory. In fact, the result of the Ho-Kalman realization algorithm from Algorithm 13 is isomorphic to the LSS returned by Algorithm 5 with the choice of the matrices  $W$  and  $V$  as described in option (C) above. That is, moment matching is just a reformulation of Ho-Kalman algorithm when the latter is applied to finite Hankel-matrices, rank of which is not maximal. The partial realization algorithm of [219] is a particular instance of this model reduction method, if  $\alpha = \beta = \{v \in Q^* \mid |v| \leq N\} \times Q$  is chosen. Furthermore, Theorem 4.2 and its counterpart for the discrete-time case [31] can be viewed as extensions of realization theory of LSSs with constrained switching [202, 203].

### Moment matching in frequency domain

By applying multivariate Laplace transform of the functions  $\{G_v^f\}_{v \in Q^+}$  we can define a sequence of functions  $\{H_v^f\}_{v \in Q^+}$  of complex variables as follows:

$$H_v^f(s_1, \dots, s_k) = \int_0^\infty \cdots \int_0^\infty G_v(t_1, \dots, t_k) e^{s_1 t_1 + \cdots + s_k t_k} dt_1 \cdots dt_k \quad (4.7)$$

for all  $\text{Re}(s_j) > s_0$  for a suitable  $s_0 \in \mathbb{R}$ , where  $k = |v|$ . If  $f$  has a realization by a LSS  $\Sigma$  of the form (2.6) then by [203],

$$G_{q_1 \dots q_k}^f(t_1, \dots, t_k) = C_{q_k} e^{A_{q_k} t_k} \cdots e^{A_{q_1} t_1} B_{q_1},$$

and hence

$$H_{q_1, q_2, \dots, q_k}^f(s_1, s_2, \dots, s_k) = C_{q_k} \Phi_{q_k}(s_k) \Phi_{q_{k-1}}(s_{k-1}) \cdots \Phi_{q_1}(s_1) B_{q_1}, \quad (4.8)$$

where  $\Phi_q(s) = (sI_n - A_q)^{-1}$ ,  $q_j \in Q$ ,  $1 \leq j \leq k$ . We call the functions  $\{H_v^f\}_{v \in Q^+}$  the *generalized transfer functions* of the input-output map  $f$  [105].

Let  $\Gamma$  and  $\Theta$  be finite sets of tuples so that

$$\Gamma, \Theta \subseteq \{(v, \underline{\mu}) \mid v \in Q^+, \underline{\mu} \in \mathbb{C}^k, k = |v|\}.$$

We will say that a LSS  $\bar{\Sigma}$  is a  $(\Gamma, \Theta)$ -partial realization of  $f$ , if for every  $(w, \underline{\mu}) \in \Gamma$ ,  $(v, \underline{\lambda}) \in \Theta$ ,

$$H_{wv}^f(\underline{\mu}, \underline{\lambda}) = H_{wv}^{Y_{\bar{\Sigma}}}(\underline{\mu}, \underline{\lambda}).$$

Intuitively,  $\bar{\Sigma}$  being a  $(\Gamma, \Theta)$ -partial realization of  $f$  means that the generalized transfer functions  $f$  and that of the input-output map of  $\bar{\Sigma}$  coincide for some switching sequences and some frequencies.

Our goal is to find a LSS  $\bar{\Sigma}$  such that  $\bar{\Sigma}$  is a  $(\Gamma, \Theta)$ -partial realization of  $f$ , and the dimension of  $\bar{\Sigma}$  is smaller than that of  $\Sigma$ .

To this end, for any  $v = q_1 \cdots q_k \in Q^+$ ,  $q_1, \dots, q_k \in Q$ , define

$$\begin{aligned} r((v, \underline{\mu})) &= \Phi_{q_k}(\underline{\mu}_k) \cdots \Phi_{q_1}(\underline{\mu}_1) B_{q_1}, \\ o((v, \underline{\mu})) &= C_{q_k} \Phi_{q_k}(\underline{\mu}_k) \cdots \Phi_{q_1}(\underline{\mu}_1). \end{aligned}$$

## 4.6. CONCLUSIONS

---

for any  $\underline{\mu} = (\mu_1, \dots, \mu_k) \in \mathbb{C}^k$ . Assume that the cardinality of  $\Gamma$  and  $\Theta$  are both  $r$  and consider an enumeration  $\Gamma = \{(w_i, \underline{\mu}_i)\}_{i=1}^r$   $\Theta = \{(v_i, \underline{\lambda}_i)\}_{i=1}^r$  of these sets. Define the matrices

$$R = [r((w_1, \underline{\mu}_1)) \quad \dots \quad r((w_k, \underline{\mu}_r))],$$

$$O = [o((v_1, \underline{\lambda}_1))^T \quad \dots \quad o((v_k, \underline{\lambda}_r))^T]^T$$

Assume that  $\text{rank } OR = r$ . We can apply Algorithm 5 with  $W = O$ ,  $V = R$  resulting in a LSS  $\bar{\Sigma}$  which will be a  $(\Gamma, \Theta)$ -partial realization of  $f$ . The formal theorem is as follows.

**Theorem 4.3** (Correctness of frequency domain moment matching [105]). *With the notation and assumptions above, the LSS  $\bar{\Sigma}$  is a  $(\Gamma, \Theta)$ -partial realization of  $f$ .*

The method described above has an alternative formulation in terms of *generalized Loewner matrices* [105], thus extending the well-known Loewner-matrix based model reduction method for linear systems.

**Further extensions** The results described above were extended to LPV systems in [107].

**Relationship with realization theory** The reformulation of this method in terms of generalized Loewner matrices yields a partial realization algorithm, as it depends on data which can directly be obtained from Laplace transforms of the input-output map. In a way, this method is the first step towards a reformulation of realization theory of LSSs in frequency domain.

## 4.6 Conclusions

In this chapter I presented a brief overview of my contribution to model reduction of linear switched systems. It is well known that for linear systems there is a deep connection between these two disciplines. As the results of this chapter demonstrate, the same holds for linear switched systems too. There are many possible directions for future research. A particularly natural one is to extend the results of this chapter to hybrid systems with state-dependent switching, for example to piecewise linear systems. The latter can be viewed as a feedback interconnection of a linear switched system with an discrete event generator, hence we are hopeful that the results of this chapter will be useful for such an extension.



## Chapter 5

# Identification of LPV systems

In this chapter I present a selection of topics related to systems identification which I worked on. I will concentrate on LPV systems, and on those topics which rely on system identification. The reason for concentrating on LPV systems is that my work on system identification mostly concerned linear switched systems with external switching and LPV systems. The former is a subclass of the latter, hence, it seems more straightforward to discuss the results for the more general system class.

System identification of LPV systems is a very active research area, without claiming completeness one can mention [18, 155, 175, 231, 88, 288, 268, 284, 289, 49, 99, 150, 273, 67, 165, 166]. Within LPV system identification, a large portion of the literature is focused on identifying input-output LPV models, see for example [49, 155, 18, 231, 69]. My own work was focused on LPV state-space representations. The existing methods in system identification of LPV state-space representations can be divided into two categories: parametric identification methods and subspace identification methods. Without claiming completeness, for parametric methods we refer to [308, 159, 158, 240, 42, 19, 192], and for subspace methods to [284, 88, 288, 289, 76, 67] and the references therein. In the case of parametric methods, the sought after state-space representation is assumed to be a member of a parameterized family of state-space representations, and the task is to estimate the correct value of the parameter from data. In the case of subspace methods, the state-space representation is assumed to be a black-box, and the system matrices are estimated up to an isomorphism. This division is not very strict, as subspace methods could be viewed as a particular instance of parametric methods, and prediction error minimization (PEM) methods [308] can be applied to black-box models. However, for the purposes of our discussion, such a classification is useful. Another way of classifying LPV state-space identification methods is whether they are local or global. Local methods aim at identifying the LTI model for each constant value of the scheduling signal and then merge the resulting family of LTI models, again, without claiming completeness we can mention [314, 99, 291] and the references therein. This is in contrast to global methods, where the LPV state-space representation is estimated using time-varying scheduling signals. In this chapter, I will focus on global methods.

My work on LPV system identification was centered around the following two problems:

- Characterizing identifiability of LPV state-space representations.
- Proving statistical consistency of subspace algorithms and overcoming the curse of dimensionality and weakening the usual observability assumptions made for subspace methods.

When I started research on LPV system identification, the existing literature did not completely solve the problem of characterizing identifiability of LPV state-space representations, despite the pioneering work [158]. Identifiability is important for parametric methods, as finding the best parameter estimate for non-identifiable parameterizations is conceptually and numerically problematic [171, 307, 285]. In addition, when I started working on LPV system identification, there were no results on statistical consistency of subspace identification algorithms. Moreover, most of the available subspace identification algorithms suffered from curse of dimensionality and

they relied on quite strong observability assumptions about the underlying data-generating model. My research in this topics was aimed at filling these gaps. In fact, similarly to the case of LTI systems, addressing both issues for LPV state-space representations required results from realization theory.

In this chapter we will consider deterministic and stochastic LPV systems. For deterministic LPV systems, we consider state-space representations with affine dependence on parameters (LPV-SSA) from Chapter 2, and we present results on identifiability analysis and a deterministic subspace identification algorithm. The results on identifiability are based on the paper [5] and PhD thesis [4], the results on subspace identification are based on [66]. The results of the deterministic case rely on realization theory of LPV-SSAs presented in Chapter 2. The stochastic case is based on the results of [172, 173] and it uses the results of Chapter 2 on realization theory of stochastic bilinear systems. In this chapter, we will deal only with *discrete-time systems*, as they are simpler and most of the existing literature on system identification of LPV and switched systems deals with the discrete-time case. As it was mentioned before, linear switched systems can be viewed as LPV systems with scheduling variables taking values in a finite set. Hence, the results of this chapter can directly be applied to linear switched systems.

The chapter is organized as follows. In Section 5.1 we deal with system identification of deterministic LPV-SSAs. In Section 5.2 we present stochastic subspace identification algorithms. In each section, we will discuss the role of realization theory in the topic discussed in that particular section.

## 5.1 System identification of deterministic LPV-SS systems

The presentation of this section is based on [66, 5]. We will use the notation and terminology introduced in Chapter 2, in particular, in Section 2.2 and Section 2.3. The system identification problem can be formulated in various ways. First, we start with an intuitive and informal problem formulation.

**Problem 5.1** (System identification: informal). *Let  $(u, p, y) \in \mathcal{U} \times \mathcal{P} \times \mathcal{Y}$  be signals such that for some  $x$ ,  $(x, u, p, y)$  is a solution of an unknown LPV-SSA  $\Sigma_{true}$ . Based on the data  $\{u(t), p(t), y(t)\}_{t=0}^N$  find a LPV-SSA  $\Sigma_N = (\mathbb{P}, \{A_i^N, B_i^N, C_i^N\}_{i=0}^{n_p})$  such that for large enough  $N$ , the input-output behavior of  $\Sigma_N$  is close enough to the input-output behavior of  $\Sigma_{true}$ .*

Intuitively,  $\Sigma_{true}$  can be thought of as the true LPV-SSA which generates the observations. The problem above is ill-posed. First of all, we did not specify what we mean my input-output behavior. Second, we did not specify how to compare them. The most natural definition of input-output behavior would be that of the behavioral approach [306], which was extended to LPV systems [273], i.e., in this case the input-output behavior of an LPV-SSA  $\Sigma$  is  $\mathcal{B}(\Sigma) = \{(\tilde{u}, \tilde{p}, \tilde{y}) \mid \exists \tilde{x} \in \mathcal{X} : (x, \tilde{y}, \tilde{u}, \tilde{p}) \text{ is a solution of } \Sigma\}$ . However, this approach presents a number of conceptual and technical difficulties. Instead, in order to keep the discussion simple, we follow the classical approach in system identification and we formalize the input-output behavior of a LPV-SSA  $\Sigma$  as the input-output map of  $\Sigma$  induced by the zero initial state. We then compare the input-output behavior of  $\Sigma_N$  and  $\Sigma_*$  by comparing the system matrices of  $\Sigma_N$  with the system matrices of a LPV-SSA realization of  $\mathfrak{F} = \mathfrak{Y}_{\Sigma_{true}, 0}$ . That is, we would like the system matrices of  $\Sigma_N$  to be close to the system matrices of a LPV-SSA  $\Sigma_*$  such that  $\Sigma_*$  is a realization of  $\mathfrak{F}$ . In this way, we ensure that for any scheduling signal and input, the output of  $\Sigma_N$  will be close to that of  $\Sigma_*$ , which, in turn, by virtue of  $\Sigma_*$  being a realization of  $\mathfrak{F}$ , will be equal to the output of the true system  $\Sigma_{true}$ . Note that the output responses of  $\Sigma_N$  and  $\Sigma_*$  could be close even if the system matrices are different. In fact, two LPV-SSAs could have the same input-output map while their system matrices are very far from each other. For this, it is sufficient to apply a linear state-space transformation to them. However, if the system matrices are close, then the output responses to the same input and scheduling signal will also be close, so the proposed approach is meaningful, even if it is conservative.

We then arrive to the following problem formulation.

**Problem 5.2** (System identification: deterministic and non-parametric). *Consider an LPV-SSA input-output map  $\mathfrak{F}$  of the form (2.4) and assume that it has an LPV-SSA realization. Let  $(u, p, y) \in \mathcal{U} \times \mathcal{P} \times \mathcal{Y}$  be signals such*

## 5.1. SYSTEM IDENTIFICATION OF DETERMINISTIC LPV-SS SYSTEMS

that  $y = \mathfrak{F}(u, p)$ . Based on the data  $\{u(t), p(t), y(t)\}_{t=0}^N$  find a LPV-SSA  $\Sigma_N = (\mathbb{P}, \{A_i^N, B_i^N, C_i^N\}_{i=0}^{n_p})$  such that for all  $i \in \mathbb{I}_{0, n_p}$ , the limits

$$A_i = \lim_{N \rightarrow \infty} A_i^N, B_i = \lim_{N \rightarrow \infty} B_i^N, \lim_{N \rightarrow \infty} C_i^N = C_i$$

exist, and  $\Sigma_* = (\mathbb{P}, \{A_i, B_i, C_i\}_{i=0}^{n_p})$  is a realization of  $\mathfrak{F}$ .

That is, based on measurements  $\{u(t), p(t), y(t)\}_{t=0}^N$  find an LPV-SSA  $\Sigma_N$  such that for large enough  $N$ , the input-output maps of  $\Sigma_N$  should be close enough to  $\mathfrak{F}$ , or, in other words,  $\Sigma_T$  should be close to a true realization of  $\mathfrak{F}$ . Often, the goal is not to find just an approximate LPV-SSA realization of  $\mathfrak{F}$ , but to find a realization with a particular structure, i.e., an element of a parameterization. In order to formalize this concept, we introduce the following terminology.

**Definition 5.1 (LPV parameterization).** Let  $\Theta \subseteq \mathbb{R}^{n_\theta}$  for some integer  $n_\theta > 0$  and let us fix the integers  $n_x, n_u, n_y, n_p$ . An LPV-SS parameterization  $\Pi$  is a map

$$\Pi : \Theta \rightarrow (\mathbb{R}^{n_x \times n_x} \times \mathbb{R}^{n_x \times n_u} \times \mathbb{R}^{n_y \times n_x})^{n_p+1}. \quad (5.1)$$

The interpretation of a parameterization is as follows. For every  $\theta \in \Theta$ ,  $\Pi(\theta)$  is of the form

$$\{(A_i(\theta), B_i(\theta), C_i(\theta))\}_{i=0}^{n_p},$$

for some  $A_i(\theta) \in \mathbb{R}^{n_x \times n_x}, B_i(\theta) \in \mathbb{R}^{n_x \times n_u}, C_i(\theta) \in \mathbb{R}^{n_y \times n_x}$  for all  $i \in \mathbb{I}_{0, n_p}$ . Then,  $\Pi(\theta)$  can be associated with an LPV-SSA

$$\Sigma_\Pi(\theta) = (\mathbb{P}, \{A_i(\theta), B_i(\theta), C_i(\theta)\}_{i=0}^{n_p}),$$

i.e.,  $\Pi$  can be viewed as a map which associates an LPV-SS model with each value  $\theta \in \Theta$  and the matrices of this LPV-SS model are functions of  $\theta$ . Note that we can also view  $\Pi$  as a map  $\Pi : \Theta \rightarrow \mathbb{R}^N$ ,  $N = n_x(n_x + n_y + n_u)(n_p + 1)$ , by storing the matrices  $\{A_i(\theta), B_i(\theta), C_i(\theta)\}_{i=0}^{n_p}$  for  $\theta \in \Theta$  as vectors. Note that if  $\Theta = \mathbb{R}^N$  and  $\Pi$  is the identity map, then the image of  $\Pi$  contains all LPV-SSA representations of dimension  $n_x$ .

The motivation for introducing parameterizations is as follows. Often, instead of looking for an approximate LPV-SSA realization of  $\mathfrak{F}$  in general, one wants to find an approximate LPV-SSA realization from a particular parameterization. This is the case, for example, if a LPV-SSA realization of  $\mathfrak{F}$  could in principle be derived from the first principles, but some of the physical quantities involved are unknown. This leads us to the following reformulation of the identification problem.

**Problem 5.3 (System identification: deterministic and parametric setting).** Let  $\Pi$  be a LPV-SSA parameterization of the form (5.1) and consider the signals  $(u, p, y) \in \mathcal{U} \times \mathcal{P} \times \mathcal{Y}$  such that  $y = \mathfrak{F}(u, p)$ . Based on the data  $\{u(t), p(t), y(t)\}_{t=0}^N$  find  $\theta_N \in \Theta$ , such that  $\lim_{N \rightarrow \infty} \theta_N = \theta_*$  and  $\Pi(\theta_*)$  is a realization of  $\mathfrak{F}$ .

Intuitively, the parametric system identification problem is to find a parameter  $\theta_N$  based on the measurements, such that the corresponding LPV-SSA  $\Pi(\theta_N)$  is close to a realization of  $\mathfrak{F}$ , at least for a large  $N$ . This is expressed by the requirement that  $\lim_{N \rightarrow \infty} \theta_N = \theta_*$  and  $\Pi(\theta_*)$  is a realization of  $\mathfrak{F}$ . The same intuition could have been formalized in a less restrictive manner, but for the purposes of this chapter the formalization above is sufficient. The parameter  $\theta_*$  can be viewed as the parameter corresponding to a true realization of  $\mathfrak{F}$ , to the exact model of underlying system.

The non-parametric system identification problem Problem 5.2 is a special case of Problem 5.3 if we take the trivial parameterization  $\Pi(\theta) = \theta, \forall \theta \in \Theta = \mathbb{R}^N$ . However, as we shall see later, viewing Problem 5.2 as a special case of Problem 5.3 is not very useful for a number of practical reasons.

The standard approach to solving problem Problem 5.3 is as follows. First, we define a loss function

$$\ell_N(\theta, \{u(t), p(t), y(t)\}_{t=0}^N)$$

which measures the difference between the actual output  $y(t)$  and the output  $\hat{y}(t)$  of the LPV-SSA  $\Pi(\theta)$ , when the latter is started from some initial state and fed the scheduling variables and inputs  $\{u(t), p(t)\}_{t=0}^N$ . A typical choice of the loss function  $\ell_N(\theta, \{u(t), p(t), y(t)\}_{t=0}^N)$  is as follows:

$$\ell_N(\theta, \{u(t), p(t), y(t)\}_{t=0}^N) = \frac{1}{N} \sum_{t=0}^N \|y(t) - \mathfrak{Y}_{\Pi(\theta),0}(u, p)(t)\|_2^2, \quad (5.2)$$

Then we choose  $\theta_T$  be the solution of the following optimization problem

$$\theta_T = \operatorname{argmin}_{\theta \in \Theta} \ell_T(\theta, \{u(t), p(t), y(t)\}_{t=0}^N). \quad (5.3)$$

That is, the identification problem of Problem 5.2 can be reduced to the optimization problem in (5.3), where  $\theta_N$  is chosen in such a way that it gives the smallest possible difference between the actual measured output and the one predicted by  $\Pi(\theta_N)$ , i.e.,  $\Pi(\theta_N)$  is model which explains the data best among all the models described by the parameterization  $\Pi$ .

So far we have not required that there should be a one-to-one relationship between the parameters  $\theta$  and the input-output behavior of  $\Pi(\theta)$ , i.e., it could happen that  $\mathfrak{Y}_{\Pi(\theta_1)} = \mathfrak{Y}_{\Pi(\theta_2)}$  for some  $\theta_1, \theta_2$ . If this is the case, then we have to deal with two problems:

1. **Non-uniqueness of the true model.** It can happen that the set  $\mathcal{D}(\mathfrak{F}) = \{\theta \mid \Pi(\theta) \text{ is a realization of } \mathfrak{F}\}$  contains several elements and hence the choice of  $\theta_*$  to which the estimates  $\theta_T$  have to converge is not unique. In principle, this need not be a problem. However, non-uniqueness of  $\theta_*$  makes the proof of convergence technically difficult, especially if  $\mathcal{D}_{\mathfrak{F}}$  is a connected set. Moreover, if the parameters have a physical meaning and we would like to estimate them from data, then this becomes impossible, as there are several possible parameter values which are consistent with any measured data.
2. **Local and global minim.a** In this case, for most of reasonably defined loss functions, in particular for the one of (5.2),  $\ell_N(\theta_1, \{u(t), p(t), y(t)\}_{t=0}^N) = \ell_N(\theta_2, \{u(t), p(t), y(t)\}_{t=0}^N)$ , and hence the optimization problem (5.3) has several solutions. In turn, this leads to numerical instability for most of the known optimization algorithms or convergence to a local minimum or lack of convergence.

For this reason, when it comes to parametric identification, one would like to avoid the situation when there are several parameters such that the corresponding models have the same input-output behavior. This prompts us to recall the following definition from [5]

**Definition 5.2 (Structural identifiability).** *An LPV-SS parameterization  $\Pi$  of the form (5.1) is called structurally identifiable if, for any two distinct parameters  $\theta_1, \theta_2 \in \Theta$ ,  $\theta_1 \neq \theta_2$ , the input-output maps of the corresponding LPV-SSAs  $\Sigma_{\Pi}(\theta_1)$  and  $\Sigma_{\Pi}(\theta_2)$  generated by zero initial states are different, i.e.,  $\mathfrak{Y}_{\Sigma_{\Pi}(\theta_1),0} \neq \mathfrak{Y}_{\Sigma_{\Pi}(\theta_2),0}$ .*

In other words, structural identifiability means that, for every two different parameters, there exists a sequence of inputs and scheduling parameter values  $(u, p) \in \mathcal{U} \times \mathcal{P}$ , such that the corresponding outputs are different. That is, in Problem 5.3 we want to deal only with structurally identifiable parameterizations. It is now easy to see why considering the trivial parameterization where  $\Pi$  is the identity is not suitable for Problem 5.3. Namely, the trivial parameterization is not structurally identifiable.

Realization theory can be used for solving Problem 5.2 and Problem 5.3 as follows. First, the Kalman-Ho-like realization algorithm can be used to get a identification algorithm for solving Problem 5.2 along with a persistence of excitation condition. Second, the theory of minimality can be used to find conditions for checking structural identifiability and to come up with structurally identifiable parameterizations.

This section is structured as follows. In Subsection 5.1.1 we present a subspace-like algorithm based on the Kalman-Ho-like algorithm from Chapter 2 and conditions for persistence of excitation. In Subsection 5.1.2 we discuss the application of realization theory to identifiability.

### 5.1.1 Deterministic subspace identification algorithm LPV-SSA

In this section we present subspace identification algorithms for deterministic LPV-SSAs. This section is based on [66].

Recall from Section 2.6, Chapter 2 that a LPV-SSA realization of  $\mathfrak{F}$  can be calculated from the Hankel-matrix  $H_{\mathfrak{F},N,N+1}$  using Algorithm 1, or from the matrices  $H_{\mathfrak{F},\alpha,\beta}$   $H_{q,\mathfrak{F},\alpha,\beta}$   $H_{q,\mathfrak{F},\alpha}$   $H_{\mathfrak{F},\beta,q}$  for some selections  $\alpha,\beta$  using Algorithm 2. Recall that these matrices can be constructed from the sub-Markov parameters of  $\mathfrak{F}$ . Hence, one approach for solving Problem 5.2 is to estimate the sub-Markov parameters from the measured data  $\{u(t), p(t), y(t)\}_{t=0}^N$  and then apply Algorithm 1 or Algorithm 2.

Notice that if  $\mathfrak{F}$  has a realization by a LPV-SSA, it has a IIR representation and hence if  $y = \mathfrak{F}(u, p)$ , then

$$y(t) = \sum_{\delta=0}^{t-1} (h_{\mathfrak{F}} \diamond p)(\delta, t) u(\delta) = \sum_{\delta=0}^{t-1} \sum_{i_{\delta}, \dots, i_{t-1} \in \mathbb{I}_{0,n_p}} \theta_{i_{t-1}, i_{t-2}, \dots, i_{\delta}, \mathfrak{F}}(i_{\delta+1} \cdots i_{t-1}) p_{i_{t-1}}(t) \cdots p_{i_{\delta}}(\delta) u(\delta). \quad (5.4)$$

In order to estimate the sub-Markov parameters of  $\mathfrak{F}$  from  $\{u(t), p(t), y(t)\}_{t=0}^N$ , equation (5.4) has to be solved with the sub-Markov parameters  $\theta_{i_{t-1}, i_{t-2}, \dots, i_{\delta}, \mathfrak{F}}(i_{\delta+1} \cdots i_{t-1})$  being treated as indeterminate to be found.

There are two ways of doing so. The first one is to assume that for some  $n_b$ , if

$$(h_{\mathfrak{F}} \diamond p)(\delta, t) = 0, \text{ if } t - \delta > n_b$$

or, which is equivalent, that

$$\theta_{i_{t-1}, i_{t-2}, \dots, i_{\delta}, \mathfrak{F}}(i_{\delta+1} \cdots i_{t-1}) = 0, \text{ if } t - \delta > n_b.$$

If  $\mathfrak{F}$  has a realization by a stable LPV-SSA of the form (2.1), then for large enough  $n_b$ , for  $t - \delta \geq n_b$ ,  $\theta_{i_{t-1}, i_{t-2}, \dots, i_{\delta}, \mathfrak{F}}(i_{\delta+1} \cdots i_{t-1})$  is going to be close to zero [66], so this assumption will approximately be true. In this case, (5.4) can be written as

$$y(t) = \sum_{\delta=1}^{n_b} \sum_{i_{t-\delta}, \dots, i_{t-1} \in \mathbb{I}_{0,n_p}} \theta_{i_{t-1}, i_{t-2}, \dots, i_{t-\delta}, \mathfrak{F}}(i_{t-\delta+1} \cdots i_{t-1}) p_{i_{t-1}}(t) \cdots p_{i_{t-\delta}}(t - \delta) u(t - \delta),$$

for all  $t \geq n_b$ . That is, we can formulate the following linear regression problem

$$Y_N = \Phi_N \theta_N \quad (5.5)$$

where

$$Y_N = [y^T(n_b + 1) \quad y^T(n_b + 2) \quad \dots \quad y^T(N)]^T$$

and  $\Phi_N$  is a suitable matrix constructed from the products  $p_{i_{t-1}}(t) \cdots p_{i_{t-\delta}}(t - \delta) u(t - \delta)$ , and  $\theta_N$  is the vector constructed from the entries of  $\theta_{i_{t-1}, i_{t-2}, \dots, i_{t-\delta}, \mathfrak{F}}(i_{t-\delta+1} \cdots i_{t-1})$ ,  $n_b \leq t \leq N$ ,  $\delta = 1, \dots, n_b$ ,  $i_{t-\delta}, \dots, i_{t-1} \in \mathbb{I}_{0,n_p}$ , see [66] for more details. Since  $Y_N$  and  $\Phi_N$  are known, the equation (5.5) can be solved in the least squares sense, leading to the following system identification algorithm by combining the solution of (5.5) with Algorithm 1 or Algorithm 2. Below we will present only the SISO case in order to avoid excessive notation, and only the version using Algorithm 2 as Algorithm 1 can be viewed as a special case of Algorithm 2:

---

#### Algorithm 6 Deterministic subspace identification algorithm based on finite impulse response

---

- 1: Solve (5.5) and denote the entry of the solution  $\theta_N$  which corresponds to  $\theta_{i_{t-1}, i_{t-2}, \dots, i_{t-\delta}, \mathfrak{F}}(i_{t-\delta+1} \cdots i_{t-1})$ , by  $\theta_{i_{t-1}, i_{t-2}, \dots, i_{t-\delta}, \mathfrak{F}}^N(i_{t-\delta+1} \cdots i_{t-1})$ , for all  $i_{t-\delta}, \dots, i_{t-1} \in \mathbb{I}_{0,n_p}$ ,  $n_b \leq t \leq N$  for all  $\delta = 1, \dots, n_b$ .
  - 2: Choose nice selection  $\alpha, \beta \subseteq \{v \in \mathbb{I}_{0,n_p}^* \mid |v| \leq n\} \times \mathbb{I}_{0,n_p}$  and construct the matrices  $\mathcal{H}_{\mathfrak{F},\alpha,\beta}^N$ ,  $\mathcal{H}_{q,\mathfrak{F},\alpha,\beta}^N$ ,  $\mathcal{H}_{\mathfrak{F},\alpha,q}^N$  and  $\mathcal{H}_{q,\mathfrak{F},\beta}^N$  by replacing in (2.20) – (2.23) every occurrence of  $\theta_{r_1, r_2, \mathfrak{F}}(h)$ ,  $r_1, r_2 \in \mathbb{I}_{0,n_p}$ ,  $h \in \mathbb{I}_{0,n_p}^*$  by  $\theta_{r_1, r_2, \mathfrak{F}}^N(h)$ .
  - 3: Let  $\Sigma_N$  be the LPV-SSA returned by Algorithm 2 when applied to  $\mathcal{H}_{\mathfrak{F},\alpha,\beta}^N$ ,  $\mathcal{H}_{q,\mathfrak{F},\alpha,\beta}^N$ ,  $\mathcal{H}_{\mathfrak{F},\alpha,q}^N$  and  $\mathcal{H}_{q,\mathfrak{F},\beta}^N$ .
  - 4: **return**  $\Sigma_N$
-



**Lemma 5.1** (Correctness of Algorithm 6). *Assume that  $\text{rank}H_{\mathfrak{F},\alpha,\beta}$  equals the dimension of a minimal LPV-SSA realization of  $\mathfrak{F}$ , and assume that  $\theta_{i_k,i_0\mathfrak{F}}(i_1 \cdots i_{k-1}) = 0$  for any  $i_0, \dots, i_k \in \mathbb{I}_{0,n_p}$ ,  $k \geq n_b$ , and that the matrix  $\Phi_N$  from (5.5) is full column rank. Then  $\Sigma_N$  returned by Algorithm 6 is a realization of  $\mathfrak{F}$ .*

That is, under suitable conditions Algorithm 6 solves Problem 5.2. Moreover, it solves it for a finite number of data points, i.e., for large enough  $N$  the estimate  $\Sigma_N$  is an exact realization of  $\mathfrak{F}$ , not an approximate one. As it is to be expected, the corresponding conditions are quite strong.

More precisely, the condition that  $\text{rank}H_{\mathfrak{F},\alpha,\beta}$  equals the dimension of a minimal LPV-SSA realization of  $\mathfrak{F}$  is not at all restrictive. In fact, it is known that if  $\mathfrak{F}$  has a minimal LPV-SSA realization of dimension  $n$ , then there exists selections  $\alpha, \beta \subseteq \mathbb{I}_{0,n_p} \times \{v \in \mathbb{I}_{0,n_p}^* \mid |v| \leq n-1\}$  such that  $\text{rank}H_{\mathfrak{F},\alpha,\beta} = n$ . That is, once the likely order of a minimal realization of  $\mathfrak{F}$  is fixed, one can always find suitable selections. In fact, the number of such selections is finite, so suitable selections can always be found by exhaustive search, although the runtime complexity of this is high. In practice, more intelligent search techniques are preferable.

The second requirement is that i

$$\theta_{i_k,i_0\mathfrak{F}}(i_1 \cdots i_{k-1}) = 0$$

for any  $i_0, \dots, i_k \in \mathbb{I}_{0,n_p}$ ,  $k \geq n_b$ . This is more problematic, since it means that long enough product of the system matrices of any minimal LPV-SSA representation of  $\mathfrak{F}$  must be zero. However, if  $\mathfrak{F}$  has a quadratically stable LPV-SSA representation, then this condition will hold approximately. Of course, in this case Lemma 5.1 no longer holds, but an asymptotic version of Lemma 5.1 can probably be formulated. The condition that  $\Phi_N$  must be of full column rank is reminiscent of persistence of excitation condition, it remains a topic of future research to investigate the choice of signals  $u, p$  which satisfy this condition. Algorithm 6 was tested on numerical examples, and it performs well, especially when combined with parametric system identification algorithms, for more details see [66].

Another option is to use the statistical properties of the signals  $u, p, y$ . More precisely, assume that  $\boldsymbol{\mu}, \mathbf{u}, \mathbf{y}$  are stochastic processes such that  $\mathbf{y}(t) = \mathcal{F}(\boldsymbol{\mu}, \mathbf{u})(t)$  for all  $t \in \mathbb{N}$  and assume that  $p, u, y$  are sample paths  $\boldsymbol{\mu}, \mathbf{u}, \mathbf{y}$ . From (5.4) it then follows that

$$\mathbf{y}(t) = \sum_{\delta=0}^{t-1} \sum_{i_\delta, \dots, i_t \in \mathbb{I}_{0,n_p}} \theta_{i_t, i_\delta \mathfrak{F}}(i_{\delta+1} \cdots i_{t-1}) \boldsymbol{\mu}_{i_t}(t) \cdots \boldsymbol{\mu}_{i_\delta}(\delta) \mathbf{u}(\delta), \quad (5.6)$$

where  $\boldsymbol{\mu}_q(t)$  denotes the  $q$ th component of  $\boldsymbol{\mu}(t)$  if  $q = 1, \dots, n_p$  and  $\boldsymbol{\mu}_0(t) = 1$  otherwise. Assume now the following:

- Assumption 1.**
1.  $\mathbf{u}(t)$  are identically distributed and independent and zero mean with covariance  $Q_{\mathbf{u}} = E[\mathbf{u}(t)\mathbf{u}^T(t)] > 0$ , and
  2. the random variables  $\boldsymbol{\mu}(t)$  are also identically distributed and independent and that  $E[\boldsymbol{\mu}_j(t)\boldsymbol{\mu}_i(t)] = 0$  for  $i \neq j$ ,  $E[\boldsymbol{\mu}_j^2(t)] = \boldsymbol{\pi}_j$ ,  $i, j \in \mathbb{I}_{0,n_p}$ , and
  3. the  $\sigma$ -algebras generated by the random variables  $\{\boldsymbol{\mu}(t)\}_{t=0}^\infty$  and  $\{\mathbf{u}(t)\}_{t=0}^\infty$  are independent.

That is,  $\mathbf{u}$  is white noise, and  $\boldsymbol{\mu}$  is either zero mean white noise or it is a binary noise, but independent of  $\mathbf{u}$ . It then follows that for any  $j_0, \dots, j_t \in \mathbb{I}_{0,n_p}$ ,

$$E[\mathbf{y}(t)\boldsymbol{\mu}_{j_t}(t) \cdots \boldsymbol{\mu}_{j_0}(0)\mathbf{u}^T(0)] = \sum_{\delta=0}^{t-1} \sum_{i_\delta, \dots, i_t \in \mathbb{I}_{0,n_p}} \theta_{i_t, i_\delta \mathfrak{F}}(i_{\delta+1} \cdots i_{t-1}) E[\boldsymbol{\mu}_{i_t}(t) \cdots \boldsymbol{\mu}_{i_\delta}(\delta)\boldsymbol{\mu}_{j_t}(t) \cdots \boldsymbol{\mu}_{j_0}(0)\mathbf{u}(\delta)\mathbf{u}^T(0)].$$

Notice that

$$E[\boldsymbol{\mu}_{i_t}(t) \cdots \boldsymbol{\mu}_{i_\delta}(\delta)\boldsymbol{\mu}_{j_t}(t) \cdots \boldsymbol{\mu}_{j_0}(0)\mathbf{u}(\delta)\mathbf{u}^T(0)] = E[\boldsymbol{\mu}_{i_t}(t) \cdots \boldsymbol{\mu}_{i_\delta}(\delta)\boldsymbol{\mu}_{j_t}(t) \cdots \boldsymbol{\mu}_{j_0}(0)]E[\mathbf{u}(\delta)\mathbf{u}^T(0)]$$

## 5.1. SYSTEM IDENTIFICATION OF DETERMINISTIC LPV-SS SYSTEMS

and

$$E[\mathbf{u}(\delta)\mathbf{u}^T(0)] = \begin{cases} 0 & \text{if } \delta \neq 0 \\ \mathbf{Q}_u & \text{if } \delta = 0 \end{cases}$$

Moreover,

$$E[\boldsymbol{\mu}_{i_t}(t) \cdots \boldsymbol{\mu}_{i_\delta}(\delta) \boldsymbol{\mu}_{j_t}(t) \cdots \boldsymbol{\mu}_{j_0}(0)] = \prod_{k=0}^t E[\boldsymbol{\mu}_{i_k}(k) \boldsymbol{\mu}_{j_k}(k)] = 0 \quad \text{if } i_0 \cdots, i_t \neq j_0 \cdots j_t.$$

it follows that under Assumption 1,

$$E[\mathbf{y}(t) \boldsymbol{\mu}_{j_t}(t) \cdots \boldsymbol{\mu}_{j_0}(0) \mathbf{u}^T(0)] = \boldsymbol{\theta}_{j_t, j_0, \mathfrak{F}}(j_1 \cdots j_{t-1}) \boldsymbol{\pi}_{j_0} \cdots \boldsymbol{\pi}_{j_t} \mathbf{Q}_u,$$

and therefore

$$\boldsymbol{\theta}_{j_t, j_0, \mathfrak{F}}(j_1 \cdots j_{t-1}) = \frac{1}{\boldsymbol{\pi}_{j_0} \cdots \boldsymbol{\pi}_{j_t}} E[\mathbf{y}(t) \boldsymbol{\mu}_{j_t}(t) \cdots \boldsymbol{\mu}_{j_0}(0) \mathbf{u}^T(0)] \mathbf{Q}_u^{-1}. \quad (5.7)$$

That is (5.7) allows us to express the sub-Markov parameters of  $\mathfrak{F}$  through the expected values

$$E[\mathbf{y}(t) \boldsymbol{\mu}_{j_t}(t) \cdots \boldsymbol{\mu}_{j_0}(0) \mathbf{u}^T(0)].$$

The latter can be estimated from the signals  $(u, p, y)$  using the empirical covariances

$$\hat{E}_{j_0 \cdots j_t, N} = \frac{1}{N-t-1} \sum_{k=t}^N y(k) p_{j_t}(k) \cdots p_{j_0}(k-t) u^T(k-t),$$

arriving at

$$\boldsymbol{\theta}_{j_t, j_0, \mathfrak{F}}^N(j_1 \cdots j_{t-1}) = \frac{1}{\boldsymbol{\pi}_{j_0} \cdots \boldsymbol{\pi}_{j_t}} \frac{1}{N-t-1} \sum_{k=t}^N y(k) p_{j_t}(k) \cdots p_{j_0}(k-t) u^T(k-t) \mathbf{Q}_u^{-1}. \quad (5.8)$$

We can then propose the following system identification algorithm based on Algorithm 1 or Algorithm 2. We will present only the SISO case in order to avoid excessive notation, and only the version using Algorithm 2 as Algorithm 1 can be viewed as a special case of Algorithm 2.

---

### Algorithm 7 Deterministic subspace identification algorithm based on covariances

---

- 1: Choose selections  $\alpha, \beta$  as defined in (2.19).
  - 2: Construct the matrices  $\mathcal{H}_{f, \alpha, \beta}^N$ ,  $\mathcal{H}_{q, \mathfrak{F}, \alpha, \beta}^N$ ,  $\mathcal{H}_{\mathfrak{F}, \alpha, q}^N$  and  $\mathcal{H}_{q, \mathfrak{F}, \beta}^N$  by replacing in (2.20) – (2.23) every occurrence of  $\boldsymbol{\theta}_{r_1, r_2, \mathfrak{F}}(h)$ ,  $r_1, r_2 \in \mathbb{I}_{0, n_p}$ ,  $h \in \mathbb{I}_{0, n_p}^*$  by  $\boldsymbol{\theta}_{r_1, r_2, \mathfrak{F}}^N(h)$  calculated from (5.8). Note that only a finite number of  $\boldsymbol{\theta}_{r_1, r_2, \mathfrak{F}}^N(h)$  has to be calculated.
  - 3: Let  $\Sigma_N$  be the LPV-SSA returned by Algorithm 2 when applied to  $\mathcal{H}_{\mathfrak{F}, \alpha, \beta}^N$ ,  $\mathcal{H}_{q, \mathfrak{F}, \alpha, \beta}^N$ ,  $\mathcal{H}_{\mathfrak{F}, \alpha, q}^N$  and  $\mathcal{H}_{q, \mathfrak{F}, \beta}^N$ .
  - 4: **return**  $\Sigma_N$
- 

We can state the following results on consistency of Algorithm 7.

**Lemma 5.2** (Consistency of Algorithm 7, adapted from [66]). *If rank  $H_{\mathfrak{F}, \alpha, \beta}$  equals the dimension  $n_m$  of a minimal LPV-SSA realization of  $\mathfrak{F}$ , and the selections  $\alpha$  and  $\beta$  contain exactly  $n_m$  elements each, and in Algorithm 2 the matrix  $O_{n_m}$  is chosen as  $O_{n_m} = \mathcal{H}_{\mathfrak{F}, \alpha, \beta}^N$  and  $R_{n_m}$  is the identity matrix, and*

$$\lim_{N \rightarrow \infty} \frac{1}{N-t+1} \sum_{k=t}^N y(k) p_{j_t}(k) \cdots p_{j_0}(k-t) u(k-t) = E[\mathbf{y}(t) \boldsymbol{\mu}_{j_t}(t) \cdots \boldsymbol{\mu}_{j_0}(0) \mathbf{u}^T(0)] \quad (5.9)$$

for all  $j_0, \dots, j_t \in \mathbb{I}_{0, n_p}$ ,  $t \geq 1$ , then the LPV-SSA  $\Sigma_N = (\mathbb{P}, \{A_i^N, B_i^N, C_i^N\}_{i=0}^{n_p})$  returned by Algorithm 7 has the following property:  $\lim_{N \rightarrow \infty} (A_i^N, B_i^N, C_i^N) = (A_i, B_i, C_i)$ ,  $i \in \mathbb{I}_{0, n_p}$  and the LPV-SSA  $\Sigma_* = (\mathbb{P}, \{A_i, B_i, C_i\}_{i=0}^{n_p})$  is a realization of  $\mathfrak{F}$ .

The condition that  $\text{rank}H_{\mathfrak{F},\alpha,\beta}$  equals the dimension of a minimal LPV-SSA realization of  $\mathfrak{F}$  is not very restrictive: in fact, we know that if  $\mathfrak{F}$  has an LPV-SSA realization of dimension  $n$  then there exist nice selection  $\alpha, \beta \subseteq \mathbb{I}_{0,n_p} \times (\mathbb{I}_{0,n_p})^{\leq n}$  of cardinality  $n$  for which this condition holds.

If  $(\mathbf{u}, \boldsymbol{\mu}, \mathbf{y})$  are jointly ergodic, then (5.9) holds with probability one i.e., for almost all choices of the sample paths  $(u, p, y)$  of  $(\mathbf{u}, \boldsymbol{\mu}, \mathbf{y})$ . It is not clear under which conditions  $(\mathbf{u}, \boldsymbol{\mu}, \mathbf{y})$  are jointly ergodic. However, in [206] it was shown that if

- $\boldsymbol{\mu}(t)$  is chosen to be binary white noise, i.e.  $\mu_i(t) = \xi(\sigma(t) = i)$ ,  $i = 1, \dots, n_p$  and  $\sigma(t)$  is an i.i.d. process taking values in  $\{1, \dots, n_p\}$  and  $P(\sigma(t) = i) = \frac{1}{n_p}$  for all  $i = 1, \dots, n_p$ , and
- $\mathbf{u}$  and  $\boldsymbol{\mu}$  satisfy Part 1 and Part 3 of Assumption 1, and
- $(u, p)$  is a sample path of  $(\mathbf{u}, \boldsymbol{\mu})$ ,

then for  $y = \mathfrak{F}(u, p)$  condition (5.9) holds, provided  $\mathfrak{F}$  can be realized by a quadratically stable LPV-SSA.

Algorithm 7 was tested on numerical examples, Algorithm 6 and it performs well, especially when combined with parametric system identification algorithms, for more details see [66]

**Remark 5.1** (Persistence of excitation). *Note that the conditions of Lemma 5.1 and Lemma 5.2 can be viewed as conditions for persistence of excitation of the scheduling signal and the inputs. In case of Lemma 5.1 the condition that  $\Phi_N$  is of full column rank is a persistence of excitation condition which depends only on the choice of the scheduling signal and the input signal. For Lemma 5.2, the condition that (5.9) holds can also be viewed as a persistence of excitation condition, as it depends on the measured data. In fact, as it was noted above, [206] provides sufficient conditions on the scheduling signal and the input signal for (5.9) to hold.*

### Role of realization theory

The subspace identification algorithms discussed in this section rely on realization theory, more precisely, on Ho-Kalman realization algorithms for LPV-SSAs. In fact, they are nothing but an application of the latter algorithms to Hankel-matrices, entries of which are estimates of the corresponding sub-Markov parameters. The consistency of these algorithms is a consequence of realization theory of LPV-SSAs.

### 5.1.2 Identifiability of LPV-SSA

In this section, we present necessary and sufficient conditions for identifiability. These conditions rely on the properties of minimal LPV-SSA realizations, in particular, on the fact the minimal LPV-SSA realizations of the same input-output map are isomorphic. The results of this section were published in [5].

As the first step, we will restrict attention to parameterizations which are so called structurally minimal.

**Definition 5.3** (Structural minimality). *A LPV-SSA parameterization  $\Pi$  of the form (5.1) is called structurally minimal if, for all  $\theta \in \Theta$ , the LPV-SSA  $\Sigma_{\Pi}(\theta)$  is minimal.*

The reason why we restrict our attention to structurally minimal parameterizations is as follows. First, any parameterization can be transformed into a structurally minimal one by minimizing the corresponding LPV-SSA. Second, for structurally minimal parameterizations, it is easy to find conditions for identifiability.

**Theorem 5.1** (Identifiability of LPV-SSAs [5]). *Let  $\Pi$  be a structurally minimal LPV-SSA parameterization of the form (5.1). Then  $\Pi$  is structurally identifiable, if and only if both conditions below are satisfied*

- (i) *the map  $\Pi$  is injective,*
- (ii) *for every  $\theta_1, \theta_2 \in \Theta$ , if  $\theta_1 \neq \theta_2$ , then there exists no isomorphism from  $\Sigma_{\Pi}(\theta_1)$  to  $\Sigma_{\Pi}(\theta_2)$ .*

## 5.1. SYSTEM IDENTIFICATION OF DETERMINISTIC LPV-SS SYSTEMS

That is, **(i)** and **(ii)** are necessary and sufficient conditions for structural identifiability of  $\Pi$ . The proof of Theorem 5.1 relies on the fact that minimal LPV-SSA realizations of the same input-output map are isomorphic. An alternative formulation of Theorem 5.1, which leads to a computationally efficient condition, can be derived as follows. Let  $\Pi$  be a structurally minimal LPV-SSA parameterization of the form (5.1). Let us introduce the following notation.

**Notation 5.1** ( $\text{vec}(\mathcal{T})$ ). For a matrix  $\mathcal{T} \in \mathbb{R}^{n \times m}$ , denote by  $\text{vec}(\mathcal{T})$  the vector obtained by stacking up the elements of  $\mathcal{T}$  column-wise, i.e.,

$$\text{vec}(\mathcal{T}) = [\mathcal{T}_{11}, \dots, \mathcal{T}_{n1}, \dots, \mathcal{T}_{1n}, \dots, \mathcal{T}_{nm}]^T.$$

**Notation 5.2** ( $\text{vec}^{-1}(X)$ ). For a vector  $X \in \mathbb{R}^{nm}$ , we denote by  $\text{vec}^{-1}(X)$  the unique matrix  $\mathcal{T} \in \mathbb{R}^{n \times m}$  such that  $X = \text{vec}(\mathcal{T})$ .

With the notation above, we identify  $\Pi(\theta) = \{A_i(\theta), B_i(\theta), C_i(\theta)\}_{i=0}^{n_p}$  with the point

$$\begin{bmatrix} \text{vec} \left[ \begin{matrix} A_0^T(\theta) \\ \vdots \\ A_{n_p}^T(\theta) \end{matrix} \right]^T \\ \text{vec} \left[ \begin{matrix} B_0^T(\theta) \\ \vdots \\ B_{n_p}^T(\theta) \end{matrix} \right]^T \\ \text{vec} \left[ \begin{matrix} C_0^T(\theta) \\ \vdots \\ C_{n_p}^T(\theta) \end{matrix} \right]^T \end{bmatrix} \in \mathbb{R}^N.$$

where  $\text{vec}(X)$  is a column-wise vector representation of the matrix  $X$ . Using this correspondence, the parameterization  $\Pi$  will be called *structurally continuous* (*structurally smooth*, *respectively*) if the map  $\Pi : \Theta \rightarrow \mathbb{R}^N$  is continuous (smooth, *respectively*). Furthermore, we will need the following notation.

**Notation 5.3** ( $\widetilde{GL}(n_x)$ ). Define the set  $\widetilde{GL}(n_x)$  as follows:  $\widetilde{GL}(n_x) = \{X \in \mathbb{R}^{n_x \times n_x} \mid \exists \mathcal{T} \in GL(n_x) : X = \text{vec}(\mathcal{T})\}$ .

Note that if  $\mathcal{T} \in GL(n_x)$ , then  $\text{vec}(\mathcal{T}^{-1})$  is well defined and it is just the vector representation of the inverse matrix of  $\mathcal{T}$ . This should not be confused with  $\text{vec}^{-1}(X)$  for some  $X \in \widetilde{GL}(n_x)$ . In the latter case,  $\text{vec}^{-1}(X)$  is the invertible matrix whose vector representation is  $X$ . That is,  $\text{vec}(\mathcal{T}^{-1})$  is a vector and  $\text{vec}^{-1}(X)$  is a matrix.

Define the map

$$F : GL(n_x) \times \Theta \rightarrow \mathbb{R}^N$$

$$(\mathcal{T}, \theta) \mapsto \begin{bmatrix} \text{vec}(\mathcal{T}A_0(\theta)\mathcal{T}^{-1}) \\ \vdots \\ \text{vec}(\mathcal{T}A_{n_p}(\theta)\mathcal{T}^{-1}) \\ \text{vec}(\mathcal{T}B_0(\theta)) \\ \vdots \\ \text{vec}(\mathcal{T}B_{n_p}(\theta)) \\ \text{vec}(C_0(\theta)\mathcal{T}^{-1}) \\ \vdots \\ \text{vec}(C_{n_p}(\theta)\mathcal{T}^{-1}) \end{bmatrix}, \quad (5.10)$$

where  $\Sigma_\Pi(\theta) = (\mathbb{P}, \{A_i(\theta), B_i(\theta), C_i(\theta)\}_{i=0}^{n_p})$  and  $N = n_x(n_x + n_y + n_u)(n_p + 1)$ . Note that for all  $\mathcal{T} \in GL(n_x)$ ,  $\mathcal{T}$  can be seen as an isomorphism from  $\Sigma_\Pi(\theta)$  to  $F(\mathcal{T}, \theta)$ .

**Theorem 5.2** (Reformulation of identifiability). *With the assumptions of Theorem 5.1,  $\Pi$  is structurally identifiable if and only if  $F$  is injective.*

The reason for reformulating Theorem 5.1 into Theorem 5.2 and introducing the map  $F$  is that it allows for a systematic procedure to verify structural and local structural identifiability of LPV-SS parameterizations in a computationally effective way. In order to present the details, we need the following definition.

**Definition 5.4** (Local structural identifiability). *Let us assume that  $\Theta$  is an open subset of  $\mathbb{R}^{n_\theta}$  and let  $\Pi$  be a LPV-SSA parameterization of the form (5.1). We say that  $\Pi$  is locally structurally identifiable at  $\theta_0$ , if there exists an open neighborhood  $W \subseteq \Theta$  of  $\theta_0$  such that the LPV-SSA parameterization  $\Pi|_W : W \ni \theta \mapsto \Pi(\theta)$  is structurally identifiable.*

Local structural identifiability is of prime interest because it is a necessary condition for the local convergence of many identification algorithms using non-linear optimization techniques (see [291, 36, 292]).

**Definition 5.5** (Local injectivity). *Assume that  $F$  is a map of the form (5.10). We say the  $F$  is locally injective at  $(I, \theta_0)$ , if there exist open sets  $S \subseteq GL(n_x)$ ,  $W \subseteq \Theta$  such that  $I \in S, \theta_0 \in W$  and the restriction  $F|_{S \times W}$  of  $F$  to  $S \times W$  is injective.*

Before presenting the characterization of local structural identifiability, we would like to investigate the partial derivatives of  $F$ . Notice that the set  $\widetilde{GL}(n_x)$  is an open subset of  $\mathbb{R}^{n_x^2}$ . Note that the correspondence  $GL(n_x) \ni X \mapsto \text{vec}(X)$  is one-to-one. Then  $F$  can be identified with the map

$$\tilde{F} : \widetilde{GL}(n_x) \times \Theta \ni (X, \theta) \mapsto F(\text{vec}^{-1}(X), \theta).$$

If  $\Pi$  is a smooth parameterization, then  $\tilde{F}$  is a smooth function defined on an open subset of  $\mathbb{R}^{n_x^2} \times \Theta$ , and then it makes sense to investigate the partial derivatives of  $\tilde{F}$ . In order to avoid excessive notation, we will tacitly identify  $F$  with  $\tilde{F}$ , and denote the Jacobian matrices  $\frac{\partial \tilde{F}}{\partial X}(X, \theta)$  and  $\frac{\partial \tilde{F}}{\partial \theta}(X, \theta)$  by  $\frac{\partial F}{\partial \mathcal{F}}(\text{vec}(X), \theta)$ ,  $\frac{\partial F}{\partial \theta}(\text{vec}(X), \theta)$ , respectively.

**Theorem 5.3** (Local identifiability). *Let  $\Pi$  be a structurally minimal LPV-SSA parameterization of the form (5.1):*

1. *Assume that  $\Pi$  is a continuous parameterization. The parameterization  $\Pi$  is locally structurally identifiable at  $\theta_0 \in \Theta$ , if and only if  $F$ , defined in (5.10), is locally injective at  $(I, \theta_0)$ .*
2. *Assume  $\Pi$  is a smooth parameterization. If the matrix*

$$M_F(\theta_0) = \begin{bmatrix} \frac{\partial F}{\partial \mathcal{F}}(I, \theta_0) & \frac{\partial F}{\partial \theta}(I, \theta_0) \end{bmatrix}, \quad (5.11)$$

*is full column rank, then  $\Pi$  is locally structurally identifiable at  $\theta_0$ .*

Note that

$$\frac{\partial F}{\partial \mathcal{F}}(I, \theta) = \begin{bmatrix} \mathcal{L}(A_0(\theta)) \\ \vdots \\ \mathcal{L}(A_{n_p}(\theta)) \\ B_0(\theta)^T \otimes I \\ \vdots \\ B_{n_p}(\theta)^T \otimes I \\ -I \otimes C_0(\theta) \\ \vdots \\ -I \otimes C_{n_p}(\theta) \end{bmatrix}, \quad \frac{\partial F}{\partial \theta}(I, \theta) = \begin{bmatrix} \frac{\partial \text{vec}(A_0(\theta))}{\partial \theta} \\ \vdots \\ \frac{\partial \text{vec}(A_{n_p}(\theta))}{\partial \theta} \\ \frac{\partial \text{vec}(B_0(\theta))}{\partial \theta} \\ \vdots \\ \frac{\partial \text{vec}(B_{n_p}(\theta))}{\partial \theta} \\ \frac{\partial \text{vec}(C_0(\theta))}{\partial \theta} \\ \vdots \\ \frac{\partial \text{vec}(C_{n_p}(\theta))}{\partial \theta} \end{bmatrix}$$

Hence, rank of the matrix  $M_F(\theta_0)$  in Eq. (5.11) can be checked numerically, and thus the condition of the theorem is computationally effective. Intuitively, the first statement of Theorem 5.3 stems from the observation

## 5.2. STOCHASTIC SUBSPACE IDENTIFICATION ALGORITHM

that local injectivity of  $F$  means that if the parameterization is restricted to a neighborhood of  $\theta_0$ , then no two LPV-SSAs corresponding to distinct parameter values can be isomorphic. The second statement of Theorem 5.3 relies on the observation that the condition that  $M_F(\theta_0)$  is full rank guarantees that  $F$  is locally injective at  $(I, \theta_0)$ .

While Theorem 5.3 gives computationally effective sufficient conditions for local structural identifiability, it does not allow us to check structural identifiability on the whole parameter space. The latter is important because local structural identifiability guarantees only that the parameter estimation problem is well-posed on a neighborhood of the nominal value. If the parameter estimation problem is formulated as an optimization problem of a cost function, then local structural identifiability may give us a hope that the optimization algorithm will converge to the true solution if it is initialized properly, *i.e.*, if it starts not far from the true value. However, local structural identifiability does not tell us anything about the global behavior of optimization algorithms: the algorithms may converge to one of several solutions, based on the initial values (which are thus required *a priori*) and the measured data. Thus, it is desired to generalize the condition of Theorem 5.3 to structural identifiability. This is done below, where we present computational effective characterization of global identifiability.

**Theorem 5.4** (Computationally effective conditions for global identifiability [5]). *Let  $\Pi$  be a structurally minimal LPV-SSA parametrization of the form (5.1), and assume that  $\Theta = \mathbb{R}^{n_\theta}$  and that  $\Pi$ , viewed as a map  $\Pi : \Theta \rightarrow \mathbb{R}^N$ , is affine. Then  $\Pi$  is structurally identifiable if*

$$\forall \theta, \tilde{\theta} \in \Theta : \det [Y^\top(\theta, \tilde{\theta})Y(\theta, \tilde{\theta})] \neq 0, \quad (5.12)$$

where

$$Y(\theta, \tilde{\theta}) = \begin{bmatrix} Z(\theta, \tilde{\theta}) & 0 & \frac{\partial F}{\partial \theta}(I, \theta) \\ 0 & Z(\tilde{\theta}, \theta) & -\frac{\partial F}{\partial \theta}(I, \theta) \end{bmatrix}, \quad (5.13)$$

and

$$Z(\theta, \tilde{\theta}) = \begin{bmatrix} A_0(\theta)^T \otimes I - I \otimes A_0(\tilde{\theta}) \\ \vdots \\ A_{n_p}(\theta)^T \otimes I - I \otimes A_{n_p}(\tilde{\theta}) \\ B_0(\theta)^T \otimes I \\ \vdots \\ B_{n_p}(\theta)^T \otimes I \\ -I \otimes C_0(\tilde{\theta}) \\ \vdots \\ -I \otimes C_{n_p}(\tilde{\theta}) \end{bmatrix}. \quad (5.14)$$

### Role of realization theory

Realization theory, especially the results on minimality allowed us to propose conditions for global and local structural identifiability. More precisely, the proposed conditions for identifiability all rely on the fact that input-output equivalent minimal LPV-SSA are isomorphic. The identifiability conditions provide conditions when there are no isomorphic elements in the parametrization.

## 5.2 Stochastic subspace identification algorithm

Since any model is wrong, it is a standard practice in system identification to aim at estimating the parameters of state-space representations which a noise term. This noise term is supposed to capture modeling errors. Historically, the noise term was assumed to be a stochastic process, in fact, it was assumed to be a white noise process with a certain variance. Traditionally, system identification algorithms also aimed at estimating this variance. For linear systems, this was useful for several reasons. For linear models the variance of the noise gives an indication

of the error between the output predicted by the model and the actual one. Moreover, stochastic linear systems with known noise variances are useful for stochastic optimal control.

For non-linear systems, the rationale for using stochastic noise terms is less clear. For example, control synthesis methods for LPV systems rely on robust control, in which the noise is assumed to be deterministic of bounded energy. Nevertheless, the system identification literature for nonlinear systems, including LPV systems, tends to use the assumption that the noise is stochastic. Reconciling this assumption with the requirements for robust control is a topic of future research.

In this section we will discuss system identification algorithms for stochastic LPV systems. More precisely, in Subsection 5.2.1 consider first the system identification problem for GBSs, generalized bilinear systems (GBSs) introduced in Chapter 3. The latter problem is motivated by the fact that identification of autonomous stochastic LPV-SSA systems can be viewed as a special case of identification of GBSs. In Subsection 5.2.2 we discuss the system identification problem for stochastic LPV-SSA with inputs, and we briefly discuss how to decompose that problem into identifying deterministic LPV-SSAs and identifying autonomous stochastic LPV-SSAs.

### 5.2.1 Stochastic subspace identification of GBSs

In this section we will consider subspace identification of stationary GBSs of the form (3.2). This section is partially based on [172]. For the sake of simplicity we consider inputs of the following two forms:

**Assumption 2.** We assume that  $Q = \{1, \dots, D\}$  and  $\boldsymbol{\mu}(t) = (\boldsymbol{\mu}_1(t), \dots, \boldsymbol{\mu}_D(t))^T$  and either

- **White noise scheduling signal:**  $\boldsymbol{\mu}_1 = 1$ ,  $\mathbf{p}(\boldsymbol{\mu}_1) = 1$ ,  $E[\boldsymbol{\mu}_i(t)] = 0$ ,  $E[\boldsymbol{\mu}_i(t)\boldsymbol{\mu}_j(t)] = 0$ ,  $i, j \neq 0$ ,  $E[\boldsymbol{\mu}_i^2(t)] = \mathbf{p}(\boldsymbol{\mu}_i) > 0$ ,  $i, j = 2, \dots, D$ , or
- **Binary white noise scheduling signal:** there exists a process  $\boldsymbol{\theta}(t)$  which is i.i.d process and which takes its values from a finite set  $\{1, \dots, n_p\}$ , and  $\boldsymbol{\mu}_q(t) = \boldsymbol{\chi}(\boldsymbol{\theta}(t) = q)$  for all  $q = 1, \dots, n_p$ ,  $t \in \mathbb{Z}$ , and  $\mathbf{p}(\boldsymbol{\mu}_i) = P(\boldsymbol{\theta}(t) = i)$ .

The former case also corresponds to bilinear systems with a white noise input or stochastic LPV-SSAs, for the latter see Subsection 5.2.2 for a detailed explanation, while the latter corresponds to stochastic switched systems. Note that in both cases,  $\{\boldsymbol{\mu}_q\}_{q \in Q}$  is an admissible input in the sense defined in Section 3.4, Chapter 3.

For GBSs we can formulate an identification problem. Intuitively, what we would like to do is to estimate the matrices of a GBS realization of  $\mathbf{y}$  based on finite number of samples of  $\mathbf{y}$  and the input process  $\boldsymbol{\mu}$ , under the hypothesis that  $\boldsymbol{\mu}$  satisfies Assumption 2 and  $\mathbf{y}$  has a realization by a stationary GBS driven by the scheduling signal  $\boldsymbol{\mu}$ .

**Problem 5.4** (System identification for GBS). Let  $\boldsymbol{\mu}$  be a process satisfying Assumption 2 and let  $\mathbf{y}$  be a process such that  $\mathbf{y}$  has a realization by a stationary GBS with input process  $\{\boldsymbol{\mu}_q\}_{q \in Q}$ . Assume that  $(y, p) \in (\mathbb{R}^p)^{\mathbb{N}} \times (\mathbb{R}^D)^{\mathbb{N}}$ , is a sample path of  $(\mathbf{y}, \boldsymbol{\mu})$ , i.e., there exists  $\boldsymbol{\omega} \in \Omega$  such that for all  $t \in \mathbb{N}$ ,  $y(t) = \mathbf{y}(t)(\boldsymbol{\omega})$ ,  $p(t) = \boldsymbol{\mu}(t)(\boldsymbol{\omega})$ . Based on the data  $\{p(t), y(t)\}_{t=0}^N$  find matrices  $(\{A_i^N, K_i^N, Q_i^N\}_{i=1}^{n_p}, C, D)$  such that for all  $i \in \mathbb{I}_{0, n_p}$ , the matrices  $A_i^N, K_i^N, Q_i^N$  converge to matrices  $A_i, K_i, C_i, Q_i$  respectively, and the matrices  $C^N, D^N$  converge to  $C, D$  respectively, as  $N \rightarrow \infty$ , such that  $\mathcal{G}_* = (n_x, n_y, n_v, \mathbf{x}, \mathbf{v}, \{\boldsymbol{\mu}_q\}_{q \in Q}, \mathbf{y}, \{A_q, K_q\}_{q \in Q}, C, D)$  is a stationary GBS realization of  $\mathbf{y}$  and  $E[\mathbf{v}(t)\mathbf{v}^T(t)\boldsymbol{\mu}_i^2(t)] = Q_i$ .

Some remarks are in order. Recall the notion of weak realization from Subsection 3.7.1 of Chapter 3. Intuitively, the problem formulation says that for large enough  $N$ , the

$$(\{A_q^N, K_q^N, Q_q^N\}_{q \in Q}, C^N, D^N)$$

is an approximation of a weak realization of  $\mathbf{y}$ , i.e., for large enough  $N$ , the second-order moments predicted by

$$(\{A_q^N, K_q^N\}_{q \in Q}, C^N, D^N)$$

## 5.2. STOCHASTIC SUBSPACE IDENTIFICATION ALGORITHM

will be close to that of  $\mathbf{y}$ ;

$$\mathcal{G}_N = (n_x, n_y, n_v, \mathbf{x}^N, \mathbf{v}^N, \{\boldsymbol{\mu}_q\}_{q \in Q}, \mathbf{y}^N, \{A_q^N, K_q^N\}_{q \in Q}, C^N, D^N),$$

with  $E[\mathbf{v}^N(t)(\mathbf{v}^N(t))^T \boldsymbol{\mu}_i^2(t)] = Q_i^N$ , is an approximate realization of  $\mathbf{y}$  in the weak sense: the covariances  $\Lambda_w^{\mathbf{y}^N}$ ,  $T_{q,q}^{\mathbf{y}^N}$  of the output  $\mathbf{y}^N$  of  $\Sigma_N$  are close to the corresponding covariances  $\Lambda_w^{\mathbf{y}}$ ,  $T_{q,q}^{\mathbf{y}}$  of  $\mathbf{y}$  for all  $w \in Q^+$ ,  $q \in Q$ . This follows from the fact that the matrices  $C^N, D^N, A_i^N, K_i^N, Q_i^N$  are close to the matrices  $C, D, A_i, K_i, Q_i$ ,  $i \in \mathbb{I}_{0, n_p}$  and the equation (3.9) applied to  $\mathcal{G}_*$  and  $\mathcal{G}_N$  respectively. That is, the output  $\mathbf{y}_N$  of  $\mathcal{G}_N$  is close to the process  $\mathbf{y}$  in the sense that some of its second order moments are close to those of  $\mathbf{y}$ . Intuitively, this means that the probability distributions of  $\mathbf{y}_N$  and  $\mathbf{y}$  are close in some sense.

This intuition is not very satisfactory, as in general close second order moments do not formally imply that the probability distributions are close. For the linear case  $n_p = 1, \boldsymbol{\mu}_1 = 1$ , this intuition is more satisfactory. Indeed, in that case, if the noise process  $\mathbf{v}_N$  and  $\mathbf{v}$  are Gaussian, then so are  $\mathbf{y}_N$  and  $\mathbf{y}$ . Then the cross covariances  $\Lambda_w^{\mathbf{y}} = E[\mathbf{y}(t + |w|)\mathbf{y}^T(t)]$ ,  $\Lambda_w^{\mathbf{y}_N} = E[\mathbf{y}_N(t + |w|)\mathbf{y}_N^T(t)]$ ,  $T_{q,q}^{\mathbf{y}} = E[\mathbf{y}(t)\mathbf{y}^T(t)]$ ,  $T_{q,q}^{\mathbf{y}_N} = E[\mathbf{y}_N(t)\mathbf{y}_N^T(t)]$  determine the joint probability distributions of  $\mathbf{y}$  respectively  $\mathbf{y}_N$  uniquely. That is, if these cross-covariances are close, then so are the probability distributions of  $\mathbf{y}$  and  $\mathbf{y}_N$  respectively.

Another way of making sense of Problem 5.4 is to use the interpretation of GBSs as predictors, as described in Subsection 3.7.1, Chapter 3. Recall that any stationary GBS can be transformed into a GBS in forward innovation form, moreover, the matrices and the noise covariance of the latter are determined by the matrices and the noise covariance of the former. In fact, recall from Subsection 3.7.1 that any the weak realization  $(\{A_q^N, K_q^N, Q_q^N\}_{q \in Q}, C^N, D^N)$  can be used to calculate the matrices of a GBS in forward innovation form using (3.18) applied to the LSS defined (3.6). Note that the matrices of the GBS in forward innovation form are continuous functions of  $(\{A_q^N, K_q^N, Q_q^N\}_{q \in Q}, C^N, D^N)$ , and hence without loss of generality we can assume that  $D^N = I$ ,  $D = I$ , and  $\mathcal{G}_*$  is in forward innovation form and

$$\mathcal{G}_N = (n_x, n_y, n_v, \mathbf{x}^N, \mathbf{v}^N, \{\boldsymbol{\mu}_q\}_{q \in Q}, \mathbf{y}^N, \{A_q^N, K_q^N\}_{q \in Q}, C^N, D^N)$$

is in forward innovation form. Then  $\mathcal{G}_*$  can be viewed as predictors, which predict the value of  $\mathbf{y}(t)$  based on past values of  $\mathbf{y}$  and  $\{\boldsymbol{\mu}_q\}_{q \in Q}$ . By replacing  $A_i, K_i, C$  by  $A_i^N, K_i^N, C^N$  in (3.38),  $\mathcal{G}_N$  also give rise to a linear predictor of  $\mathbf{y}(t)$  based on past values of  $\mathbf{y}(s)$  and  $\{\boldsymbol{\mu}_q(s)\}_{q \in Q}$ ,  $s \in [t - t_f, t)$ .

More precisely, let us denote by  $\mathbf{x}_{\mathcal{G}_N}(t | t_f)$  and  $\mathbf{y}_{\mathcal{G}}(f | t_f)$  the processes defined as in (3.38), but with  $A_i, K_i, C$  replaced by  $A_i^N, K_i^N, C^N$ , i.e.,  $\mathbf{x}_{\mathcal{G}_N}(t_f | t_f) = 0$  and for all  $k \geq t_f$

$$\begin{aligned} \mathbf{x}_{\mathcal{G}_N}(t + 1 | t_f) &= \sum_{q \in Q} (A_q^N - K_q^N C^N) \mathbf{x}_{\mathcal{G}_N}(t | t_f) + K_q \mathbf{y}(t) \boldsymbol{\mu}_q(t), \\ \mathbf{y}_{\mathcal{G}_N}(t | t_f) &= C^N \mathbf{x}_{\mathcal{G}_N}(t | t_f). \end{aligned} \tag{5.15}$$

It then follows that  $\lim_{N \rightarrow \infty} \mathbf{x}_{\mathcal{G}_N}(t | t_f) = \mathbf{x}_{\mathcal{G}}(t | t_f)$ ,  $\lim_{N \rightarrow \infty} \mathbf{y}_{\mathcal{G}_N}(t | t_f) = \mathbf{y}_{\mathcal{G}}(t | t_f)$ , i.e., for large enough  $N$ ,  $\mathcal{G}_N$  produces predictions of  $\mathbf{y}(t)$  which is arbitrarily close to the predictions produced by  $\mathcal{G}$ , and the latter is asymptotically optimal as the prediction horizon goes to  $\infty$ .

Recall from Subsection 3.6.3, Chapter 3 that a stationary GBS realization of  $\mathbf{y}$  in forward innovation form can be computed from the covariances  $\{T_{q,q}^{\mathbf{y}}\}_{q \in Q}$  and  $\{\Lambda_w^{\mathbf{y}}\}_{w \in Q^+, |w| \leq 2n-1}$ . More precisely, recall from Section 3.3, Chapter 3 that we can associate the deterministic input-output map  $f_{\mathbf{y}}$  defined in (3.10) with the process  $\mathbf{y}$ , and recall that there is a correspondence between LSSs realizations of  $f_{\mathbf{y}}$  and stationary GBS realizations of  $\mathbf{y}$ . It is easy to see that if in Step 2 of the covariance realization algorithm Algorithm 4 from Subsection 3.6.3, Algorithm 13 from Appendix B is used, then instead of  $H_{\mathbf{y}, k, k+1}$ , it is sufficient to use the Hankel-like matrices  $\mathcal{H}_{f_{\mathbf{y}}, \alpha, \beta}, \mathcal{H}_{q, f_{\mathbf{y}}, \alpha, \beta}, \mathcal{H}_{q, f_{\mathbf{y}}, \alpha}, \mathcal{H}_{f_{\mathbf{y}}, \beta, q}$  from (B.14)–(B.17) for some selections  $\alpha, \beta$ , such that  $\text{rank } \mathcal{H}_{f_{\mathbf{y}}, \alpha, \beta} = n$  where  $n$  is the dimension of a minimal stationary GBS realization of  $\mathbf{y}$ .

The latter version of the covariance realization algorithm inspires the following system identification algorithm:



1. we choose selections  $\alpha, \beta$  such that  $\text{rank } \mathcal{H}_{f_y, \alpha, \beta} = n$ , where  $n$  is the dimension of a minimal stationary GBS realization of  $\mathbf{y}$ ,
2. We approximate the covariances necessary to construct the matrices  $\mathcal{H}_{f_y, \alpha, \beta}, \mathcal{H}_{q, f_y, \alpha, \beta}, \mathcal{H}_{q, f_y, \alpha}, \mathcal{H}_{f_y, \beta, q}$  by empirical covariances, and apply the covariance realization algorithm from Subsection 3.6.3 to the thus obtained matrices.

In order to formalize the algorithm, we introduce the following notation: For all  $w = q_1 q_2 \cdots q_r \in \mathcal{Q}^+, r > 0$ , define

$$\begin{aligned} p_w(t) &= p_{q_1}(t-k+1)p_{q_2}(t-k+2) \cdots p_{q_r}(t) \\ z_w(t) &= y(t-|w|)p_w(t-1) \frac{1}{\sqrt{\mathbf{p}(\boldsymbol{\mu}_w)}} \end{aligned} \quad (5.16)$$

and define the empirical covariances

$$\begin{aligned} {}^N\Lambda_w &= \frac{1}{N-|w|} \sum_{t=|w|}^N y(t)z_w(t) \\ {}^N T_{q,q}^y &= \frac{1}{N+1} \sum_{t=0}^N z_q(t)z_q^T(t) \end{aligned} \quad (5.17)$$

for every  $w \in \mathcal{Q}^+, q \in \mathcal{Q}$ . The resulting algorithm is formalized in Algorithm 8, for the *single output case*  $p = 1$ . Below, we will state statistical consistency of the subspace identification algorithm described in Algorithm 8.

**Theorem 5.5** (Consistency of subspace identification [172]). *Assume that*

1. *The nice selection pair  $(\alpha, \beta)$  is such that  $\text{rank } \mathcal{H}_{f_y, \alpha, \beta} = n_m$ , and the cardinality of  $\alpha$  and of  $\beta$  is the same and it equals  $n_m$ , where  $n_m$  is the state-space dimension of a minimal stationary GBS realization of  $\mathbf{y}$ . Moreover, in Step 3 of Algorithm 8, Algorithm 13 is used with the following choice of matrices  $O_{n_m}$  and  $R_{n_m}$ :  $O_{n_m} = {}^N H_{\alpha, \beta}$  and  $R_{n_m}$  is the  $n_m \times n_m$  identity matrix.*
2. *The signals  $y : \mathbb{N} \rightarrow \mathbb{R}^p$  and  $p : \mathbb{N} \rightarrow \mathbb{R}^D$  are sample paths of of the processes  $\mathbf{y}$  and  $\boldsymbol{\mu}$  respectively such that for all  $w \in \mathcal{Q}^+, q \in \mathcal{Q}$ ,*

$$\Lambda_w^y = \lim_{L \rightarrow \infty} {}^L \Lambda_w T_{q,q}^y = \lim_{L \rightarrow \infty} {}^L T_{q,q}^y \quad (5.18)$$

where  ${}^L \Lambda_w, {}^L T_{q,q}^y$  are defined as in (5.17)

Then the result of Algorithm 8 satisfies the following for all  $q \in \mathcal{Q}$ :

$$\begin{aligned} \hat{K}_q &= \lim_{M \rightarrow \infty} \lim_{N \rightarrow \infty} {}^N \hat{K}_q^M, \quad \hat{Q}_q = \lim_{M \rightarrow \infty} \lim_{N \rightarrow \infty} {}^N \text{hat} Q_q^M, \\ \hat{P}_q &= \lim_{M \rightarrow \infty} \lim_{N \rightarrow \infty} {}^N \hat{P}_q^M, \quad \hat{A}_q = \lim_{N \rightarrow \infty} {}^N \hat{A}_q, \quad \hat{C} = \lim_{N \rightarrow \infty} {}^N \hat{C}, \end{aligned}$$

and the GBS

$$\mathcal{G} = (n, p, p, \mathbf{x}, \mathbf{e}, \{\boldsymbol{\mu}_q\}_{q \in \mathcal{Q}}, \mathbf{y}, \{\hat{A}_\sigma, \hat{K}_\sigma\}_\sigma, \hat{C}, I_p)$$

is a minimal realization of  $\mathbf{y}$  in forward innovation form, and for all  $q \in \mathcal{Q}$ ,

$$\begin{aligned} \hat{Q}_q &= E[\mathbf{e}(t)\mathbf{e}^T(t)\boldsymbol{\mu}_q^2(t)], \\ \hat{P}_q &= E[\mathbf{x}(t)\mathbf{x}^T(t)\boldsymbol{\mu}_q^2(t)]. \end{aligned}$$

## 5.2. STOCHASTIC SUBSPACE IDENTIFICATION ALGORITHM

---



---

### Algorithm 8 Identification of GBSs

---

**Input:** Observations sequence  $\{y(t), p(t)\}_{t=0}^N$ , and nice selection  $(\alpha, \beta)$  of the form (2.19), maximum number of iterations  $M > 0$ .

- 1: Compute approximate covariances  ${}^N\Lambda_w, {}^N T_{q,q}^y$  are defined as in (5.17), for all  $q \in \mathcal{Q}$  and for every  $w \in \mathcal{Q}^+$ , such that  $w = ivu$  or  $w = iv$  or  $w = iu$  or  $w = ivqu$  for some words  $v, u \in \mathcal{Q}^*$ ,  $i, q \in \mathcal{Q}$ ,  $(u, k) \in \alpha$ ,  $(v, l) \in \beta$  for some  $k, l \in \mathcal{Q}$ , and
- 2: Construct *empirical* Hankel matrix  ${}^N H_{\alpha, \beta}, {}^N H_{q, \alpha, \beta}, {}^N H_{q, \alpha}, {}^N H_{\beta, q} \sigma \in \mathbb{I}_{0, n_p}$ , by replacing  $S_{q_1, q_2}^{f_y}(v)$  by  ${}^N \Lambda_{q_2 v}$  in (B.14)–(B.17), i.e., for all  $i = 1, \dots, n, j = 1, \dots, l$ ,

$$\begin{aligned} [{}^N H_{\alpha, \beta}]_{i, j} &= {}^N \Lambda_{\sigma_{jv} u_i} \\ [{}^N H_{q, \alpha, \beta}]_{i, j} &= {}^N \Lambda_{\sigma_{jv} q u_i}, \quad i = 1, \dots, n, j = 1, \dots, l \\ [{}^N H_{\alpha, q}]_i &= {}^N \Lambda_{q u_i}, \quad i = 1, \dots, n \\ [{}^N H_{q, \beta}]_j &= {}^N \Lambda_{\sigma_{jv}}, \quad j = 1, \dots, l \end{aligned}$$

- 3: Run Algorithm 13 from Chapter 2 with:  ${}^N H_{\alpha, \beta}$  instead of  $\mathcal{H}_{\alpha, \beta}^f$ ;  ${}^N H_{q, \alpha, \beta}$  instead of  $\mathcal{H}_{q, \alpha, \beta}^f$ ,  ${}^N H_{q, \alpha}$  instead of  $\mathcal{H}_{q, \alpha}^f$ ,  ${}^N H_{\beta, q}$  instead of  $\mathcal{H}_{\beta, q}^f$ ,  $q \in \mathbb{I}_{0, n_p}$ , and let the LSS returned by Algorithm 13 be  $\hat{\Sigma}_T = (n_m, \{({}^N \hat{A}_q, {}^N \hat{B}_q, {}^N \hat{C} \mid q \in \mathcal{Q})\})$ .
- 4: Set  ${}^N \hat{P}_q^0 = 0$  and  
**for**  $i = 1, \dots, M$  **do**

$$\begin{aligned} {}^N \hat{P}_q^{i+1} &= \sum_{q_1 \in \mathbb{I}_{0, n_p}} \mu_{q_1} ({}^N \hat{A}_{q_1} {}^N \hat{P}_q^i ({}^N \hat{A}_{q_1})^T + ({}^N \hat{K}_{q_1}^i) \hat{Q}_{q_1}^i ({}^N \hat{K}_{q_1}^i)^T) \\ {}^N \hat{Q}_q^i &= p \sigma {}^N T_{\sigma, \sigma}^y - {}^N \hat{C} ({}^N \hat{P}_q^i) ({}^N \hat{C})^T \\ {}^N \hat{K}_q^i &= ({}^N \hat{B}_q \sqrt{p \sigma} - {}^N \hat{A}_q {}^N \hat{P}_q^i ({}^N \hat{C})^T) ({}^N \hat{Q}_q^i)^{-1} \end{aligned}$$

- 5: **return**  $(\{{}^N \hat{A}_q, {}^N \hat{K}_q^M, {}^N \hat{Q}_q^M, {}^N \hat{P}_q^M\}_{q \in \mathcal{Q}}, {}^N \hat{C})$
-

**Remark 5.2** (Algorithm 8). *Theorem 5.5 was deliberately formulated in a non statistical language. It can readily be reformulated into the language of statistics as follows. Let us fix selections  $\alpha, \beta$  such that  $\text{rank } \mathcal{H}_{f_y, \alpha, \beta} = n_m$ , where  $n_m$  is the state-space dimension of a minimal GBS realization of  $\mathbf{y}$ . Then Algorithm 8 can be viewed as a map*

$$\mathcal{E}_N : \{\mathbf{y}(t), p(t)\}_{t=0}^N \mapsto (\{\hat{A}_q, \hat{K}_q^M, \hat{Q}_q^M\}_{q \in \mathcal{Q}}, \hat{C}).$$

We can view  $\mathcal{E}_N$  as a series of statistical estimators which estimate the parameters

$$\theta = (\{A_q, K_q, Q_q\}_{q \in \mathcal{Q}}, C, I)$$

of a weak GBS realization of  $\mathbf{y}$ , which, in turn, determine the second order moments of  $\mathbf{y}$ . If  $(\mathbf{y}, \boldsymbol{\mu})$  are jointly ergodic, then Theorem 5.5 says that

$$\lim_{N \rightarrow \infty} \mathcal{E}_N(\mathbf{y}(t), \boldsymbol{\mu}(t)) = \theta_0 = (\{\hat{A}_q, \hat{K}_q, \hat{Q}_q\}_{q \in \mathbb{I}_{0, n_p}}, \hat{C}, I),$$

where  $\theta_0$  is the vector of parameters of a stationary GBS realization of  $\mathbf{y}$ , i.e.,  $\mathcal{E}_N$  is a consistent estimator of a stationary GBS realization of  $\mathbf{y}$ .

That is, Algorithm 8 is a statistically consistent subspace identification algorithm for GBSs. To the best of our knowledge, Algorithm 8 is the first subspace identification algorithm for GBSs, whose statistical consistency has been proven formally.

## 5.2.2 Identification of stochastic LPV-SSA with inputs and its relationship with identification of GBSs and of deterministic LPV-SSA

Below we will explain the relationship between identification of stochastic LPV-SSA and that of GBSs explained in Subsection 5.2.1. This section is partially based on [173]. As it was pointed out above, a GBS can be viewed as an autonomous stochastic LPV-SSA, i.e., LPV-SSA without control inputs. Below we will discuss the relationship between identifying stochastic LPV-SSAs with control inputs and identifying GBSs.

Consider *stochastic LPV-SSA* is a system of the form

$$\begin{aligned} \mathbf{x}(t+1) &= \sum_{i=0}^{n_p} (A_i \mathbf{x}(t) + B_i u(t) + K_i \mathbf{v}(t)) p_i(t), \\ \mathbf{y}(t) &= C \mathbf{x}(t) + D u(t) + G \mathbf{v}(t) \end{aligned} \quad (5.19)$$

where,  $A_i \in \mathbb{R}^{n_x \times n_x}$ ,  $B_i \in \mathbb{R}^{n_x \times n_u}$ ,  $K_i \in \mathbb{R}^{n_x \times n_v}$ ,  $i \in \mathbb{I}_{0, n_p}$ ,  $C \in \mathbb{R}^{n_y \times n_x}$  and  $D \in \mathbb{R}^{n_y \times n_u}$ ,  $i \in \mathbb{I}_{0, n_p}$ ,  $G \in \mathbb{R}^{n_y \times n_v}$  are real constant matrices,  $\mathbf{v}$  is a white noise process, i.e.,  $E[\mathbf{v}(t) \mathbf{v}^T(s)] = 0$ ,  $s \neq t$ ,  $u(t)$  is the control input,  $\mathbf{x}(t)$  is the state process and  $\mathbf{y}(t)$  is the output,  $p(t) = (p_1(t), \dots, p_{n_p}(t))^T$ ,  $p_0(t) = 1$ , is the scheduling signal.

In order to deal with identification of (5.19), it will be useful to think of a stochastic LPV-SSA as the combination of a deterministic component:

$$\begin{aligned} \mathbf{x}^d(t+1) &= \sum_{i=0}^{n_p} (A_i^d \mathbf{x}^d(t) + B_i^d u(t)) p_i(t) \\ \mathbf{y}^d(t) &= C^d \mathbf{x}^d(t) + D^d u(t) \end{aligned} \quad (5.20)$$

and a stochastic components

$$\begin{aligned} \mathbf{x}^s(t+1) &= \sum_{i=0}^{n_p} (A_i^s \mathbf{x}^s(t) + K_i^s \mathbf{v}(t)) p_i(t), \\ \mathbf{y}^s(t) &= C^s \mathbf{x}^s(t) + G^s \mathbf{v}(t) \end{aligned} \quad (5.21)$$

## 5.2. STOCHASTIC SUBSPACE IDENTIFICATION ALGORITHM

---

Clearly, these two views are equivalent: a system of the form (5.19) can be rewritten as (5.20), (5.21) and (5.22) by taking  $A_i^s = A_i^d = A_i$ ,  $C_i^s = C_i^d = C_i$ ,  $B_i^d = B_i$ ,  $K_i^s = K_i$ ,  $i \in \mathbb{I}_{0, n_p}$ ,  $D^d = D$ ,  $G^s = G$ . In this case

$$\mathbf{x}(t) = \mathbf{x}^d(t) + \mathbf{x}^s(t),$$

and in which case the state and output process satisfy

$$\mathbf{y}(t) = \mathbf{y}^d(t) + \mathbf{y}^s(t). \quad (5.22)$$

That is, the output of a stochastic LPV-SSA is the sum of a deterministic and a stochastic components. The deterministic component is generated by LPV-SSA driven by the control input, while the stochastic component is generated by a LPV-SSA input of which is a noise processes. Intuitively, the deterministic component describes the response of a nominal system to the control input and scheduling signal, while the stochastic component models the model error and measurement uncertainty. Under suitable assumptions the decomposition of  $\mathbf{y}$  into the sum of  $\mathbf{y}^d$  and  $\mathbf{y}^s$  can be shown to be independent of the choice of the state-space representation (5.19), see [173] for more details

Conversely, from (5.20), (5.21) and (5.22) we can construct a system of the form (5.19) as follows: and

$$\mathbf{x}(t) = \begin{bmatrix} \mathbf{x}^d(t) \\ \mathbf{x}^s(t) \end{bmatrix}, A_i = \begin{bmatrix} A_i^d & 0 \\ 0 & A_i^s \end{bmatrix}, B_i = \begin{bmatrix} B_i^d \\ 0 \end{bmatrix}, K_i = \begin{bmatrix} 0 \\ K_i^s \end{bmatrix}, \\ D = D^d, C = [C^d \quad C^s]$$

and the output  $\mathbf{y}$  of the constructed stochastic LPV-SSA (5.19) will satisfy (5.22).

That is, we can identify a stochastic LPV-SSA (5.19) with a tuple

$$(\Sigma_d, \Sigma_s, \mathbf{v})$$

where

$$\Sigma_d = (\mathbb{P}, \{A_i^d, B_i^d, C_i^d, D_i^d\}_{i=0}^{n_p}), C_0^d = C^d, D_0^d = D^d, (C_i^d, D_i^d) = (0, 0), i = 1, \dots, n_p \\ \Sigma_s = (\mathbb{P}, \{A_i^s, K_i^s, C_i^s, G_i^s\}_{i=0}^{n_p}), C_0^s = C^s, G_0 = G, (C_i^s, G_i^s) = (0, 0), i = 1, \dots, n_p$$

are LPV-SSA, and  $\mathbf{v}$  is the stochastic noise process. In the sequel, we will identify a LPV-SSA

$$(\mathbb{P}, \{\tilde{A}_i, \tilde{B}_i, \tilde{C}_i, \tilde{D}_i\}_{i=0}^{n_p})$$

for which the matrices  $\tilde{C}_i, \tilde{D}_i$  are zero for all  $i = 1, \dots, n_p$  and  $\tilde{C}_0 = \tilde{C}$ ,  $\tilde{D}_0 = \tilde{D}$  with the tuple

$$(\mathbb{P}, \{\tilde{A}_i, \tilde{B}_i\}_{i=0}^{n_p}, \tilde{C}, \tilde{D}).$$

In particular, since the matrices  $C_i^d, D_i^d, G_i, C_s^i$  are zero for  $i = 1, \dots, n_p$ , we will use the notation

$$\Sigma_d = (\mathbb{P}, \{A_i^d, B_i^d\}_{i=0}^{n_p}, C^d, D^d) \\ \Sigma_s = (\mathbb{P}, \{A_i^s, K_i^s\}_{i=0}^{n_p}, C^s, G)$$

and we will refer to  $\Sigma_d$  as the deterministic component, and  $\Sigma_s$  as the stochastic component of the stochastic LPV-SSA.

The correspondence above allows us to specify what we mean by the input-output behavior of the stochastic LPV-SSA (5.19). In turn, the latter is important in order to formulate the system identification problem.

More precisely, we will proceed as follows: we will define the input-output behavior of (5.19) via the input-output functions of the deterministic LPV-SSA  $\Sigma_s$  and  $\Sigma_d$  defined above. Note that we could identify the input-output behavior of a deterministic LPV-SSA with the input-output map generated by the zero initial state, and we could define all the signals on the positive time axis. However, if we assume the presence of a stochastic noise, then there is no reason to assume that the noise has ever been absent in the past. On the contrary, the natural assumption is that the noise has always been present, hence the state and output at the beginning of observations represent some sort of steady state behavior which arises from feeding in the noise in the infinite past. For linear time-invariant systems the situation is similar, and the classical solution is to assume that the state and output signals involved are defined for negative time instances too, and to assume that the system is stable. In this case, we can define the behavior of the system started in the infinite past as a limit of behaviors starting from some finite time instance  $t_0$  in the past, as  $t_0 \rightarrow +\infty$ . In addition, it solves the problem of possibly non-zero initial state, as the effect of any initial state is absent in this case. We will take a similar approach for LPV systems.

To this end, we introduce the following definitions. We will say that a LPV-SSA

$$(\mathbb{P}, \{\tilde{A}_i, \tilde{B}_i, \tilde{C}_i, \tilde{D}_i\}_{i=0}^{n_p})$$

is  $\{\mathbf{p}_q\}_{q=1}^{n_p}$ -stable, if the matrix

$$(n_p + 1)(\tilde{A}_0^s \otimes \tilde{A}_0^s + (\sum_{q=1}^{n_p} \mathbf{p}_q^2 \tilde{A}_q^s \otimes \tilde{A}_q^s))$$

is stable.

Let us call a tuple  $(\Sigma_s, \mathbf{v})$  *stationary stochastic autonomous LPV-SSA*, if  $\Sigma_s$  is a LPV-SSA such that

- $\Sigma_s$  is  $\{\mathbf{p}_q\}_{q=1}^D$ -stable, and
- $\mathbb{P} = [-\mathbf{p}_1, \mathbf{p}_1] \times \cdots \times [-\mathbf{p}_{n_p}, \mathbf{p}_{n_p}]$ , and
- $\mathbf{v}$  is a white noise process taking values in the space  $\mathbb{R}^{n_v}$ , i.e.,  $\mathbf{v}$  is zero mean square integrable wide-sense stationary with  $E[\mathbf{v}(s)\mathbf{v}^T(t)] = 0$ ,  $s \neq t$  and  $E[\mathbf{v}(t)\mathbf{v}^T(t)] = \mathbf{Q}$ ,  $s, t \in \mathbb{Z}$ .

Let us call the triple  $(\Sigma_d, \Sigma_s, \mathbf{v})$  a *stationary stochastic LPV-SSA*, if  $(\Sigma_s, \mathbf{v})$  is a stationary stochastic autonomous LPV-SSA.

We will first define the input-output behavior of stationary stochastic autonomous LPV-SSA, and then we will use it to define the input-output behavior of stationary stochastic LPV-SSAs.

To third end, for a set  $X$ , let us denote by  $X^{\mathbb{Z}}$  the set of all functions  $\mathbb{Z} \rightarrow X$ . Then for any  $p \in \mathbb{P}^{\mathcal{Z}}$ , if  $p_i(t) \in [-\mathbf{p}_i, \mathbf{p}_i]$ ,  $i = 1, \dots, n_p$ , and  $\mathbf{v}$  is a white noise process, then similarly to the proof of [226, Lemma 3] it can be shown that for all  $t \in \mathbb{Z}$  the limit

$$\begin{aligned} \mathbf{y}_s(t) &= \lim_{t_0 \rightarrow \infty} \mathfrak{Y}_{\Sigma_s, 0}(\sigma^t p, \sigma^t \mathbf{v})(t_0 + t) \\ \mathbf{x}_s(t) &= \lim_{t_0 \rightarrow \infty} \mathfrak{X}_{\Sigma_s, 0}(\sigma^t p, \sigma^t \mathbf{v})(t_0 + t) \end{aligned}$$

converges in mean square sense, where  $\mathfrak{Y}_{\Sigma_s, 0}$  and  $\mathfrak{X}_{\Sigma_s, 0}$  are the input-output and input-to-state maps of  $\Sigma_s$  induced by the zero initial state, as defined in Chapter 2, and  $\sigma^t p(s) = p(s+t)$ ,  $\sigma^t \mathbf{v}(s+t)$ <sup>1</sup>.

Let us denote by  $\mathbf{Y}$  the set of all random variables with values in  $\mathbb{R}^p$  and assume that

$$\mathbb{P} = [-\mathbf{p}_1, \mathbf{p}_1] \times \cdots \times [-\mathbf{p}_{n_p}, \mathbf{p}_{n_p}].$$

Then for a stationary autonomous LPV-SSA  $(\Sigma_s, \mathbf{v})$ , we define the map

$$\mathcal{B}_{\Sigma_s, \mathbf{v}} : \mathbb{P}^{\mathbb{Z}} \rightarrow \mathbf{Y}^{\mathbb{Z}}$$

<sup>1</sup>To be very precise, note that  $\sigma^t p : \mathbb{N} \ni s \mapsto p(s+t) \in \mathbb{P}$  and we can define the map  $\sigma^t \mathbf{v} : \Omega \ni \omega \mapsto (\mathbb{N} \ni s \mapsto \mathbf{v}(s+t)(\omega)) \in (\mathbb{R}^{n_v})^{\mathbb{N}}$  and hence  $\mathfrak{Y}_{\Sigma_s, 0}(\sigma^t p, \sigma^t \mathbf{v})(t_0 + t)$  and  $\mathfrak{X}_{\Sigma_s, 0}(\sigma^t p, \sigma^t \mathbf{v})(t_0 + t)$  can naturally be identified with the random variables  $\Omega \ni \omega \mapsto \mathfrak{Y}_{\Sigma_s, 0}(\sigma^t p, \sigma^t \mathbf{v}(\omega))(t_0 + t)$  and  $\Omega \ni \omega \mapsto \mathfrak{X}_{\Sigma_s, 0}(\sigma^t p, \sigma^t \mathbf{v}(\omega))(t_0 + t)$  respectively.

## 5.2. STOCHASTIC SUBSPACE IDENTIFICATION ALGORITHM

---

as follows

$$\mathcal{Y}_{\Sigma_s, \mathbf{v}}(p)(t) = \lim_{t_0 \rightarrow \infty} \mathfrak{Y}_{\Sigma_s, 0}(\sigma^t p, \sigma^t \mathbf{v})(t_0 + t).$$

We will call the map  $\mathcal{Y}_{\Sigma_s, \mathbf{v}}$  the *input-output map* of  $(\Sigma_s, \mathbf{v})$ . We will say that  $(\Sigma_s, \mathbf{v})$  is a *realization* of a function  $\mathcal{F}_s : \mathbb{P}^{\mathbb{Z}} \rightarrow \mathbf{Y}^{\mathbb{Z}}$ , if  $\mathcal{F}_s = \mathcal{Y}_{\Sigma_s, \mathbf{v}}$ .

Let  $(\Sigma_d, \Sigma_s, \mathbf{v})$  be a stationary stochastic LPV-SSA. Then we define its input-output map as the map

$$\mathcal{Y}_{\Sigma_d, \Sigma_s, \mathbf{v}} : \mathcal{U} \times \mathcal{P}^{\mathbb{Z}} \rightarrow \mathbf{Y}^{\mathbb{Z}}$$

such that

$$\mathcal{Y}_{\Sigma_d, \Sigma_s, \mathbf{v}}(u, p)(t) = \begin{cases} \mathfrak{Y}_{\Sigma_d}(u, p|_{\mathbb{N}})(t) + \mathcal{Y}_{\Sigma_s, \mathbf{v}}(p)(t) & t \geq 0 \\ \mathcal{Y}_{\Sigma_s, \mathbf{v}}(p)(t) & t < 0 \end{cases}$$

where  $p|_{\mathbb{N}}$  denotes the restriction of  $p$  to  $\mathbb{N}$ , i.e.,  $p|_{\mathbb{N}} : \mathbb{N} \ni s \mapsto p(s)$ .

It is then easy to see that for the system (5.19), if  $\mathbf{x}^d(0) = 0$  and  $u(t) = 0$  for  $t < 0$ , then

$$\mathbf{y}(t) = \mathcal{Y}_{\Sigma_d, \Sigma_s, \mathbf{v}}(u, p)(t),$$

for all  $t > 0$ .

That is, for negative times the output of a stochastic LPV-SSA is its steady state response due to the presence of noise, while for nonnegative time its output is the sum of the response to control inputs and the response to the noise.

We will say that  $(\Sigma_d, \Sigma_s, \mathbf{v})$  is a *realization* of a function  $\mathcal{F} : \mathcal{U} \times \mathcal{P}^{\mathbb{Z}} \rightarrow \mathbf{Y}^{\mathbb{Z}}$ , if

$$\mathcal{Y}_{\Sigma_d, \Sigma_s, \mathbf{v}} = \mathcal{F}.$$

Notice that  $\mathcal{F}$  can be realized by a stochastic LPV system, if and only if we can decompose it as follows:

$$\mathcal{F}(u, p) = \mathcal{F}_d(u, p|_{\mathbb{N}}) + \mathcal{F}_s(p) \tag{5.23}$$

such that  $\mathcal{F}_d$  is an input-output map of a LPV-SSA and  $\mathcal{F}_s$  is the input-output map of an autonomous stochastic LPV-SSA. In particular,  $\mathcal{F}(0, p)(t) = \mathcal{F}_s(p)(t)$  and  $\mathcal{F}_d(u, p|_{\mathbb{N}})(t) = \mathcal{F}(u, p)(t) - \mathcal{F}(0, p)(t)$  for all  $t \in \mathbb{N}$ .

The idea behind this definition is as follows:  $\mathcal{Y}_{\Sigma_d, \Sigma_s, \mathbf{v}, \boldsymbol{\mu}, \mathbf{u}} = \mathcal{F}$  captures the output response of (5.19) to any scheduling signal and control input, without assuming that latter signals being stochastic. By saying that we would like to find approximate realizations of  $\mathcal{F}$  we capture the intuition that the estimated model should approximate the output response of the underlying system to any scheduling signal and control input, and not only to the signal  $\boldsymbol{\mu}$  and  $\mathbf{u}$  used for system identification.

Assume now that we would like to estimate a stationary stochastic LPV-SSA realization  $(\Sigma_d, \Sigma_s, \mathbf{v})$  of  $\mathcal{F}$  from data collected during system identification experiments. For the sake of simplicity assume that we can make two system identification experiments:

- one with zero control inputs, and
- one with a nonzero control input.

and that the scheduling signal in both experiments is the same and it is a sample of a stochastic process. Moreover, we assume that the data is generated by a stationary stochastic LPV-SSA realization of  $\mathcal{F}$ .

**Assumption 3** (Identification experiment stochastic LPV-SSA). *Assume that  $\mathcal{F}$  has a realization by a stochastic LPV-SSA realization  $(\Sigma_d, \Sigma_s, \mathbf{v})$  and assume there exist a stochastic process  $\boldsymbol{\mu}$  defined on the same  $\alpha$  underlying probability space  $(\Omega, P, \mathcal{F})$  as the noise process  $\mathbf{v}$ , such that the following holds:*

- $\boldsymbol{\mu}$  satisfies the conditions of Assumptions 2, and
- $(\Sigma_s, \mathbf{v})$  is a stationary autonomous stochastic LPV-SSA, and

- $\Sigma_s$  is a minimal LPV-SSA,  $G = I_{n_y}$ , and
- $\sum_{q \in Q} (\mathbf{p}(\boldsymbol{\mu}_q))^2 (A_q^s - K_q^s C^s) \otimes (A_q^s - K_q^s C^s)$  is a stable matrix, i.e. all its eigenvalues are inside the unit disk, and
- $\boldsymbol{\mu}(t) \in \mathbb{P}$ ,  $t \in \mathbb{Z}$ , and
- $\mathbf{v}$  is an ZMWSSI processes w.r.t.  $\{\boldsymbol{\mu}_q\}_{q \in Q}$ , see Chapter 3 for the definition of ZMWSSI processes.

Furthermore, assume that we observe the signals

$$p \in \mathbb{P}^{\mathbb{Z}}, u \in \mathcal{U}, y, y^s \in \mathcal{Y} \quad (5.24)$$

which are samples of the stochastic processes  $\boldsymbol{\mu}$ ,  $\mathbf{u}$ ,  $\mathbf{y}$  and  $\mathbf{y}^s$  respectively, i.e. for some  $\omega \in \Omega$ ,

$$p(t) = \boldsymbol{\mu}(t)(\omega), u(t) = \mathbf{u}(t)(\omega), y(t) = \mathbf{y}(t)(\omega), y^s(t) = \mathbf{y}^s(t)(\omega) \quad (5.25)$$

where for all  $t \in \mathbb{Z}$ ,

$$\begin{aligned} \mathbf{y}^s(t) &= \mathcal{F}(0, \boldsymbol{\mu})(t) \\ \mathbf{y}(t) &= \mathcal{F}(u, \boldsymbol{\mu})(t) \end{aligned} \quad (5.26)$$

<sup>2</sup> That is,  $\hat{y}$  is a sample of  $\mathcal{F}(0, \boldsymbol{\mu})$  and  $y$  is a sample of  $\mathcal{F}(u, \boldsymbol{\mu})(t)$  and  $p$  is the sample of  $\boldsymbol{\mu}$ , corresponding to the same random event.

**Problem 5.5** (Identification problem). *If Assumption 3 holds, given any  $N$ , compute from the dataset*

$$\{y(t), y^s(t), p(t), u(t)\}_{t=1}^N$$

the matrices  $\{\hat{A}_i^{d,N}, \hat{A}_i^{s,N}, \hat{B}_i^{d,N}, \hat{K}_i^{s,N}, Q_i^N\}_{i=0}^{n_p}, \hat{C}_i^{d,N}, \hat{C}_i^{s,N}, \hat{D}^N, \hat{G}^N$  such that the limits below exist,

$$(\hat{A}_i^s, \hat{K}_i^s, Q_i) = \lim_{N \rightarrow \infty} (\hat{A}_i^{s,N}, \hat{K}_i^{s,N}, Q_i^N), i \in \mathbb{I}_{0, n_p}, (\hat{C}^s, \hat{G}) = \lim_{N \rightarrow \infty} (\hat{C}^{s,N}, \hat{G}^N) \quad (5.27)$$

$$(\hat{A}_i^d, \hat{B}_i^d) = \lim_{N \rightarrow \infty} (\hat{A}_i^{d,N}, \hat{B}_i^{d,N}), i \in \mathbb{I}_{0, n_p}, (\hat{C}^d, \hat{D}) = \lim_{N \rightarrow \infty} (\hat{C}^{d,N}, \hat{D}^N) \quad (5.28)$$

and for

$$\hat{\Sigma}_s = (\mathbb{P}, \{\hat{A}_i^s, \hat{K}_i^s\}_{i=0}^{n_p}, \hat{C}^s, \hat{D}^s) \quad (5.29)$$

$$\hat{\Sigma}_d = (\mathbb{P}, \{\hat{A}_i^d, \hat{B}_i^d\}_{i=0}^{n_p}, \hat{C}^d, \hat{D}^d) \quad (5.30)$$

there exists stochastic process  $\hat{\mathbf{v}}$  such that

$$(\hat{\Sigma}_d, \hat{\Sigma}_s, \hat{\mathbf{v}})$$

is a stationary stochastic LPV-SSA realization of  $\mathcal{F}$ , and

$$E[\hat{\mathbf{v}}(t)\hat{\mathbf{v}}(t)\boldsymbol{\mu}^2(t)] = Q_i, i \in \mathbb{I}_{0, n_p}. \quad (5.31)$$

Recall the decomposition (5.23) of  $\mathcal{F}$ . Note that in the problem formulation above  $(\hat{\Sigma}_s, \hat{\mathbf{v}})$  is necessarily a realization of the stochastic component  $\mathcal{F}_s$  of  $\mathcal{F}$ , and  $\hat{\Sigma}_d$  is necessarily a realization of the deterministic component  $\mathcal{F}_d$  of  $\mathcal{F}$ . Moreover,  $\hat{\Sigma}_s$  and the variances  $\{Q_i\}_{i=0}^{n_p}$  of  $\{\hat{\mathbf{v}}(t)\boldsymbol{\mu}_i^2(t)\}_{i=0}^{n_p}$  depend only on  $\{\hat{A}_i^{s,N}, \hat{K}_i^{s,N}, Q_i^N\}_{i=0}^{n_p}, \hat{C}_i^{s,N}, \hat{G}^N$ . Likewise,  $\hat{\Sigma}_d$  depends only on the matrices  $\{\hat{A}_i^{d,N}, \hat{B}_i^{d,N}\}_{i=0}^{n_p}, \hat{C}_i^{d,N}, \hat{D}^N$ .

That is, in order to solve Problem 5.5, it is sufficient to solve the following two problems.

<sup>2</sup>  $\mathcal{F}(0, \boldsymbol{\mu})(t)$  and  $\mathcal{F}(u, \boldsymbol{\mu})(t)$  can naturally be identified with the random variable:  $\Omega \ni \hat{\omega} \mapsto \mathcal{F}(0, \boldsymbol{\mu}(\hat{\omega}))(t)(\hat{\omega})$  and  $\Omega \ni \hat{\omega} \mapsto \mathcal{F}(u, \boldsymbol{\mu}(\hat{\omega}))(t)(\hat{\omega})$ , where  $\boldsymbol{\mu}(\hat{\omega}) : \mathbb{Z} \ni s \mapsto \boldsymbol{\mu}(s)(\hat{\omega})$ .

## 5.2. STOCHASTIC SUBSPACE IDENTIFICATION ALGORITHM

**Problem 5.6** (Identification of the deterministic component). *Compute the matrices  $\{\hat{A}_i^{d,N}, \hat{B}_i^{d,N}\}_{i=0}^{n_p}$ ,  $\hat{C}^{d,N}$ ,  $\hat{D}^N$  from  $\{y(t) - y^s(t), p(t), u(t)\}_{t=0}^N$  such that the limits (5.28) exist and  $\hat{\Sigma}_d$  from (5.30) is a realization of  $\mathcal{F}_d$ .*

**Problem 5.7** (Identification of the stochastic component). *Compute the matrices  $\{\hat{A}_i^{s,N}, \hat{K}_i^{s,N}, Q_i^N\}_{i=0}^{n_p}$ ,  $\hat{C}^{s,N}$ ,  $\hat{G}^N$  from  $\{y^s(t), p(t)\}_{t=0}^N$  such that the limits (5.27) exist and for  $\hat{\Sigma}_s$  from (5.29) and for some noise process  $\hat{\mathbf{v}}$  for which (5.31),  $(\hat{\Sigma}_s, \hat{\mathbf{v}})$  is a stationary autonomous stochastic LPV-SSA which is a realization of  $\mathcal{F}_s$ .*

Solving Problem 5.6 amounts to solving Problem 5.2 and can be done using the methods discussed in Subsection 5.1.1. We will explain how to reduce Problem 5.7 to solving the identification Problem 5.4 for GBSs. To this end, notice that if the stochastic autonomous LPV-SSA  $(\Sigma_s, \mathbf{v})$  satisfies Assumption 3, then we can define GBS

$$\mathcal{G}_{\Sigma_s, \mathbf{v}} = (n_x, n_y, n_v, Q, \mathbf{x}^s, \mathbf{v}, \{\mu_q\}_{q \in Q}, \mathbf{y}^s, C^s, D^s, \{A_i^s, K_i^s\}_{i=1}^D)$$

where  $Q = \{0, \dots, D\}$ , where  $\mathbf{y}^s(t) = \mathfrak{Y}_{\Sigma_s, \mathbf{v}}(\mu)(t) = \mathcal{F}_s(\mu)(t) = \mathcal{F}(0, \mu)(t)$  and  $\mathbf{x}^s(t) = \mathfrak{X}_{\Sigma_s, \mathbf{v}}(\mu)(t)$ , and this GBS will be stationary. That is, with assumptions of Assumption 3, the stationary GBS  $\mathcal{G}_{\Sigma_s, \mathbf{v}}$  is a realization of  $(\mathbf{y}^s, \mu)$ .

Recall from Subsection 3.7.2 the correspondence between LPV-SSAs and stationary GBSs. It then follows that  $\Sigma_s$  is the LPV-SSA corresponding to  $\mathcal{G}_{\Sigma_s, \mathbf{v}}$ . Recall from Assumption 3 that  $\Sigma_s$  is a minimal LPV-SSA and it satisfies the conditions above and the condition that  $G = I_{n_y}$  and  $\sum_{q \in Q} (\mathbf{p}(\mu_q))^2 (A_q^s - K_q^s C^s) \otimes (A_q^s - K_q^s C^s)$  is stable. As it was noted in Subsection 3.7.2, then  $\mathcal{G}_{\Sigma_s, \mathbf{v}}$  is a minimal realization of  $\mathbf{y}^s$  in forward innovation form, and hence  $\mathbf{v}$  is the innovation process of  $\mathbf{y}^s$  in the sense defined in Chapter 3. Hence, any minimal stationary GBS realization  $\mathcal{G}$  of  $\mathbf{y}^s$  will yield an LPV-SSA  $\Sigma_{\mathcal{G}}$  such that  $\Sigma_{\mathcal{G}}$  is isomorphic to  $\Sigma_s$ . In particular,  $(\Sigma_{\mathcal{G}}, \mathbf{v})$  will be a stochastic autonomous LPV-SSA which is a realization of  $\mathfrak{Y}_{\Sigma_s, \mathbf{v}} = \mathfrak{F}_s$ .

Assume now that the matrices  $\{\hat{A}_i^{s,N}, \hat{K}_i^{s,N}, Q_i^N\}_{i=0}^{n_p}$ ,  $\hat{C}^{s,N}$ ,  $\hat{G}^N$  computed from  $\{y^s(t), p(t)\}_{t=0}^N$  are such that  $G_N = I_{n_y}$  and the limits (5.27) exist and

$$\hat{\mathcal{G}} = (n, n_y, n_y, Q, \hat{\mathbf{x}}, \hat{\mathbf{v}}, \mathbf{y}^s, \hat{C}^s, I_{n_y}, \{\hat{A}_q^s, \hat{K}_q^s\}_{q \in Q}),$$

is a realization of  $\mathbf{y}^s$  in forward innovation form and  $\hat{\mathbf{v}}$  is the innovation process of  $\mathbf{y}^s$  and it satisfies (5.31). Consider  $\hat{\Sigma}_s$  from (5.29). Then  $\hat{\Sigma}_s$  is the LPV-SSA associated with  $\hat{\mathcal{G}}$ , and by the discussion above  $(\hat{\Sigma}_s, \hat{\mathbf{v}})$  is a stationary autonomous stochastic LPV-SSA which is a realization of  $\mathcal{F}_s$ . That is, a solution to Problem 5.4 gives rise to a solution of Problem 5.7.

To sum up, in order to solve Problem 5.5, it is sufficient to

1. solve Problem 5.2 for data  $\{u(t), p(t), y(t) - y^s(t)\}_{t=0}^N$  using the methods discussed in Subsection 5.1.1.
2. solve Problem 5.4 for data  $\{p(t), y^s(t)\}_{t=0}^N$  using the methods discussed in Subsection 5.2.1.

In other words, under some mild assumptions, the problem of identifying a stochastic autonomous LPV-SSA can be reduced to the problem of identifying a stationary GBS. The most important of these assumptions is that the observed scheduling signal behaves as a sample path of a stochastic process. In fact, even this assumption could be further weakened by requiring that the sample covariances constructed using the observed scheduling signal converge to covariances of some stochastic process, [174]. We can view these assumptions as persistence of excitation assumptions for the scheduling signal. In particular, when applied to stochastic LPV-SSAs with inputs, this assumption does not mean that the scheduling signal of the estimated model is assumed to be stochastic. Hence, the usual techniques of robust control still apply. Finally, if the scheduling signal is an external input (this might be the case for switched systems, for example), then we can actually generate scheduling signals as sample paths of suitable stochastic processes. The only slight issue is that the thus generated signals are applied starting from a certain initial time instance, not from  $-\infty$ . In practice, this is not an issue, as it is sufficient to run the system long enough with the generated scheduling signal.

We note that in the discussion above we assumed that two distinct identification experiments can be performed: one with zero control input, and another one with a sufficiently rich non-zero control input. In fact, the



principle that the identification of a stochastic LPV-SSA can be reduced to the identification of the deterministic and stochastic components can be extended to the case of one single control experiment, see [174] for details.

Finally, the discussion above sheds some more light on the notion of stochastic LPV-SSAs in innovation form: namely, if the stochastic component of an LPV-SSA satisfies the conditions which guarantee that the corresponding stationary GBS is a minimal one in forward innovation form, then from the point of view of identification it behaves similarly to stochastic linear time-invariant state-space representations in forward innovation form. This suggests that we define the innovation form of an LPV-SSA by requiring that the stochastic component should be such that the corresponding GBS is in forward innovation form. The implications of such a definition remain a topic of future research.

### 5.2.3 Role of realization theory

We could see that the covariance realization algorithm from Subsection 3.6.3 can be used to derive a subspace identification algorithm GBSs and stochastic LPV-SSAs, and to show that the derived algorithm is a statistically consistent estimator. This is not the only use of realization theory. Another consequence of realization theory that it provides a mathematical formalization of the concept of innovation representation for GBS and for stochastic LPV-SSAs. This also justifies and formalizes in a rigorous manner the widespread assumption made in LPV subspace identification literature that the generator of the output is in innovation form [284, 76, 87]. More generally, it helps to propose a coherent formalization of the identification problem for LPV systems and find assumptions which are relevant for both control and for system identification. For example, it allows to reduce the problem of identification of stochastic LPV-SSAs to the problem of identification of GBSs, i.e., to relate the case when the scheduling signal is stochastic with the case of deterministic scheduling signals. This work is not yet finished, and a lot more remains to be done.

## 5.3 Conclusions and further work

In this chapter I have presented a selection of my contributions to identification of LPV systems. Namely, I presented selected results from my work on identifiability analysis of deterministic LPV-SSAs and on subspace identification of deterministic LPV-SSAs and of autonomous stochastic LPV-SSAs. These contributions relied on realization theory of LPV and stochastic generalized bilinear systems (GBS). More precisely, realization theory of deterministic LPV-SSAs was shown to be useful for identifiability analysis and deterministic subspace algorithms which are provenly consistent. Realization theory of GBSs was shown to be useful for statistically consistent subspace identification algorithms for GBSs and for stochastic LPV-SSAs. There are other potential applications of realization theory to system identification. For example, in [222, 214] the notion of distance between linear switched systems was investigated and it was shown that the space of minimal linear switched systems forms a smooth (analytic, Nash) manifold. Moreover, explicit expression for the coordinate charts were presented and it was shown that these coordinate charts also represent identifiable canonical forms. In particular, it shows that there are no global identifiable parameterizations for linear switched systems, even in the SISO case. These results relied heavily on realization theory, in particular, on realization algorithms using selections. In fact, coordinate charts of the manifold of minimal state-space representations correspond to different choices of selections. Another application of realization theory are conditions for persistence of excitation of linear switched systems [206], and conditions on minimality and identifiability of switched autoregressive models [207, 208].

## Chapter 6

# Reverse engineering of network structure for stochastic linear and bilinear systems

### 6.1 Introduction

In this chapter we are interested in the problem of reverse engineering of the network graph of a stochastic dynamical system. By reverse engineering of the network graph we mean finding out the network graph of a system based on the observed output of the system. This problem arises in several domains such as systems biology [191, 138, 146, 278], neuroscience [242], smart grids [48, 315], etc.

To solve this problem, we first need to understand when the observed behavior can be realized by a system with a specific network graph. More precisely, we would like to decide, based on observed data, if the observed behavior can be realized by a system with a specific network graph. To this end, we need to:

- propose a formal mathematical definition of the concept of network graph, and
- relate this definition to the statistical properties of the observed data.

Informally, by the *network graph* of a system we mean a directed graph, whose nodes correspond to subsystems, such that each subsystem generates a component of the output process. There is an edge from one node to the other, if the subsystem corresponding to the source node sends information to the subsystem corresponding to the target node.

In this chapter, we will consider two types of dynamical systems: stochastic *linear state-space representations* without inputs (*abbreviated by sLTI-SS*) and *generalized bilinear systems* (*abbreviated as GBSs*) from Chapter 3.

For these systems network graphs are defined as follows. Let  $\mathbf{y}$  be an output process of a sLTI-SS or of a GBS and denote the system that represents  $\mathbf{y}$  by  $\mathcal{S}$ . Assume that  $\mathbf{y}$  is partitioned such that  $\mathbf{y} = [\mathbf{y}_1^T, \dots, \mathbf{y}_n^T]^T$  and consider the subsystems  $\mathcal{S}_i$ ,  $i = 1, \dots, n$  of the system  $\mathcal{S}$  such that  $\mathcal{S}_i$  generates the component  $\mathbf{y}_i$ . Then, the network graph has nodes  $\{1, \dots, n\}$  and there is an edge from node  $i$  to node  $j$ , if the noise and the state process of  $\mathcal{S}_i$  serve as an input of  $\mathcal{S}_j$ . In fact, for the systems at hand, an edge  $(i, j)$  in the network graph corresponds to non-zero blocks in certain matrix parameters of the system. Similarly, the lack of this edge corresponds to zero blocks in those matrices. Intuitively, an edge in the network graph means that information can flow from the subsystem corresponding to a source node to the subsystem corresponding to the target node, but there is no information flowing the other way around. Figure 6.1 illustrates the network graph of a sLTI-SS representation having the three-node star graph as its network graph. This approach offers an intuitive mechanistic explanation of how one component of the output process influences the other. However, the same output process can be generated by systems with different network graphs. As a result, the presence of an interaction between two

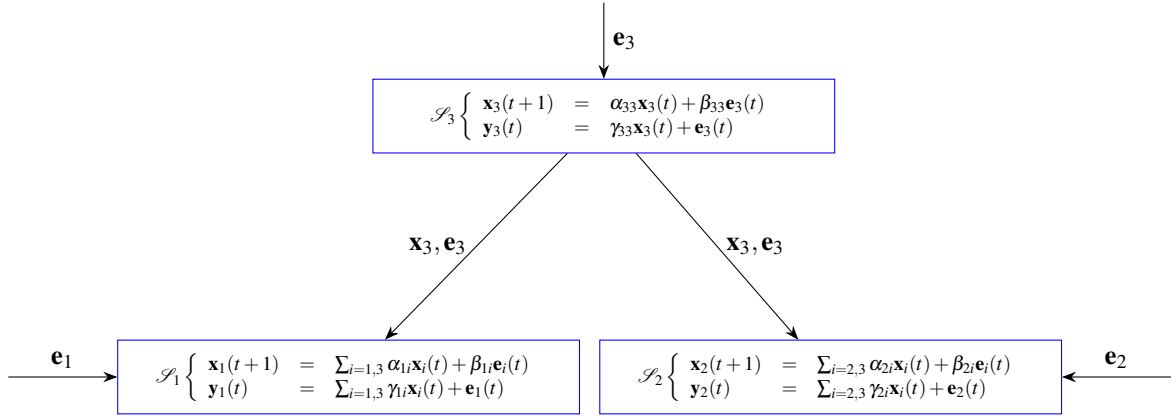


Figure 6.1: LTI-SS representation of a process  $\mathbf{y} = [\mathbf{y}_1^T, \mathbf{y}_2^T, \mathbf{y}_3^T]^T$  with the three-node star graph as its network graph: The state and noise process of subsystem  $\mathcal{S}_3$  serves as an input to subsystems  $\mathcal{S}_1$  and  $\mathcal{S}_2$ .

output components depends on the exact dynamical system representing the output process. For this reason, it is not a-priori clear how to relate statistical properties of the outputs to the existence of a sLTI-SS or GBSs.

In order to translate existence of a state-space representation with a certain network graph to statistical properties of outputs, we use Granger causality [110]. Intuitively,  $\mathbf{y}_1$  Granger causes  $\mathbf{y}_2$ , if the best linear predictions of  $\mathbf{y}_2$  based on the past values of  $\mathbf{y}_1$  and  $\mathbf{y}_2$  are better than those only based on the past values of  $\mathbf{y}_2$ . More generally, consider an output process  $\mathbf{y} = [\mathbf{y}_1^T, \dots, \mathbf{y}_n^T]^T$  of a dynamical system. We consider a graph nodes of which correspond to the components  $\mathbf{y}_i$  and there is no edge from the node labeled by  $\mathbf{y}_i$  to the node labeled by  $\mathbf{y}_j$ , if  $\mathbf{y}_i$  does not Granger-cause  $\mathbf{y}_j$  in a suitable sense (*conditional Granger-causality*, *GB-Granger causality*, etc.). We will try to relate this graph with the network graph of a suitable state-space representation.

Conditional Granger non-causality is a general form of Granger non-causality: Informally,  $\mathbf{y}_i$  conditionally does not Granger cause  $\mathbf{y}_j$  with respect to  $\mathbf{z}$  if the knowledge of the past values of  $\mathbf{y}_i$ ,  $\mathbf{y}_j$  and  $\mathbf{z}$  does not yield a more accurate prediction of the future values of  $\mathbf{y}_j$  than the knowledge of the past values of only  $\mathbf{y}_j$  and  $\mathbf{z}$ . In this case, when defining the graph above, we will require that there is no edged from the node  $\mathbf{y}_i$  to the node  $\mathbf{y}_j$ , if  $\mathbf{y}_i$  does not conditional Granger-cause  $\mathbf{y}_j$  with respect to a suitably defined variable  $\mathbf{z}$ , components of which are chosen from the output components  $\mathbf{y}_k$ ,  $k \neq i, k \neq j$ .

The notion of GB-Granger causality will be used only for outputs generated by GBSs driven by some input process  $\{\boldsymbol{\mu}_q\}_{q \in Q}$ . Intuitively,  $\mathbf{y}_1$  does not GB-Granger cause  $\mathbf{y}_2$ , if the knowledge of the products of past value of  $\mathbf{y}_1$  and  $\mathbf{y}_2$  with the past vales of  $\{\boldsymbol{\mu}_q\}_{q \in Q}$  does not yield a more accurate prediction of future values  $\mathbf{y}_2$  than the knowledge of the products of past values of  $\mathbf{y}_2$  with the past value of  $\{\boldsymbol{\mu}_q\}_{q \in Q}$ .

**Relationship between the two approaches: main results of the chapter** In this chapter we relate Granger-causality with existence of a state-space representations with a certain network graph. The chapter is based on the thesis [133], and the publications [136, 137, 134, 135].

In particular, we show that a process  $\mathbf{y} = [\mathbf{y}_1^T, \mathbf{y}_2^T]^T$  has a sLTI-SS realization in the so-called block triangular form if and only if  $\mathbf{y}_1$  does not Granger cause  $\mathbf{y}_2$ . Informally, a sLTI-SS representation in block triangular form is a system whose network graph has two nodes, corresponding to two subsystems generating  $\mathbf{y}_1$  and  $\mathbf{y}_2$ , and an edge from the node associated with  $\mathbf{y}_2$  to the node associated with  $\mathbf{y}_1$ . We can also give conditions for the minimality of the representations and present algorithms on the construction of the representations.

This result can be extended to more general graphs with 3 and more nodes [136, 134]. More precisely, let  $\mathbf{y} = [\mathbf{y}_1^T, \mathbf{y}_2^T, \dots, \mathbf{y}_n^T]^T$  and let us define the graph  $G_{\mathbf{y}}$  associated with  $\mathbf{y}$  as follows. We associate each component  $\mathbf{y}_i$  of  $\mathbf{y}$  with the node  $i$  of  $G_{\mathbf{y}}$  and there is no edge from node  $i$  to node  $j$ , if  $\mathbf{y}_i$  does not conditionally Granger cause  $\mathbf{y}_j$  with respect to the collection of the components of  $\mathbf{y}$  that correspond to the parent nodes of  $i$ . We then show

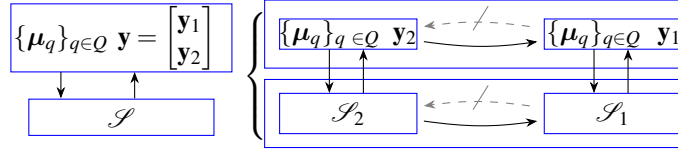


Figure 6.2: Illustration of the results: Cascade interconnection structure in a GBS  $S$  with input  $\{\mu_q\}_{q \in Q}$  and output  $\mathbf{y}$  decomposed into subsystems  $S_1$  and  $S_2$  in the presence of GB-Granger non-causality from  $\mathbf{y}_1$  to  $\mathbf{y}_2$  with respect to  $\{\mu_q\}_{q \in Q}$

that there exist an LTI-SS representation of  $\mathbf{y}$  whose network graph equals  $G_{\mathbf{y}}$ , and conversely, if there exists a LTI-SS representation of  $\mathbf{y}$  with a network graph  $G'$ , then  $G'$  coincides with  $G_{\mathbf{y}}$ .

The relationship described above for graphs with 2 nodes was extended to GBSs in [137]. More precisely, it can be shown that a process  $\mathbf{y} = [\mathbf{y}_1^T, \mathbf{y}_2^T]^T$  admits a specific GBS realization in block triangular form if and only if  $\mathbf{y}_1$  does not GB-Granger cause  $\mathbf{y}_2$ . The network graph of a GBS realization of  $\mathbf{y}$  in block triangular form has two nodes, corresponding to two subsystems generating  $\mathbf{y}_1$  and  $\mathbf{y}_2$ , and an edge from the node associated with  $\mathbf{y}_2$  to the node associated with  $\mathbf{y}_1$ , see Figure 6.2.

The results of this chapter partially settle a long dispute in neuroscience [242, 278, 93, 102]. There, the purpose is to detect and model interactions between brain regions using e.g., fMRI, EEG, MEG data. For this purpose, both statistical method (Granger causality) based methods [102] and state-space based methods [93] were used. In the former case, the presence of an interaction was identified with the presence of a statistical relationship (Granger causality) between the outputs associated with various brain regions. In the latter case, the presence of an interaction was interpreted as the presence of an edge in the network graph of a state-space representation, whose parameters were identified from data. However, the formal relationship between these methods was not always clear. This has led to a lively debate regarding the advantages/disadvantages of both methods [278, 70, 241]. The results of this chapter indicate that the two approaches are equivalent under some assumptions.

Besides the concept of network graph introduced in this chapter, there are several other notions for describing the structure of a system or the network of subsystems in a system. Examples of such notions are: feedback modeling [54, 53, 98, 124], dynamical structure function [104, 123, 310], dynamic causal modeling [93, 117, 201], and causality graphs [78, 79]. Also, besides Granger and GB-Granger causality, there are several examples for statistical notions that have essential role to understand the relation between stochastic processes. We can mention here conditional orthogonality [55, 56], transfer entropy [20] and directional mutual information [170, 151]. For a detailed discussion on the relationship between the results of this chapter and the cited papers see [133]. We would also like to mention [148, 149, 237, 196] on coordinated LTI-SSs, which served as inspiration for certain classes of sLTI-SSs studied in this chapter.

Finally, there is a large body of literature on identifying transfer functions which are interconnected according to some graph [297, 68, 298, 280, 50] and where all the signals exchanged by the transfer functions can be observed. This latter problem is quite different from the problem considered in this chapter.

**Outline:** In Section 6.2 we present an overview of main results on the relationship between Granger-causality and network structure of sLTI-SSs. In Section 6.3 we present an overview of the results relating GB-Granger causality with the network structure of GBSs.

## 6.2 Granger causality: linear case

We will present an overview of the main result relating the graph defined by Granger-causality relationships with the network graph of sLTI-SSs. We start with the case of two processes in Subsection 6.2.2. Then in Subsection

6.2.3 we continue with the case of arbitrary number of processes network graph of which is start shaped. The most general case is treated in Subsection 6.2.4. We conclude by discussing briefly in Subsection 6.2.5 the application of the results to reverse engineering of network graphs. Before presenting the main results, in Subsection 6.2.1 we will introduce the necessary notation and terminology and briefly recall the relevant results from realization theory of sLTI-SS.

### 6.2.1 Technical preliminaries: linear stochastic realization theory

The notation and presentation of this section follows those of [136].

The discrete-time axis is the set of integers  $\mathbb{Z}$ . The random variable of a process  $\mathbf{z}$  at time  $t$  is denoted by  $\mathbf{z}(t)$ . If  $\mathbf{z}(t)$  is  $k$ -dimensional (for all  $t \in \mathbb{Z}$ ), then we write  $\mathbf{z} \in \mathbb{R}^k$  and we call  $k = \dim(\mathbf{z})$  the dimension of  $\mathbf{z}$ . The  $n \times n$  identity matrix is denoted by  $I_n$  or by  $I$  when its dimension is clear from the context. We denote by  $\mathcal{H}$  the Hilbert space of zero-mean square-integrable random variables, where the inner product between two random variables  $\mathbf{y}, \mathbf{z}$  is the covariance matrix  $E[\mathbf{y}\mathbf{z}^T]$ . The Hilbert space generated by a set  $U \subset \mathcal{H}$  is the smallest (w.r.t. set inclusion) closed subspace of  $\mathcal{H}$  which contains  $U$ . Consider a zero-mean square-integrable process  $\mathbf{z} \in \mathbb{R}^k$ . Then  $\mathcal{H}_{t-}^{\mathbf{z}}, \mathcal{H}_{t+}^{\mathbf{z}}, \mathcal{H}_t^{\mathbf{z}}, t \in \mathbb{Z}$  are the Hilbert spaces generated by the sets  $\{\ell^T \mathbf{z}(s) \mid s \in \mathbb{Z}, s < t, \ell \in \mathbb{R}^k\}$ ,  $\{\ell^T \mathbf{z}(s) \mid s \in \mathbb{Z}, s \geq t, \ell \in \mathbb{R}^k\}$ , and  $\{\ell^T \mathbf{z}(t) \mid \ell \in \mathbb{R}^k\}$ , respectively. If  $\mathbf{z}_1, \dots, \mathbf{z}_n$  are vector valued processes, then  $\mathbf{z} = [\mathbf{z}_1^T, \dots, \mathbf{z}_n^T]^T$  denotes the process defined by  $\mathbf{z}(t) = [\mathbf{z}_1^T(t), \dots, \mathbf{z}_n^T(t)]^T, t \in \mathbb{Z}$ .

If  $\mathbf{z}(t) \in \mathcal{H}$  is a random variable and  $U$  is a closed subspace in  $\mathcal{H}$ , then we denote by  $E_i[\mathbf{z}(t)|U]$  the orthogonal projection of  $\mathbf{z}(t)$  onto  $U$ . The orthogonal projection onto  $U$  of a random variable  $\mathbf{z}(t) = [\mathbf{z}_1(t), \dots, \mathbf{z}_k(t)]^T$  taking values in  $\mathbb{R}^k$  is denoted by  $E_i[\mathbf{z}(t)|U]$  and defined element-wise as  $E_i[\mathbf{z}(t)|U] := [\hat{\mathbf{z}}_1(t), \dots, \hat{\mathbf{z}}_k(t)]^T$ , where  $\hat{\mathbf{z}}_i(t) = E_i[\mathbf{z}_i(t)|U], i = 1, \dots, k$ . That is,  $E_i[\mathbf{z}(t)|U]$  is the random variable with values in  $\mathbb{R}^k$  obtained by projecting the coordinates of  $\mathbf{z}(t)$  onto  $U$ . Accordingly, the orthogonality of a multidimensional random variable to a closed subspace in  $\mathcal{H}$  is meant element-wise. The orthogonal projection of a closed subspace  $U \subseteq \mathcal{H}$  onto a closed subspace  $V \subseteq \mathcal{H}$  is written by  $E_i[U|V] := \{E_i[u|V], u \in U\}$ . Note that for jointly Gaussian processes  $\mathbf{y}$  and  $\mathbf{z}$  the orthogonal projection  $E_i[\mathbf{y}(t)|\mathcal{H}_t^{\mathbf{z}}$  is equivalent with the conditional expectation of  $\mathbf{y}(t)$  given  $\mathbf{z}(t)$ . A stochastic process is called *zero-mean square-integrable with rational spectrum (abbreviated by ZMSIR)* if it is weakly-stationary, square-integrable, zero-mean, full rank, purely non-deterministic, and its spectral density is a proper rational function.

A stochastic stochastic time-invariant linear state-space representation (sLTI-SS for short) is a stochastic dynamical system of the form

$$\begin{aligned} \mathbf{x}(t+1) &= \mathbf{A}\mathbf{x}(t) + \mathbf{B}\mathbf{v}(t) \\ \hat{\mathbf{y}}(t) &= \mathbf{C}\mathbf{x}(t) + \mathbf{D}\mathbf{v}(t) \end{aligned} \tag{6.1}$$

where  $\mathbf{A} \in \mathbb{R}^{n \times n}, \mathbf{B} \in \mathbb{R}^{n \times m}, \mathbf{C} \in \mathbb{R}^{p \times n}, \mathbf{D} \in \mathbb{R}^{p \times m}$  for  $n \geq 0, m, p > 0$  and  $\mathbf{x} \in \mathbb{R}^n, \mathbf{y} \in \mathbb{R}^p, \mathbf{v} \in \mathbb{R}^m$  are ZM-SIR processes. The processes  $\mathbf{x}, \mathbf{y}$  and  $\mathbf{v}$  are called state, output and noise process, respectively. Furthermore, we require that  $\mathbf{A}$  is stable (all its eigenvalues are inside the open unit circle) and that for any  $t, k \in \mathbb{Z}, k \geq 0, E[\mathbf{v}(t)\mathbf{v}^T(t-k-1)] = 0, E[\mathbf{v}(t)\mathbf{x}^T(t-k)] = 0$ , i.e.,  $\mathbf{v}(t)$  is white noise and uncorrelated with  $\mathbf{x}(t-k)$ . In (6.1) the state process  $\mathbf{x}$  is uniquely determined by the noise process  $\mathbf{v}$  and the system matrices  $\mathbf{A}, \mathbf{B}, \mathbf{C}, \mathbf{D}$  so that  $\mathbf{x}(t) = \sum_{k=0}^{\infty} \mathbf{A}^k \mathbf{B} \mathbf{v}(t-k)$ , where the convergence of the infinite sum is understood in the mean square sense. Hence, an sLTI-SS of the form (6.1) will be identified with the tuple  $(\mathbf{A}, \mathbf{B}, \mathbf{C}, \mathbf{D}, \mathbf{v})$ . Following the classical terminology, we call the dimension of the state process the *dimension* of (6.1). We will say that the sLTI-SS of the form (6.1) is a *realization* of a process  $\mathbf{y}$ , if  $\hat{\mathbf{y}} = \mathbf{y}$ , if the output process of the sLTI-SS equals  $\mathbf{y}$ . A sLTI-SS realization of  $\mathbf{y}$  is called *minimal* realization of  $\mathbf{y}$ , if it has minimal dimension among all sLTI-SSs which are realizations of  $\mathbf{y}$ .

sLTI-SS realizations of a given process  $\mathbf{y}$  are strongly related to LTI-SS realizations of the covariance sequence

$$\{\Lambda_k^{\mathbf{y}} := E[\mathbf{y}(t+k)\mathbf{y}^T(t)]\}_{k=0}^{\infty},$$

see [163, Section 6.2]. Below we briefly sketch this relationship, as it plays an important role in deriving the

## 6.2. GRANGER CAUSALITY: LINEAR CASE

results of the paper. Consider a sLTI-SS  $(A, B, C, D, \mathbf{v})$  which is a realization of  $\mathbf{y}$ . Denote the (time-independent<sup>1</sup>) noise variance matrix by  $\Lambda_0^{\mathbf{y}} = E[\mathbf{v}(t)\mathbf{v}^T(t)]$ . Then, the variance matrix  $\Lambda_0^{\mathbf{x}} = E[\mathbf{x}(t)\mathbf{x}^T(t)]$  of the state process  $\mathbf{x}$  of  $(A, B, C, D, \mathbf{v})$  is the unique symmetric solution of the Lyapunov equation

$$\Sigma = A\Sigma A^T + B\Lambda_0^{\mathbf{y}}B^T \quad (6.2)$$

and the covariance  $G := E[\mathbf{y}(t)\mathbf{x}^T(t+1)]$  satisfies

$$G = C\Lambda_0^{\mathbf{x}}A^T + D\Lambda_0^{\mathbf{y}}B^T. \quad (6.3)$$

In light of this, the Markov parameters of the input-output map of the LTI-SS  $(A, G^T, C, \Lambda_0^{\mathbf{y}})$  are equal to the covariances  $\{\Lambda_k^{\mathbf{y}}\}_{k=0}^{\infty}$ . More precisely,

$$\Lambda_k^{\mathbf{y}} = CA^{k-1}G^T \quad k > 0. \quad (6.4)$$

Therefore, sLTI-SS realizations of  $\mathbf{y}$  yield LTI-SSs Markov parameters of which are the covariances  $\{\Lambda_k^{\mathbf{y}}\}_{k=0}^{\infty}$  of  $\mathbf{y}$ . Conversely, deterministic LTI-SS systems whose Markov parameters are the covariances  $\{\Lambda_k^{\mathbf{y}}\}_{k=0}^{\infty}$  yield an stochastic linear system of  $\mathbf{y}$ . To this end, we use the following terminology: Recall that  $\mathcal{H}_{t-}^{\mathbf{y}}$  denotes the Hilbert space generated by  $\mathbf{y}(t-k)$ ,  $k > 0$ . We call the process

$$\mathbf{e}(t) := \mathbf{y}(t) - E_t[\mathbf{y}(t) | \mathcal{H}_{t-}^{\mathbf{y}}], \quad \forall t \in \mathbb{Z}$$

the (*forward*) *innovation process* of  $\mathbf{y}$ . Assume now that  $(A, G^T, C, \Lambda_0^{\mathbf{y}})$  is a stable minimal deterministic LTI-SS system whose Markov parameters are the covariances of  $\mathbf{y}$ , i.e., (6.4) holds. Let  $\Sigma_{\mathbf{x}}$  be the minimal symmetric solution<sup>2</sup> of the algebraic Riccati equation

$$\Sigma = A\Sigma A^T + (G^T - A\Sigma C^T)(\Delta(\Sigma))^{-1}(G^T - A\Sigma C^T)^T, \quad (6.5)$$

where  $\Delta(\Sigma) = (\Lambda_0^{\mathbf{y}} - C\Sigma C^T)$  and set  $K$  as

$$K := (G^T - A\Sigma_{\mathbf{x}}C^T)(\Lambda_0^{\mathbf{y}} - C\Sigma_{\mathbf{x}}C^T)^{-1}. \quad (6.6)$$

**Proposition 6.1.** [147, Section 7.7] *Let  $K$  be as in (6.6) and  $\mathbf{e}$  be the innovation process of  $\mathbf{y}$ . Then the following sLTI-SS realization of  $\mathbf{y}$  is minimal:*

$$(A, K, C, I, \mathbf{e}) \quad (6.7)$$

Note that if  $\mathbf{x}$  is the state of  $(A, K, C, I, \mathbf{e})$ , then

$$\begin{aligned} \Sigma_{\mathbf{x}} &= E[\mathbf{x}(t)\mathbf{x}^T(t)] \\ E[\mathbf{e}(t)\mathbf{e}^T(t)] &= (\Lambda_0^{\mathbf{y}} - C\Sigma_{\mathbf{x}}C^T) \\ K &= E[\mathbf{x}(t+1)\mathbf{e}^T(t)]E[\mathbf{e}(t)\mathbf{e}^T(t)]^{-1} \end{aligned}$$

and  $K$  is the gain of the steady-state Kalman filter [163, Section 6.9]. This motivates the following definition: Let  $\mathbf{e}, \mathbf{y} \in \mathbb{R}^p$  be ZMSIR processes and  $A \in \mathbb{R}^{n \times n}, K \in \mathbb{R}^{n \times p}, C \in \mathbb{R}^{p \times n}, D \in \mathbb{R}^{p \times p}$ . A sLTI-SS  $(A, K, C, D, \mathbf{e})$  such that it is a realization of  $\mathbf{y}$  and where  $\mathbf{e}$  is the innovation process of  $\mathbf{y}$  and  $D = I_p$  is called *forward innovation representation of  $\mathbf{y}$*  or a *stochastic LTI realization of  $\mathbf{y}$  in forward innovation form*. A forward innovation representation of  $\mathbf{y}$ , called *minimal forward innovation representation*, if it is a minimal dimensional stochastic LTI realization of  $\mathbf{y}$ . The representation in Proposition 6.1 is a minimal forward innovation representation, thus we conclude that

**Proposition 6.2.** *Every ZMSIR process  $\mathbf{y}$  has a minimal forward innovation representation.*

<sup>1</sup>stationarity implies that the (co)variance matrices are time-independent

<sup>2</sup>for any other symmetric solution  $\tilde{\Sigma}$ , the matrix  $\tilde{\Sigma} - \Sigma$  is positive definite

In the sequel, we will use the following covariance realization algorithm for computing a minimal forward innovation representation of  $\mathbf{y}$ .

---

**Algorithm 9 Covariance realization algorithm**

**Inputs:** an integer  $N > 0$  and the covariances  $\{\Lambda_k^{\mathbf{y}} := E[\mathbf{y}(t+k)\mathbf{y}^T(t)]\}_{k=0}^{N+1}$

**Output:** linear system  $\Sigma_N$

---

Construct the Hankel-matrix

$$H_{N,N+1} = \begin{bmatrix} \Lambda_1^{\mathbf{y}} & \Lambda_2^{\mathbf{y}} & \cdots & \Lambda_{N+1}^{\mathbf{y}} \\ \Lambda_2^{\mathbf{y}} & \Lambda_3^{\mathbf{y}} & \cdots & \Lambda_{N+1}^{\mathbf{y}} \\ \vdots & \vdots & \cdots & \vdots \\ \Lambda_N^{\mathbf{y}} & \Lambda_{N+1}^{\mathbf{y}} & \cdots & \Lambda_{2N}^{\mathbf{y}} \end{bmatrix}$$

Compute a decomposition  $H_{f,N,N+1} = \mathbf{O}\mathbf{R}$ , where  $\mathbf{O} \in \mathbb{R}^{Np \times n}$  and  $\mathbf{R} \in \mathbb{R}^{n \times (N+1)p}$  and  $\text{rank } \mathbf{R} = \text{rank } \mathbf{O} = n$ , Consider the decomposition

$$\mathbf{R} = [\mathbf{C}_1, \quad \dots, \quad \mathbf{C}_{N+1}],$$

such that  $\mathbf{C}_i \in \mathbb{R}^{n \times p}$ ,  $i = 1, 2, \dots, N+1$ , i.e., they are the block columns of  $\mathbf{R}$ . Define  $\bar{\mathbf{R}}, \hat{\mathbf{R}} \in \mathbb{R}^{n \times Np}$ ,

$$\begin{aligned} \bar{\mathbf{R}} &= [\mathbf{C}_1, \quad \dots, \quad \mathbf{C}_N], \\ \hat{\mathbf{R}} &= [\mathbf{C}_2, \quad \dots, \quad \mathbf{C}_{N+1}]. \end{aligned}$$

Construct  $\Sigma_N = (A, G, C)$  such that

$$\begin{aligned} G &= \text{the first } p \text{ columns of } \mathbf{R} \\ C &= \text{the first } p \text{ rows of } \mathbf{O} \\ A &= \hat{\mathbf{R}}\bar{\mathbf{R}}^+, \end{aligned}$$

where  $\bar{\mathbf{R}}^+$  is the Moore-Penrose pseudo-inverse of  $\bar{\mathbf{R}}$ .

**Step 4** Find the minimal symmetric solution  $\Sigma_x$  of the Riccati equation (6.5) (see e.g., [147, Section 7.4.2]).

**Step 5** Set  $K$  as in (6.6) and define  $\Lambda_0^{\mathbf{e}} := \Lambda_0^{\mathbf{y}} - C\Sigma_x C^T$ .

---

**Remark 6.1** (Correctness of Algorithms 9). Consider a ZMSIR process  $\mathbf{y}$  with covariance sequence  $\{\Lambda_k^{\mathbf{y}}\}_{k=0}^{\infty}$  and a sLTI-SS  $(\tilde{A}, \tilde{B}, \tilde{C}, \tilde{D}, \mathbf{v})$ , which is a realization of  $\mathbf{y}$ . Let  $\mathbf{e}$  be the innovation process of  $\mathbf{y}$  and  $N$  be larger than or equal to the dimension of a minimal stochastic linear system of  $\mathbf{y}$ . Then it follows from [147, Lemma 7.9, Section 7.7] that if  $\{A, K, C, \Lambda_0^{\mathbf{e}}\}$  is the output of Algorithm 9 with input  $\{\Lambda_k^{\mathbf{y}}\}_{k=0}^{2N}$  then  $(A, K, C, I, \mathbf{e})$  is a minimal forward innovation representation of  $\mathbf{y}$  and  $\Lambda_0^{\mathbf{e}} = E[\mathbf{e}(t)\mathbf{e}^T(t)]$ .

Minimal forward innovation representations have the following properties.

**Proposition 6.3.** [163, Proposition 8.6.3] A sLTI-SS  $(A, K, C, I, \mathbf{e})$  realization of  $\mathbf{y}$  in forward innovation form is minimal if and only if  $(A, K)$  is controllable and  $(A, C)$  is observable.

Proposition 6.3 shows that minimality of a sLTI-SS  $(A, K, C, I, \mathbf{e})$  realization of  $\mathbf{y}$  in forward innovation form can be characterized by minimality of the deterministic system  $(A, K, C, I)$ . In general, the characterization of minimality in stochastic linear systems is more involved, and it is related to the minimality of the deterministic LTI-SS system  $(A, G^T, C, \Lambda_0^{\mathbf{y}})$  associated with the stochastic stochastic linear system ([163, Corollary 6.5.5]). The following Proposition shows that minimal sLTI-SS realizations of  $\mathbf{y}$  in forward innovation form are *isomorphic*. Isomorphism is defined as follows: two sLTI-SS  $(A, K, C, I, \mathbf{e})$  and  $(\tilde{A}, \tilde{K}, \tilde{C}, I, \mathbf{e})$  are isomorphic if there exists a non-singular matrix  $T$  such that  $A = T\tilde{A}T^{-1}$ ,  $K = T\tilde{K}$  and  $\tilde{C} = CT^{-1}$ . Again, in general, the result does not apply for stochastic linear systems.

**Proposition 6.4.** [163, Theorem 6.6.1] *If  $(A, K, C, I, \mathbf{e})$  and  $(\tilde{A}, \tilde{K}, \tilde{C}, I, \mathbf{e})$  are minimal sLTI-SS realizations of  $\mathbf{y}$  in forward innovation form, then they are isomorphic.*

### 6.2.2 Classical Granger causality between two processes

Below we discuss the relationship between Granger-causality and existence of an sLTI-SS realization in so called block triangular form. The latter class of sLTI-SS will be defined later. The systems of that class have the property that their network graph has two nodes and one edge. We will follow the presentation of [136].

We start by defining the notion of Granger causality. Informally,  $\mathbf{y}_1$  does not Granger cause  $\mathbf{y}_2$ , if the best linear predictions of  $\mathbf{y}_2$  based on the past values of  $\mathbf{y}_2$  are the same as those based only on the past values of  $\mathbf{y}$ . Recall that  $\mathcal{H}_{t-}^{\mathbf{y}_2}$  denotes the Hilbert space generated by the past  $\{\mathbf{z}(t-k)\}_{k=1}^{\infty}$  of  $\mathbf{z}$ . Then, Granger causality is defined as follows:

**Definition 6.1** (Granger causality). *Consider a zero-mean square integrable, wide-sense stationary process  $\mathbf{y} = [\mathbf{y}_1^T, \mathbf{y}_2^T]^T$ . We say that  $\mathbf{y}_1$  does not Granger cause  $\mathbf{y}_2$  if for all  $t, k \in \mathbb{Z}$ ,  $k \geq 0$   $E_t[\mathbf{y}_2(t+k) | \mathcal{H}_{t-}^{\mathbf{y}_2}] = E_t[\mathbf{y}_2(t+k) | \mathcal{H}_{t-}^{\mathbf{y}}]$ . Otherwise, we say that  $\mathbf{y}_1$  Granger causes  $\mathbf{y}_2$ .*

The class of sLTI-SSs that Granger causality is associated with later on in this section is defined below.

**Definition 6.2.** *A sLTI-SS  $(A, K, C, I, \mathbf{e} = [\mathbf{e}_1^T, \mathbf{e}_2^T]^T)$  in forward innovation form, where  $\mathbf{e}_i \in \mathbb{R}^{r_i}$ ,  $i = 1, 2$ , is called a sLTI-SS in block triangular form, if*

$$A = \begin{bmatrix} A_{11} & A_{12} \\ 0 & A_{22} \end{bmatrix} \quad K = \begin{bmatrix} K_{11} & K_{12} \\ 0 & K_{22} \end{bmatrix} \quad C = \begin{bmatrix} C_{11} & C_{12} \\ 0 & C_{22} \end{bmatrix}, \quad (6.8)$$

where  $A_{ij} \in \mathbb{R}^{p_i \times p_j}$ ,  $K_{ij} \in \mathbb{R}^{p_i \times r_j}$ ,  $C_{ij} \in \mathbb{R}^{r_i \times p_j}$  and  $p_i \geq 0$  for  $i, j = 1, 2$ . If, in addition,  $(A_{22}, K_{22}, C_{22}, I_{r_2}, \mathbf{e}_2)$  is a minimal sLTI-SS realization of  $\mathbf{y}_2$  in forward innovation form, then  $(A, K, C, I, \mathbf{e}, \mathbf{y})$  is called a sLTI-SS in causal block triangular form.

The results that relate Granger causality to sLTI-SS in block triangular form are presented next, see [136, Theorem 1].

**Theorem 6.1** (Granger causality and network graphs with 2 nodes). *Consider the following statements for a ZMSIR process  $\mathbf{y} = [\mathbf{y}_1^T, \mathbf{y}_2^T]^T$ :*

1.  $\mathbf{y}_1$  does not Granger cause  $\mathbf{y}_2$ ;
2. there exists a minimal sLTI-SS of  $\mathbf{y}$  in causal block triangular form;
3. there exists a minimal sLTI-SS of  $\mathbf{y}$  in block triangular form;
4. there exists a sLTI-SS of  $\mathbf{y}$  in block triangular form;

Then  $1 \iff 2$ . If  $\mathbf{y}$  is coercive, then  $1 \iff 2 \iff 3 \iff 4$ .

Algorithm 10 below shows how to calculate minimal sLTI-SS in causal block triangular form from the covariance sequence of the output process.



**Algorithm 10** Minimal Kalman representation in causal block triangular form based on output covariances

**Input**  $\{\Lambda_k^y := E[\mathbf{y}(t+k)\mathbf{y}^T(t)]\}_{k=0}^{2N}$ : Covariance sequence of  $\mathbf{y} = [\mathbf{y}_1^T, \mathbf{y}_2^T]^T$

**Output**  $\{A, K, C\}$ : System matrices of (6.8)

**Step 1** Apply Algorithm 9 with input  $\{\Lambda_k^y\}_{k=0}^{2N}$  and denote its output by  $\{\hat{A}, \hat{K}, \hat{C}\}$ .

**Step 2** Let  $\hat{C} = [\hat{C}_1^T \ \hat{C}_2^T]^T$  be such that  $\hat{C}_i \in \mathbb{R}^{r_i \times n}$ . Calculate a non-singular matrix  $T$  such that

$$T\hat{A}T^{-1} = \begin{bmatrix} A_{11} & A_{12} \\ 0 & A_{22} \end{bmatrix}, \quad \hat{C}_2T^{-1} = [0 \ C_{22}], \quad (6.9)$$

where  $(A_{22}, C_{22})$  is observable.

**Step 3** Set  $A := T\hat{A}T^{-1}$ ,  $K := T\hat{K}$ ,  $C := \hat{C}T^{-1}$ .

**Remark 6.2** (Correctness of Algorithm 10). Consider a ZMSIR process  $\mathbf{y} = [\mathbf{y}_1^T, \mathbf{y}_2^T]^T$  with covariance sequence  $\{\Lambda_k^y\}_{k=0}^{\infty}$ . Let  $\mathbf{e}$  be the innovation process of  $\mathbf{y}$  and  $N$  be any number larger than or equal to the dimension of a minimal sLTI-SS realization of  $\mathbf{y}$ . If  $\mathbf{y}$  satisfies condition 1 of Theorem 6.1 and  $\{A, K, C\}$  is the output of Algorithm 10 with input  $\{\Lambda_k^y\}_{k=0}^{2N}$ , then  $(A, K, C, I, \mathbf{e})$  is a minimal sLTI-SS realization of  $\mathbf{y}$  in causal block triangular form.

### 6.2.3 Conditional Granger causality and sLTI-SS in coordinated form

The result above can be generalized to more than two output processes. Let us start by presenting the generalization to the case when the network graph is star like Figure 6.3.

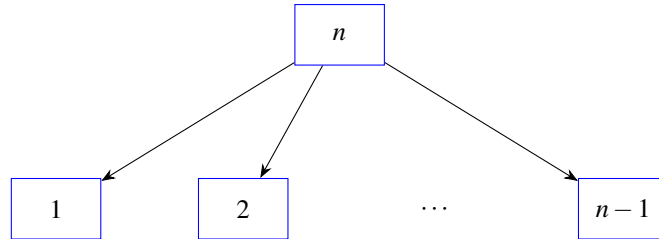


Figure 6.3: Star-like network graph

To this end, in the sequel, we assume that  $\mathbf{y} = [\mathbf{y}_1^T, \dots, \mathbf{y}_n^T]^T$  is a ZMSIR process, where  $n \geq 2$ ,  $\mathbf{y}_i \in \mathbb{R}^{r_i}$ , and  $r_i > 0$  for  $i = 1, \dots, n$ . We introduce following definition, which describes a subset of those sLTI-SSs, network graph of which is as in Figure 6.3.

**Definition 6.3.** A sLTI-SS  $(A, K, C, I, \mathbf{e} = [\mathbf{e}_1^T, \dots, \mathbf{e}_n^T]^T, \mathbf{y})$  realization of  $\mathbf{y}$ , where  $\mathbf{e}_i \in \mathbb{R}^{r_i}$ ,  $i = 1, \dots, n$ , is called a

## 6.2. GRANGER CAUSALITY: LINEAR CASE

sLTI-SS in coordinated form, if  $\mathbf{e}$  is the innovation process of  $\mathbf{y}$ , and

$$A = \left[ \begin{array}{cccc|c} A_{11} & 0 & \cdots & 0 & A_{1n} \\ 0 & A_{22} & \cdots & 0 & A_{2n} \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ 0 & 0 & \cdots & A_{(n-1)(n-1)} & A_{(n-1)n} \\ \hline 0 & 0 & \cdots & 0 & A_{nn} \end{array} \right] \quad K = \left[ \begin{array}{cccc|c} K_{11} & 0 & \cdots & 0 & K_{1n} \\ 0 & K_{22} & \cdots & 0 & K_{2n} \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ 0 & 0 & \cdots & K_{(n-1)(n-1)} & K_{(n-1)n} \\ \hline 0 & 0 & \cdots & 0 & K_{nn} \end{array} \right]$$

$$C = \left[ \begin{array}{cccc|c} C_{11} & 0 & \cdots & 0 & C_{1n} \\ 0 & C_{22} & \cdots & 0 & C_{2n} \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ 0 & 0 & \cdots & C_{(n-1)(n-1)} & C_{(n-1)n} \\ \hline 0 & 0 & \cdots & 0 & C_{nn} \end{array} \right] \quad (6.10)$$

where  $A_{ij} \in \mathbb{R}^{p_i \times p_j}$ ,  $K_{ij} \in \mathbb{R}^{r_i \times r_j}$ ,  $C_{ij} \in \mathbb{R}^{r_i \times p_j}$  and  $p_i \geq 0$  for  $i, j = 1, \dots, n$ . If, in addition, for each  $i = 1, \dots, n-1$

$$\left( \begin{bmatrix} A_{ii} & A_{in} \\ 0 & A_{nn} \end{bmatrix}, \begin{bmatrix} K_{ii} & K_{in} \\ 0 & K_{nn} \end{bmatrix}, \begin{bmatrix} C_{ii} & C_{in} \\ 0 & C_{nn} \end{bmatrix}, I_{r_i+r_n}, \begin{bmatrix} \mathbf{e}_i \\ \mathbf{e}_n \end{bmatrix} \right) \quad (6.11)$$

is a minimal sLTI-SS realization of  $[\mathbf{y}_1^T, \mathbf{y}_n^T]^T$  in causal block triangular form, then  $(A, K, C, I, \mathbf{e}, \mathbf{y})$  is called a sLTI-SS in causal coordinated form.

If  $n = 2$ , then Definition 6.3 coincides with Definition 6.2 of sLTI-SS in block triangular form. Furthermore, if  $(A, K, C, I, \mathbf{e})$  is a sLTI-SS in causal coordinated form, then the dimensions of the block matrices  $A_{ij}, K_{ij}, C_{ij}$ ,  $i, j = 1, \dots, n$  are uniquely determined by  $\mathbf{y}$ . Definition 6.3 is based on the deterministic terminology [148, 237] and on the definition of Gaussian coordinated systems [148, 196]. The term *coordinated* is used because the sLTI-SS at hand can be viewed as consisting of several subsystems; one of which plays the role of a coordinator and the others play the role of agents. More precisely, let  $(A, K, C, I, \mathbf{e}, \mathbf{y})$  be a sLTI-SS in coordinated form as in (6.10) and let  $\mathbf{x} = [\mathbf{x}_1^T, \dots, \mathbf{x}_n^T]^T$  be its state such that  $\mathbf{x}_i \in \mathbb{R}^{p_i}$ ,  $i = 1, \dots, n$ . Then, for  $i = 1, \dots, n-1$

$$\mathcal{S}_{a_i} \begin{cases} \mathbf{x}_i(t+1) &= \sum_{j=\{i,n\}} A_{ij} \mathbf{x}_j(t) + K_{ij} \mathbf{e}_j(t) \\ \mathbf{y}_i(t) &= \sum_{j=\{i,n\}} C_{ij} \mathbf{x}_j(t) + \mathbf{e}_i(t) \end{cases} \quad (6.12)$$

$$\mathcal{S}_c \begin{cases} \mathbf{x}_n(t+1) &= A_{nn} \mathbf{x}_n(t) + K_{nn} \mathbf{e}_n(t) \\ \mathbf{y}_n(t) &= C_{nn} \mathbf{x}_n(t) + \mathbf{e}_n(t) \end{cases} \quad (6.13)$$

Notice that subsystem  $\mathcal{S}_{a_i}$  generates  $\mathbf{y}_i$  as output, has  $\mathbf{x}_i, \mathbf{e}_i$  as its state and noise process and takes  $\mathbf{x}_n, \mathbf{e}_n$  as its inputs, thus takes inputs from subsystem  $\mathcal{S}_c$ . In contrast,  $\mathcal{S}_c$  is autonomous, generating  $\mathbf{y}_n$  as output and having  $\mathbf{x}_n, \mathbf{e}_n$  as its state and noise process but not taking input from subsystems  $\mathcal{S}_{a_i}$ ,  $i = 1, \dots, n-1$  (see Figure 6.4). We call  $\mathcal{S}_c$  the *coordinator* and  $\mathcal{S}_{a_i}$  with  $i = 1, \dots, n-1$  the *agents*. Intuitively, the agents do not communicate with each other, only the coordinator sends information ( $\mathbf{x}_n$  and  $\mathbf{e}_n$ ) to all agents and does not receive information from them. Next, we define the notion of conditional Granger-causality, which will play a central role in characterizing existence of sLTI-SS realizations in coordinated form.

**Definition 6.4.** Consider a ZMSIR process  $\mathbf{y} = [\mathbf{y}_1^T, \mathbf{y}_2^T, \mathbf{y}_3^T]^T$ . We say that  $\mathbf{y}_1$  conditionally does not Granger cause  $\mathbf{y}_2$  with respect to  $\mathbf{y}_3$ , if for all  $t, k \in \mathbb{Z}$ ,  $k \geq 0$

$$E_t[\mathbf{y}_2(t+k) \mid \mathcal{H}_t^{\mathbf{y}_2, \mathbf{y}_3}] = E_t[\mathbf{y}_2(t+k) \mid \mathcal{H}_t^{\mathbf{y}_1, \mathbf{y}_2, \mathbf{y}_3}].$$

Otherwise, we say that  $\mathbf{y}_1$  conditionally Granger causes  $\mathbf{y}_2$  with respect to  $\mathbf{y}_3$ .

New we are ready to state the main result relating conditional Granger-causality with existence of sLTI-SSs in coordinated form.

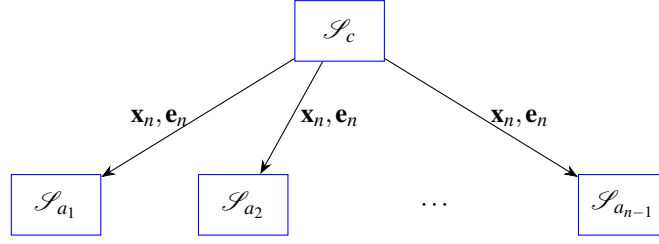


Figure 6.4: Network graph of a sLTI-SS in coordinated form:  $\mathcal{S}_c$  is the coordinator (6.13) and  $\mathcal{S}_{a_i}, i = 1, \dots, n-1$  are the agents (6.12).

**Theorem 6.2** (Conditional Granger-causality and star-shaped network graphs, [136, 133]). *Consider the following statements for a ZMSIR process  $\mathbf{y} = [\mathbf{y}_1^T, \dots, \mathbf{y}_n^T]^T$ :*

1.  $\mathbf{y}_i$  does not Granger cause  $\mathbf{y}_n, i = 1, \dots, n-1$ ;
2.  $\mathbf{y}_i$  conditionally does not Granger cause  $\mathbf{y}_j$  with respect to  $\mathbf{y}_n, i, j = 1, \dots, n-1, i \neq j$ ;
3. there exists a minimal sLTI-SS realization of  $\mathbf{y}$  in causal coordinated form;
4. there exists a sLTI-SS realization of  $\mathbf{y}$  in causal coordinated form;
5. there exists a sLTI-SS realization of  $\mathbf{y}$  in coordinated form;

Then, the following hold:

- 1 and 2  $\iff$  4.
- If, in addition,  $\mathbf{y}$  is coercive, then we have: 1 and 2  $\iff$  4  $\iff$  5.

It is possible to formulate an algorithm for computing a sLTI-SS in coordinated form from the covariances of  $\mathbf{y}$  or any sLTI-SS realization of  $\mathbf{y}$ , see [136, 133]. In fact, Theorem 6.2 and the corresponding algorithm for calculating a sLTI-SS in coordinated form can be used to formulate statistical tests for checking (conditional) Granger causality relationships, see [133, Chapter 7].

## 6.2.4 Directed acyclic graphs induced by Granger causality relations

To conclude, we present the general case, where the graph induced by Granger causality relations is a directed acyclic graph. To this end, we define the class of transitive acyclic graphs.

**Definition 6.5** (TADG). *A directed graph  $G = (V, E)$ , with set of nodes  $V = \{1, \dots, k\}$  and set of directed edges  $E \subseteq V \times V$  is called acyclic if there is no cycle i.e., closed directed path. Furthermore, it is transitive if for  $i, j, l \in V$  the implication  $(i, j), (j, l) \in E \implies (i, l) \in E$  holds. The class of transitive acyclic directed graphs is denoted by TADG.*

For convenience we make the following assumption that applies for all ZMSIR processes throughout this section.

**Assumption 4.** *For a process  $\mathbf{y} = [\mathbf{y}_1^T, \dots, \mathbf{y}_n^T]^T$ , we assume that none of the components of  $\mathbf{y}$  is a white noise process, or equivalently, the dimension of a minimal sLTI-SS realization of  $\mathbf{y}_i$  is strictly positive for all  $i \in \{1, \dots, n\}$ .*

## 6.2. GRANGER CAUSALITY: LINEAR CASE

For a TADG  $G = (V = \{1, \dots, n\}, E)$ , the set of nodes  $V$  has a so-called *topological ordering*. By topological ordering we mean an ordering on  $V$  such that if  $(i, j) \in E$  is a directed edge then  $i > j$ . Throughout this chapter we use integers to represent nodes of graphs and, without the loss of generality, we assume the following:

**Assumption 5.** Consider a TADG  $G = (V, E)$  where  $V = \{1, \dots, n\}$ . Then  $(i, j) \in E$  implies  $i > j$ .

The class of TADGs will be used to define the class of sLTI-SS in forward innovation form network graph of which is a TADG. To define this class of sLTI-SSs formally, we need to introduce some new terminology.

**Notation 6.1** (parent and non-parent succeeding nodes). Let  $G = (V = \{1, \dots, n\}, E)$  be a TADG and consider a node  $j \in V$ . The set of parent nodes  $\{i \in V | (i, j) \in E\}$  of  $j$  is denoted by  $I_j$ . In addition, the set of non-parent succeeding (with respect to the topological ordering of  $V$ ) nodes  $\{i \in V | i > j, (i, j) \notin E\}$  of  $j$  is denoted by  $\bar{I}_j$ .

The topological ordering on the set of nodes of a TADG graph implies that  $I_j, \bar{I}_j \subseteq \{j+1, \dots, n\}$  for all  $j \in \{1, \dots, n-1\}$ . Furthermore, from the definition of  $\bar{I}_j$ , we have that  $I_j \cup \bar{I}_j = \{j+1, \dots, n\}$ . The next notation helps in referring to components of processes beyond the original partitioning of those processes.

**Notation 6.2** (sub-process). Consider the finite set  $V = \{1, \dots, n\}$  and a tuple  $J = (j_1, \dots, j_l)$  where  $j_1, \dots, j_l \in V$ . Then for a process  $\mathbf{y} = [\mathbf{y}_1^T, \dots, \mathbf{y}_n^T]^T$ , we denote the sub-process  $[\mathbf{y}_{j_1}^T, \dots, \mathbf{y}_{j_l}^T]^T$  by  $\mathbf{y}_{j_1, \dots, j_l}$  or by  $\mathbf{y}_J$ . By abuse of terminology, if  $J$  is a subset of  $V$  and not a tuple, then  $\mathbf{y}_J$  will mean process  $\mathbf{y}_\alpha$ , where  $\alpha$  is the tuple obtained by taking the elements of  $J$  in increasing order, i.e. if  $J = \{j_1, \dots, j_k\}$ ,  $j_1 < j_2 < \dots < j_k$ , then  $\alpha = (j_1, \dots, j_k)$ . However,  $\mathbf{y}_{\alpha, \beta}$  always means  $[\mathbf{y}_\alpha^T, \mathbf{y}_\beta^T]^T$  regardless the topological order between the elements of  $\alpha$  and  $\beta$ .

Next, we introduce what we mean by partition of matrices. Call the set  $\{p_i, q_i\}_{i=1}^k$  a *partition* of  $(p, q)$ , where  $p, q > 0$ , if  $\sum_{i=1}^k p_i = p$  and  $\sum_{i=1}^k q_i = q$ , where  $p_i, q_i > 0$  for  $i = 1, \dots, k$ .

**Definition 6.6** (partition of a matrix). Let  $\{p_i, q_i\}_{i=1}^k$  be a partition of  $(p, q)$  for some  $p, q > 0$ . Then the partition of a matrix  $M \in \mathbb{R}^{p \times q}$  with respect to  $\{p_i, q_i\}_{i=1}^k$  is a collection of matrices  $\{M_{ij} \in \mathbb{R}^{p_i \times q_j}\}_{i,j=1}^k$ , such that

$$M = \begin{bmatrix} M_{11} & \cdots & M_{1k} \\ \vdots & \ddots & \vdots \\ M_{k1} & \cdots & M_{kk} \end{bmatrix}.$$

In Definition 6.6, the indexing of matrix  $M$  refers to the blocks of  $M$  and does not refer directly to the elements of  $M$ . It is parallel to the component-wise indexing of processes where the components can be multidimensional.

**Notation 6.3** (sub-matrix). Consider the partition  $\{M_{ij} \in \mathbb{R}^{p_i \times q_j}\}_{i,j=1}^k$  of a matrix  $M \in \mathbb{R}^{p \times q}$  with respect to the partition  $\{p_i, q_i\}_{i=1}^k$  of  $(p, q)$ . Furthermore, consider the tuples  $I = (i_1, \dots, i_n)$  and  $J = (j_1, \dots, j_m)$  where  $i_1, \dots, i_n, j_1, \dots, j_m \in \{1, \dots, k\}$ . Then by the sub-matrix of  $M$  indexed by  $IJ$  we mean

$$M_{IJ} := \begin{bmatrix} M_{i_1 j_1} & \cdots & M_{i_1 j_m} \\ \vdots & \ddots & \vdots \\ M_{i_n j_1} & \cdots & M_{i_n j_m} \end{bmatrix}$$

We are now ready to define sLTI-SSs which have a so-called TADG-zero structure:

**Definition 6.7** ( $G$ -zero structure). Consider a process  $\mathbf{y} = [\mathbf{y}_1^T, \dots, \mathbf{y}_n^T]^T$  and a TADG

$$G = (V = \{1, \dots, n\}, E).$$

Let

$$(A, K, C, I, \mathbf{e})$$

be a  $p$ -dimension sLTI-SS realization of  $\mathbf{y} \in \mathbb{R}^r$  in forward innovation form. Consider the partition of

- $A$  with respect to  $\{p_i, p_i\}_{i=1}^n$ ,
- $K$  with respect to  $\{p_i, r_i\}_{i=1}^n$ ,
- $C$  with respect to  $\{r_i, p_i\}_{i=1}^n$

where  $\{p_i, r_i\}_{i=1}^n$  is a partition of  $(p, r)$ . Then we say that  $(A, K, C, I, \mathbf{e})$  has  $G$ -zero structure if

$$A_{ij} = 0, K_{ij} = 0, C_{ij} = 0 \text{ for all } (j, i) \notin E.$$

If, in addition, for all  $j \in V$ , the tuple  $J := (j, \bar{I}_j, I_j)$  defines a sLTI-SS

$$(A_{JJ}, K_{JJ}, C_{JJ}, I, [\mathbf{e}_j^T, \mathbf{e}_{\bar{I}_j}^T, \mathbf{e}_{I_j}^T]^T)$$

which is a realization of  $[\mathbf{y}_j^T, \mathbf{y}_{\bar{I}_j}^T, \mathbf{y}_{I_j}^T]^T$  in causal coordinated form (see Definition 6.3), then we say that  $(A, K, C, I, \mathbf{e})$  has causal  $G$ -zero structure.

Besides saying that a sLTI-SS has  $G$ -zero structure or causal  $G$ -zero structure, we also say, *sLTI-SS with  $G$ -zero structure or with causal  $G$ -zero structure*.

Consider the TADGs  $G_1 = (\{1, 2\}, \{(2, 1)\})$  and  $G_2 = (\{1, 2, \dots, n\}, \{(n, 1), (n, 2), \dots, (n, n-1)\})$ . If the graph  $G$  in Definition 6.7 is  $G_1$ , then Definition 6.7 coincides with Definition 6.2 considering ZMSIR processes that satisfy Assumption 4. In a similar manner, if the graph  $G$  in Definition 6.7 is  $G_2$  then it coincides with Definition 6.3 considering ZMSIR processes that satisfy Assumption 4.

If a  $p$ -dimensional sLTI-SS  $(A, K, C, I, \mathbf{e})$  of  $\mathbf{y} \in \mathbb{R}^r$  has causal  $G$ -zero structure, where  $G = (V, E)$  is a TADG, then the partition  $\{p_i, r_i\}_{i=1}^n$  of  $(p, r)$  in Definition 6.7 is uniquely determined by  $\mathbf{y}$ .

A sLTI-SS with TADG-zero structure can be viewed as consisting of subsystems where each subsystem generates a component of  $\mathbf{y} = [\mathbf{y}_1^T, \dots, \mathbf{y}_n^T]^T$ . More precisely, let  $G = (V = \{1, \dots, n\}, E)$  be a TADG and  $(A, K, C, I, \mathbf{e}, \mathbf{y})$  be a  $p$ -dimensional sLTI-SS with  $G$ -zero structure where  $A, K$  and  $C$  are partitioned with respect to a partition  $\{p_i, q_i\}_{i=1}^k$  of  $(p, q)$ . Furthermore, let  $\mathbf{x} = [\mathbf{x}_1^T, \dots, \mathbf{x}_n^T]^T$  be its state such that  $\mathbf{x}_i \in \mathbb{R}^{p_i}$ ,  $i = 1, \dots, n$ . Then the sLTI-SS with output  $\mathbf{y}_j$ ,  $j \in V$  is in the form of

$$\mathcal{S}_j \begin{cases} \mathbf{x}_j(t+1) = A_{jj}\mathbf{x}_j(t) + (A_{jI_j}\mathbf{x}_{I_j}(t) + K_{jI_j}\mathbf{e}_{I_j}(t)) + K_{jj}\mathbf{e}_j(t) \\ \mathbf{y}_j(t) = C_{jj}\mathbf{x}_j(t) + C_{jI_j}\mathbf{x}_{I_j}(t) + \mathbf{e}_j(t). \end{cases} \quad (6.14)$$

Notice that if  $(i, j) \in E$ , i.e.,  $i$  is a parent node of  $j$ , then subsystem  $\mathcal{S}_j$  takes inputs from subsystem  $\mathcal{S}_i$ , namely the state and noise processes of  $\mathcal{S}_i$ . In contrast, if  $(j, i) \notin E$ ,  $\mathcal{S}_j$  does not take input from  $\mathcal{S}_i$ . Intuitively, it means that the subsystems communicate with each other as it is allowed by the directed paths of the graph  $G$ . Note that from transitivity, if there is a directed path from  $i \in V$  to  $j \in V$  then there is also an edge  $(i, j) \in E$ .

**Example 6.1.** Take the TADG graph

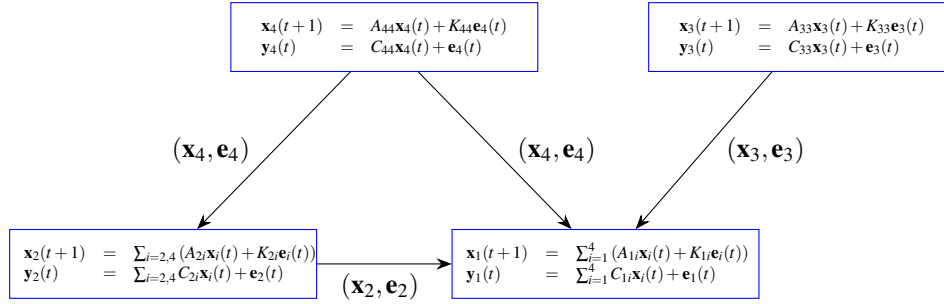
$$G = (\{1, 2, 3, 4\}, \{(4, 1), (4, 2), (3, 1), (2, 1)\})$$

and a process

$$[\mathbf{y}_1^T, \mathbf{y}_2^T, \mathbf{y}_3^T, \mathbf{y}_4^T]^T$$

with innovation process  $[\mathbf{e}_1^T, \mathbf{e}_2^T, \mathbf{e}_3^T, \mathbf{e}_4^T]^T$ . Then a sLTI-SS with  $G$ -zero structure of  $[\mathbf{y}_1^T, \mathbf{y}_2^T, \mathbf{y}_3^T, \mathbf{y}_4^T]^T$  is given by

$$\begin{aligned} \begin{bmatrix} \mathbf{x}_1(t+1) \\ \mathbf{x}_2(t+1) \\ \mathbf{x}_3(t+1) \\ \mathbf{x}_4(t+1) \end{bmatrix} &= \begin{bmatrix} A_{11} & A_{12} & A_{13} & A_{14} \\ 0 & A_{22} & 0 & A_{24} \\ 0 & 0 & A_{33} & 0 \\ 0 & 0 & 0 & A_{44} \end{bmatrix} \begin{bmatrix} \mathbf{x}_1(t) \\ \mathbf{x}_2(t) \\ \mathbf{x}_3(t) \\ \mathbf{x}_4(t) \end{bmatrix} + \begin{bmatrix} K_{11} & K_{12} & K_{13} & K_{14} \\ 0 & K_{22} & 0 & K_{24} \\ 0 & 0 & K_{33} & 0 \\ 0 & 0 & 0 & K_{44} \end{bmatrix} \begin{bmatrix} \mathbf{e}_1(t) \\ \mathbf{e}_2(t) \\ \mathbf{e}_3(t) \\ \mathbf{e}_4(t) \end{bmatrix} \\ \begin{bmatrix} \mathbf{y}_1(t) \\ \mathbf{y}_2(t) \\ \mathbf{y}_3(t) \\ \mathbf{y}_4(t) \end{bmatrix} &= \begin{bmatrix} C_{11} & C_{12} & C_{13} & C_{14} \\ 0 & C_{22} & 0 & C_{24} \\ 0 & 0 & C_{33} & 0 \\ 0 & 0 & 0 & C_{44} \end{bmatrix} \begin{bmatrix} \mathbf{x}_1(t) \\ \mathbf{x}_2(t) \\ \mathbf{x}_3(t) \\ \mathbf{x}_4(t) \end{bmatrix} + \begin{bmatrix} \mathbf{e}_1(t) \\ \mathbf{e}_2(t) \\ \mathbf{e}_3(t) \\ \mathbf{e}_4(t) \end{bmatrix}, \end{aligned} \quad (6.15)$$


 Figure 6.5: Network graph of the sLTI-SS (6.15) with  $G$ -zero structure

where  $A_{ij} \in \mathbb{R}^{p_i \times p_j}$ ,  $K_{ij} \in \mathbb{R}^{p_i \times r_j}$ ,  $C_{ij} \in \mathbb{R}^{r_i \times p_j}$  and  $\mathbf{y}_i, \mathbf{e}_i \in \mathbb{R}^{r_i}$ ,  $\mathbf{x}_i \in \mathbb{R}^{p_i}$  for some  $p_i > 0$ ,  $i, j = 1, 2, 3, 4$ . The network graph of this sLTI-SS is the network of the sLTI-SSs  $\mathcal{S}_1, \mathcal{S}_2, \mathcal{S}_3, \mathcal{S}_4$  defined in (6.14), generating  $\mathbf{y}_1, \mathbf{y}_2, \mathbf{y}_3$  and  $\mathbf{y}_4$ , respectively. See Figure 6.5 for illustration of this network graph.

**Theorem 6.3** (General network structure, [133, 134]). Consider the following statements for a TADG  $G = (V = \{1, \dots, n\}, E)$  and a process  $\mathbf{y} = [\mathbf{y}_1^T, \dots, \mathbf{y}_n^T]^T$ :

1.  $\mathbf{y}$  has  $G$ -consistent causality structure;
2. there exists a sLTI-SS realization of  $\mathbf{y}$  with causal  $G$ -zero structure;
3. there exists a sLTI-SS realization of  $\mathbf{y}$  with  $G$ -zero structure;

Then, the following hold:

- $1 \implies 3$ ;
- $2 \implies 1$ .
- If, in addition,  $\mathbf{y}$  is coercive, then we have  $1 \iff 2 \iff 3$ .

The proof of Theorem 6.3 is constructive and it leads to an algorithm for computing a sLTI-SS realization of  $\mathbf{y}$  from any sLTI-SS realization of  $\mathbf{y}$  or from the covariances of  $\mathbf{y}$ , see [134, 133] for more details.

### 6.2.5 Reverse engineering of the network graph

The results described above were used to develop statistical hypothesis tests for checking various Granger-causality based on data. The essence of those methods is to use empirical covariances of the observed processes to calculate a sLTI-SS in causal block triangular form, causal coordinated form, or with causal  $G$ -zero structure respectively. In the ideal case, when true covariances are used, suitable blocks of the matrices of such a sLTI-SS should be zero, if the corresponding Granger-causality relationships hold. When approximate covariance are used, the corresponding blocks are not zero, due to approximation error. The statistical hypothesis testing relies on checking if those blocks are significantly different from zero. Details of this approach can be found in [133, Chapter 7]. This statistical hypothesis testing procedure can also be viewed as a procedure for reverse engineering of the network graph. Note that the problem of discovering the network graph from data is in principle ill-posed, as systems with different network graphs can generate the same observed behavior. However, let us consider the weaker problem of deciding if observed behavior is consistent with the hypothesis that the underlying system has a particular network graph. That is, we want to decide if there exists a system with the given network graph which can generate the observed data. This problem is well-posed. The hypothesis testing procedure described above could be used to solve this weaker problem.

### 6.3 Granger causality: bilinear case

In this section we will use the terminology and notation of Chapter 3. In particular, we will use the notion of admissible inputs, ZMWSSI processes, and the processes  $\mathbf{z}_v^{y^+}$ ,  $\mathbf{z}_v^y$  defined in (3.15) and (3.20) respectively. More precisely, we consider the processes  $(\{\boldsymbol{\mu}_q\}_{q \in Q}, \mathbf{y} = [\mathbf{y}_1^T, \mathbf{y}_2^T]^T)$ , where  $\{\boldsymbol{\mu}_q\}_{q \in Q}$  is admissible and  $\mathbf{y}$  is ZMWSSI w.r.t.  $\{\mathbf{u}_q\}_{q \in Q}$ .

We would like to relate the existence of a GBS realization of  $\mathbf{y}$  with a network graph consisting of two nodes and one edge to Granger-causality-like properties of  $\mathbf{y}_1$  and  $\mathbf{y}_2$ . Recall from [57, 163, 147] that sLTI-SS realizations of  $\mathbf{y}$  in forward innovation form can be viewed as optimal predictors which map past values of  $\mathbf{y}$  to its best linear prediction. Since the classical definition of Granger-causality imposed conditions on such optimal linear predictors, it was possible to translate Granger-causality among components of  $\mathbf{y}$  to properties of sLTI-SS realizations of  $\mathbf{y}$ . If we try to extend the same idea to GBSs, then we have to deal with the problem that a GBS realization of  $\mathbf{y}$  in forward innovation form is not a linear predictor based on past values of  $\mathbf{y}$ , but it is a linear predictor from the past outputs and inputs  $\mathbf{z}_v^y(t)$  to the future outputs and inputs  $\mathbf{z}_v^{y^+}(t)$ , see This motivates our extension of Granger causality, where we use the process  $\mathbf{z}_w^{y^+}(t)$  rather than  $\mathbf{y}(t + |w|)$  and  $\mathbf{z}_v^y(t)$  rather than  $\mathbf{y}(t - |v|)$ ,  $v, w \in Q^+$ :

**Definition 6.8** (GB–Granger causality). *We say that  $\mathbf{y}_1$  does not GB–Granger cause  $\mathbf{y}_2$  w.r.t.  $\{\boldsymbol{\mu}_q\}_{q \in Q}$  if for all  $v \in Q^*$  and  $t \in \mathbb{Z}$*

$$E_t[\mathbf{z}_v^{y^+}(t) \mid \{\mathbf{z}_w^y(t)\}_{w \in Q^+}] = E_t[\mathbf{z}_v^{y^+}(t) \mid \{\mathbf{z}_w^y(t)\}_{w \in Q^+}]. \quad (6.16)$$

*Otherwise,  $\mathbf{y}_1$  GB–Granger causes  $\mathbf{y}_2$  w.r.t.  $\{\boldsymbol{\mu}_q\}_{q \in Q}$ . Here we used Notation 3.1 from Chapter 3.*

Informally,  $\mathbf{y}_1$  does not GB–Granger cause  $\mathbf{y}_2$ , if the best linear predictions of the future of  $\mathbf{y}_2$  w.r.t.  $\{\mathbf{u}_q\}_{q \in Q}$  along  $v$  is the same based on the past of  $\mathbf{y}$  or based on the past of  $\mathbf{y}_2$  w.r.t.  $\{\mathbf{u}_q\}_{q \in Q}$  along  $\{w\}_{w \in Q^+}$ .

**Remark 6.3.** *If  $\mathbf{y}_1$  does not GB–Granger cause  $\mathbf{y}_2$  then it implies that  $\mathbf{y}_1$  does not Granger causes  $\mathbf{y}_2$ . Moreover, in the specific case, when  $Q = \{1\}$  and  $\mathbf{u}_1(t) \equiv 1$ ,  $\mathbf{z}_v^{y^+}(t) = \mathbf{y}(t + |v|)$  and  $\mathbf{z}_w^y(t) = \mathbf{y}(t - |w|)$  and thus Definitions 6.1 and 6.8 coincide. The relationship between GB–Granger causality and other concepts of causality, such as conditional independence [9], seems to be more involved and remains a topic of future research.*

Next, we present the result on the relationship between GB–Granger causality and network graphs of GBSs. The GBS in question are minimal ones in forward innovation form that can be constructed algorithmically (see Algorithm 11 later on in this section).

**Theorem 6.4** (GB-Granger causality and network graphs with 2 nodes ). *Consider a GBS realization of  $\mathbf{y} = [\mathbf{y}_1^T, \mathbf{y}_2^T]^T$  and let  $\mathbf{e} = [\mathbf{e}_1^T, \mathbf{e}_2^T]^T$  be the innovation process of  $\mathbf{y}$  w.r.t.  $\{\mathbf{u}_q\}_{q \in Q}$ , where  $\mathbf{e}_i \in \mathbb{R}^{k_i}, i = 1, 2$ . Then,  $\mathbf{y}_1$  does not GB–Granger cause  $\mathbf{y}_2$  w.r.t.  $\{\mathbf{u}_q\}_{q \in Q}$  if and only if there exists a GBS  $\mathcal{G}$  of the form (3.2), such that  $\mathcal{G}$  is a minimal realization of  $\mathbf{y}$  in forward innovation form and for all  $q \in Q$ ,*

$$A_q = \begin{bmatrix} A_{q,11} & A_{q,12} \\ 0 & A_{q,22} \end{bmatrix}, K_q = \begin{bmatrix} K_{q,11} & K_{q,12} \\ 0 & K_{q,22} \end{bmatrix}, C = \begin{bmatrix} C_{11} & C_{12} \\ 0 & C_{22} \end{bmatrix}, \quad (6.17)$$

where  $A_{q,ij} \in \mathbb{R}^{n_i \times n_j}$ ,  $K_{q,ij} \in \mathbb{R}^{n_i \times k_j}$ ,  $C_{ij} \in \mathbb{R}^{k_i \times n_j}$ ,  $q \in Q$ ,  $i, j = 1, 2$  for some  $n_1 \geq 0$ ,  $n_2 > 0$  and

$$\mathcal{G}_2 = (n_2, p, p, Q, \mathbf{x}, \mathbf{e}, \{\boldsymbol{\mu}_q\}_{q \in Q}, \mathbf{y}_2, C_{22}, I, \{A_{q,22}, K_{q,22}\}_{q \in Q}),$$

*is a minimal GBS realization of  $\mathbf{y}_2$  in forward innovation form.*

If  $Q = \{1\}$  and  $\mathbf{u}_1(t) \equiv 1$ , then GBS reduce to sLTI-SSs and Definitions 6.1 and 6.8 coincide. As a result, Theorem 6.17 reduces to earlier results on sLTI-SSs and Granger causality (see [136, Theorem 1]).

### 6.3. GRANGER CAUSALITY: BILINEAR CASE

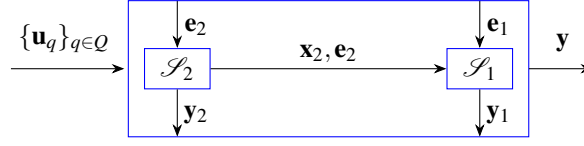


Figure 6.6: Cascade interconnection of a GBS realization in forward innovation form with system matrices as in (6.17).

A GBS realization of  $\mathbf{y}$  of the form (3.2) in forward innovation form which satisfies (6.17) can be viewed as a cascade interconnection of two subsystems. Define the subsystems

$$\begin{aligned} \mathcal{S}_1 \begin{cases} \mathbf{x}_1(t+1) &= \sum_{q \in Q} (A_{q,11} \mathbf{x}_1(t) + K_{q,11} \mathbf{e}_1(t)) \mathbf{u}_q(t) + \sum_{q \in Q} (A_{q,12} \mathbf{x}_2(t) + K_{q,12} \mathbf{e}_2(t)) \mathbf{u}_q(t) \\ \mathbf{y}_1(t) &= \sum_{i=1}^2 C_{1i} \mathbf{x}_i(t) + \mathbf{e}_1(t) \end{cases} \\ \mathcal{S}_2 \begin{cases} \mathbf{x}_2(t+1) &= (A_{q,22} \mathbf{x}_2(t) + K_{q,22} \mathbf{e}_2(t)) \mathbf{u}_q(t) \\ \mathbf{y}_2(t) &= C_{22} \mathbf{x}_2(t) + \mathbf{e}_2(t) \end{cases} \end{aligned}$$

Notice that  $\mathcal{S}_2$  sends its state  $\mathbf{x}_2$  and noise  $\mathbf{e}_2$  to  $\mathcal{S}_1$  as an external input while  $\mathcal{S}_1$  does not send information to  $\mathcal{S}_2$ . The corresponding network graph is illustrated in Figure 6.6.

The necessity part of the proof of Theorem 6.4 is constructive, and it is based on calculating a GBS realization in innovation form described in Theorem 6.4 from the observed behavior, i.e., from output covariances. For this calculation, we present Algorithm 11.

---

**Algorithm 11** Block triangular minimal GBS

---

**Input**  $\{\Lambda_w^{\mathbf{y}} := E[\mathbf{y}(t)(\mathbf{z}_w^{\mathbf{y}}(t))^T]\}_{w \in Q^+, |w| \leq n}$  and  $\{T_{q,q}^{\mathbf{y}} := E[\mathbf{z}_q^{\mathbf{y}}(t)(\mathbf{z}_q^{\mathbf{y}}(t))^T]\}_{q \in Q}$

**Output**  $(\{A_q, K_q\}_{q \in Q}, C)$

---

**Step 1** Apply Algorithm 4 from Subsection 3.6.3, Chapter 3 to compute a minimal GBS realization

$$\tilde{\mathcal{G}} = (n, p, m, Q, \mathbf{x}, \mathbf{e}, \{\boldsymbol{\mu}_q\}_{q \in Q}, \mathbf{y}, C, I, \{\tilde{A}_q, \tilde{K}_q\}_{q \in Q})$$

of  $\mathbf{y}$  in forward innovation form.

**Step 2** Define the sub-matrix consisting of the last  $k_2$  rows of  $\tilde{C}$  by  $\tilde{C}_2 \in \mathbb{R}^{k_2 \times n}$  and define the observability matrix  $\tilde{O}_{\mathbf{N}(n)}$

$$\tilde{O}_{\mathbf{N}(n)} = [(\tilde{C}_2 \tilde{A}_{v_1})^T \quad \dots \quad (\tilde{C}_2 \tilde{A}_{v_k})^T]^T,$$

where we used the lexicographic ordering ( $\prec$ ) on  $Q^*$  from Chapter 2, Notation 2.1. If  $\tilde{O}_{\mathbf{N}(n)}$  is not of full column rank then define the non-singular matrix  $T^{-1} = [T_1 \quad T_2]$  such that the columns of  $T_1 \in \mathbb{R}^{n \times n_1}$  is the kernel of  $\tilde{O}_{\mathbf{N}(n)}$ . If  $\tilde{O}_{\mathbf{N}(n)}$  is of full column rank, then set  $T = I$ .

Define the matrices  $A_q = T \tilde{A}_q T^{-1}$ ,  $K_q = T \tilde{K}_q$  for  $q \in Q$  and  $C = \tilde{C} T^{-1}$ .

**return**  $(\{A_q, K_q\}_{q \in Q}, C)$ .

---

**Remark 6.4.** It follows that if  $\mathbf{y}_1$  does not GB-Granger cause  $\mathbf{y}_2$ , then Algorithm 11 calculates the system matrices of a GBS realization described in Theorem 6.4, if  $\mathbf{y}$  can be realized by a stationary GBS of dimension  $n$ . Hence, Algorithm 11 provides a constructive proof of the necessity part of Theorem 6.4, by calculating a minimal GBS realization in forward innovation form that characterizes GB-Granger non-causality.



**Remark 6.5** (Checking GB–Granger causality). *Algorithm 11 can be used for checking GB–Granger causality as follows. Apply Algorithm 11 and check if the matrices  $\{A_q, K_q\}_{q \in Q}$  and  $C$  returned by Algorithm 11 satisfy,*

$$A_q = \begin{bmatrix} A_{q,11} & A_{q,12} \\ 0 & A_{q,22} \end{bmatrix}, \quad C = \begin{bmatrix} C_{11} & C_{12} \\ 0 & C_{22} \end{bmatrix},$$

$$K_q = \begin{bmatrix} K_{q,11} & K_{q,12} \\ 0 & K_{q,22} \end{bmatrix},$$

where  $A_{q,ij} \in \mathbb{R}^{n_i \times n_j}$ ,  $C_{ij} \in \mathbb{R}^{k_i \times n_j}$ ,  $K_{i,j} \in \mathbb{R}^{n_i \times k_j}$ ,  $i, j = 1, 2$  for some  $n_1 \geq 0$ ,  $n_2 > 0$ , and if

$$\mathcal{S}_2 = (n_2, p, p, Q, \mathbf{x}_2, \mathbf{e}, \{\boldsymbol{\mu}_q\}_{q \in Q}, \mathbf{y}_2, C_{22}, I, \{A_{q,22}, K_{q,22}\}_{q \in Q})$$

is a minimal GBS realization of  $\mathbf{y}_2$  in forward innovation form. By Theorem 6.4 both tests are positive, if and only if  $\mathbf{y}_1$  does not GB–Granger cause  $\mathbf{y}_2$ . We check whether  $\mathcal{S}_2$  is a minimal GBS realization in innovation form as follows. We use Subsection 3.6.3 compute a minimal GBS realization  $\tilde{\mathcal{S}}_2$  of  $\mathbf{y}_2$  in forward innovation form and the covariances  $\tilde{Q}_q = E[\mathbf{v}(t)\mathbf{v}^T(t)\mathbf{u}_q^2(t)]$ ,  $q \in Q$  of the innovation process  $\mathbf{v}$  of  $\mathbf{y}_2$ . Then,  $\mathcal{S}_2$  is a minimal GBS realization in forward innovation form, if and only if  $\mathcal{S}_2$  and  $\tilde{\mathcal{S}}_2$  have the same dimension and the same noise process, i.e.,  $\mathbf{v} = \mathbf{e}_2$ . For checking the latter, we remark that  $\mathbf{v}(t) = \mathbf{e}_2(t)$  if and only if for all  $i = 1, \dots, k_2$ ,  $\sum_{q \in Q} \alpha_q^2 \tilde{Q}_{q,ii} = \sum_{q \in Q} \alpha_q^2 Q_{q,(k_1+i)(k_1+i)}$ , where  $\{\alpha_q\}_{q \in Q}$  are such that  $\sum_{q \in Q} \alpha_q \boldsymbol{\mu}_q = \mathbf{1}$ <sup>3</sup> and  $Q_{q,rl}$ ,  $\tilde{Q}_{q,kl}$  denotes the  $(k, l)$ th entry of the matrix  $Q_q, \tilde{Q}_q$  respectively.

Note that the covariances used as inputs of Algorithm 11 could be estimated from sampled data, the procedure above could be a starting point of a statistical test for checking GB–Granger causality, similar to the one of Granger causality in [133]. This remains a topic of future research.

## 6.4 Role of realization theory

The results of Section 6.2 depend on stochastic realization theory [163] of sLTI-SSs. In particular, the existence of minimal of sLTI-SSs in forward innovation form is the main tool which allows us to related Granger-causality with the network graph of sLTI-SS realizations. Note that the notion of Granger-causality plays an important role in realization theory of sLTI-SSs with inputs. The relationship between the results of Section 6.2 and realization theory of sLTI-SSs with inputs remains a topic of future research. Likewise, the results of Section 6.3 rely on realization theory of GBSs, and especially on the existence of minimal GBSs in forward innovation form. Both for sLTI-SSs and for GBSs, the algorithms for checking Granger-causality rely on covariance realization and minimization algorithms.

## 6.5 Conclusions and future work

In this chapter we have presented an overview of results on the relationship between network graphs of state-space representations and Granger causality relations among their outputs. We have discussed in detail the results on sLTI-SS and GBSs. Similar results can be formulated for the relationship between transfer functions and Granger causality relations, [135]. These results could be used for developing statistical hypothesis testing for deciding if the network graph of the system which generates the data has certain interconnections [133].

There are many potential directions for further research. First of all, it would be desirable to extend the results to sLTI-SSs and to LPV-SSAs/LSSs with control inputs. Reformulations of the results in purely deterministic setting would also be of interest. The latter would require reformulating Granger-causality for the deterministic setting. Another direction would be the use of statistical notions which are different from Granger-causality. That remains topic of future research as well. Finally, more work needs to be done on statistical hypothesis

<sup>3</sup>Such  $\{\alpha_q\}_{q \in Q}$  exist due to the definition of admissible inputs, see Section 3.4

## 6.5. CONCLUSIONS AND FUTURE WORK

---

testing methods for deciding if the data could be generated by a system with a certain network graph. The latter problem is essentially the well-posed version of the generally ill-posed problem of reverse engineering of network graphs. First steps towards this goal were made in [133]. However, a formal theoretical analysis and experimental validation of the methods from [133] is still lacking.



## **Part III**

# **Future research**



## Chapter 7

# Future research directions

Recall that my global research objective is to provide algorithms and theory for building models for control. In fact, I plan to continue this research program, since there exist several open problems which need to be solved. In this chapter, I will first describe some of these open research problems which I plan to work on. This line of research represents **Axis A** (reliable models for control of cyber-physical systems) of my future research. In addition, I will also work on some new research topics, which represent a departure from my previous research: namely

- **Axis B: Reliable AI (artificial intelligence) for cyber-physical systems.**

There is an increasing need for solving decision making problems for cyber-physical systems which involve models from artificial intelligence. Traditionally, this topic is studied in computer science/artificial intelligence, and significant progress has been made. However, when AI algorithms are applied to cyber-physical systems, the behavior of physical components should be taken into account. This calls for methods and theory for constructing such models from data and for simplifying them. In turn, this is likely to require elements of control theory to be integrated into the analysis of such algorithms. Think for example of obstacle detection and collision avoidance of autonomous vehicles. In a way, the general research program of Subsection 1.1 still applies, with the difference that instead of gearing modeling methods and theory towards controller synthesis we gear them towards more general decision making algorithms. The challenges remain the same: we would like to develop methods for learning models from data, such that the resulting models can be proven to be useful for decision making. In particular, I plan to focus on analyzing machine learning algorithms which are applied to dynamical systems. This includes learning recurrent neural networks and reinforcement learning.

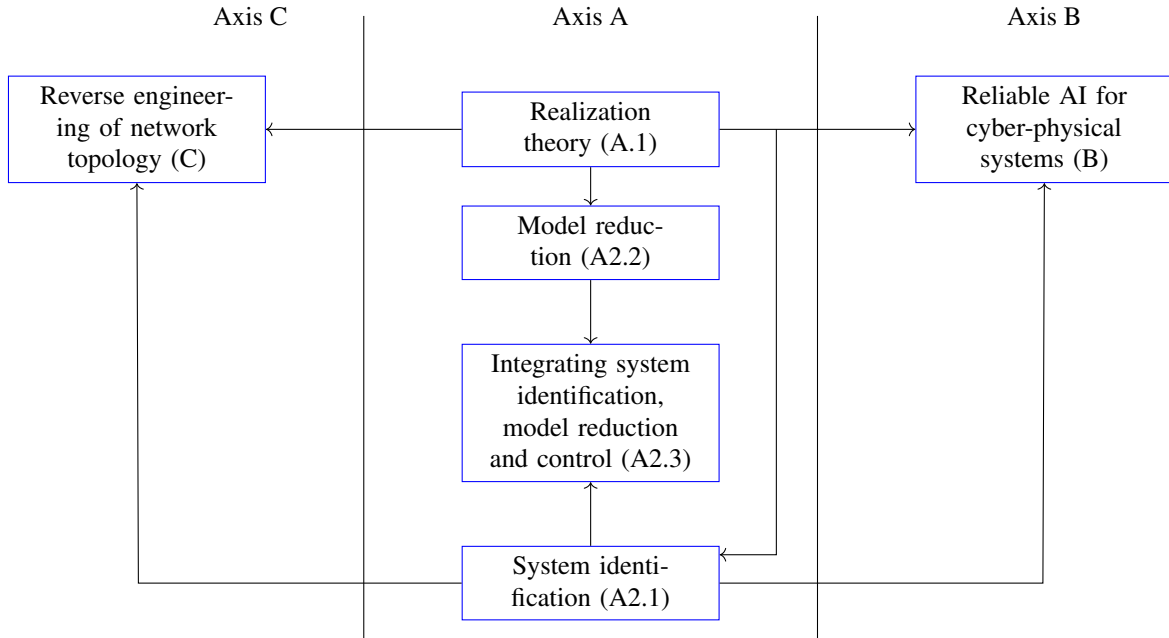
- **Axis C: Reverse engineering of network topology of interconnected systems.**

The goal is to provide theoretical guarantees and algorithms for learning the network topology of interconnected systems. This is motivated by the large number of applications of this problem in systems biology and neuroscience.

Note that while these two topics seem distant, they both require the same methodological tools, hence my interest in them. In particular, both axis B and C will use results from axis A, more precisely, from sub-axis A1 which deals with structural properties of cyber-physical systems. The relationship between different axes is illustrated in Figure 7.1.

In Section 7.1 I will describe my research program for pursuing the objective of Subsection 1.1. In Section 7.2 I will discuss the research program for the two emerging topics discussed above.

Figure 7.1: Relationship between axes,



## 7.1 Axis A: reliable models for control of cyber-physical systems

This research axis represents the continuation of the past research activities. Recall that the high-level goal of the past research activities was to come up with model reduction and system identification algorithms for which there are theoretical guarantees that the resulting models are useful for control.

When it comes to cyber-physical systems, existing results are still inadequate and there are still many open questions. For the simplicity of presentation we will concentrate on cyber-physical systems which can be modeled by polynomial, LPV and hybrid systems, as they have been the focus of my past research efforts. Note that while the category of cyber-physical systems is an engineering concept, their theoretical analysis is done by choosing a model class which can represent them sufficiently faithfully and analyzing the mathematical properties of the elements of that model class. In our case, we will concentrate on polynomial, LPV and hybrid systems, as they are capable of modeling a wide range of cyber-physical systems.

While there is a great body of literature on control, system identification and model reduction of polynomial, LPV and hybrid systems [273, 268, 177, 154, 103, 260] the existing results do not yet form a comprehensive theoretical package. While the literature on control is relatively mature, the existing results on identification and model reduction tend to focus more on algorithms, and less so on their theoretical justification. Moreover, the theoretical results are not always of the form which is informative for control. More precisely, recall that the ultimate goal is to produce theoretical guarantees for the whole processes of control design, starting with modeling based on data and ending up with the actual controller synthesis. This means that we should be able to provide guarantees that if the cyber-physical system can be modeled by a certain class of polynomial/LPV/hybrid systems, then the modeling step will yield a model such that the controller synthesized based on that model will work when interconnected with the true hypothetical model.

As it was explained in Subsection 1.1, for systems identification, the theoretical guarantees usually take the form of *consistency results*, i.e., proofs that the model produced by system identification algorithms converges

to a true model as the number of data points available for learning increases. For model reduction, theoretical guarantees tend to take the form of analytic error bounds which characterize the difference between the input-output behavior of the original model and the reduced one, or proofs that the reduced order model has exactly the same output response as the original one for some inputs. Despite the past research efforts, there are still many open problems. In particular, I would like to address the following challenges:

- **Extending theoretical guarantees to a larger class of system identification and model reduction algorithms.**

For many popular system identification and model reduction algorithms no analytical error bounds exist. Moreover, there is a gap between the class of models for which there exist theoretical error bounds for system identification and model reduction algorithms and the class of models used for control design. For example, in LPV system identification often stochastic models are considered, and the statistical parameters (variance, etc.) of the stochastic component are analyzed, while controller design methods often work with deterministic uncertainties.

- **Few theoretical results on how to convert theoretical guarantees for the model into guarantees for controllers.**

There are few explicit results relating the error bounds for model reduction and system identification with the correctness of control design. Even when error bounds for system identification and model reduction algorithms are available, there is little systematic theory explaining how to use these error bounds for controller design.

- **For certain applications, the models and assumptions used in system identification/model reduction are too restrictive.**

For instance, most of the results on LPV/hybrid system identification assume that the scheduling signal/switching signal is externally generated and that it satisfies some persistence of excitation condition. However, in many cases these signals often depend on the state of the underlying system. In turn, this means that they cannot be chosen arbitrarily, and hence it may be impossible to design identification experiments for which the persistence of excitation condition holds. Often, the the persistence of excitation condition involves the measured output too, which means that it is completely unclear how to design the experiment in order to get persistently exciting measurement data. Another problem is that in hybrid systems the switching signal may be unobservable and dependent on the unmeasured state of the underlying system. For this case, there are few theoretical guarantees, and they all tend to assume some richness of the measured data (output). However, it is not at all clear how to design the corresponding identification experiment to achieve it.

In order to address these shortcomings, the research will aim at achieving the following goals.

- **(O1)** Control-oriented theoretical guarantees for system identification and model reduction algorithms.
- **(O2)** Theoretical guarantees for controller synthesis which take into account the modeling error of the models produced by system identification and model reduction algorithms.
- **(O3)** Developing theoretically sound system identification and model reduction algorithms for more realistic model classes.

The research can be divided into fundamental and methodological parts. The goal of the methodological part is to achieve the cited research objectives. The goal of the fundamental part is to lay the theoretical foundations for the methodological part. More precisely, fundamental research will be directed to developing realization theory for model classes for which such a theory is lacking and applying it to understand the structural properties of these model classes. The methodological part will be directed towards applying the results of the fundamental research to achieve the stated goals.



### 7.1.1 Axis A1: Fundamental research: further research on realization theory and structure theory

The goal of this research axis is to study the realization theory and the related fundamental structural of various classes of state-space representations. This axis can be further divided into the following axis:

- Extending realization theory to more general model classes.
- Applying realization theory for studying structural properties which are relevant for modeling (system identification and model reduction).
- Applying realization theory for studying structural properties which are relevant for control synthesis.

The main idea is that we develop realization theory for those model classes for which such a theory does not exist, and then we use realization theory to develop further fundamental theoretical results, which in turn will be useful for proposing an integrated approach for modeling and control. Below we discuss these research axes one by one.

#### Axis A1.1: Extending realization theory to more general classes of LPV and hybrid systems

The goal of this research axis is to extend realization theory to model classes which are more realistic. In particular, I plan to address the following topics

##### Axis A1.1.1: Realization theory of hybrid systems with autonomous switching.

This problem remains open, yet it is important as hybrid systems with autonomous switching are widely used in applications. Moreover, a widely used subclass of recurrent neural networks also falls into the category of hybrid systems with autonomous switching, so the topic is also relevant for machine learning.

##### Axis A1.1.2: Realization theory of quasi-LPV systems.

A quasi-LPV system is an LPV system where the scheduling parameter depends on the input and state. Such systems are widespread in practice, as non-linear systems are often modeled as quasi-LPV systems. Once a quasi-LPV model is obtained, for control purposes it can be treated as an LPV model, and the controller designed for the LPV model will also work for the original quasi-LPV model, since the trajectories of the latter are contained in the set of the trajectories of the former. Despite its wide-spread use, there are few systematic procedures for transforming a non-linear model to a quasi-LPV, or constructing a quasi-LPV model from data, and little is known about the relationship between the structural properties of the quasi-LPV (minimality, observability, controllability) and those of the original non-linear model. Realization theory of quasi-LPVs would shed a light on these questions. Note that if we allow the scheduling parameter to be a discontinuous function of the state and the scheduling parameter takes values from a finite set, then quasi-LPV becomes a piecewise-linear hybrid system with autonomous switching. Hence, the proposed research topic is closely related to realization theory of hybrid systems with autonomous switching.

##### Axis A1.1.3: Realization theory for hybrid/LPV systems with stochastic noise.

Most of existing results on realization theory of hybrid/LPV systems deals with models which contain no noise. There is a recent extension of realization theory to hybrid/LPV systems with stochastic noise, see [226], but the results are still incomplete as they do not allow for deterministic linear inputs. For this reason, I plan to work further on realization theory of hybrid/LPV systems with stochastic noise.

##### Axis A1.1.4: Realization theory for hybrid/LPV systems with bounded deterministic noise and its relationship with stochastic noise.

For many applications the assumption that the noise is stochastic is not necessarily justified. For these applications, it is often more natural to assume that the noise is deterministic and has a bounded amplitude

or energy. Moreover, there is a rich theory and tool-set for robust control of systems with bounded noise, especially for LPV and piecewise-affine systems. For this reasons, it would be desirable to develop realization theory for hybrid/LPV systems with bounded noise. Furthermore, it would be desirable to establish a formal equivalence between the case of deterministic and stochastic noise. One option is to think of a stochastic noise as the modeling error which arises due to the use of stochastic inputs.

This would be especially useful for system identification of hybrid/LPV systems. Right now, the standard assumption in hybrid/LPV system identification is that the noise is stochastic, and a lot of attention is devoted to estimating the statistical properties of this stochastic noise. However, system identification is done in order to have models for control, and most of available tools for hybrid/LPV control can handle only bounded deterministic noise. This is especially true for LPV models, as they are often used exactly because they are convenient for using tools from robust control. So there is a discrepancy between the class of models considered in system identification and the class of models used in control design. I believe that the existing system identification algorithms are probably quite adequate even when they are applied to systems with a bounded deterministic noise. In this case, the estimated noise statistics is likely to represent a bound on the amplitude/energy of the deterministic noise. In fact, the presence of stochastic noise could be explained by assuming that the input used for the identification experiment is a sample path of a stochastic process and that the noise process represents the prediction error of the output. In turn, using stochastic inputs for identification might be necessary to ensure persistence of excitation.

I am especially optimistic about LPV system identification algorithms, as they have been extensively tested on practical applications, and hence are likely to deliver adequate models in most of cases. So the task is not so much to design completely new system identification algorithms, but to reinterpret existing ones. In order to do so, parallels between stochastic and deterministic case have to be established, which amounts to developing realization theory of hybrid/LPV systems with a bounded deterministic noise. The results presented in [229, 317]. are expected to be useful for achieving this goal.

### **Axis A1.1.5: Realization theory and algorithms for polynomial systems.**

I plan to continue working on realization theory and algorithms for polynomial systems, especially for semi-algebraic systems, i.e. systems determined by polynomial equations and inequalities. These systems lack a complete realization theory and the existing procedures for minimization and construction of a realization are not algorithmically effective. This line of research overlaps with realization theory of recurrent neural networks (Section 7.2.1), hybrid systems with autonomous switching, and quasi-LPV systems, as all these system classes overlap with the class of semi-algebraic systems. Indeed, piecewise-linear hybrid systems are semi-algebraic by definition, and so are quasi-LPV systems where the matrices are polynomial/rational functions of the scheduling parameters.

### **Axis A1.1.6: Behavioral approach for realization theory of hybrid/polynomial/LPV systems.**

Most of the existing work on realization theory uses input-output functions as a formalization of the observed behavior of a state-space representation. However, in general, this formalization is not satisfactory, as it tacitly assumes that the solutions of the state-space representation start from a particular initial state. Intuitively, this might be approximately true for state-space representations which are stable in a suitable sense. This assumption is also suitable for identification purposes, if only one long time-series is available for identifying a state-space representation. However, in general, this assumption is not the most realistic one, especially for control. Instead, it is much more natural to model the observed behavior as sets of timed signals, i.e., adopt the behavioral approach of J.C. Willems [306]. All the questions of realization theory (existence of a state-space representation, minimality, etc.) can easily be reformulated in the language of the behavioral approach. For LTI state-space representations a fairly complete realization theory in the behavioral setting was developed [300, 301, 302] and partially extended to various other system classes [311], in particular to LPV systems [273, 276]. However, much remains to be done, even for hybrid and LPV systems. For instance, the results of [273, 276] concern LPV state-space representation with a dynamic and rational dependence on the scheduling parameters. However, for control purposes, LPV state-space repre-

representations with a static and affine dependence are preferable, and the results of [273, 276] do not yield a realization theory for the latter class of state-space representations.

**Axis A1.2: Application of realization theory to structure theory of models relevant for model reduction and system identification**

The goal of this research axis is to apply realization theory for studying certain structural properties of models. These properties are useful for analyzing system identification and model reduction algorithms and they are as follows:

**Axis A1.2.1: Manifold structure of spaces of systems and distances between systems.**

The goal is to study the topology and geometry of spaces of state-space representations belonging to various model classes, for example LPV state-space representations with an affine dependence on parameters, linear switched systems, etc. More precisely, let us fix a model class and a state-space dimension and let us identify any two minimal dimensional state-space representations from the fixed model class and of the fixed dimension, if these two state-space representations have the same input-output behavior. The corresponding space will be composed of equivalence classes of state-space representations with the same input-output behavior. Alternatively, we might identify each point of this space with the input-output behavior of a state-space representation. We would like to see if we can show that this space has the structure of a finite-dimensional (smooth, algebraic, etc.) manifold, if it is metrizable and if the corresponding distance has a system-theoretic interpretation (for example, in the sense of induced  $L_2$  norm, etc.). Moreover, there should be a finite number of local coordinate charts and they should be computable. Ideally, the corresponding distance should be a Riemannian one and computable too.

Realization theory is necessary for this for the following reasons: first, it allows to characterize equivalence classes of minimal state-space representations, second, realization algorithm might give rise to computable coordinate charts [214, 222].

While some preliminary results for linear switched systems exist, [214, 222], I would like to work out this topic further and to extend it to other model classes such as (quasi-)LPV systems, polynomial systems and hybrid systems with autonomous switching.

**Axis A1.2.2: Identifiability of parameterizations, identifiable canonical forms.**

The goal is to study conditions for identifiability of parameterizations of various model classes. Recall from Chapter 5 that a parameterization is identifiable, if any two elements of the parameterizations have different input-output behaviors, i.e., their output responses are different for some input. Clearly, non-identifiable parameterizations are not suitable for system identification, as it was discussed in Chapter 5. This brings the following two questions: how to decide if a given parameterization is identifiable and how to come up with identifiable parameterizations. In Chapter 5 it was shown how realization theory can be used to answer the first question for LPV/linear switched state-space representations. I would like to use the same proof technique to analyze identifiability for other classes of models, such as hybrid systems with autonomous switching and polynomial systems.

The second question, i.e., the existence and construction of identifiable canonical forms has not been studied systematically even for linear switched systems or LPV systems. One obvious option for choosing such parameterizations is to choose local coordinate charts of the manifold of systems as defined in Axis A1.2.3. Other options remain to be explored.

**Axis A1.2.3: Persistence of excitation.**

The goal of this research axis is to explore the notion of universal inputs and persistence of excitation for hybrid, LPV, and polynomial systems. Some preliminary work was done in [206], but much more needs to be done. There are many technical definitions of persistence of excitation. Here by a persistently exciting input signal we mean an input signal such that it is possible to construct a model of the whole input-output

behavior from the output response of the system to this particular input signal. Clearly, the class of persistently exciting signals depends on the choice of the class of models. The concept of persistence of excitation is necessary for input design for the system identification experiments. Whatever input signal is used for system identification, it should be persistently exciting, since the goal of any system identification experiment is to generate data for estimating a model of the underlying system. Realization theory can be used for this purpose as follows. In the process of developing realization theory various generalizations of the notion of Markov-parameters emerge. The generalizations of Markov-parameters have two things in common: first, they allow to encode the input-output behavior by a sequence of numbers (generalized Markov-parameters), second these numbers can be constructed from the input-output behavior. Persistently exciting input signals then are those input signals which allow the construction of these generalized Markov-parameters from the corresponding output response. This approach was used in [206] for linear switched systems, and I am hopeful that it can be extended to other model classes too.

### **Axis A1.2.4: Minimization algorithms, calculating observable/reachable/minimal subsystems and their properties.**

The goal of this research axis is to further develop algorithms for transforming a state-space representation to a minimal one while preserving its input-output behavior, and to study the theoretical properties of these algorithms. In particular, it is of interest to understand how to calculate observable/reachable/minimal subsystems of a state-space representation. Usually, minimization algorithms boil down to a rank/dimension test, and the latter boils down to deciding if a number is zero. The numerical implementation of such a procedure then depends on the choice of a cut-off threshold. In fact, such algorithms can be viewed as simple model reduction algorithms. Furthermore, many model reduction algorithms can be viewed as implementations of a minimization algorithm with a large cut-off threshold. For these reasons, it is important to understand these algorithms and their properties.

### **Axis A1.2.5: (Partial-) realization algorithms and their properties: continuity, correctness, etc.**

The goal of this research axis is to develop further realization algorithms, including partial realization algorithms, and to study their theoretical properties. Usually, realization algorithms act on a finite number of input data (Markov-like parameters) and they return a state-space realization of the underlying input-output map if the input data is adequate, more specifically, enough Markov-like parameters are provided. The minimal number of Markov-like parameters often depends on the dimension of the minimal state-space representation of the underlying input-output map and they algorithm returns a minimal dimensional state-space representation.

Realization algorithms can be used for system identification. The idea is to estimate the necessary Markov-like parameters from data and to use the estimated values of these parameters as inputs to a realization algorithm. The model returned by the realization algorithm is then the output of the system identification algorithm. Partial realization algorithms can be used for (data-driven) model reduction: the reduced-order model can be calculated by applying the realization algorithm to a smaller number of Markov-like parameters than the one which is necessary to calculate an exact state-space representation of the underlying input-output behavior.

In order to analyze the correctness of such system identification and model reduction algorithms, the following questions need to be answered:

- We need to determine the number of Markov-like parameters which is necessary to calculate a minimal state-space representation. If the number of Markov-like parameters is smaller than this number, then we need to understand the relationship between the input-output map of the states-space representation calculated by the realization algorithm and the true input-output map. Ideally, the former should be an approximation of the latter in some suitable topology, for example the one studied in Axis 1.2.1.

- We need to understand if the realization algorithm is continuous in Markov-like parameters, if we use the topology of state-space representations studied in Axis 1.2.1. That is, if a small perturbations in the input parameters will lead to small changes in the model produced by the realization algorithm.

**Axis A1.2.6: Reinterpreting stochastic realization theory in terms of deterministic input-output behaviors.**

Traditionally, in stochastic realization theory stochastic state-space representations are viewed as representations of a pair of input and output processes, or a parameterization of the probability of the output process. This interpretation is justified from the point of view of the subsequent use of stochastic models in system identification. Indeed, in system identification, the assumption is that the data is a finite portion of a sample path of the input and output processes. While the behavior of the underlying system is well defined for all inputs, what we measure is a sample of its response to a particular input process. Hence, it is quite natural to aim at finding a representation of a particular pair of input and output processes. Therefore, the usual formalization of stochastic realization problem represents a natural abstraction of the system identification problem. Moreover, this abstraction allows us to use the language and tools of statistics to analyze system identification algorithms. In turn, stochastic realization theory is a necessary theoretical tool for using statistical concepts.

While such a point of view is justified from the point of view of statistics, it is not always useful for controller synthesis.

First of all, the control input need not be stochastic, and in fact the underlying physical system need not be stochastic. Second, even if the stochasticity assumption is justified, is not sufficient that the model is a valid representation of a certain pair of input and output processes. For control we need something stronger, namely, that for all inputs (not necessarily stochastic ones) the output response of the model should be close to the true one. This is especially problematic for model classes for which robust control techniques are used, for example for LPV/hybrid models. In this case, the stochastic model generated by stochastic realization algorithms/system identification algorithms cannot be directly used for control.

The goal of this research axis is to bridge this gap, and to argue that under suitable assumptions stochastic state-space representations of the same pair of input and output processes have the same input-output behavior, i.e., the two state-space representations generate the same output for all inputs, not only for a particular stochastic input.

The main idea is to argue that stochasticity arises from the use of stochastic inputs. That is, we assume that the physical process can be modeled by models which admit deterministic inputs (and which are possibly equipped with a deterministic or stochastic noise process). However, by feeding a stochastic input to these models we get stochastic outputs, and such models then become representations of a concrete pair of stochastic input and output processes. In turn, the motivation for using stochastic inputs is that the sample paths of stochastic inputs can be viewed as means to generate persistently exciting input signals; for linear systems this is well-known, for switched and LPV systems see [206, 66]. Conversely, state-space representations of the pair of the stochastic input and output processes can be reinterpreted as models defining input-output relations.

Under suitable minimality assumptions, and assuming that the noise process driving the system is the innovation noise, we can then show that minimal state-space representations of the stochastic input and output processes generate the same input-output relation as the original system.

Note that there is a long tradition in system identification to view models as predictors of outputs based on past outputs and inputs, [164]. Here, this view becomes an important technical tool. Indeed, we need to assume that the original system can be modeled by a model where the noise is the innovation process, at least when the designated stochastic input signal is used. Furthermore, we have to assume that the model returned by system identification algorithms is in innovation form. In turn, models in innovation form can be viewed as predictors.

Note that the original noise process need not be assumed to be stochastic either. If it is assumed to be a function of past outputs and inputs, which is the case if it represents unmodeled dynamics, then under suitable technical stability assumptions, the noise becomes stochastic, if stochastic input is used.

The stochasticity of the input-output pair then means that the identification problem can legitimately viewed as a statistical estimation problem, and all the statistical analysis is justified and meaningful. However, the resulting model can be shown to be an adequate representation of the entire input-output behavior of the underlying system.

In other words, from *the fact that system identification algorithms are statistically consistent we can derive that the models returned by system identification algorithms will be adequate models for the underlying, possibly deterministic, input-output behavior, provided that the number of data points is large enough.*

These ideas were partially developed in Subsection 3.7.2, Chapter 3 and for LPV-SSA models in Subsection 5.2.2, Chapter 5.

In order to accomplish the technical part of this research program, (stochastic) realization theory will be necessary, that is, we will need the results of axes A1.1.1-A1.1.4. We will also need the results of axes A2.1.1-A2.1.3. Finally, axes A1.1.6 will also be necessary for a proper formalization of the results in the language of behaviors [306].

**Axis A1.2.7: Relationship between input-output and state-space models for hybrid, polynomial and (quasi) LPV systems.**

Many dynamical models come in input-output form, i.e., in the form of equations relating high-order derivatives or time shifts of outputs and inputs. Such models often arise from the first principles and they are often preferred in system identification, as estimating such models tends to be easier than estimating state-space representations. However, for controller synthesis, state-space models are preferable. Hence, it is of interest to be establish equivalence between various input-output equations and state-space representations along with algorithms for converting state-space representations to input-output equations and vice versa. While for LTI systems such an equivalence and algorithms are well known, this remains an open problem for LPV, hybrid and certain classes of polynomial systems.

**Axis A1.3: Structural properties of models relevant for controller synthesis**

The goal of this research axis is to investigate those structural properties which are relevant for controller synthesis. Traditionally, state-space representations are used for controller synthesis, hence in this axis only models in state-space representation form are considered. The main goal is to find a subclass  $\mathcal{M}$  of state-space representations from the designated class  $\mathcal{S}$ , such that

- (A1.3.I) Any state-space representation from  $\mathcal{S}$  can be transformed to a state-space representation from  $\mathcal{M}$  while preserving input-output behavior.
- (A1.3.II) If a control synthesis algorithm can generate a suitable controller for a plant model from  $\mathcal{M}$ , then the same algorithm will generate a suitable controller for a model of the plant such that this model belongs to  $\mathcal{S}$ . By suitable we mean that the interconnection of the controller with the plant model satisfies the the control objectives. In particular, if a control synthesis problem has a solution for a state-space representation from  $\mathcal{M}$ , then it has a solution for some input-output equivalent state-space representation from  $\mathcal{S}$ .
- (A1.3.III) If a controller achieves the control objectives when interconnected with a model  $m$  from the class  $\mathcal{M}$ , then it achieves the similar (close enough) control objectives when interconnected with any other model from  $\mathcal{M}$  input-output behavior of which is close enough to that of  $m$ .

That is, we would like to show that for controller synthesis, it is sufficient to work with state-space representations from  $\mathcal{M}$ , and for the outcome of controller synthesis is does not matter which element of  $\mathcal{M}$  is used

as long as it represents the input-output behavior of the plant in question. If  $\mathcal{S}$  is the class of LTI state-space representations, then the class  $\mathcal{M}$  is the class of minimal-dimensional state-space representations and the criteria (A1.3.I)-(A1.3.III) are met. Roughly speaking, the goal of this research axis is to show that for other choices of  $\mathcal{S}$  (linear switched systems, LPV with affine dependence on parameters, etc.), the class  $\mathcal{M}$  of minimal dimensional state-space representations has the same property. The reason why minimal dimensional models are expected to play the role of  $\mathcal{M}$  is as follows. If  $\mathcal{M}$  allows models which are not minimal dimensional, then these models will necessarily contain components which do not depend on the input-output behavior of the model. The presence of such components mean that the requirement (A1.3.III) cannot be guaranteed, since those components can be chosen arbitrarily, and hence it is impossible to say anything about the behavior of those components under the influence of the controller at hand.

These properties are relevant for integrating modeling (systems identification and model reduction) with control for the following reasons. Different system identification and model reduction algorithms applied to the same data/system tend to result in different models, but all these models have roughly the same input-output behavior. Hence, the controller synthesized based on any such model should achieve the control objective for the true plant. A necessary condition for this is that the controller provenly achieves the control objective when interconnected with any model from the class  $\mathcal{M}$ , as long as the latter model has the same or similar input-output behavior as the model which was used to calculate the controller. This then means that the outcome of **any** correct system identification/model reduction algorithm can be used for controller synthesis, and the thus calculated controller will work for the plant. By correctness of a system identification we mean that as the number of data points grow, the input-output behavior of the model returned by the algorithm converges to the input-output behavior of the data generating system. That is, the returned model can represent the true input-output behavior with any desired accuracy if a sufficiently large number of data points are used. By correctness of a model reduction algorithm we mean that the distance between the input-output behavior of the model returned by the algorithm and that of the original model is within a certain designated accuracy level. The right notion of distance to be used is application specific. The tacit assumption behind this reasoning is that the plant can be identified with a model from  $\mathcal{M}$ . This assumption may be problematic, but in the absence of any further knowledge about the system there is no way to falsify it.

In order to realize this program, we will organize the research along the following axis.

**Axis A1.3.1; Observability, reachability, existence of observers/controllers.**

The goal of this research axis is to relate structural properties such as minimality, controllability and reachability of hybrid/LPV/polynomials systems with the existence of a controller (possibly with partial observation) achieving a certain control objective and with the existence of an observer. Here, observability and reachability are understood in the sense of realization theory, i.e., as the structural properties which are necessary and sufficient for a state-space representation to be minimal dimensional. More precisely, we would like to show that any controller can influence only on the reachable subsystem of the state-space representation and any observer (output based controller) reacts only to the observable subsystem of the state-space representation. From this it then follows that any controller with partial observations acts only on the minimal subsystem of the plant model. This research axis is going to be useful to show that minimal state-space representations satisfy a weaker version of (A1.2.II), i.e., that if a control objective can be achieved for a state-space representation, then it can be achieved for a minimal state-space representation.

**Axis A1.3.2: Relationship between dissipativity, input-to-output stability and internal stability of minimal state-space representations.**

The goal of this axis is to relate dynamic properties of the input-output behavior such as input-to-output stability and dissipativity with the corresponding properties of their hybrid/LPV/polynomial state-space representations. In particular, we would like to show that input-to-output stability/dissipativity implies existence of a Lyapunov function/storage function of the minimal state-space representation. As a consequence, input-to-output stability then implies internal stability for minimal state-space representations. That is, we are interested in counterparts of Bounded Real and Positive Real Lemma (with possibly with

non-quadratic storage functions). These properties are useful for understanding the pitfall of using non-minimal models. They are also useful for proving property (A2.1.II): indeed, many controller synthesis algorithms attempt to find a Lyapunov function or a storage function for the closed-loop system, hence it is of interest to show that if the closed-loop input-output behavior is input-to-output stable or dissipative, and the plant is modeled by a minimal state-space representation, then such a Lyapunov function or storage function exists.

These results will also be useful for showing (A2.1.III): they can also be used to show that the same controller can achieve the control objectives for two different state-space representations with close enough input-output behavior. The difficult part in such reasonings is that control objectives often include internal stability. Note it is relatively easy to conclude if two models have a close enough input-output behavior, then the closed-loop input-output behaviors will also be close enough, if the models are interconnected with the same controller. Hence, it is relatively easy to argue that if a controller meets the control objectives for the input-output behavior with one plant then it meets those objectives when interconnected with another plant with a similar input-output behavior. It is much more difficult to argue that the control objectives which concern the properties of the state-space representation, such as internal stability, will also be satisfied. The results of this research axis allow us to do just that, by connecting the properties of the state-space representation with those of its input-output behavior.

### **Axis A1.3.3: Robustness of controllers in the topology of systems.**

Finally, we would like to show that if a controller satisfies the control objectives when interconnected with a plant modeled by a state-space representation  $m$  from  $\mathcal{M}$ , then it will also meet sufficiently similar control objectives when interconnected a plant which can be modeled by an element of  $\mathcal{M}$  which is close enough to  $m$  in the topology of spaces of systems to be investigated in Axis A1.2.1. This then helps to show property (A2.1.III) completely. Ideally, explicit error bounds using the distances defined in Axis A1.2.1 should be derived, such that if the distance between two models is smaller than the error bound, then the controller which achieves the control objectives for one of the models, will achieve the close enough control objectives for the other one. Moreover, a suitable topology on the control objectives should be found too.

## **7.1.2 Axis A2: Methodological research: control-oriented and theoretically sound system identification and model reduction algorithms**

The goal of this research axis is to derive theoretical guarantees for system identification and model reduction algorithms, such that the derived theoretical guarantees are likely to be useful for control. This will be achieved by developing theoretical guarantees for existing algorithms and by formulating new algorithms.

### **Axis A2.1: System identification: algorithms and theoretical guarantees**

In system identification I plan to organize the research along the following topics:

#### **Axis A2.1.1: Statistical analysis of existing system identification algorithms: consistency, variance.**

The goal is to analyze the statistical properties of existing system identification algorithm for LPV/hybrid/polynomial systems. We view system identification algorithms as estimators, and we investigate its statistical properties such asymptotic consistency and variance. That is, we want to find conditions which guarantee that the model returned by a system identification algorithm converges to a true one as the number of data points tend to infinity, and we would also like to have estimate on the variance of the estimated models.

There are two approaches for proving such results:

- For system identification algorithms which are based on realization algorithms (subspace identification, covariance realization) we try to prove that as the number of data points tend to infinity, the identification algorithm behaves more and more like a realization algorithm. For example, if the algorithm is based on estimating output covariances and applying a covariance realization algorithm



to the estimates, then we will try to show that the estimates converge to the true covariances and hence, if the realization algorithm is continuous, then the model generated by the identification algorithm converges a true one. This approach was taken for example for a particular version of subspace algorithms for linear systems [163] and for LPV systems [172, 173].

- For parametric system identification algorithms the proof might be constructed as follows. Such algorithms are based on minimizing an empirical cost function which depends on data. Then, under the assumption of identifiability of the parameterizations and informativity of the signals, one has to show that the minimum of the empirical cost function converges to the true parameter. The latter can be shown by showing the the empirical cost function converges to a cost function unique minimum of which is the true parameter.

This axis relies on axis A1.2. Indeed, realization algorithms from A1.2.5 allow the proof of consistency of identification algorithms which are based on realization algorithms from axis A1.2.5. Furthermore, in order to formalize the notion of consistency, the concepts on distances and topology for the space of models from axis A1.2.1 will be necessary. In fact, we define consistency as follows: the sequence of models returned by the identification algorithm when applied to an increasing number of data points converges to a true model, and the convergence is understood in the sense of the topology from axis A1.2.1.

Moreover, the concepts of persistence of excitation (axis A1.2.3) and the continuity of realization algorithms (axis A1.2.4) will be indispensable for formulating the proofs of consistency. The former is necessary to show that the identification algorithm is related to the realization algorithm, as persistence of excitation means exactly that the Markov-parameter-like numbers can be recovered from data. Continuity of realization algorithms will be indispensable for showing that as the number of data points increase, the result of applying the system identification algorithm to estimates of Markov-parameters will converge to the application of the realization algorithm to the true Markov-parameters.

Finally, for parametric identification algorithms the assumption that the parameterization is identifiable is likely to be necessary for proving consistency. In order to ensure that this assumption is satisfied, the use of identifiable canonical forms is likely to be required. These topics are investigated in axis A1.2.2.

Axis A1.2.7 is relevant because many of the system identification algorithms are formulated for input-output models, and it is of interest to reinterpret them for state-space models.

The research of this axis is a continuation of the classical research in system identification.

#### **Axis A2.1.2: Non-asymptotical (PAC,PAC-Bayesian) error bounds for system identification algorithms inspired by machine learning.**

The goal of this research axis is to develop error bounds which characterize the difference between the true behavior of the system and the behavior of a model returned by the system identification algorithm. Again, the focus is on model classes for which realization theory exists and on existing system identification algorithms. In particular, inspired by machine learning [112], we aim at developing Probably Approximately Correct (PAC) [254] and PAC-Bayesian [112] error bounds for system identification of several known model classes, and to analyze existing system identification algorithms in the light of those error bounds. Such error bounds were recently proposed for ARX model identification [253] we hope to extend them to other model classes. The advantage of such error bounds is that they express the difference between the estimated and true model as a function of the number of data points and experimental conditions (e.g., persistence of excitation of inputs). This allows more precise statements regarding the number of data points and experimental conditions which are required for estimating a model which is accurate enough.

This research axis can be seen as a continuation of axis A2.1.1: axis A2.1.1 aims at showing that as we are taking more and more data points, the resulting model will get closer and closer to a true one, while the present research axis attempts to give an upper bound on the distance between the learned model and a true one based on the number of data points.

## 7.1. AXIS A: RELIABLE MODELS FOR CONTROL OF CYBER-PHYSICAL SYSTEMS

---

The role of axes A1.2 is the same as for A2.1.2, with the exception that the notion of distance between models (axis A1.2.1) is even more indispensable.

### **Axis A2.1.3: Theoretically sound new system identification algorithms for more realistic model classes.**

The objective is to derive new system identification algorithms which apply to more realistic model classes and then execute for them the research program described in axes A2.1.1-A2.1.2. That is, this axis can be viewed as an extension of A2.1.1-A2.2.2 to new system identification algorithms and new model classes.

Possible extensions include

- Black-box/gray-box system identification algorithms for LPV systems based on realization algorithms, together with the corresponding theoretical guarantees, preliminary results in that direction are in [172, 173] and they were also discussed in Chapter 5.
- Realization theory based system identification algorithms for polynomial/quasi-LPV systems and hybrid systems with autonomous switching, for some preliminary steps see [224, 299].
- System identification for aperiodically sampled sampled-data and quantized systems, in particular, networked control systems, [295, 95, 118, 119]: such systems can be represented as LPV/hybrid systems, but they have a number of particular properties.
- System identification for finite-dimensional approximations of infinite dimensional models. This is relevant for identifying systems which can be modeled by PDE: in practice, one uses finite dimensional approximations of such PDEs, and hence it makes more sense to try to estimate a finite-dimensional approximation from data, rather than to estimate the parameters of a PDE.

This axis relies on the realization algorithms to be developed and analyzed in A1.2.5. As it was explained in the description of axis A2.1.1, in order to reformulate these realization algorithms as system identification algorithms and prove their consistency, we will need the notion of persistence of excitation (axis A1.2.3). Moreover, in order to formally state consistency results, we will have to use the notion of topology for the space of models from axis A1.2.1.

### **Axis A2.1.4: Control-oriented reinterpretation system identification algorithms in terms of (deterministic) input-output behaviors.**

The goal of this axis is to bridge the gap between a purely statistical interpretation of the models generated by system identification algorithms, and the needs of controller synthesis. As such, this research axis is a logical continuation of the research axis A2.1.1-A2.1.3.

In axes A2.1.1-A2.1.3 we followed the classical approach of viewing system identification as a statistical estimation problem. In particular, the underlying models are assumed to be essentially stochastic, and they are interpreted as parameterizations of the probability distributions of stochastic processes from which the data used for identification was sampled. That is, even if stochastic consistency is shown, it just says that the identified model will represent approximately correctly *the output response to the particular stochastic input*, sample of which was used for system identification. *It says nothing about the response of the model to other inputs.* As it was pointed out in axis A1.2.6 this is not a sufficient guarantee for control purposes.

There are two ways to deal with it: to dismiss altogether the use of the statistical approach for models which are intended for deterministic control, and with it a large body of the existing literature, or to try to reinterpret the statistical results from a deterministic perspective. Many of the existing system identification algorithms have a statistical justification but their result is used for deterministic controller synthesis. For this reason, I believe that the second approach is more reasonable, and that the discrepancy described above is often an artifact, which is a consequence of the use of (pseudo-random) inputs rather than a consequence of physical reality.

In this axis we plan to use the results of axis A1.2.6 for showing that stochastic state-space representations returned by system identification algorithms are approximations for the entire input-output behavior of the

underlying system, i.e., they recreate approximately not only the output generated to the stochastic input used for system identification, but the output for any other possible control input.

**Axis A2.1.5: System identification algorithms for deterministic systems.**

The goal of this research axis is to develop system identification algorithms and to analyze their correctness in a deterministic setting. It is assumed that the models at hand are purely deterministic, uncertainty is modeled as a deterministic noise, and the goal is to show that the model returned by the system identification algorithm converges to a true one as the number of data points tends to infinity.

This research axis is complementary to the previous ones: axes A2.1.1-A2.1.3 aim at studying system identification algorithms in a stochastic setting, while axis A2.1.4 aims at translating stochastic results into a deterministic setting. In contrast, the aim of the present axis is to skip statistical analysis altogether. This approach is motivated by the behavioral approach to system identification [169] and bears certain similarity with the system identification methods with bounded noise [176]. In a way, we aim at further extending [169] to LPV/hybrid models. Note that there is a significant literature on the extension of the methods [176] to LPV models, without claiming completeness see for example [61].

This research axis will require realization theory of models with bounded deterministic noises and realization theory with a behavioral approach, hence

**Axis A2.2: Model reduction: algorithms and theoretical guarantees**

The goal is similar to the one formulated for system identification. Namely, we would like to derive theoretical guarantees for the existing model reduction algorithms and to derive new model reduction algorithms which are applicable to more realistic classes of models. More precisely, I plan to investigate the following topics.

**Axis A2.2.1: Theoretical guarantees for model reduction algorithms: global error bounds.**

The goal of this research axis is to provide analytical bounds for the difference between the input-output behavior of the original and the reduced-order models. Roughly speaking, for any input signal we would like to have an analytic upper bound on the difference between the output responses of the original and reduced-order model to this input. Examples of such error bounds are the ones derived for balanced truncation [12], and extended to linear time-varying, LPV and switched systems [248, 309, 40, 227, 106]. Most of the existing error bounds are  $L_2$  error bounds, which are not always useful for control synthesis. Hence, it is of interest to extend them to other norms. Moreover, for many model reduction algorithms and model classes there are no comparable error bounds. For deriving global error bounds, the various distances of models from axis A1.2.1 will be necessary. Moreover, global error bounds tacitly imply stability, and their proofs rely on the notion of dissipativity, hence the results of axis A1.2.2 will be useful for deriving global error bounds. Furthermore, balanced truncation is intimately related to observability and reachability and minimization algorithms, hence the corresponding results from axis A1.2.4 and axis A1.3.1 will be necessary.

**Axis A2.2.2: Theoretical guarantees for model reduction algorithms: restricted input-output behavior.**

Another type of theoretical guarantee is a proof that the reduced-order model and the original model generate the same (possibly filtered) response to some inputs. These types of guarantees are natural for the so called moment matching algorithms, and they were derived for various algorithms for linear, nonlinear, and switched systems [12, 14, 30, 31, 105]. The goal of this axis is to continue research for such guarantees, one possible direction is to derive similar results for Loewner-based model reduction for switched systems [105], and for LPV/bilinear systems [228].

In order to derive the desired results, the notion of Markov-parameters and their relationship with the input-output behavior will be required. Hence, we will need realization theory and partial realization algorithms from axis A2.1.6. We might also run into issues related to persistence of excitation, i.e. the relationship between Markov-parameters and output response to certain inputs. That is, axis A1.2.3 might be useful.

### **Axis A2.2.3: Extension of model reduction algorithms to more realistic models.**

The goal of this axis is to extend model reduction algorithms to new model classes. In particular, I plan to work on model reduction for hybrid systems with autonomous switching, polynomial and LPV state-space representations. For the latter class, the challenge is to develop methods which are more computationally efficient than the current methods based on balanced truncation, and which can also be applied to models which are not quadratically stable. Candidates for such algorithms are methods based on moment matching, in particular, I plan to extend the model reduction methods developed for switched systems [30, 31, 105]. In the light of the close relationship between LPV state-space representations and switched systems described in Chapter 2, the extension is likely to be feasible. A notable subclass of LPV/hybrid models are those which arise in networked systems, where the plant can be modeled by a linear system. In those models, the scheduling variable/switching represents delays and communication constraints, and those models have a very rich structure. First steps towards model reduction algorithms were already made in [27, 26, 25].

The notion of Markov-parameters, minimization and realization algorithms were central to moment matching for linear [12] linear switched [30, 31] and LPV/bilinear systems [28, 228], and we expect the same for other system classes. Moreover, the notions of minimality, reachability, observability were necessary for methods based on balanced truncation. That is, the results of axes A1.2.4-A1.2.5 are indispensable for the present axis. Furthermore, the notion of internal stability and dissipativity are central for methods based on balanced truncation, hence axes A1.3.2 will also be necessary for the present axis. Finally, the distances developed in axis A1.2.1 will also be necessary for global error bounds.

### **Axis A2.2.4: Combining system identification and model reduction: data-driven model reduction.**

The goal is to combine system identification and the model reduction algorithms into one single algorithm which could be used for calculating a reduced-order model directly from data. This is especially interesting for systems which can be modeled by PDEs; in that case it is of interest to calculate a finite-dimensional approximate model directly from data. However, for such a model to be useful for control and simulation, its size should be reasonable, and hence a tradeoff between modeling accuracy and size should be achieved. The latter is central to model reduction too. Hence this axis is closely related to Axis A2.1.3. A particularly promising candidate for data-driven model reduction is the Loewner-framework [12, 10], which was recently extended to switched systems [105]. The challenge is to extend these methods to LPV systems and hybrid systems with autonomous switching and to adapt them to the use of realistic data. Note that the extension [105] uses data which is difficult to measure in practice.

### **Axis A2.3: Integrating system identification, model reduction and control**

The goal of this axis is to integrate the results of axes A2.1 and A2.2 into control design. That is, we would like to provide theoretical guarantees that the models obtained via system identification or model reduction are good enough for control. More precisely, we would like to show that the controller designed based on a hybrid/LPV/polynomial model obtained by system identification or model reduction will also work for the true system, assuming that the true system can adequately be described by a hybrid/LPV/polynomial model.

This research axis is divided into the following directions:

#### **Axis A2.3.1: Asymptotic guarantees for using system identification for control.**

The main idea is to use the continuity of controllers in the topology of models and the asymptotic consistency of identification algorithms to show that for a large enough number of data points, the controller synthesized based on the identified model will achieve the control objectives when applied to the true plant.

The argument is as follows: if a large enough number of data points is used, the identified model will be sufficiently close to a true model of the plant. If the controller synthesis algorithm is provenly correct, then the interconnection of controller with the model will satisfy the control objectives. The consistency of the identification algorithm means that for a sufficiently large number of data points, the model is close enough to a true model of the plant. Hence, if the controller is continuous in the topology of models, then the

behavior of the interconnection of the controller with the plant will be sufficiently close to the behavior of the interconnection of the controller with the identified model. Hence, if the control objectives are robust, i.e., if a model satisfies them then a sufficiently close model will also satisfy them, then the interconnection of the controller with the plant should satisfy these control objectives too.

That is, in this axis, axes A2.1.1 and A1.3.3 meet. There several challenges, the following two issues seem to be the most significant:

- **Formalization.**

The argument described above needs to be formalized mathematically. In particular, suitable formal definitions for distances between models should be proposed, and a formal proof should be provided that as the model used for controller synthesis converges to the true one, then the behavior of the closed-loop system converges to the desired one. In particular, formal statement should be provided which characterize the modeling error under which the controller computed from the identified model still achieves the control objectives for the true system.

- **Reconciling assumptions of system identification and control design: stochastic vs. deterministic, input-output maps vs. behaviors.**

As it was noted before, consistency results for system identification algorithms tend to be stochastic, and the formalization of the input-output behavior tends to be done in terms of input-output functions. However, for control purposes, it is often more adequate to work with deterministic systems (with possibly bounded deterministic noise) and with deterministic error bound/consistency guarantees. Moreover, the formalization in terms of behaviors is preferable to that of input-output maps, as it is more realistic for control, see axis A1.1.7.

The stochastic nature of consistency analysis of identification algorithms has to be dealt with. One option is to provide stochastic guarantees, guarantees which hold with certain probability. Note that even if the model is deterministic, the input signal may very well be stochastic, hence the consistency analysis of the system identification algorithm will be stochastic.

For the reinterpretation in terms of behaviors, the results of A2.1.4 will have to be used together with A2.1.1. Alternatively, the results of A2.1.5 might be used directly too. Moreover, system identification for new model classes will be necessary, i.e., A2.1.3 will be used. Finally, for the formalization of the results and the formal proofs, we will use axis A1.3, as we will need the properties (A1.3.I)–(A1.3.III).

- **Dealing with realistic model classes.**

Another issue is that for this axis, the model classes generated by system identification should be realistic and should be useful for control. In particular, model classes such as hybrid systems with autonomous switching and quasi-LPV should be covered. That is, system identification for new model classes will be necessary, i.e., A2.1.3 will be used

This research axis is likely to rely heavily on axis A2.1.3.

### **Axis A2.3.2: Non-asymptotic guarantees for using system identification for control.**

The goal of this research axis is similar to the one of A2.3.1, but instead of just showing that for a large enough number of data points the controller built using the identified model will meet the control objectives, we would like to have more precise results regarding the number of data points which is necessary. In a way, the results of A2.3.1 are just a sanity check, showing that combining system identification and control synthesis makes sense. However, they do not give the user any guidance on the number of data points which has to be chosen. In this axis, the goal is to derive bounds on the number of data points which guarantee that the controller synthesized based on the plant model learned from those data points will satisfy the control objectives for the true plant. Note that the controller synthesis could be done with modified control objectives in order to take into account model uncertainty: for instance, if we know that the  $L_2$  distance between the true model and the identified one is less than  $d_1$  and we want the  $L_2$  norm of the closed

loop system to be smaller than  $\gamma$ , then the controller synthesized for the identified plant should achieve a closed-loop  $L_2$  gain which is smaller than  $\gamma - d_1$ . In order to achieve this goal, the results of axes A2.1.2, A2.1.3 and A1.3.3 should be combined. The main challenges are similar to that of axis A2.3.1, namely, the formalization of the argument and the combination of stochastic error bounds of system identification algorithms with the deterministic guarantees of controller synthesis algorithms. Consequently, we will need the results of axes A2.1.2, A2.1.3, A2.1.4, A2.1.5 and axis A1.3.

### **Axis A2.3.3: Integrating theoretical guarantees for model reduction algorithms and controller synthesis.**

The goal of this research axis is to combine theoretical guarantees for model reduction algorithms and for controller synthesis, i.e., to show that the controller synthesized using a reduced-order model will achieve the desired control objectives for the original model too. As in the case of Axis 2.3.2, this may require changing the control objectives used for controller synthesis, so that they take into account the modeling error. Recall that the theoretical guarantees for model reduction can take the form of error bounds for the difference between the original and reduce-order model, or they can come in the form of guarantees that the restricted input-output behavior the reduced-order model and the original one coincide. Both approaches should be explored. The latter approach is especially useful for formal methods and symbolic control. One of the bottlenecks in applying formal method and symbolic control for large-scale systems is that the corresponding methods do not scale-up sufficiently well for large systems. Model reduction of hybrid systems could be very useful for solving this problem. Preliminary steps in this direction were made in [31]. I would like to explore further the use of model reduction for producing models which are sufficiently small for existing tools in formal methods and symbolic control.

This research will use the results of axes A2.2.1 and A2.2.2 and A2.2.3.

### **Axis A2.3.4: Integrating theoretical guarantees for model reduction, system identification and controller synthesis into one framework.**

The goal of this axis is to integrate the results of system identification, model reduction and controller synthesis into one framework. That is the goal is to show that if we

- learn a model from a large enough number of data points,
- apply model reduction to it, and then
- perform controller synthesis based on the reduced-order model with possibly adapted control objectives so that they take into account modeling error,

then the resulting controller will achieve the original control objectives for the plant. That is, we plan to combine axes A2.3.1, A2.3.3, A2.3.3 and possibly A2.2.4 (data-driven model reduction) into one single theoretical framework.

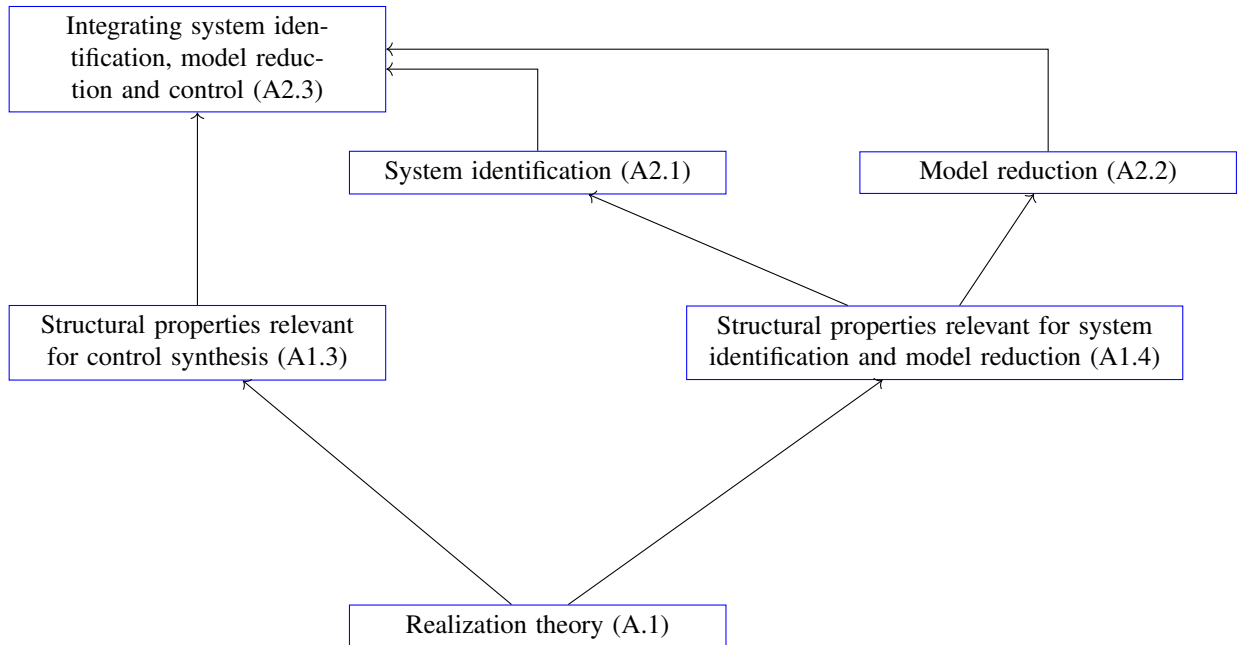
### **7.1.3 Interplay between fundamental and methodological research**

The fundamental and theoretical research axes interact in a variety of ways. As it was pointed out, axis A1 is necessary to accomplish the tasks described in A2. The detailed relationship between various axes was discussed in their description. However, to make the relationship more clear, in Figure 7.2 we summarize it in a diagram.

## **7.2 Emerging research directions:axes B and C**

In addition to the research explained in Section 7.1, I plan to work on a few research topics, which represent a departure from the topics of Section 7.1. The first such is a direct continuation of Section 7.1, in the sense that instead of focusing on reliable models for controller synthesis, its high-level objective is to generate reliable models for more high-level decision making algorithms of the type for which artificial intelligence is used. This research direction is described in Subsection 7.2.1. The second emerging topic is the detection of the graph

Figure 7.2: Relationship between sub-axes of axis A,



of interconnections of a network of dynamical systems. My interest in the latter topic is motivated by need of systems biology, especially of neuroscience. Of course, this topic might be of interest in many other domains, such as economics or sociology, which try to represent a black box system as an interconnection of several simpler subsystems. In particular, such results can be of interest for control, as they may allow estimating distributed models of the plant, which in turn, may allow the use of decentralized controllers. However, this is more of a useful side effect than the main objective of the research. What both directions have in common is that structure theory in general, and realization theory in particular, seems to be useful for both topics.

### 7.2.1 Axis B: Reliable AI for cyber-physical systems: using control theory for machine learning

Machine learning and control theory are two closely related subjects with common roots. Nevertheless, in the past couple of decades, they developed separately. Recently, the two topics started to converge again: control theorists are becoming increasingly interested in using machine learning techniques, while researcher in machine learning start looking at control problems and at possibilities to use results from control theory for machine learning.

In the control theory community, machine learning techniques are increasingly adopted in system identification, see the description of axis A2.1.2 and [168, 197, 290, 89, 51, 15, 239, 232, 156, 253] for some examples.

In fact, major control theory conferences include tracks on machine learning, speakers from the machine learning community are invited as plenary speakers to major control theory conference and entire research projects which involve machine learning are financed.

From the side of machine learning, there is a renewed interest in control theory. Part of this interest stems from the popularity of reinforcement learning [52, 262, 263], which shares its roots with adaptive control [39, 180, 236]. Other source of interest is the desire to apply machine learning to cyber-physical systems. The challenge there is twofold. First, these applications tend to be safety critical, and hence there is an increased need in theoretical

guarantees. Second, the physical part of such systems tends to be dynamic and it can often be modeled from the first principles. In turn, the control community has experience in dealing with dynamical systems which partially arise from first principles, and proving their properties. This experience is likely to be relevant. In particular, researchers in machine learning have also shown interest in control theory due to the need to:

- learn dynamical models of time-series [249, 264],
- analyze correctness and robustness of machine learning algorithms [116, 160, 41], especially when applied to cyber-physical systems.

The objective of this research axis is to explore the application of control and systems theory to the analysis of machine learning algorithms. More precisely, I plan to investigate the following topics.

### **Axis B1: Theoretical limits for learning dynamical models from time series**

The goal of this research axis is to combine systems identification and realization theory with the general results of the general results of [246, 247] to understand which classes of dynamical systems can at all be learned from data. This axis relies both on system identification (Axis A2.1) and on realization theory (Axis A1). The latter is necessary both indirectly, via axis A2.1 and directly, in order to be able to use the topological and geometric structure of the space equivalence classes of minimal systems (Axis A1.2.1).

### **Axis B2: Theoretical guarantees for learning recurrent neural networks using realization theory**

Recurrent neural networks (RNNs) are the most popular class of models used in deep learning when one needs to process sequential data. The high-level objective of this axis is to come up with theoretical guarantees for the correctness of learning algorithms for RNNs. In order to achieve it, we propose to use results from control theory and systems identification. More precisely, note that an RNN can be viewed as a dynamical system equipped with an internal state, input and output. Learning such an RNN from data is equivalent to estimating the parameters of the RNN, viewed as a dynamical system. Estimating parameters of dynamical systems from data is the subject of system identification. In particular, the research program of axis A2.1, especially A2.1.1-A2.1.3 becomes relevant, and it can be carried out for RNNs. That is, we would like to show asymptotic consistency for learning algorithms, and to derive PAC and PAC-Bayesian error bounds for the problem of learning RNNs.

In order to do, so we need to develop realization theory for RNNs. That is, we would like to answer the following questions:

1. Which class of observed behaviors (input-output maps) can be represented by a RNN.
2. How can we characterize minimal RNNs (RNNs of the least complexity) representing a certain observed behavior ?
3. What is the appropriate definition of minimality (smallest number of neurons, etc.) for RNNs, are minimal RNNs are related by some transformation ?
4. Is there are constructive procedure for constructing an RNN representation from input-output behavior which can be proven to be mathematically correct ?

We expect that the answer to these questions will allow us to answer the questions described in axes A1.2.1-A1.2.4 and then use them for solving the research problems described in A2.1.1-A2.1.2 for RNNs. In turn, the solution of the research problems described in A2.1.1-A2.2.2 are exactly the theoretical guarantees which are sought after. In fact, axis A2.1.2 can be viewed as an integral part of the present research axis. Axis B1 could be relevant in order to understand learnability issues for RNNs.

There are already some preliminary results in this direction, see [259, 3, 2, 72, 253].



**Axis B3: Theoretical limits of reinforcement learning algorithms.**

The objective of this axis is to study the theoretical limits of reinforcement learning algorithms from machine learning [139], using results from adaptive control and system identification. In particular, we would like to investigate reinforcement learning algorithms applied to linear and piecewise-linear dynamical systems. The latter represent a simple but powerful class of state-space representations with infinite and uncountable state-spaces. Note that the use of finite state- and action-spaces is a limiting factor in scalability of reinforcement learning algorithms, so this line of research has the potential to deliver algorithms which scale up for larger systems. Since reinforcement learning and adaptive control are closely related, based on the analogy with adaptive control it is likely that this research axis will rely on such concepts as identifiability, minimality, persistence of excitation, i.e., concepts developed in research axis A1.2. It is also expected that learnability results from axis B1 will be useful too.

**7.2.2 Axis C: Reverse engineering of network topology of interconnected systems**

I plan to continue working on the problem of reverse engineering of the network topology of interconnected systems, by further extending the results presented in Chapter 6.

More precisely, in Chapter 6 existence of a state-space representation with a certain network structure was characterized by the presence of certain statistical relationships (Granger-causality). This opens up the possibility of reverse engineering the network structure of a system based on data. More precisely, the presence of Granger-causality relationships can be checked based on observed data. In fact, it can be cast into the framework of hypothesis testing. If the hypothesis of the presence of Granger-causality is accepted based on the data, then we can say that there exists a state-space representation of the observed behavior which has a certain network graph structure. This means that the true underlying system *may* have such a network graph structure. If the hypothesis is rejected, that there does not exist a state-space representation with such a network structure, hence, the true system *cannot* have such a network structure either. In other words, the results of Chapter 6 allow us to test if a certain hypothesis regarding the network structure of the true system is *consistent* with data. This is not entirely reverse engineering, as we do not reconstruct the network structure, we merely say if a certain network structure is possible based on our observations. However, systems with different network structures can generate the same observations, so in general the problem of reverse engineering of the network structure is not well posed. Moreover, for applications in system biology, the presence of a certain network structure has to be proven experimentally anyway. It is then useful to have a procedure to eliminate hypotheses on the presence of a certain network structure by statistical means, as it allows to guide the experiments and avoid the ones which are not likely to lead to a positive result.

This research program was partially carried out for linear systems: the relationship between Granger-causality and the existence of a certain network structure is quite well understood, at least for the case of autonomous stochastic state-space representations. There are some preliminary steps on hypothesis testing for checking Granger-causality [133, Chapter 7], see also Subsection 6.2.5 of Chapter 6, but much more work needs to be done.

The next step would be to extend the results of Chapter 6 to linear systems with inputs, nonlinear, LPV and switched systems, and to develop statistically sound hypothesis testing for deciding if a certain network graph is consistent with data. Finally, it would be of interest to apply the results real data from neuroscience (fMRI, EEG) and systems biology.

More precisely, I plan to work along the following axes:

**Axis C1: Relationship between Granger-causality and the network structure for stochastic nonlinear, LPV and switched systems with inputs**

The results described in Chapter 6 apply only to linear and bilinear systems without linear input terms. We would like to extend those results to systems with inputs, and to more general models, such as polynomial/LPV/hybrid

models. The latter is especially useful, as one of the main application of this research is the study of brain connectivity [242, 278, 93, 102]. There, EEG and fMRI data is used to build hypothesis on the connectivity of brain regions. However, fMRI data is normally modeled as an output of cascade interconnection of a bilinear system with hemodynamic model, and the bilinear systems themselves contain linear input terms. Hence, it would be desirable to extend those results to models which can cover completely the existing models used for fMRI data.

This research axis will rely heavily on (stochastic) realization theory of these model classes, to be developed in axes A1.1.1-A1.1.5.

### **Axis C2: Deterministic reformulation of Granger-causality and its relationship with the network structure of linear, nonlinear, hybrid and LPV models**

For many systems, the assumption that the system is stochastic is not valid. Even when it could be valid, in many experimental settings the amount of data is not sufficient to estimate stochastic models in a reliable manner. This might be the case for some applications in neuroscience and in system biology. In this case, deterministic models are a better choice. However, the existing concepts of Granger-causality apply only to stochastic models. The challenge is to extend these concepts to the deterministic case in such a manner that the extension represents a meaningful formalization of causality and that it is equivalent to connectivity structure of the model.

For this research axis, we are likely to need elements of behavioral approach, hence, in addition to axes A1.1.1-A1.1.5 we will also use axis A1.1.7.

### **Axis C3: Testing the consistency of a network graph with data**

The previous axes C1-C2 only establish an equivalence between Granger-causality relationships of the observed signals, and existence of a state-space representation with a certain network structure. As it was explained above, this relationship could be used for deriving a hypothesis testing procedure for the presence of Granger-causality relationships among the observed processes. Checking Granger-causality relationship can be done by comparing variances of the innovation noise for various processes, see [133, Chapter 7], see also Subsection 6.2.5 of Chapter 6. The goal of this axis is to further extend the existing results on testing hypotheses regarding the presence of a certain network structure. The main research goals are to make the existing algorithms statistically sound and to extend them to other classes of models (linear systems with inputs, LPV and switched systems, etc.).

This axis will rely on the results of axes C1-C2. For checking Granger-causality, we will need the notion of state-space realization in forward innovation form, and the ability to calculate such state-space representations from data, see [133, Chapter 7]. Hence, we will need (stochastic) realization algorithms, and this axis A1 will be required for this axis. Moreover, we will have to calculate state-space models from data, and hence the results of axis A2 will be used.

### **Axis C4: Applications to neuroscience and systems biology**

The goal of this research axis is to apply the results of the previous axes to exploring the network graph for neuroscience and systems biology. As it was pointed out in Chapter 6, this problem arises in neuroscience, when studying brain connectivity, [242, 278, 93, 102], and in system biology, when studying gene regulation networks [191, 138, 146]. In particular, the use of Granger-causality and network graphs of state-space representations are both well established in neuroscience [278, 70, 241]. However, the mathematical foundations of both methods have not been investigated in depth. The previous axes C1-C3 can be viewed as the theoretical foundations for applying these methods. The goal of this axis is then apply the theoretical results to real data and to evaluate their relevance and efficiency.

Note that for systems biology and for neuroscience the theoretical foundations of the methods are important, because in both cases the goal is to find the truth about the system under study, and in both cases experimental validation of the results is difficult. The latter is especially the cases in neuroscience, where the goal is to study

the connectivity between brain regions in humans. There, the results of experiments on animals are difficult to extrapolate to humans, and such experiments are costly, and it is impossible to make experiments on humans. Hence, experimental validation of the ground truth can take place only by observing neurological disorders, where it is known that the connection between brain regions becomes broken, and hence it becomes a byproduct of trying to give medical treatment to such patients. When methods are used which are theoretically not sound, then the probability of deriving a false conclusion from data is high, and it is then very difficult to falsify such false conclusions using experiments. For this reason, it is preferable to avoid methods which are theoretically questionable.

### 7.2.3 Interplay between research axes A,B,C

While research axes B and C are quite distinct, they both rely on fundamental structural properties developed in Axis A. As such, they can be viewed as particular applications of Axis A. The relationship between various axis is illustrated on Figure 7.3 below.

## 7.3 Conclusions

In this chapter I have presented my plan for future research. The high-level goal of future research is to provide theoretical foundations for model-based design of decision making algorithms, i.e., algorithms which use mathematical models of the underlying phenomenon to predict its behavior and change according to some objectives. To this end, two questions have to be dealt with:

1. assuming that the model is an adequate representation of the phenomenon, can we guarantee that the decision making algorithm based on this model will achieve the desired objectives,
2. can we guarantee that the model is an adequate representation of the underlying phenomenon.

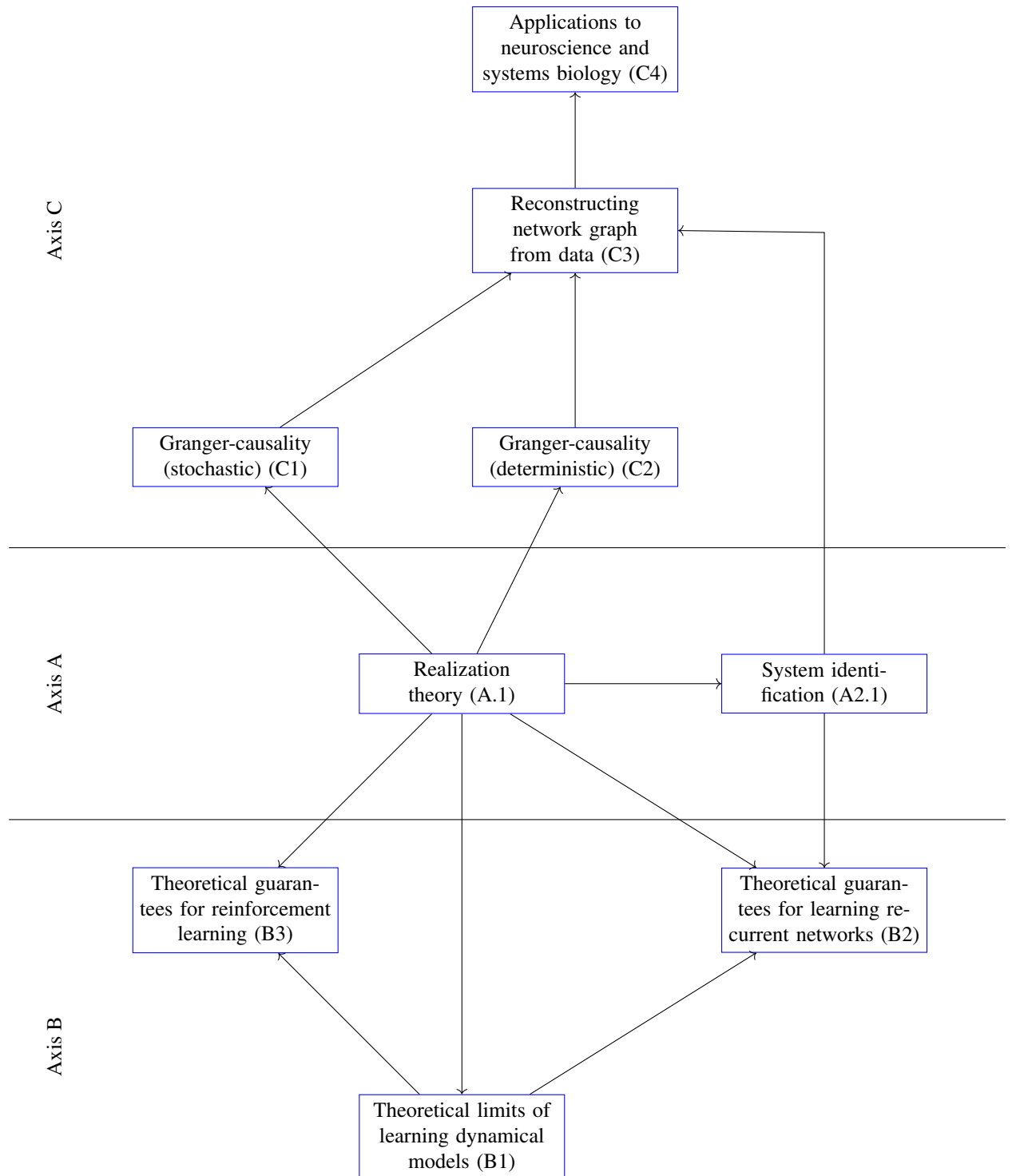
I plan to concentrate on the latter problem: how to get model from data or from other more complex models in such a manner that they represent adequately the underlying phenomenon.

Controller synthesis is a particular case of decision making algorithms, and due to my background, a significant part of my efforts will be directed for contribution theoretical foundations of modeling for control. In particular, I plan to continue contributing to theoretical foundations of systems identification and model reduction for control. What distinguishes the proposed research from the one already carried by other researchers is that it relies heavily on structural, algebraic properties of models, in particular on realization theory. In fact, part of the proposed research focuses on further developing realization theory. That is, the proposed research could be summed up as follows: developing realization theory and applying it theoretical foundations of system identification and model reduction algorithms. Moreover, I plan to concentrate on those theoretical questions which are relevant for controller synthesis.

In addition to theoretical foundations of modeling for control, I plan to work on two other topics: namely, application of control and systems theory to machine learning (PAC, PAC-Bayesian, RNN, reinforcement learning), and reverse engineering of network graphs of interconnected dynamical systems. While these two topics are not related at a first glance, realization theory is relevant for both topics.

That is, future research is organized around realization theory and its applications. This choice is motivated by my desire to have a high degree of continuity with the past research activities while exploring significantly new topics.

Figure 7.3: Relationship between axes A, B and C,





**Part IV**

**General conclusions**



## Chapter 8

# Summary and general conclusions

In this manuscript I have presented some selected topics I have worked on in the past decade. More precisely, I have presented an overview of my work on realization theory and some of its applications to system identification and model reduction. In my research I focused on classes of dynamical systems (hybrid systems, linear parameter-varying systems, bilinear systems) which can be used to represent cyber-physical systems, i.e., systems which combine physical systems with digital computers.

The manuscript starts with a general introduction in Chapter 1, where I explain my research philosophy, my motivation to study realization theory and past research activities. The first chapter is followed by 5 technical chapters, grouped in 2 parts: the first part presents some selected topics on realization theory, the second part presents some applications of realization theory to system identification and model reduction. The manuscript is concluded with the description of future research plans. The manuscript contains two appendices: one appendix contains my detailed CV, the other one contains some background technical material on realization theory of linear switched systems.

As it was discussed in Section 1.1, Chapter 1, the motivation for studying realization theory and its applications is the need to develop mathematically correct methods for building models of cyber-physical systems. Moreover, the built models should be suitable for designing decision algorithms. In particular, the models should be suitable for controller synthesis and fault detection. Such models can be built either from data or from more complex models, which are derived from domain knowledge. Building models from data is the topic of system identification. Computing simplified models from complex one, while preserving relevant properties, is the topic of model reduction. That is, what we need are formal proofs of correctness of system identification and model reduction algorithms. In turn, for the latter tasks it is useful to thoroughly understand the relationship between observed behavior and structure of models representing them, which is the topic of realization theory. In addition, realization theory helps to relate properties of input-output behaviors with those of the underlying models, which is itself useful for controller synthesis.

In the past decade my goal was to apply realization theory of hybrid systems to model reduction and system identification, and to extend realization theory to other system classes.

In order for realization theory of hybrid systems to be useful for system identification, it had to be extended to the discrete-time case, and stochastic noise and switching had to be taken into account. The need to extend realization theory to discrete-time hybrid systems was motivated my work on realization theory of discrete-time switched systems. I also contributed to other realization theory of some other classes of hybrid systems (semi-algebraic discrete-time hybrid systems, continuous-time linear systems with unknown switching, etc.).

It turned out that the realization problem linear parameter-varying (LPV) systems can be developed using realization theory of linear switched systems. Considering the widespread use of LPV systems in applications and the increased need for system identification of LPV systems, this observation motivated me to explore realization theory and system identification of LPV systems. I presented this work in Chapter 2 of this manuscript. In technical terms, these results required non-trivial application of the theory of Fliess-series expansion of input-



output maps of nonlinear systems.

The need to include stochastic noise and stochastic switching for analyzing identification algorithms for hybrid systems lead to the study of jump-Markov systems. The latter inspired the development of realization theory of generalized stochastic bilinear systems, which includes jump-Markov systems and stochastic LPV systems as a special case. In technical terms, this work required a non-trivial extension of the geometric (Hilbert-space based) realization theory of stochastic linear systems, and its merger with realization theory of discrete-time linear switched systems. For this reason, these results were included in Chapter 3, where I presented an overview of realization theory of generalized stochastic bilinear systems, and its application to realization theory of jump-Markov systems.

In parallel to extending realization theory, I also worked on applying existing results to model reduction and system identification. In particular, I worked on applying realization theory of linear switched systems to analyzing model reduction algorithms and to propose new ones. An overview of my work on model reduction of linear switched systems was presented in Chapter 4. My research on model reduction was centered around the following two approaches: balanced truncation and moment matching. In balanced truncation states which are difficult to control and to observer are removed. My work on balanced truncation was aimed at proving analytical bounds for the difference between the input-output behavior of the original and the reduced models. Realization theory served as an important theoretical tool for showing that balanced truncation methods are well-posed (their result does not depend on the choice of the state-space representation) and for providing intuition for the definition of reachability/observability grammians. The proofs of the error bounds relied on proving that the error system is dissipative for a well-chosen storage function. Moment matching was based on calculating a reduced-order model while preserving certain Markov-parameters of the original model. The very idea of this approach relies on realization theory, namely, on the notion of Markov-parameters. The corresponding model reduction algorithms are variations of the realization algorithm developed for linear switched systems.

Another major application of realization I worked was system identification of hybrid systems, and later LPV systems. First I focused on persistence of excitation and structural identifiability of linear switched systems. However, due to the close relationship between realization theories of LPV and switched systems, I soon started focusing more on structural identifiability and subspace identification of LPV systems. For the latter, realization theory of generalized stochastic bilinear systems turned out to be useful, as it allowed to develop elements of realization theory of stochastic LPV systems and to formulate statistically consistent subspace identification methods. Note that the subspace identification algorithms relied on first estimating Markov-parameters/output covariances and then applying the realization algorithm for LPV/generalized stochastic bilinear systems. In that respect, they differ from many other subspace identification algorithms for LPV systems found in the literature. An overview of my research on identifiability and subspace identification of LPV systems was presented in Chapter 5.

In addition to classical system identification for LPV systems, I also worked on reverse engineering of the network structure of interconnected systems. This work was presented in Chapter 6. The aim was to establish a relationship between statistical properties (Granger-causality) of the observed output and the communication graph of the model which generates those outputs. First, the case of outputs generated by stochastic LTI models was considered, later similar results were derived for outputs generated by stochastic generalized bilinear systems. This work is a non-trivial application of stochastic realization theory. Moreover, due to the use of Granger-causality, which is an equivalent formulation of feedback freeness, these results are expected to be useful for analyzing identification algorithms for generalized bilinear systems with additive inputs. The latter includes stochastic LPV systems and jump-Markov linear systems. This research direction can be viewed as a particular case of system identification in the broad sense of the word, as the original goal was to indeed find the network structure from data. However, the latter problem is ill-posed, as in general, the network structure cannot be uniquely determined from data. Instead, the best what we can do is to determine if a certain network structure is consistent with data, i.e., if there exists a model of the observed data with a given network structure. Even for the latter case, the derived results represent only first steps and further research is required to derive efficient and well-founded statistical hypothesis testing methods.

As it was mentioned in Section 1.3 of Chapter 1, in addition to the topics mentioned above, I also worked on several others, such as theory of descriptor systems, symbolic control of hybrid systems, discrete-event systems

---

and their applications, applications of control theory to water networks, stability analysis and observer design of sampled-data systems, machine learning. The results were not included in the manuscript as they are either recent research directions or they are less representative of my main research interests.

My plans for future research can be divided into the following directions. First, I would like to continue working on reliable modeling for control of cyber-physical systems. In particular, I would like to further develop realization theory of hybrid and LPV systems, and use it for developing the theoretical foundations of system identification and model reduction of these systems. In addition, I would like to explore the interplay between the structural properties of these systems (controllability, observability, minimality) and control synthesis. The second research direction is the development of reliable machine learning algorithms for cyber-physical systems. In particular, I would like to work on formal mathematical proof of correctness (consistency) of machine learning algorithms when applied to cyber-physical systems. At a first glance it represents a departure from my previous research, however, this is not entirely the case, as machine learning applied to physical systems can be viewed as a particular case of system identification. In fact, many of the challenges are similar. In particular, I expect that developing realization theory for neural networks will be useful for this research direction. I have already done some preliminary work in this direction. Another future research direction is the reverse engineering of network structures, that is, the continuation of the topic discussed in Chapter 6. In addition to theoretical work I also plan to explore applications of reverse engineering of network structures to neuroscience and systems biology. A detailed description of my plans for future research was presented in Chapter 7.

The results discussed in Chapter 2 - 6 were derived in collaboration with a large number of researchers from several countries, and are partially based on the work of PhD students and postdocs I co-advised. In particular, Chapter 4 – 6 each include results from PhD theses I co-advised. More precisely:

Chapter 2 (realization theory of LPV systems) was developed in collaboration with Guillaume Mercère (Université de Poitiers, France) and Roland Tóth (Eindhoven University of Technology, The Netherlands).

Chapter 3 (realization theory of generalized stochastic bilinear systems) was carried out in collaboration with René Vidal (Johns Hopkins University, USA), and was started when I had a postdoc position under his supervision.

Chapter 4 (model reduction of hybrid systems):

- the results on moment matching using Markov-parameters were part of the PhD thesis of Mert Bastug (Aalborg University, Denmark and IMT-Lille-Douai, France), whom I co-advised with John Leth and Rafael Wisniewski (Aalborg University, Denmark)
- the results on balanced truncation and moment matching in frequency domain were derived in collaboration with John Leth, Rafael Wisniewski, Ion Victor Gosea (Max Planck Institute, Germany), Athansios C. Antoulas (Max Planck Institute, Germany and Rice University, USA) and Christophe Fiter (CRISAL, France).

Chapter 5 (identifiability and subspace identification of LPV systems):

- Identifiability of LPV systems was the topic of the PhD thesis of Ziad Alkhoury (Université de Poitiers and IMT-Lille-Douai, France), whom I co-advised with Guillaume Mercère,
- The results on reduced basis subspace identification algorithm for deterministic LPV systems were developed in collaboration with Roland Tóth and Pepijn Cox (Eindhoven University of Technology, The Netherlands).
- Subspace identification algorithms for stochastic LPV were developed in collaboration with Manas Mejari (CRISAL, France), of whom I was the advisor during his postdoctoral appointment,

Chapter 6 (reverse engineering of networks) is based on the PhD thesis of Mónika Józsa (Groningen University, The Netherlands, IMT-Lille-Douai, France), whom I co-advised with Kanat Camlibel (Groningen University, The Netherlands).

A more detailed overview of the PhD students and postdocs I advised is presented in Appendix A. In addition to the collaborations described above, I also worked together with a number of other researchers on topics which were not discussed in the manuscript. For a detailed overview of my other collaborations see Appendix A.

# **Part V**

## **CV**



# Appendix A

## CV

### A.1 Professional experience

- **October 1, 2015 – present** Chargé de recherche CNRS 1er classe, Centre de Recherche en Informatique, Signal et Automatique de Lille (CRIStAL), UMR CNRS 9189, France.
- **2011 September 1 – September 30, 2015**  
Assistant Professor (Maître assistant), Département Informatique et Automatique, École des Mines de Douai (IMT-Lille-Douai), France.
- **2009 August 1 – 2011 August 31**  
Assistant Professor, Department of Knowledge Engineering, Maastricht University, The Netherlands.
- **2007 June 15 – 2009 July 31**  
Post-doc, Systems Engineering, Department of Mechanical Engineering, Eindhoven University of Technology, The Netherlands.
- **2006 August 7 – 2007 June 9**  
Post-doc, Center for Imaging Science, Johns Hopkins University, USA.
- **2005 October – 2006 March**  
CTS Fellow at INRIA Sophia-Antipolis, France
- **2002 August 1 – 2006 July 31**  
PhD student at CWI, The Netherlands.

### A.2 Education

- **2006 June 22** : PhD, Vrije Universiteit, Amsterdam, The Netherlands. Title of the PhD thesis: *Realization theory of hybrid systems*.
- **2002 August 14**: Masters (MSc.) Computer Science, Vrije Universiteit, Amsterdam, The Netherlands

### Summary of research activities since 2006

- Co-author of 27 international journal papers, over 50 papers in conference proceedings, 5 book chapters. Published in IEEE Transactions on Automatic Control (7), Automatica (8), Systems & Control Letters (3),

SIAM Journal of Optimal Control (1), ESAIM COCV (2), International Journal of Robust and Nonlinear Control (1), Nonlinear Analysis: Hybrid Systems (1), and leading conferences CDC, ACC, ECC, IFAC World Congress, HSCC, ADHS, MTNS, NOLCOS.

- Advising PhD students: 3 graduated
- Advising postdocs: 6 postdocs.

## A.3 Supervision of postdocs and PhD students

### A.3.1 Supervision of PhD students

I have advised 3 PhD students, the percentage of my involvement in the supervision was 50%. Below I will present the list of PhD students I co-advised.

**Mert Bastug (April 2013 – April 2016)** His PhD was carried jointly at École des Mines de Douai (IMT-Lille-Douai) and University of Aalborg. He was co-advised by John Leth (Aalborg University) and Rafael Wisniewski (Aalborg University) and by me. The topic of his PhD thesis was model reduction of linear switched and LPV systems. He defended his thesis, entitled *Model reduction of hybrid systems*, on February 26, 2016. His thesis defense committee was a high profile one: Prof. Athanasios Antoulas (Rice University, Texas, USA & Jacobs University, Bremen, Germany), Prof. Zhendong Sun (Chinese Academy of Sciences, Beijing, China). The former is a leading researcher in model reduction, the latter is one of the leading researchers in theory of switched systems. Mert Bastug received his degree from Aalborg University. He continued his career as a postdoc at SyNeR, working on the ANR project ROCC-SYS, under the supervision of Laurentiu Hetel, and in 2018 he joined the ASITIX company in Lille as an engineer on a permanent contract. This thesis resulted in 2 journal papers (1 IEEE TAC full paper, 1 Automatica brief paper) and 3 conference papers.

**Ziad Alkhoury (December 2014 – November 2017)** His PhD was carried out jointly at École des Mines de Douai (IMT-Lille-Douai) and at Université de Poitiers. His other co-advisor is Guillaume Mercère from Université de Poitiers. The topic of his PhD thesis is identification of LPV systems. Ziad Alkhoury defend his thesis, entitled *Minimality, input-output equivalence and identifiability of LPV systems in State-Space and Linear Fractional Representations* on November 9, 2017, and was awarded his PhD degree by Université de Poitiers. His thesis defense committee was also high profile one: Marion Gilson (CRAN, Université de Lorraine), Xavier Bombois (CNRS, École Centrale de Lyon), Herbert Werner (Hamburg University of Technology), Thierry Poinot (LIAS, Université de Poitiers), Roland Tóth (Eindhoven University of Technology). After graduation, Ziad Alkhoury joined the research laboratory ICube at University of Strasbourg as a postdoc. This thesis resulted in 1 journal paper (Automatica, full paper) and 2 conference papers.

**Mónika Józsa (September, 2014 – September 2018)** Her PhD was carried out jointly at École des Mines de Douai (IMT-Lille-Douai) and at University of Groningen. Her other co-advisor is Kanat Camlibel from University of Groningen, The Netherlands. She is expected to defend her thesis, entitled *Relationship between Granger non-causality and network graph of state-space representations*, on February 25, 2019 and be awarded a PhD degree by the University of Groningen. Her thesis assessment committee is a high profile one too: Arjan van der Schaft (University of Groningen, The Netherlands), Peter Caines (McGill University, Canada), Ralf Peeters (Maastricht University, The Netherlands). The same is true for her thesis defense committee: Arjan van der Schaft, Ralf Peeters, Jan H. van Schuppen (Delft University of Technology, The Netherlands), Paul van den Hof (Eindhoven University of Technology, The Netherlands). She continued as a Research Associate in Computational and Theoretical Neuroscience, Department of Engineering, Cambridge University. This thesis lead to 2 journal papers (1 IEEE TAC full paper, 1 IEEE TAC technical note) and 3 conference papers.

### A.3. SUPERVISION OF POSTDOCS AND PHD STUDENTS

---

**Deividas Eringis (October 2020 – ongoing)** His PhD thesis deals with applying control theory to machine learning, he is co-supervised with Rafael Wisniewski, John Leth and Zheng-Hua Tan. This project ongoing, it has resulted in 1 conference paper, 1 journal submission, and 1 conference submission.

**Elie Rouphael (October 2021 – ongoing)** His PhD deals with application of realization theory of LPV and hybrid systems to system identification. This thesis is co-supervised with Prof. Lotfi Belkour from University of Lille. The thesis is ongoing.

#### A.3.2 Supervision of postdocs

**Hassan Omran** I advised him in the period April 2014 – December 2014, during this period he was a postdoc at École des Mines de Douai (IMT-Lille-Douai), he was recruited for regional project 'Estimation distribuée de systèmes dynamiques' (ESTIREZ). He is currently an Associate Professor at Université de Strasbourg. This postdoc results in 1 journal paper (Automatica brief paper)

**Lucien Etienne** He was recruited first as a research engineer (February 2016 – April 2016) and later (he defended his PhD thesis in April) as a post-doc (starting from May 1, 2016) at SyNeR, for the regional project ESTIREZ, financed by Région Hauts de France. Laurentiu Hetel and Denis Efimov were his principal advisors were. In 2017 he joined IMT Lille Douai as a tenured associate professor. This postdoc resulted in 1 journal paper (Automatica brief paper) and 1 conference paper.

**Mert Bastug** He was recruited on April 1, 2016 as a postdoc at SyNeR, for the ANR project ROCC-SYS. His principal advisor was Laurentiu Hetel. He is my former PhD student. In 2018 he joined the ASITIX company in Lille as an engineer on a permanent contract. This postdoc resulted in 2 conference papers and 1 book chapter.

**Manas Dilip Mehari** He was recruited on October 1, 2018 on the project CPER Data MLC, he left in June, 2019 for a postdoc position at Dalle Molle Institute for Artificial Intelligence, Lugano, Switzerland. This postdoc resulted in 2 conference papers.

**Thibault Defourneau** He was recruited on October 1, 2018 on the project CPER Data MLC, her left in June, 2019 for a data scientist position at Trinove, Paris, France. This postdoc resulted in 1 conference paper.

**Alireza Fakhrizadeh-Esfahani** He was recruited in July, 2019 on the project CPER Data MLC, he left on June 30, 2020. He is currently a postdoc at IMT-Lille-Douai. This collaboration resulted in 2 conference papers, one in the highly ranked machine learning conference AAAI.

**Martin Gonzales** He was recruited on January 1, 2022 on the project 'Reliable AI for cyber-physical systems using control theory' with this a joint project with IRT System-X within the programme. call 'Confiance.ia'. He is currently employed by this project. This collaboration has resulted in 1 submitted journal paper and 1 accepted workshop paper at the flagship conference ICML 2022.



## A.4 Scientific collaboration

Table A.1: Summary of the principal national and international collaborations

Name	Affiliation	Publications, projects, etc.
Guillaume Mercère	Université de Poitiers, France	Publications [5, 7, 6, 213, 216] PhD of Ziad Alkhoury: 2014–2017
Laurent Bako	École Centrale de Lyon, France	Publications [206, 210, 209, 207, 208]
Stéphane Lecoecuche	IMT Lille Douai, France	Publications [207, 208, 113]
Mohamed Djemai	Université de Valenciennes, France	Publications [13]
John Leth	Aalborg University, Denmark	Publications : [30, 31, 29, 28, 28, 228, 227, 82, 108, 212]
Rafael Wisniewski		PhD of Mert Bastug: 2013-2016 PhD of Deividas Eringis: 2020 – ongoing
René Vidal	Johns Hopkins University, USA	Publications: [226, 223, 222, 224, 81]
Kanat Camlibel	University of Groningen, The Netherlands	Publications [137, 136, 134, 135] PhD of Monika Jozsa: 2014-2018
Serhiy Zhuk	IBM Research, Dublin, Ireland	Publications: [229, 317]
Ralf Peeters, Ronald Westra	Maastricht University, The Netherlands	Publications: [214, 299]
Jan H. van Schuppen	Delft University of Technology, The Netherlands	Publications: [185, 183, 186, 187, 188, 221, 220, 209, 210, 218, 217, 219, 215, 211, 272]
Jana Nemcova	Institute of Chemical Technology, Prague, Czech Republic	Publications: [182, 185, 181, 183, 186, 187, 188]
Roland Tóth,	Eindhoven University of Technology, The Netherlands	Publications: [28, 216, 66, 1, 65]
Ion Victor Gosea Anthansios C. Antoulas	Max Planck Institute for Dynamics of Complex Technical Systems  Magdeburg, Germany	Publications: [105, 106, 108, 107, 230]
Dirk A. van Beek, Ramon R. Schiffelers  Rolf J. M. Theunissen	Jacobus E. Rooda, Eindhoven University of Technology, The Netherlands	publications [272, 211, 215, 38]
Eric Duviella Lala Rajaoarisoa Klaudia Horvath	IMT-Lille-Douai IMT-Lille-Douai IMT-Lille-Douai	Publications [121, 122, 120]
Pascal Germain	Université Laval, Canada	PEPS Blanc 2019, publications [253]

## A.5. FUNDED PROJECTS

---

Table A.2: Collaboration with the members of CRISAL

Name	Affiliation	Publications, projects, etc.
Laurentiu Hetel	CO2,SHOC	Publications: [84, 83, 193, 27, 26, 25] Supervision of postdocs: Lucien Etienne Regional project: ESTIREZ
Christophe Fiter	CO2, SHOC	Publications: [106]
Jean-Pierre Richard	CO2, VALSE	Publications: [193] Regional project: ESTIREZ
Denis Efimov	CO2, VALSE	Publications: [28] Regional project: ESTIREZ

## A.5 Funded projects

- 2022-2023: Co-PI of ‘Reliable AI for cyber-physical systems using control theory’, joint project with IRT System-X, in the program ‘Confiance.ia’, Other co-PIs: Hatem Hajri (IRT System-X), Geoffrey D. Delhomme (Airbus).  
Funding: 140 000 EUR,
- 2019: Co-PI of PEPS Blanc 2019, ‘PAC-Bayesian theory for recurrent neural networks: a control theoretic approach’, other PI: Pascal Germain, INRIA Nord.  
Funding: 8000 EUR.
- 2018 - 2020: Co-PI of the regional project CPER Data ‘Machine learning meets control’, other PI: Daniil Ryabko, INRIA Nord.  
Funding: 109 00 EUR.
- 2013-2017: Coordinator of regional project ‘Estimation distribuée de systèmes dynamiques en réseaux’, financed by the region Hauts de France. Participating teams: École des Mines de Douai (IMT-Lille-Douai), INRIA Non-A, CNRS CRISAL SYNER.  
Funding: 262 578 EUR.
- 2013-2014: participant in PHC VAN GOGH project 29342QL, together with Vincent Laurain (CRAN,France) and Roland Toth (TUE, The Netherlands)  
Funding: 2500 EUR.

## A.6 Organizational activities and service to the scientific community

**Organization of invited session, workshops, conferences** I was also involved in the organization of invited sessions and workshops and I was member of program committees of several conferences, below is a detailed list of these activities:

- Member of the local organizing committee of JAMACS 2016, ‘Journées d’Automatique du GDR MACS 2016’, to be held in Lille, 15-16 November 2016
- Organizer of the workshop ‘Realization theory and its role in system identification’, 55th Conference on Decision and Control, Las Vegas, USA, 2016.
- IPC member of the Sixth Workshop on Design, Modeling and Evaluation of Cyber Physical Systems, CyPhy’16, Pittsburgh, PA, USA, 2016.
- IPC member IFAC Conference on Analysis and Design of Hybrid Systems, ADHS 2018, Oxford, UK, ADHS 2021, Brussels, Belgium.

**Membership in scientific committees** I am a member of the technical committee (TC) on System identification & Adaptive Control of CSS (Control Systems Society) of IEEE.

#### **Editorial activities**

- Associate editor of Systems & Control Letters, 2019 – present.
- Associate editor of Nonlinear Analysis: Hybrid Systems, 2020 – present.

#### **Review of publications and project proposals**

- I have been a reviewer for the major journals and conferences of control theory (IEEE Transactions on Automatic Control, Automatic, Systems & Control Letters, IEEE Conference on Decision and Control, American Control Conference, etc.)
- I evaluated several project proposals for the region Grand Est (France).

#### **Membership in PhD defense committees**

- Opponent PhD thesis of Dobriborsci Dmitrii, ITMO University, St. Petersburg, December 2019.
- Assesor of the PhD thesis of Gabrielle Fiore, University of L'Aquila, May 2017.
- PhD defense and evaluation committee of Ivo Bleylevens, December 09, 2010, Maastricht University.
- PhD defense committee of Jana Nemcova, December 02, 2009, Vrije Universiteit, Amsterdam.

## **A.7 Teaching experience**

#### **PhD level courses**

- Course "System identification", France Excellence Summer School ACES 2021, ACES 2022.
- Course "Hybrid Dynamical Systems", lab classes, (together with Laurentiu Hetel, Christophe Prieur, Alexander Kruszewski), PhD School of H2020 Project UCoCoS, 2019.
- Course "Switched Systems and Observations: theory and applications" (together with Jean-Pierre Barbot, Marc Jungers, Laurentiu Hetel), JD-JN MAC'19, 2019.
- Course "Geometric Linear Control Theory" (together with Rafael Wisniewski and John Leth). Doctoral School The Doctoral School of Engineering and Science (2014, 2020) /Electrical and Electronic Engineering, Aalborg University.

#### **Master level courses**

- Programming in Python II: lectures and lab classes (SKEMA Bussiness School, Spring 2021,2022).
- Programming in Python I: lectures and lab classes (SKEMA Bussiness School, Fall 2020,2021).
- Machine learning, statistics, econometrics: lectures and lab classes (SKEMA Bussiness School, Fall 2021).
- Machine learning: lectures and lab classes, part of the module 'Autonomous transportation systems' (Ecole Centrale de Lille, Fall 2019, 2020).

## A.7. TEACHING EXPERIENCE

---

- Sampled data control and observers: lectures and lab classes, part of the module ‘Autonomous transportation systems’ (Ecole Centrale de Lille, Fall 2019, 2020).
- Programming in C: lectures and lab classes, part of the module ‘Autonomous transportation systems’ (Ecole Centrale de Lille, Fall 2019, 2020).
- Programming in Python: lectures and lab classes (SKEMA Business School, Fall 2018, Spring 2020).
- Microcontroller programming: lectures and lab classes, part of the module ‘Autonomous transportation systems’ (Ecole Centrale de Lille, Fall 2018).
- Control of dynamical systems (LMI & LPV, elements of robust control, Lyapunov theory): lectures and exercise classes (Ecole Centrale de Lille, Winter 2015-2016, 2016-2017, 2017-2018).
- Linear control, part of the module: Control of dynamical systems with application to robotics, lectures, lab and exercises classes (Ecole Centrale de Lille, Spring 2018).
- Tools for control of complex systems (Kalman-filters): lecture and exercise classes (Ecole Centrale de Lille, Spring 2016, 2017).
- Modelling for control of physical systems (linear state-space control): lecture and exercise classes (Institute de Genie Informatique et Industriel, Ecole Centrale de Lille, Spring 2016, 2017, 2018).
- Architecture of embedded systems: lab classes (IMT Lille Douai, Fall 2014, 2015, 2016, 2017, 2018, 2019,202).
- C Programming: lectures and lab classes (Ecole des Mines de Douai, Spring 2015).
- Linear control: lectures and lab classes (Ecole des Mines de Douai, Spring 2014, 2015).
- Systems Identification (Maastricht University, Spring 2010, 2011)
- Topics in Computation and Control (Maastricht University, Spring 2011)
- Stochastic Decision Making (Maastricht University, Fall 2010)

### **Bachelor level courses**

- Calculus I (Maastricht University, Fall 2009, 2010).
- Calculus II (Maastricht University, Spring 2010,2011).
- Capita Selecta – Selected topics (Maastricht University, Spring 2010, 2011).
- Advanced Programming (Java), exercise classes (Ecole des Mines de Douai, Fall 2011, 2012).
- Programming and algorithms (Java) (Ecole des Mines de Douai, Fall 2012, 2013).
- Automatic control (discrete-event systems), exercise classes (Ecole des Mines de Douai, Fall 2012, Fall 2013).
- Programming and algorithms (Java):online course (Ecole des Mines de Douai, Fall 2014).
- Matlab for engineers, (Ecole des Mines de Douai, Winter 2013, Winter 2014).
- Programming and algorithms (Java), lab and exercise classes (IMT Lille Douai, Fall 2018).

**Co-supervised MSc. and BSc. thesis projects****2009**

‘Supervisory control theory applied to exception handling in Océ printers’, Esmeé Bertens, MSc. thesis, Eindhoven University of Technology, Océ.

**2011**

1. Lian Nouwen, ‘Automatic branch matching of multiple angiographic artery projections to perform 3D reconstruction’ MSc. thesis, Maastricht University and Pie Medical Imaging B.V.
2. Ufuk Gundogmus ‘Contour detection in left ventricular x-ray angiographic images’, MSc. thesis, Maastricht University and Pie Medical Imaging B.V.
3. Remco Bras, ‘Connectivity structure of dynamical systems’, BSc. thesis, Maastricht University.

**A.8 List of publications which appeared after completing my PhD****Journal Papers**

1. Hossam Abbas, Roland Toth, Mihaly Petreczky, Nader Meskin, Javad Mohammadpour Velni, Patrick J.W. Koelewijn.  
*LPV Modeling of Nonlinear Systems: A Multi-Path Feedback Linearization Approach.*  
International Journal of Robust and Nonlinear Control, Wiley, 31(18):9436 – 9465, 2021.
2. Mihaly Petreczky, Laurent Bako, Stéphane Lecoeuche, Koffi M. D. Motchon.  
*Minimality and identifiability of discrete-time switched autoregressive exogenous systems.*  
International Journal of Robust and Nonlinear Control,30: 5871 – 5891, 2020.
3. Monika Jozsa, Mihaly Petreczky, Kanat M. Camlibel. *Causality and network graph in general bilinear state-space representations.*  
IEEE Transactions on Automatic Control, 65(8), 2623–2630, 2020.
4. Monika Jozsa, Mihaly Petreczky, Kanat M. Camlibel.  
*Relationship between Granger non-causality and network graph of state-space representations,*  
IEEE Transactions Automatic Control, 64(3):912-927, 2019.
5. Ion Gosea, Mihaly Petreczky, Athanasios Antoulas.  
*Data-Driven Model Order Reduction of Linear Switched Systems in the Loewner Framework.*  
SIAM Journal on Scientific Computing, Society for Industrial and Applied Mathematics, 40 (2):572–610, 2018.
6. Ion Victor Gosea, Mihaly Petreczky, Athanasios C. Antoulas, Christophe Fiter.  
*Balanced truncation for linear switched systems.*  
Advances in Computational Mathematics, 44:1845-1886, 2018.
7. Pepijn Cox, Roland Tóth, Mihaly Petreczky.  
*Towards Efficient Maximum Likelihood Estimation of LPV-SS Models.*  
Automatica, 97:392–403, 2018.
8. Mihaly Petreczky, René Vidal.  
*Realization Theory of Generalized Bilinear Systems.*  
IEEE Transactions on Automatic Control, 63(1):69-84 2018.

#### A.8. LIST OF PUBLICATIONS WHICH APPEARED AFTER COMPLETING MY PHD

---

9. Lucien Etienne, Laurentiu Hetel, Denis Efimov, Mihaly Petreczky.  
*Observer synthesis under time-varying sampling for Lipschitz nonlinear systems.*  
Automatica, 85:433-440, 2017.
10. Ziad Alkhoury, Mihaly Petreczky, Guillaume Mercère  
*Identifiability of Affine Linear Parameter-Varying Models,*  
Automatica, 80:62-74, 2017.
11. Mihaly Petreczky, Serhiy Zhuk.  
*Solutions of differential-algebraic equations as outputs of LTI systems: Application to LQ control problems.*  
Automatica, 84:166-173, 2017.
12. Serhiy Zhuk, Mihaly Petreczky.  
*Minimax Observers for Linear Differential-Algebraic Equations.*  
IEEE Transactions on Automatic Control, 62(8):4104–4108, 2017.
13. Mihaly Petreczky, Roland Tóth, Guillaume Mercère  
*Realization Theory for LPV State-Space Representations with Affine Dependence,*  
IEEE Transactions on Automatic Control, 62(9):4667-4674, 2017.
14. Mert Bastug, Mihaly Petreczky, Rafael Wisniewski, John Leth.  
*Reachability and Observability Reduction for Linear Switched Systems with Constrained Switching*  
Automatica, 74:162-170, 2016.
15. Hassan Omran, Laurentiu Hetel, Mihaly Petreczky, Jean-Pierre Richard, Françoise Lamnabhi-Lagarigue.  
*Dissipativity-based approach for stability analysis of nonlinear sampled-data systems with input-affine control.*  
Automatica, 70:266-274, 2016.
16. Mert Bastug, Mihaly Petreczky, Rafael Wisniewski, John Leth.  
*Model Reduction by Moment Matching for Linear Switched Systems.*  
IEEE Transactions on Automatic Control 61(11): 3422-3437, 2016.
17. Rolf J.M. Theunissen, Mihaly Petreczky, Ramon R.H. Schifferers, Dirk A. van Beek, Jacobus E. Rooda.  
*Application of Supervisory Control Synthesis to a Patient Support Table of a Magnetic Resonance Imaging Scanner,*  
IEEE Transaction on Automation Science and Engineerig, 11(1): 20 –32, 2014.
18. Mihaly Petreczky, Laurent Bako, Jan H. van Schuppen.  
*Realization theory of discrete-time linear switched systems.*  
Automatica, 49(11):3337-3344, 2013.
19. Jana Nemcova, Mihaly Petreczky, Jan H. van Schuppen.  
*Realization theory of Nash systems.*  
SIAM J. Control Optim. (51-5):3386-3414, 2013.
20. Mihaly Petreczky, Rafael Wisniewski, John J. Leth.  
*Theoretical Analysis of Balanced Truncation for Linear Switched Systems.*  
Nonlinear Analysis: Hybrid Systems, 10:4-20, 2013.
21. Mihaly Petreczky, Jan H. van Schuppen.  
*Partial realization theory for linear switched systems: a formal power series approach.*  
Automatica, 47:2177-2184, 2011.

22. Mihaly Petreczky.  
*Realization theory for linear and bilinear switched systems: Formal power series approach - Part I: Realization theory of linear switched systems*  
ESAIM Control, Optimization and Calculus of Variations, 17:410-445, 2011.
23. Mihaly Petreczky.  
*Realization theory for linear and bilinear switched systems: Formal power series approach - Part II: Bilinear switched systems.*  
ESAIM Control, Optimization and Calculus of Variations, 17:472-492, 2011.
24. Mihaly Petreczky, Jan H. van Schuppen.  
*Generating series for bilinear hybrid systems*  
Systems & Control Letters, 59:218-225, 2010.
25. Mihaly Petreczky, Jan H. van Schuppen.  
*Span-reachability and observability of bilinear hybrid systems*  
Automatica, 46:501-509, 2010.
26. Mihaly Petreczky, Jan H. van Schuppen.  
*Realization Theory For Linear Hybrid Systems*  
IEEE Transactions on Automatic Control, 55:2282 - 2297, 2010.
27. Mihaly Petreczky.  
*Realization theory for linear switched systems: Formal power series approach.*  
Systems and Control Letters, 56:588-595, 2007

#### **Book chapters**

1. Mihaly Petreczky and Ion Victor Gosea.  
*Model reduction and realization theory of linear switched systems.*  
In Realization and Model Reduction of Dynamical Systems: A Festschrift in Honor of the 70th Birthday of Thanos Antoulas. Springer International Publishing, 2022.
2. Jana Nemcova, Mihaly Petreczky, Jan van Schuppen.  
*Towards a system theory of rational systems*  
Operator Theory, Analysis and the State Space Approach. Operator Theory: Advances and Applications, 2018,
3. Mert Bastug, Laurentiu Hetel, Petreczky Mihaly.  
*Minimal- and Reduced-Order Models for Aperiodic Sampled-Data Systems*  
In *Control Subject to Computational and Communication Constraints*, Lecture Notes in Control and Information Sciences, vol. 475, S. Tarbouriech, A. Girard, L. Hetel (eds), Springer-Verlag, ISBN: 978-3-319-78448-9, 2018.
4. Mihaly Petreczky, Aneel Tanwani and Stephan Trenn  
*Observability of Switched Linear Systems*  
In *Hybrid Dynamical Systems: Observation and control*, Lecture Notes in Control and Information Sciences, vol. 457, M. Djemai, M. Defoort (eds), Springer Verlag, ISBN: 978-3-319-10795-0, 2015
5. Mihaly Petreczky  
*Realization theory of linear hybrid systems*  
In *Hybrid Dynamical Systems: Observation and control*, Lecture Notes in Control and Information Sciences, vol. 457, M. Djemai, M. Defoort (eds), Springer Verlag, ISBN: 978-3-319-10795-0, 2015

## A.8. LIST OF PUBLICATIONS WHICH APPEARED AFTER COMPLETING MY PHD

---

6. Klaudia Horváth, Lala Rajaoarisoa, Eric Duviella, Joaquim Blesa, Mihály Petreczky, Karine Chuquet *Enhancing inland navigation by model predictive control of water level – the Cuinchy-Fontinettes case*, In *Transport of Water versus Transport over Water - Exploring the dynamic interplay between transport and water*, Operations Research/Computer Science Interfaces Series, vol. 58, Carlos Ocampo-Martinez, Rudy Negenborn (eds), Springer-Verlag, ISBN 978-3-319-16132-7, 2015.

### Conference proceedings

1. Deividis Eringis, John Leth, Zheng-Hua Tan, Rafal Wisniewski, Alireza Fakhrizadeh Esfahani, Mihaly Petreczky  
*PAC-Bayesian theory for stochastic LTI systems*  
60th IEEE Conference on Decision and Control, 2021.
2. Mihaly Petreczky, John Leth, Rasmus Pedersen and Rafael Wisniewski.  
*Model reduction for linear switched systems with autonomous switching*.  
2021 IEEE American Control Conference (ACC), 2021.
3. Victor Gosea, Mihaly Petreczky, Athanasios C. Antoulas.  
*Reduced-order modeling of LPV systems in the Loewner framework*.  
60th IEEE Conference on Decision and Control, 2021.
4. Ion Victor Gosea, Mihaly Petreczky, John Leth, Rafal Wisniewski, Athanasios C. Antoulas.  
*Model Reduction of Linear Hybrid Systems*.  
59th IEEE Conference on Decision and Control, 2020.
5. Vera Shalaeva, Alireza Fakhrizadeh Esfahani, Pascal Germain, Mihaly Petreczky.  
*Improved PAC-Bayesian Bounds for Linear Regression*.  
Thirty-Fourth AAAI Conference on Artificial Intelligence, 2020.
6. Manas Mejari, Mihaly Petreczky.  
*Consistent and computationally efficient estimation for stochastic LPV state-space models: realization based approach*.  
58th IEEE Conference on Decision and Control, 2019.
7. Manas Mejari, Mihaly Petreczky.  
*Realization and identification algorithm for stochastic LPV state-space models with exogenous inputs*.  
3rd IFAC Workshop on Linear Parameter-Varying Systems, 2019.
8. Thibault Defourneau, Mihaly Petreczky.  
*Realization theory of recurrent neural networks and rational systems*.  
58th IEEE Conference on Decision and Control, 2019.
9. Monika Jozsa, Mihaly Petreczky, Kanat Camlibel.  
*Causality based graph structure of stochastic linear state-space representations*.  
56th IEEE Conference on Decision and Control, 2017.
10. Mert Bastug, Laurentiu Hetel, Mihaly Petreczky.  
*Model Reduction for Aperiodically Sampled Data Systems*.  
20th IFAC World Congress, 2017.
11. Ziad Alkhoury, Mihaly Petreczky, Guillaume Mercère.  
*Comparing Global Input-Output Behavior of Frozen-Equivalent LPV State-Space Models*.  
20th IFAC World Congress, 2017.



12. Monika Jozsa, Mihaly Petreczky, Kanat Camlibel.  
*Relationship between Causality of Stochastic Processes and Zero Blocks of Their Joint Innovation Transfer Matrices.*  
20th IFAC World Congress, 2017.
13. Lucien Etienne, Laurentiu Hetel, Denis Efimov, Mihaly Petreczky.  
*Observer Analysis and Synthesis for Lipschitz Nonlinear Systems under Discrete Time-Varying Measurements.*  
20th IFAC World Congress, 2017.
14. Mert Bastug, Mihaly Petreczky, Laurentiu Hetel.  
*Minimality of Aperiodic Sampled Data Systems.*  
American Control Conference (ACC), 2017.
15. Jana Nemcova, Mihaly Petreczky, Jan H. van Schuppen.  
*Observability reduction algorithm for rational systems.*  
55th Conference on Decision and Control, 2016.
16. Ziad Alkhoury, Mihaly Petreczky, Guillaume Mercère.  
*Structural properties of LPV to LFR transformation: minimality, input-output behavior and identifiability.*  
55th Conference on Decision and Control, 2016.
17. Jana Nemcova, Mihaly Petreczky, Jan H. van Schuppen.  
*Rational observers of rational systems.* 55th Conference on Decision and Control, 2016.
18. Mihaly Petreczky, Rafal Wisniewski, John Leth.  
*Moment Matching for Bilinear Systems with Nice Selections.*  
10th IFAC Symposium on Nonlinear Control Systems, 2016.
19. Mónika Józsa, Mihaly Petreczky, Kanat Camlibel.  
*Towards Realization Theory of Interconnected Linear Stochastic Systems. Extended abstract.*  
22nd International Symposium on Mathematical Theory of Networks and Systems, Minnesota, Minneapolis, MN, USA during July 11-15, 2016.
20. Mert Bastug, Mihaly Petreczky, Roland Tóth, Rafal Wisniewski, John Leth, Denis Efimov.  
*Moment Matching Based Model Reduction for LPV State-Space Models.*  
54th IEEE Conference on Decision and Control, 2015.
21. Faycal Arichi, Mihaly Petreczky, Mohamed Djemai, Brahim Cherki.  
*Observability and Observer Design of Partially Observed Petri Nets.*  
5th IFAC Conference on Analysis and Design of Hybrid Systems (ADHS), 2015.
22. Pepijn Bastiaan Cox, Roland Tóth, Mihaly Petreczky.  
*Estimation of LPV-SS Models with Static Dependency Using Correlation Analysis.*  
1st IFAC Workshop on Linear Parameter Varying Systems, 2015.
23. Jana Nemcova, Mihaly Petreczky, Jan H. van Schuppen.  
*An Algorithm for System Identification of a Discrete-Time Polynomial System without Inputs.*  
17th IFAC Symposium on System Identification, 2015.
24. Denis Efimov, Wilfrid Perruquetti, Mihaly Petreczky  
*On Necessary Conditions of Instability and Design of Destabilizing Controls,*  
53rd IEEE Conference on Decision and Control, 2014.

#### A.8. LIST OF PUBLICATIONS WHICH APPEARED AFTER COMPLETING MY PHD

---

25. Blaise Guepie, Mihaly Petreczky, Stéphane Lecoeuche  
*Eddy current signatures classification by using time series: a system modeling approach*,  
Annual Conference of the Prognostics and Health Management Society 2014
26. Jana Nemcova, Ewa Pawluszewicz, Mihaly Petreczky  
*Unified framework for continuous and discrete time Nash realizations*, Extended Abstract  
Symposium on Mathematical Theory of Networks and Systems 2014
27. Klaudia Horváth, Eric Duviella, Lala Rajaoarisoa, Mihály Petreczky, Karine Chuquet  
11th International Conference on Hydroinformatics, 2014.
28. Klaudia Horváth, Mihaly Petreczky, Lala Rajaoarisoa, Eric Duviella, Karine Chuquet  
*MPC control of water level in a navigation canal - The Cuiinchy-Fontinettes case study*,  
European Control Conference, 2014
29. Hossam S. Abbas, Roland Tóth, Mihaly Petreczky, Nader Meskin, Javad Mohammadpour  
*Embedding of Nonlinear Systems in a Linear Parameter-Varying Representation*,  
IFAC World Congress 2014
30. Mert Bastug, Mihaly Petreczky, Rafael Wisniewski, John Leth  
*Model Reduction of Linear Switched Systems by Restricting Discrete Dynamics*,  
53rd IEEE Conference on Decision and Control, 2014
31. Mert Bastug, Mihaly Petreczky, Rafael Wisniewski, John Leth  
*Model reduction by moment matching for linear switched systems*,  
American Control Conference, 2014
32. Sergiy Zhuk and Mihaly Petreczky.  
*Infinite horizon control and observer design for linear DAEs*.  
52th IEEE Conference on Decision and Control (CDC), 2013.
33. Jana Nemcova and Mihaly Petreczky.  
*Local Nash Realizations*.  
In 18th International Conference on Methods and Models in Automation and Robotics, Miedzyzdroje,  
Poland, 2013.
34. Mihaly Petreczky and Guillaume Mercère.  
*Affine LPV systems: realization theory, input-output equations and relationship with linear switched systems*.  
In 51th IEEE Conference on Decision and Control, 2012.
35. Mihaly Petreczky, Laurent Bako, Stéphane Lecoeuche.  
*Minimality and identifiability of SARX systems*.  
Proceedings of 16th IFAC Symposium on System Identification (SYSID), Brussel, 2012.
36. Jana Nemcova, Mihaly Petreczky, Jan H. van Schuppen.  
*System reduction and system identification of rational Systems*.  
Proceedings of 16th IFAC Symposium on System Identification (SYSID), Brussel, 2012.
37. Mihaly Petreczky, Rafael Wisniewski, John Leth.  
*Theoretical analysis of balanced truncation for linear switched systems*.  
Proceedings of 4th IFAC Conference on Analysis and Design of Hybrid Systems, Eindhoven, 2012.

- 
38. Mihaly Petreczky and Laurent Bako.  
*On the notion of persistence of excitation for linear switched systems.*  
Proceedings of 50th IEEE Conference on Decision and Control (CDC), 2011.
  39. Ronald Westra, Mihaly Petreczky, Ralf Peeters.  
*Identification of Piecewise Linear Models of Complex Dynamical Systems.*  
Proceedings 18th IFAC World Congress 2011.
  40. Mihaly Petreczky, Laurent Bako and Jan H. van Schuppen.  
*Identifiability of discrete-time linear switched systems.*  
In Hybrid Systems: Computation and Control (HSCC) 2010.
  41. Mihaly Petreczky and Jan H. van Schuppen.  
*Observability reduction of piecewise-affine hybrid systems.*  
Proceedings 19th International Symposium on Mathematical Theory of Networks and Systems, 203–210, 2010.
  42. Mihaly Petreczky and Ralf Peeters.  
*Spaces of nonlinear and hybrid systems representable by recognizable formal power series*  
Proceedings 19th International Symposium on Mathematical Theory of Networks and Systems, pages 1051–1058, 2010
  43. Jana Nemcova, Mihaly Petreczky and Jan H. van Schuppen  
*Realization Theory of Nash systems*  
In Proceedings 48th IEEE Conference on Decision and Control (CDC), 2009.
  44. Mihaly Petreczky, Peter Collins, Dirk A. Van Beek, Jan H. van Schuppen, Jacobus Rooda.  
*Sampled-data control of hybrid systems with discrete inputs and outputs*  
In Proceedings of 3rd IFAC Conference on Analysis and Design of Hybrid Systems (ADHS09)
  45. Mihaly Petreczky and Jan H. van Schuppen  
*Realization Theory of Discrete-Time Linear Hybrid System.*  
In Proceedings 15th IFAC Symposium on System Identification, 2009.
  46. Mihaly Petreczky, Rolf Theunissen, Rong Su, Dirk A. Van Beek, Jan H. van Schuppen, Jacobus E. Rooda  
*Control of input/output discrete-event systems.*  
In Proceedings of European Control Conference, 2009.
  47. Ehsan Elhamifar, Mihaly Petreczky, René Vidal  
*Rank Test for Observability of Discrete-Time Jump Linear Systems with Inputs.*  
In American Control Conference, 2009.
  48. E. Bertens, R. Fabel, M. Petreczky, D.A. van Beek, J.E. Rooda.  
*Supervisory control synthesis for exception handling in printers.*  
In Proceedings Philips Conference on Applications of Control Technology. 2009.
  49. Mihaly Petreczky and René Vidal.  
*Realization Theory of Discrete-Time Semi-Algebraic Hybrid Systems*  
Hybrid Systems: Computation and Control (HSCC), LNCS 4981, 2008.
  50. Mihaly Petreczky and René Vidal.  
*Metrics and Topology For Nonlinear and Hybrid Systems.*  
Hybrid Systems: Computation and Control (HSCC 2007), 2007, LNCS 4416.

#### A.8. LIST OF PUBLICATIONS WHICH APPEARED AFTER COMPLETING MY PHD

---

51. Mihaly Petreczky and Rene Vidal.  
*Realization Theory of Stochastic Jump-Markov Linear Systems.*  
In Proceedings 46th IEEE Conference on Decision and Control, 2007.



## Appendix B

# Review of realization theory of linear switched systems

### B.1 Introduction

For the convenience of the reader in this chapter we present an overview of realization theory of linear switched systems published in [203, 219, 209, 202].

### B.2 Linear switched systems: basic definition

We follow the presentation of [203, 227].

A linear switched system (LSS) is a control system of the form

$$\Sigma \begin{cases} (\xi x)(t) = A_{\sigma(t)}x(t) + B_{\sigma(t)}u(t) \\ y(t) = C_{\sigma(t)}x(t) \end{cases} \quad (\text{B.1})$$

where  $x(t) \in \mathbb{R}^n$  is the state at time  $t$ ,  $\sigma(t) \in Q$  is the discrete mode at time  $t$ ,  $y(t) \in \mathbb{R}^p$  is the output at time  $t$ , and  $u(t) \in \mathbb{R}^m$  is the continuous-valued input at time  $t$ , and  $(\xi x)(t) = \frac{d}{dt}x(t)$  in the continuous-time case, and  $(\xi x)(t) = x(t+1)$  in the discrete-time. The set  $Q$  is a finite one, and it is referred to as the set of discrete modes. Moreover,  $A_q \in \mathbb{R}^{n \times n}$ ,  $B_q \in \mathbb{R}^{n \times m}$ ,  $C_q \in \mathbb{R}^{p \times n}$  are the matrices of the linear system in the discrete state  $q \in Q$ .

The following short-hand notation

$$\Sigma = (n, \{(A_q, B_q, C_q) \mid q \in Q\}) \quad (\text{B.2})$$

is used as a short-hand representation for LSSs of the form (B.1). The number  $n$  is called the *dimension (order)* of  $\Sigma$  and will be denoted by  $\dim \Sigma$ .

Intuitively, an LSS is just a control system which switches among finitely many linear time-invariant systems. The switching signal is part of the input. Whenever a switch occurs, the continuous state remains the same, only the differential equation governing the state and output evolution changes. That is, whenever we switch to a new linear system, we start the new linear system from the state which is the final state of the previous linear system. For all this to make sense, all the linear systems should have the same input, output and state-spaces. Finally, we do not have discrete outputs, only continuous ones.

In order to define formally the solution of a switched system (B.1), we have to put some additional restrictions of  $\sigma, x, y, u$ . To this end, recall the notation of Section 2.2 and let us introduce the following notation.

**Notation B.1** (Spaces  $\mathcal{X}, \mathcal{U}, \mathcal{Q}, \mathcal{Y}$ ). We write  $\mathcal{U}, \mathcal{Q}, \mathcal{X}$  and  $\mathcal{Y}$  to denote either  $PC(\mathbb{R}_+, \mathbb{R}^m), PC(\mathbb{R}_+, \mathcal{Q}), AC(\mathbb{R}_+, \mathbb{R}^n)$  and  $PC(\mathbb{R}_+, \mathbb{R}^p)$  (in continuous-time) or  $(\mathbb{R}^m)^\mathbb{N}, \mathcal{Q}^\mathbb{N}, (\mathbb{R}^n)^\mathbb{N}$  and  $(\mathbb{R}^p)^\mathbb{N}$  (in discrete time).

If we fix an initial state  $x_0 \in \mathbb{R}^n$ , then every input  $u$  and switching signal  $\sigma$  give rise to a unique state trajectory  $x$  and output trajectory  $y$ , such that  $(x, u, \sigma, y)$  satisfies (B.1) and such that  $x(0) = x_0$ . We can define the *input-output map* of an LSS induced by the initial state  $x_0$  as the map  $Y_{\Sigma, x_0} : \mathcal{U} \times \mathcal{Q} \rightarrow \mathcal{Y}$  such that  $Y_{\Sigma, x_0}(u, \sigma) = y$ , where  $(x, u, \sigma, y)$  is the unique solution of the LSS (B.1) such that  $x(0) = x_0$ . That is, every initial state of the the LSS gives rise to an input-output map.

There are several ways to model the input-output behavior of an LSS  $\Sigma$ :

- as the set  $\mathcal{B}(\Sigma)$  of tuples  $(u, \sigma, y)$  such that there exists a state trajectory  $x$  of  $\Sigma$  such that  $(x, u, \sigma, y)$  is a solution of  $\Sigma$ ,
- as the set  $\{Y_{\Sigma, x} \mid x \in \mathbb{R}^n\}$  of input-output maps of  $\Sigma$  induced by all possible initial states of  $\Sigma$ ,
- as the input-output map  $Y_{\Sigma, x_0}$  induced by a particular initial state  $x_0$  of  $\Sigma$ .

For the sake of simplicity, we will adopt the third approach. In principle, this would imply that in addition specifying the matrices of  $\Sigma$  we would have to specify an initial state too. Again, for the sake of simplicity, we will consider the the initial state is zero. That is, we formalize the input-output behavior of  $\Sigma$  as the input-output map of  $\Sigma$  induced by the initial state  $x_0 = 0$ . We call the input-output map  $Y_{\Sigma, 0}$  induced by the initial state 0 of  $\Sigma$  the *input-output map* of  $\Sigma$ , and we denote  $Y_{\Sigma, 0}$  by  $Y_\Sigma$ .

We would like to stress that the assumption that the initial state is zero is not essential, it is made in order to avoid notational and terminological difficulties, the general case is treated in [203].

We model the input-output behavior of a system (not necessarily of a finite-dimensional LSS) as a function

$$f : \mathcal{U} \times \mathcal{Q} \rightarrow \mathcal{Y}. \quad (\text{B.3})$$

In the rest of this section, functions of the form (B.3) will be called *input-output maps*. Such a function captures the behavior of a black-box, which reacts to piecewise-continuous inputs and switching sequences by generating outputs in  $\mathbb{R}^p$ . Next, we define what it means that this black-box can be modeled as an LSS, i.e. that an LSS is a realization of  $f$ . The LSS  $\Sigma$  is a *realization of an input-output map  $f$  of the form (B.3)*, if  $Y_\Sigma = f$ , i.e. if the input-output map of  $\Sigma$  coincides with  $f$ . If  $\Sigma$  is a realization of  $f$ , then  $\Sigma$  is a *minimal realization of  $f$* , if for any LSS realization  $\hat{\Sigma}$  of  $f$ ,  $\dim \Sigma \leq \dim \hat{\Sigma}$ . Two LSSs  $\Sigma_1, \Sigma_2$  are said to be *input-output equivalent*, if their input-output maps are equal, i.e.  $Y_{\Sigma_1} = Y_{\Sigma_2}$ . A LSS  $\Sigma$  is said to be *minimal*, if it is a minimal realization of its own input-output map  $f = Y_\Sigma$ .

### B.3 Characterization of minimality by reachability and observability

We start by presenting the main results on minimality of LSSs. To this end, we need to introduce the notions of observability, span-reachability and isomorphism. In the subsequent discussion,  $\Sigma$  denotes a LSS of the form (B.1). An LSS  $\Sigma$  is said to be *observable*, if for any two distinct states  $x_1 \neq x_2 \in \mathbb{R}^n$ , the input-output maps induced by  $x_1$  and  $x_2$  are different, i.e., if  $\forall x_1, x_2 \in \mathbb{R}^n : x_1 \neq x_2 \implies Y_{\Sigma, x_1} \neq Y_{\Sigma, x_2}$ .

Let  $Reach_0(\Sigma) \subseteq \mathbb{R}^n$  denote the reachable set of the LSS  $\Sigma$  from the initial condition  $x_0 = 0$ , i.e.,  $Reach_0(\Sigma) = \{x(t) \in \mathbb{R}^n \mid \text{there exists } \sigma \in \mathcal{Q}, u \in \mathcal{U}, y \in \mathcal{Y}, (x, u, \sigma, y) \text{ solution of } \Sigma\}$ .

The LSS  $\Sigma$  is said to be *reachable*, if  $Reach_0(\Sigma) = \mathbb{R}^n$ . The LSS  $\Sigma$  is *span-reachable* if  $\mathbb{R}^n$  is the smallest vector space containing  $Reach_0(\Sigma)$ . We note that span-reachability and reachability are the same in continuous-time, see [260].

Next, we recall the notion of LSS morphism. Consider two LSSs

$$\Sigma_1 = (n, \{(A_q, B_q, C_q) \mid q \in \mathcal{Q}\})$$

and

$$\Sigma_2 = (n^a, Q, \{(A_q^a, B_q^a, C_q^a) \mid q \in Q\}).$$

An LSS morphism  $\mathcal{S}$  from  $\Sigma_1$  to  $\Sigma_2$ , denoted by  $\mathcal{S} : \Sigma_1 \rightarrow \Sigma_2$ , is an  $n^a \times n$  matrix, such that

$$\forall q \in Q: \quad A_q^a \mathcal{S} = \mathcal{S} A_q, \quad B_q^a = \mathcal{S} B_q, \quad C_q^a \mathcal{S} = C_q.$$

The LSS morphism  $\mathcal{S}$  is said to be an *isomorphism*, if the matrix  $\mathcal{S}$  is square and invertible.

The following theorem summarizes various results on minimality, see [202, 203, 209].

**Theorem B.1** (Minimality, [202, 203, 209]). *A LSS  $\Sigma$  is minimal, if and only if it is span-reachable and observable. If  $\Sigma_1$  and  $\Sigma_2$  are two minimal LSSs, and  $\Sigma_1$  and  $\Sigma_2$  are input-output equivalent, then  $\Sigma_1$  and  $\Sigma_2$  are isomorphic.*

The usefulness of Theorem B.1 becomes more apparent after presenting an algorithm for minimization of LSSs, i.e. for converting an LSS into a minimal one while preserving its input-output map. This means that as far as the external behavior is concerned, we can always replace an LSS with a minimal one. Moreover, this minimal LSS will have such nice properties as observability and span-reachability. Finally, the fact that minimal and equivalent LSSs are isomorphic is important for system identification: it means that while several LSSs can produce the same observed behavior, as long as we restrict attention to minimal LSSs, all possible models fitting the observed behavior are essentially the same (isomorphic).

In fact, any LSS can be transformed to a minimal LSS with the same input-output behavior. In order to formulate the minimization algorithm, we will present a geometric and algebraic characterization of span-reachability and observability. This will be done in the next section.

## B.4 Algebraic and geometric characterization of reachability and observability, minimization algorithm

Below we present an algebraic characterization of span-reachability and observability. This characterization is similar to the well-known conditions on the rank of controllability and observability matrices of LTI systems, and similarly to the LTI case, it allows us to formulate a minimization algorithm for LSSs.

In order to present these conditions, we recall from [227] the definition of the following spaces

**Definition B.1** ( $\mathcal{V}^*$  and  $\mathcal{W}^*$ ). *Let  $\mathcal{V}^* = \mathcal{V}^*(\Sigma)$  be the smallest subspace (with respect to the inclusion) of  $\mathbb{R}^n$  which satisfies  $A_q \mathcal{V}^* \subseteq \mathcal{V}^*$  and  $\text{Im} B_q \subseteq \mathcal{V}^*$  for any  $q \in Q$ . We will call  $\mathcal{V}^*(\Sigma)$  the reachable subspace of  $\Sigma$ . Let  $\mathcal{W}^* = \mathcal{W}^*(\Sigma)$  be the largest subspace (with respect to inclusion), such that  $\mathcal{W}^* \subseteq \ker C_q$  and  $A_q \mathcal{W}^* \subseteq \mathcal{W}^*$ , for any  $q \in Q$ . We will call  $\mathcal{W}^*$  the unobservable subspace of  $\Sigma$ .*

**Remark B.1** (Computing  $\mathcal{V}^*, \mathcal{W}^*$ ). *It is not difficult to see that the spaces  $\mathcal{V}^*$  and  $\mathcal{W}^*$  can be computed as follows. Set  $V_0$  to the space spanned by  $x_0$  and the columns of  $B_q$  and define  $V_k$  recursively as follows:  $V_k = V_0 + \sum_{q \in Q} A_q V_{k-1}$ . It then follows that  $V_{n-1} = \mathcal{V}^*$ . Similarly, if we set  $W_0 = \bigcap_{q \in Q} \ker C_q$  and  $W_k = W_0 \cap \bigcap_{q \in Q} A_q^{-1}(W_{k-1})$ , then  $W_{n-1} = \mathcal{W}^*$ .*

The above steps can be implemented, see [202, 29].

We can also define an explicit matrix representation of the spaces  $\mathcal{V}^*$  and  $\mathcal{W}^*$  too. To this end, we will use the notation and terminology for sequences of letters, their length and lexicographic ordering from Section 2.2. In particular,  $Q^*$  will denote the set of all finite sequences of the elements of  $Q$ , and *without loss of generality we assume that  $Q = \{1, \dots, D\}$*  and let  $v_1 \prec v_2 \prec \dots$  be lexicographic ordering on the sequences from  $Q^*$  as defined in Definition 2.1 for  $X = Q$ . Recall that  $\mathbf{N}(M)$  denotes the number of sequences from  $Q^*$  of length at most  $M$ , and that  $|v_i| \leq M$  if and only if  $i \leq \mathbf{N}(M)$ . Furthermore, we will need the following notation for products of square



matrices  $\{A_q\}_{q \in Q}$  indexed by sequences from  $Q^*$ , as defined in Notation 2.1, Section 2.2. We are now in position to define the matrix representation of  $\mathcal{V}^*$  and  $\mathcal{W}^*$ .

Define the *controllability matrix* of  $\Sigma$  as

$$\mathcal{R}(\Sigma) = \begin{bmatrix} A_{v_1} \tilde{B}, & A_{v_2} \tilde{B}, & \dots, & A_{v_{N(n-1)}} \tilde{B} \end{bmatrix} \in \mathbb{R}^{n \times (mD+1)N(n-1)} \quad (\text{B.4})$$

with  $\tilde{B} = [B_1, B_2, \dots, B_D] \in \mathbb{R}^{n \times (Dm+1)}$ .

Define the *observability matrix* of  $\Sigma$  as

$$\mathcal{O}(\Sigma) = \begin{bmatrix} (\tilde{C}A_{v_1})^T, & (\tilde{C}A_{v_2})^T, & \dots, & (\tilde{C}A_{v_{N(n-1)}})^T \end{bmatrix}^T \in \mathbb{R}^{pDN(n-1) \times n} \quad (\text{B.5})$$

with  $\tilde{C} = [C_1^T, C_2^T, \dots, C_D^T]^T \in \mathbb{R}^{pD \times n}$ . The relationship between  $\mathcal{W}^*$ ,  $\mathcal{V}^*$  and the matrices defined above is as follows.

**Proposition B.1.**  $\mathcal{V}^* = \text{Im} \mathcal{R}(\Sigma)$  and  $\mathcal{W}^* = \ker \mathcal{O}(\Sigma)$ .

In other words, the controllability matrix can be viewed as a matrix representation of  $\mathcal{V}^*$  and the observability matrix can be viewed as a matrix representation of the orthogonal complement of  $\mathcal{W}^*$ . Theoretically, the controllability and the observability matrices could be used to compute the spaces  $\mathcal{W}^*$ ,  $\mathcal{V}^*$ , but this approach would not be very practical, as the size of the matrices involved is exponential in the number of continuous states. For this reason, it is more practical to use the ideas of Remark B.1. We are now ready to state the algebraic and geometric conditions for span-reachability and observability.

**Theorem B.2** ([260, 203, 205, 209]). *Let  $\Sigma = (n, \{(A_q, B_q, C_q) \mid q \in Q\})$  be an LSS.*

1. **Span-Reachability:** *The following three statements are equivalent:*

1.  $\Sigma$  is span-reachable,
2.  $\mathcal{V}^* = \mathbb{R}^n$ ,
3.  $\text{rank} \mathcal{R}(\Sigma) = n$ .

2. **Observability:** *The following are equivalent:*

1.  $\Sigma$  is observable,
2.  $\mathcal{W}^* = \{0\}$ ,
3.  $\text{rank} \mathcal{O}(\Sigma) = n$ .

**Remark B.2** (Minimal LSS may have non-minimal subsystems). *Note that observability (span-reachability) of an LSS does not imply observability (reachability) of any of its linear subsystems. In fact, it is easy to construct a counter example [203]:  $\Sigma = (n, \{(A_q, B_q, C_q) \mid q \in Q\})$ ,  $Q = \{q_1, q_2\}$ ,  $p = m = 1$  and*

$$A_{q_1} = \begin{bmatrix} 0 & 1 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 1 \end{bmatrix}, \quad B_{q_1} = \begin{bmatrix} 0 \\ 1 \\ 0 \end{bmatrix}, \quad C_{q_1} = \begin{bmatrix} 1 \\ 1 \\ 0 \end{bmatrix}^T, \quad A_{q_2} = \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 1 & 0 \end{bmatrix}, \quad B_{q_2} = \begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix}, \quad C_{q_2} = \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix}^T$$

*It is easy to see that  $\Sigma$  is span-reachable and observable, yet none of the subsystems are reachable or observable.*

*Together with Theorem B.1, which states that minimal realizations are unique up to isomorphism, this implies that there exist LSSs which cannot be converted to an equivalent LSS where all (or some) of the linear subsystems are observable (or reachable).*

## B.5. EXISTENCE OF A REALIZATION, KALMAN-HO ALGORITHM

The proof of Theorem B.2 can be found in [260, 203, 227]. The intuition behind it is as follows:  $\mathcal{V}^*$  is simply the span of all reachable states, and  $Y_{\Sigma, x_1} = Y_{\Sigma, x_2} \iff x_1 - x_2 \in \mathcal{W}^*$ .

We are now able to recall from [227] the following algorithms for converting a LSS to a reachable, observable and minimal one respectively, while preserving input-output behavior. We follow the presentation of [227].

**Procedure B.1** (Reachability reduction). *Assume that  $\dim \mathcal{V}^* = r$  and choose a basis  $b_1, \dots, b_n$  of  $\mathbb{R}^n$  such that  $b_1, \dots, b_r$  span  $\mathcal{V}^*$ . It is easy to see that in this new basis, the matrices  $A_q, B_q, C_q$  can be rewritten as*

$$A_q = \begin{bmatrix} A_q^R & A_q' \\ 0 & A_q'' \end{bmatrix}, C_q = [C_q^R, C_q'], B_q = \begin{bmatrix} B_q^R \\ 0 \end{bmatrix}, \quad (\text{B.6})$$

where  $A_q^R \in \mathbb{R}^{r \times r}$ ,  $B_q^R \in \mathbb{R}^{r \times m}$ , and  $C_q^R \in \mathbb{R}^{p \times r}$ .

Define the LSS  $\Sigma^R = (r, Q, \{(A_q^R, B_q^R, C_q^R) \mid q \in Q\})$ .

It is easy to see that  $\Sigma^R$  is span-reachable, and it is input-output equivalent to  $\Sigma$ . Intuitively,  $\Sigma^R$  is obtained from  $\Sigma$  by restricting the dynamics and the output map of  $\Sigma$  to the subspace  $\mathcal{V}^*$ .

**Procedure B.2** (Observability reduction). *Assume that  $\dim \mathcal{W}^* = n - o$ , and let  $b_1, \dots, b_n$  be a basis in  $\mathbb{R}^n$  such that  $b_{o+1}, \dots, b_n$  span  $\mathcal{W}^*$ . In this new basis, the matrices  $A_q, B_q, C_q$  can be rewritten as*

$$A_q = \begin{bmatrix} A_q^O & 0 \\ A_q' & A_q'' \end{bmatrix}, C_q = [C_q^O, 0], B_q = \begin{bmatrix} B_q^O \\ B_q' \end{bmatrix}, \quad (\text{B.7})$$

where  $A_q^O \in \mathbb{R}^{o \times o}$ ,  $B_q^O \in \mathbb{R}^{o \times m}$ , and  $C_q^O \in \mathbb{R}^{p \times o}$ .

Define LSS  $\Sigma^O = (o, Q, \{(A_q^O, B_q^O, C_q^O) \mid q \in Q\})$ .

It then follows that  $\Sigma^O$  is observable and it is input-output equivalent to  $\Sigma$ . If  $\Sigma$  is span-reachable, then so is  $\Sigma^O$ . Intuitively,  $\Sigma^O$  is obtained from  $\Sigma$  by merging any two states  $x_1, x_2$  of  $\Sigma$ , for which  $x_1 - x_2 \in \mathcal{W}^*$ .

Finally, by combining Procedures B.1 – B.2 and using Theorem B.1, we can formulate the following procedure for minimization of LSSs.

**Procedure B.3** (Minimization). *Transform  $\Sigma$  to a reachable LSS  $\Sigma^R$  by Procedure B.1. Subsequently, transform  $\Sigma^R$  to an observable LSS  $\Sigma^M = (\Sigma^R)^O$  using Procedure B.2. Then  $\Sigma^M$  is a minimal LSS which is input-output equivalent to  $\Sigma$ .*

A more detailed description of the algorithms described in Procedure B.1 – B.3 can be found in [202].

## B.5 Existence of a realization, Kalman-Ho algorithm

In this section we present conditions on existence of a LSS realization of an input-output map and an algorithm for computing a LSS realization from input-output data. *In this section,  $f$  denotes an input-output map of the form (B.3).* Note that at this point we do not assume that  $f$  is an input-output map of an LSS. We start with defining the concept of Markov-parameters and the Hankel-matrix of  $f$ . The former is used to define the latter. Similarly to the the linear case, these concepts are defined using only input-output data. The Hankel-matrices are then used for characterizing the existence of a LSS realization and for computing such a realization from input-output data. This is precisely the reason that the Hankel-matrix is defined without the assumption that an LSS realization exists. *In this section, we will tacitly assume that  $Q = \{1, 2, \dots, D\}$ .* This can be done without loss of generality.

We start with defining the Markov-parameters of  $f$ . To this end, we assume that  $f$  has a *generalized kernel representation* (in the sequel abbreviated as **GKR**), that is, there exists a function  $G^f : \mathcal{Q} \times T \times T \rightarrow \mathbb{R}^{p \times m}$  such that for all  $u \in \mathcal{U}$ ,  $y \in \mathcal{Y}$ , such that

$$f(u, \sigma)(t) = \begin{cases} \int_0^t G^f(\sigma, s, t) u(s) ds & \text{continuous time} \\ \sum_{s=0}^{t-1} G^f(\sigma, s, t) u(s) & \text{discrete time} \end{cases}$$

and  $G$  satisfies a number of technical conditions. For the continuous-time case, these conditions are elaborated in [203, 202], for the discrete-time case these conditions are described in [209]. These conditions imply that there exists maps  $S_{q_1, q_2}^f : \mathcal{Q}^* \rightarrow \mathbb{R}^p$  such that for the discrete-time case

$$G^f(\sigma, s, t) = \begin{cases} S_{\sigma(t), \sigma(s)}(\varepsilon) & s = t - 1 \\ S_{\sigma(t), \sigma(s)}(\sigma(s+1) \cdots \sigma(t-1)) & s < t - 1 \end{cases}$$

For the continuous-time case, the conditions on  $G^f$  imply that  $G^f(\sigma, s, t)$  depends analytically  $t_1, \dots, t_k$ , where  $s = t_0 \leq t_1 < t_2 < \cdots < t_k < t = t_{k+1}$  and  $\sigma$  restricted to  $[t_i, t_{i+1})$  equals the constant  $q_i \in \mathcal{Q}$ ,  $i = 0, \dots, k$  and  $S_{q_0, q_k}^f$  are the coefficients of the Taylor series expansion of  $G^f(\sigma, s, t)$ , if the latter is viewed as an analytic function of  $\tau_i = t_i - t_{i-1}$ ,  $i = 1, \dots, k$ , i.e.,

$$S_{q, q_0}^f(\varepsilon) = G_{q_0 q}^f(0, 0), \quad S^f(q, q_0)(q_1 \cdots q_k) = \frac{d}{dt_1} \cdots \frac{d}{dt_k} G_{q_0 q_1 \cdots q_k}^f(0, t_1, \dots, t_k, 0) \Big|_{t_1 = \dots = t_k = 0}$$

For the precise definition, see [203], or [205, Definition 25].

In both cases (CT and DT), the functions  $\{S_{r, q}^f\}_{r, q \in \mathcal{Q}}$  uniquely determine  $G^f$  and hence  $f$ , and conversely, the values of  $\{S_{r, q}^f\}$  can uniquely be recovered from the values of  $f$ . For the discrete-time case,

$$S_{r, q}^f(v) = [f(u_1, \sigma)(t) \quad \cdots \quad f(u_m, \sigma)(t)],$$

where  $u_i(0)$  is the  $i$ th standard unit vector of  $\mathbb{R}^m$ , and  $u_i(s) = 0$  for all  $s > 0$ ,  $i = 1, \dots, m$ , and  $t$  and  $\sigma$  are defined as follows: if  $v = \varepsilon$ , then  $t = 1$  and  $\sigma$  is any element of  $\mathcal{Q}$  such that  $\sigma(0) = q$  and  $\sigma(1) = r$ ; if  $v = q_1 \cdots q_k$ ,  $q_1, \dots, q_k \in \mathcal{Q}$  and  $k > 0$ , then  $t = k + 1$  and  $\sigma \in \mathcal{Q}$  is such that  $\sigma(0) = q$ ,  $\sigma(t) = r$  and  $\sigma(i) = q_i$  for all  $i = 1, \dots, t - 1$ .

For the continuous-time case, similar formulas can be stated: for any  $v = q_1 q_2 \cdots q_k \in \mathcal{Q}^*$ ,  $q_1, \dots, q_k \in \mathcal{Q}$ ,  $k > 0$  and  $u \in \mathcal{U}$ , define the map  $f_{u, v} : \mathbb{R}_+^k \rightarrow \mathbb{R}^p$  as follows

$$f_{u, v}(t_1, t_2, \dots, t_k) = f(u, \sigma)\left(\sum_{j=1}^k t_j\right).$$

where  $\sigma \in \mathcal{Q}$  is such that for any  $i = 0, \dots, k - 1$ , the restriction of  $\sigma$  to  $[T_i, T_{i+1})$  is  $q_{i+1} \in \mathcal{Q}$ ,  $T_0 = 0$ ,  $T_i = \sum_{j=1}^i t_j$ . In other words,  $f_{u, q_1 \cdots q_k}(t_1, \dots, t_k)$  is the output before the  $k + 1$ th switch, if in every discrete mode  $q_i$  the system stays for  $t_i$  time. Let  $u_j$  be the constant function which takes as value the  $j$ th standard unit vector of  $\mathbb{R}^m$ .

$$S_{q, q_0}^f(v) = \frac{d}{dt_0} \frac{d}{dt_1} \cdots \frac{d}{dt_k} [f_{u_1, q_0 v q}(t_0, t_1, \dots, t_k, 0) \quad \cdots \quad f_{u_m, q_0 v q}(t_0, t_1, \dots, t_k, 0)] \Big|_{t_0 = \dots = t_k = 0} \quad \text{if } |v| = k > 0,$$

$$S_{q, q_0}^f(\varepsilon) = \frac{d}{dt_0} [f_{u_1, q_0 q}(t_0, 0) \quad \cdots \quad f_{u_m, q_0 q}(t_0, 0)] \Big|_{t_0 = 0}$$

(B.8)

Then for any  $v \in \mathcal{Q}^*$ ,  $M^f(v)$  is defined as

$$M^f(v) = \begin{bmatrix} S_{1,1}^f(v), & \cdots, & S_{1,D}^f(v) \\ S_{2,1}^f(v), & \cdots, & S_{2,D}^f(v) \\ \vdots & \cdots & \vdots \\ S_{D,1}^f(v), & \cdots, & S_{D,D}^f(v) \end{bmatrix}. \quad \text{(B.9)}$$

The value of  $M^f$  are called the *Markov parameters* of  $f$ , and  $M^f(v)$  is called the Markov parameter of  $f$  indexed by  $v \in \mathcal{Q}^*$ . Note that the Markov-parameters of  $f$  determine  $f$  uniquely,

## B.5. EXISTENCE OF A REALIZATION, KALMAN-HO ALGORITHM

If  $\Sigma$  be of the form (B.1), and  $\Sigma$  is a realization of  $f$ , then

$$G^f(\sigma, s, t) = C_{\sigma(t)} \Phi_{\sigma}(t, s) B_{\sigma(s)}$$

$\Phi_{\sigma} : \{(t, \tau) \in \mathbb{T} \times \mathbb{T} \mid \tau \leq t\} \rightarrow \mathbb{R}^{n \times n}$ , such that in DT,  $\Phi_{\sigma}(t, s) = A(\sigma(t-1))A(\sigma(t-2)) \cdots A(\sigma(s+1))$ , and in CT  $X_{\sigma, s} : [s, \infty) \ni t \mapsto \Phi_{\sigma}(t, s)$  is the unique absolutely continuous matrix-valued function which satisfies  $\frac{d}{dt} X_{\sigma, s} = A(\sigma(t))X_{\sigma, s}$  and  $X_{\sigma, s}(s) = I_n$ .

Similarly to the linear case, the Markov-parameters  $M^f$  can be written as products of the matrices of an LSS realization. To present this result, we need the following notation. If  $v = q_1 \cdots q_k$  with  $q_1, \dots, q_k \in Q$ ,  $k > 0$ , then

$$A_v = A_{q_k} A_{q_{k-1}} \cdots A_{q_1} \text{ if } k > 0, \quad A_{\varepsilon} = I_n$$

**Lemma B.1** ([202, 209, 203]). *Then  $\Sigma$  is a realization of  $f$ , if and only if  $f$  has a GKR, and for all  $v \in Q^*$ ,*

$$M^f(v) = \tilde{C} A_v [B_1, B_2, \dots, B_D], \quad (\text{B.10})$$

where  $\tilde{C} = [C_1^T, C_2^T, \dots, C_D^T]^T \in \mathbb{R}^{pD \times n}$ .

Using the notation of (B.8), we can rewrite (B.10) as  $S_{q, q_0}^f(v) = C_q A_v B_{q_0}$ ,  $\forall q, q_0 \in Q$ .

**Example B.1.** *Consider the linear switched system of the form (B.1), where  $Q = \{1, 2\}$ , and*

$$A_1 = \begin{bmatrix} 0 & 0 & 0 \\ 0 & 1 & 0 \\ 1 & 0 & 1 \end{bmatrix}, B_1 = \begin{bmatrix} 0 \\ 1 \\ 1 \end{bmatrix}, C_1 = \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix}^T, \\ A_2 = \begin{bmatrix} 0 & 1 & 1 \\ 0 & 0 & 0 \\ 0 & 0 & 1 \end{bmatrix}, B_2 = \begin{bmatrix} 0 \\ 1 \\ 0 \end{bmatrix}, C_2 = \begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix}^T,$$

Consider the input-output map  $f = Y_{\Sigma}$  of  $\Sigma$ . Let us compute the Markov-parameters  $M^f(\varepsilon)$ ,  $M^f(1)$  and  $M^f(2)$  of  $f$ . Since  $\Sigma$  is a realization of  $f$ , we can use (B.10)

$$M^f(\varepsilon) = \begin{bmatrix} C_1 B_1 & C_1 B_2 \\ C_2 B_1 & C_2 B_2 \end{bmatrix} = \begin{bmatrix} 1 & 0 \\ 0 & 0 \end{bmatrix} \\ M^f(i) = \begin{bmatrix} C_1 A_i B_1 & C_1 A_i B_2 \\ C_2 A_i B_1 & C_2 A_i B_2 \end{bmatrix} = \begin{bmatrix} 1 & 0 \\ 0 & 0 \end{bmatrix}, \quad (\text{B.11})$$

where  $i = 1, 2$ .

Next we define the notion of the Hankel-matrix of  $f$ . Similarly to the linear case, the entries of the Hankel-matrix are formed by the Markov parameters of  $f$ . For the definition of the Hankel-matrix of  $f$ , we will use lexicographical ordering on the set of sequences  $Q^*$ .

**Definition B.2** (Hankel-matrix). *Consider the lexicographic ordering  $\prec$  of  $Q^*$  from Define 2.1. Define the Hankel-matrix  $H_f$  of  $f$  as the following infinite matrix*

$$H_f = \begin{bmatrix} M^f(v_1 v_1) & M^f(v_2 v_1) & \cdots & M^f(v_k v_1) & \cdots \\ M^f(v_1 v_2) & M^f(v_2 v_2) & \cdots & M^f(v_k v_2) & \cdots \\ M^f(v_1 v_3) & M^f(v_2 v_3) & \cdots & M^f(v_k v_3) & \cdots \\ \vdots & \vdots & \cdots & \vdots & \cdots \end{bmatrix},$$

i.e. the  $pD \times (mD + 1)$  block of  $H_f$  in the block row  $i$  and block column  $j$  equals the Markov-parameter  $M^f(v_j v_i)$  of  $f$ . The rank of  $H_f$ , denoted by  $\text{rank} H_f$ , is the dimension of the linear span of its columns.

In other words, the  $l$ th block column of  $H_f$  is the sequence of Markov-parameters  $\{M^f(v_l v_k)\}_{k=1}^\infty$ . If  $|Q| = 1$ , then  $H_f$  coincides with the Hankel-matrix as it was defined for linear systems.

**Theorem B.3** (Existence, [202, 209, 203]). *The input-output map  $f$  has a realization by an LSS if and only if  $f$  has a GKR and  $\text{rank}H_f < +\infty$ . A minimal realization of  $f$  can be constructed from  $H_f$ .*

In fact, we can formulate a Kalman-Ho-like realization algorithm for LSSs. To this end, for every  $M, L \in \mathbb{N}$ , we define the following sub-matrix of the Hankel-matrix  $H_f$ :

$$H_{f,L,M} = \begin{bmatrix} M^f(v_1 v_1) & M^f(v_2 v_1) & \cdots & M^f(v_{\mathbf{N}(M)} v_1) \\ M^f(v_1 v_2) & M^f(v_2 v_2) & \cdots & M^f(v_{\mathbf{N}(M)} v_2) \\ \vdots & \vdots & \cdots & \vdots \\ M^f(v_1 v_{\mathbf{N}(L)}) & M^f(v_2 v_{\mathbf{N}(L)}) & \cdots & M^f(v_{\mathbf{N}(M)} v_{\mathbf{N}(L)}) \end{bmatrix}. \quad (\text{B.12})$$

Recall from Definition 2.1 that  $\mathbf{N}(K)$  is the unique integer such that  $\{v_1, \dots, v_{\mathbf{N}(K)}\}$  is the set of all element of  $Q^*$  of length at most  $K$ . Intuitively,  $H_{f,L,M}$  is the sub-matrix of  $H_f$ , obtained by keeping the columns of  $H_f$  indexed by words of length at most  $M$  and keeping the rows indexed by words of length at most  $L$ . In contrast to  $H_f$ , the matrix  $H_{f,L,M}$  is a finite matrix, albeit a very large one: its size is exponential in  $M$  and  $L$ . We are now read to state the realization algorithm in Algorithm 12.

---

#### Algorithm 12 Realization algorithm

**Inputs:** an integer  $N > 0$  and the Hankel-matrix  $H_{f,N,N+1}$ .

**Output:** LSS  $\Sigma_N$

---

- 1: Compute a decomposition  $H_{f,N,N+1} = \mathbf{O}\mathbf{R}$ , where  $\mathbf{O} \in \mathbb{R}^{I_N \times n}$  and  $\mathbf{R} \in \mathbb{R}^{n \times J_{N+1}}$  and  $\text{rank}\mathbf{R} = \text{rank}\mathbf{O} = n$ ,  $I_N = \mathbf{N}(N)pD$  and  $J_{N+1} = \mathbf{N}(N+1)(mD)$ .
- 2: Consider the decomposition

$$\mathbf{R} = \begin{bmatrix} \mathbf{C}_{v_1} & \cdots & \mathbf{C}_{v_{\mathbf{N}(N+1)}} \end{bmatrix},$$

such that  $\mathbf{C}_{v_i} \in \mathbb{R}^{n \times (Dm)}$ ,  $i = 1, 2, \dots, \mathbf{N}(N+1)$ , i.e.  $\mathbf{C}_{v_i} \in \mathbb{R}^{n \times (Dm)}$ ,  $i = 1, 2, \dots, \mathbf{N}(N+1)$  are the block columns of  $\mathbf{R}$ . Define  $\bar{\mathbf{R}}, \mathbf{R}_q \in \mathbb{R}^{n \times J_N}$ ,  $J_N = \mathbf{N}(N)mD$ ,  $q \in Q$  as follows

$$\begin{aligned} \bar{\mathbf{R}} &= \begin{bmatrix} \mathbf{C}_{v_1} & \cdots & \mathbf{C}_{v_{\mathbf{N}(N)}} \end{bmatrix}, \\ \mathbf{R}_q &= \begin{bmatrix} \mathbf{C}_{v_1 q} & \cdots & \mathbf{C}_{v_{\mathbf{N}(N)} q} \end{bmatrix}. \end{aligned}$$

Note that for any  $i \in \{1, \dots, \mathbf{N}(N)\}$  there exists  $j = j(i, q) \in \{2, \dots, \mathbf{N}(N+1)\}$  such that  $v_i q = v_j$ , hence  $\mathbf{R}_q$  is well defined.

- 3: Construct  $\Sigma_N = (n, \{(A_q, B_q, C_q) \mid q \in Q\})$  such that

$$\begin{aligned} [B_1, \dots, B_D] &= \\ &\text{the first } mD \text{ columns of } \mathbf{R} \\ [C_1^T, C_2^T, \dots, C_D^T]^T &= \text{the first } pD \text{ rows of } \mathbf{O} \\ \forall q \in Q: A_q &= \mathbf{R}_q \bar{\mathbf{R}}^+, \end{aligned}$$

where  $\bar{\mathbf{R}}^+$  is the Moore-Penrose pseudo-inverse of  $\bar{\mathbf{R}}$ .

- 4: Return  $\Sigma_N$
- 

The following theorem gives conditions under which the state-space representation returned by Algorithm 12 is a realization of the map  $f$ .

## B.5. EXISTENCE OF A REALIZATION, KALMAN-HO ALGORITHM

**Theorem B.4** ([202, 219]). *If  $\text{rank}H_{f,N,N} = \text{rank}H_f$ , then Algorithm 12 returns a minimal realization of  $f$ . The condition  $\text{rank}H_{f,N,N} = \text{rank}H_f$  holds for a given  $N$ , if there exists a LSS realization  $\Sigma$  of  $f$  such that  $\dim\Sigma \leq N+1$ .*

**Remark B.3** (Determining a correct value for  $N$ ). *In order to ensure that Algorithm 12 yields a correct realization of  $f$ , we need to select a value of  $N$ , such that  $\text{rank}H_{f,N,N} = \text{rank}H_f$  holds. If  $\text{rank}H_{f,N,N} = \text{rank}H_f$  is not satisfied, then the LSS returned by Algorithm 12 might fail to be a realization of  $f$ . By Theorem B.4, one option is to select  $N$  such that  $N$  is larger than the dimension of a LSS realization of  $f$ . Another, more practical, possibility is to assume that  $f$  has a realization by a minimal LSS of dimension  $n$  and then choose the smallest  $N$  such that  $\text{rank}H_{f,N,N} = n$ . Numerical experiments show that in this way one can often take  $N$  to be much smaller than  $n$ . This is important, since the size of  $H_{f,N,N}$  is exponential in  $N$ . Of course, if no knowledge is available on LSS realization of  $f$ , then choosing  $N$  becomes difficult.*

Note that  $H_{f,N,N}$  can be computed from the responses of  $f$ . However, in principle, the computation of  $H_{f,N,N}$  requires an exponential number of input/output experiments involving different switching sequences. This is clearly not very practical. It would be more practical to build  $H_{f,N,N}$  based on the response of  $f$  to a single switching sequence. Preliminary results on the latter approach, for the discrete-time case, can be found in [206].

**Remark B.4.** *One way to compute the factorization  $H_{f,N,N+1} = \mathbf{OR}$  in Algorithm 12 is as follows. If  $\text{rank}H_{f,N,N+1} = n$  and  $H_{f,N,N+1} = USV$  is the SVD decomposition of  $H_{f,N,N+1}$  with  $S$  being the  $n \times n$  diagonal matrix, then define  $\mathbf{O} = US^{1/2}$  and  $\mathbf{R} = S^{1/2}V$ .*

Note that the size of the matrix  $H_{f,N,N+1}$  grows exponentially with  $N$ . This renders Algorithm 12 impractical. There is a way to get around it, by using so called *selections*, which will play an important role in the application of realization theory to model reduction and system identification.

We will present only the SISO case ( $p = m = 1$ ), the full MIMO case was discussed in [214, 30]. We will call any subset  $\alpha \subseteq Q^* \times Q$  a *selection*. Finite selections will be used to define Hankel-like matrices, entries of which are Markov parameters.

Assume that  $\alpha$  and  $\beta$  are selection respectively and assume that  $\alpha$  and  $\beta$  are both finite sets of cardinality  $n$  and  $l$  respectively. Fix an enumeration of the elements of  $\alpha$  and  $\beta$  as follows.

$$\alpha = \{(u_i, q_i)\}_{i=1}^n, \beta = \{(v_j, \sigma_j)\}_{j=1}^l, \quad (\text{B.13})$$

Let us now define the matrix  $\mathcal{H}_{f,\alpha,\beta} \in \mathbb{R}^{n \times l}$  as follows:

$$[\mathcal{H}_{f,\alpha,\beta}]_{i,j} = S_{q_i, \sigma_j}^f(v_j u_i) \quad i = 1, \dots, n, \quad j = 1, \dots, l \quad (\text{B.14})$$

If  $\alpha, \beta$  has the same cardinality, then the matrix  $H_{f,\alpha,\beta}$  is a square one. Intuitively, the rows of  $\mathcal{H}_{f,\alpha,\beta}$  are indexed the elements of  $\alpha$ , and the columns by the elements of  $\beta$ .

In order to present the algorithm, we define the matrices  $\mathcal{H}_{q,f,\alpha,\beta} \in \mathbb{R}^{n \times l}$ ,  $\mathcal{H}_{f,\alpha,q} \in \mathbb{R}^{n \times 1}$  and  $\mathcal{H}_{q,f,\beta} \in \mathbb{R}^{1 \times l}$ :

$$[\mathcal{H}_{q,f,\alpha,\beta}]_{i,j} = S_{q_i, \sigma_j}^f(v_j q u_i), \quad i = 1, \dots, n, \quad j = 1, \dots, l \quad (\text{B.15})$$

$$[\mathcal{H}_{f,\alpha,q}]_i = S_{q_i, q}^f(u_i), \quad i = 1, \dots, n \quad (\text{B.16})$$

$$[\mathcal{H}_{q,f,\beta}]_j = S_{q, \sigma_j}^f(v_j) \quad j = 1, \dots, l \quad (\text{B.17})$$

---

### Algorithm 13 Realization algorithm for LSSs with nice selections

---

Consider the factorization  $H_{f,\alpha,\beta} = O_{n_m} R_{n_m}$  such that  $O_{n_m}$  is full column rank,  $R_{n_m}$  is full row rank and  $\text{rank}O_n = \text{rank}R_n = n_m$ .

Define

$$\hat{A}_q = O_{n_m}^+ \mathcal{H}_{q,f,\alpha,\beta} R_{n_m}^+, \quad \hat{B}_q = O_{n_m}^+ \mathcal{H}_{f,\alpha,q}, \quad \hat{C}_q = \mathcal{H}_{q,f,\beta} R_{n_m}^+$$

and  $O_{n_m}^+, R_{n_m}^+$  is the Moore-Penrose inverse of  $O_n$  and  $R_n$  respectively.

**return**  $\hat{\Sigma} = (n_m, \{\hat{A}_q, \hat{B}_q, \hat{C}_q \mid q \in Q\})$ .

---

**Lemma B.2** (Adapted from [66], Ho-Kalman algorithm). *If  $n_m$  is the dimension of a minimal LSS realization of  $f$ , then the LSS  $\hat{\Sigma}$  defined in Algorithm 13 is a minimal realization of  $f$ . Moreover, if  $n_m$  is the dimension of a minimal LSS realization of  $f$ , then there exists a pair of selections  $\alpha, \beta \subseteq Q^* \times Q$  such that the cardinality of the sets  $\alpha, \beta$  is  $n_m$  and  $\text{rank } \mathcal{H}_{f, \alpha, \beta} = n_m$ .*

From [219] it follows that we can choose  $\alpha_N = \beta_N = \{(v, q) \mid v \in Q^*, |v| \leq N, q \in Q\}$ , where  $N$  is any integer not smaller than the dimension of a minimal LSS realization of  $f$ .

## B.6 Stability and minimality

Stability of LSSs is a classical research topic with a very rich literature [161, 260]. The purpose of this section is to recall some results on preservation of stability under minimization, i.e., to recall results which state that if an LSS is stable in some sense, then the corresponding minimal LSS will also be stable.

More precisely, we will concentrate on quadratically stable LSSs. Informally, a quadratically stable LSS is a LSS which admits a common quadratic Lyapunov function. Formally, a LSS of the form (B.1) *quadratically stable*, if there exists a positive definite matrix  $P > 0$  such that

$$\forall q \in Q : \mathbf{S}(q, \Sigma, P) < 0, \quad (\text{B.18a})$$

where

- in continuous-time (Lyapunov equation)

$$\mathbf{S}(q, \Sigma, P) = A_q^T P + P A_q, \quad (\text{B.18b})$$

- in discrete-time (Stein equation)

$$\mathbf{S}(q, \Sigma, P) = A_q^T P A_q - P. \quad (\text{B.18c})$$

For discrete-time LSSs it will be useful to introduce the notion of *strong stability* [227], which will be used later on for jump-Markov systems and for model reduction: LSS  $\Sigma$  of the form (B.1) is *strongly stable*, if the matrix  $\sum_{q \in Q} A_q^T \otimes A_q^T$  is a stable matrix (all its eigenvalues lie inside the unit disc), where  $\otimes$  denotes Kronecker product. From [64] it follows that if  $\Sigma$  is strongly stable if and only if for any  $\mathcal{G} > 0$ , there exists a unique  $P > 0$  such that  $P = \sum_{q \in Q} A_q^T P A_q + \mathcal{G}$ . In particular,  $P - A_q^T P A_q > 0$  for all  $q \in Q$ , i.e.,  $\Sigma$  is quadratically stable.

It then follows [161, 260] that a quadratically stable LSS is globally uniformly asymptotically stable and its input-output map has a finite  $L_2$  gain, and it is also BIBO stable. We can then state the following result.

**Theorem B.5** ([227]). *If an input-output map  $f$  has a realization by a quadratically stable LSS (respectively strongly stable LSS), then all minimal LSS realizations of  $f$  are quadratically stable (respectively strongly stable).*

# Bibliography

- [1] H.S. Abbas, R. Tóth, M. Petreczky, N. Meskin, and J. Mohammadpour. Embedding of nonlinear systems in a linear parameter-varying representation. In *IFAC World Congress*, 2014.
- [2] F. Albertini and E.D. Sontag. For neural networks, function determines form. *Neural Networks*, 6(7):975–990, 1993.
- [3] F. Albertini, E.D. Sontag, and V. Maillot. Uniqueness of weights for neural networks. In R. Mammone, editor, *Artificial Neural Networks for Speech and Vision*, pages 115–125. Chapman and Hall, London, 1994.
- [4] Z. Alkhoury. *Minimality, input-output equivalence and identifiability of LPV systems in state-space and linear fractional representations*. PhD thesis, 2017. Thèse de doctorat dirigée par Mercère, Guillaume et Petreczky, Mihály Image, signal et automatique Poitiers 2017.
- [5] Z. Alkhoury, M. Petreczky, and G. Mercère. Structural properties of affine LPV to LFR transformation: minimality, input-output behavior and identifiability. In *Proceedings of the IEEE Conference on Decision and Control*, Las Vegas, USA, 2016.
- [6] Z. Alkhoury, M. Petreczky, and G. Mercère. Comparing global input-output behavior of frozen-equivalent lpv state-space models. *IFAC-PapersOnLine*, 50(1):9766 – 9771, 2017.
- [7] Z. Alkhoury, M. Petreczky, and G. Mercère. Identifiability of affine linear parameter-varying models. *Automatica*, 80:62 – 74, 2017.
- [8] R. Alur. *Principles of Cyber-Physical Systems*. The MIT Press. MIT Press, 2015.
- [9] P.-O. Amblard and O. J. Michel. On directed information theory and granger causality graphs. *Journal of computational neuroscience*, 30(1):7–16, 2011.
- [10] A C. Antoulas, S. Lefteriu, and A C. Ionita. *A tutorial introduction to the Loewner Framework for Model Reduction*, pages 335–376. *Model Reduction and Approximation for Complex Systems*, Edited by P. Benner, A. Cohen, M. Ohlberger, and K. Willcox, Series: Computational Science & Engineering. SIAM, 2017.
- [11] A.C. Antoulas. *Approximation of Large-Scale Dynamical Systems*. Advances in Design and Control. 2009.
- [12] C. A. Antoulas. *Approximation of Large-Scale Dynamical Systems*. SIAM, 2005.
- [13] F. Arichi, M. Petreczky, M. Djemai, and B. Cherki. Observability and observer design of partially observed petri nets. In *5th IFAC Conference on Analysis and Design of Hybrid Systems (ADHS)*, 2015.
- [14] A. Astolfi. Model reduction by moment matching for linear and nonlinear systems. *IEEE Transactions on Automatic Control*, 55(10):2321–2336, 2010.



- 
- [15] L. Bako, K. Boukharouba, and S. Lecoeuche. Recovering piecewise affine maps by sparse optimization. *IFAC Proceedings Volumes*, 45(16):356–361, 2012.
- [16] J.A. Ball, G. Groenewald, and T. Malakorn. Structured noncommutative multidimensional linear systems. *SIAM J. on Control and Optimization*, 44(4):1474–1528, 2005.
- [17] B. Bamieh and L. Giarré. Identification of linear parameter varying models. *International Journal of Robust and Nonlinear Control*, 12:841–853, 2002.
- [18] B. Bamieh and L. Giarré. Identification of linear parameter varying models. *International Journal of Robust and Nonlinear Control*, 12:841–853, 2002.
- [19] Yajie Bao, Javad Mohammadpour Velni, and Mahdi Shahbakhti. Epistemic uncertainty quantification in state-space lpv model identification using bayesian neural networks. *IEEE Control Systems Letters*, 5(2):719–724, 2021.
- [20] L. Barnett, A. B. Barrett, and A. K. Seth. Granger causality and transfer entropy are equivalent for Gaussian variables. *Phys. Rev. Lett.*, 103:238701, 2009.
- [21] L. Barnett and A. K. Seth. Granger causality for state space models. *Physical Review E*, 91(4):737–739, 2015.
- [22] Z. Bartosiewicz. Realizations of polynomial systems. In *Algebraic and geometric methods in nonlinear control theory*, volume 29 of *Math. Appl.*, pages 45–54. Reidel, Dordrecht, 1986.
- [23] Z. Bartosiewicz. Rational systems and observation fields. *Systems and Control Letters*, 9:379–386, 1987.
- [24] M. Bastug. *Model Reduction of Linear Switched Systems and LPV State-Space Models*. PhD thesis, Aalborg University, 2016.
- [25] M. Bastug, L. Hetel, and M. Petreczky. Minimality of aperiodic sampled data systems. In *American Control Conference (ACC)*, 2017.
- [26] M. Bastug, L. Hetel, and M. Petreczky. Model reduction for aperiodically sampled data systems. In *20th IFAC World Congress*, 2017.
- [27] M. Bastug, L. Hetel, and M. Petreczky. Minimal- and reduced-order models for aperiodic sampled-data systems. In *Control Subject to Computational and Communication Constraints*, volume 475 of *Lecture Notes in Control and Information Sciences*. Springer-Verlag, 2018.
- [28] M. Bastug, M. Petreczky, R. Tóth, R. Wisniewski, J. Leth, and D. Efimov. Moment matching based model reduction for lpv state-space models. In *Decision and Control (CDC), 2015 IEEE 54rd Annual Conference on*, 2015.
- [29] M. Bastug, M. Petreczky, R. Wisniewski, and J. Leth. Model reduction by moment matching for linear switched systems. In *American Control Conference (ACC), 2014*, pages 3942–3947, June 2014.
- [30] M. Bastug, M. Petreczky, R. Wisniewski, and J. Leth. Model reduction by moment matching for linear switched systems. *IEEE Transactions on Automatic Control*, 61:3422–3437, 2016.
- [31] M. Bastug, M. Petreczky, R. Wisniewski, and J. Leth. Reachability and observability reduction for linear switched systems with constrained switching. *Automatica*, 74:162–170, 2016.
- [32] C. L. Beck. On formal power series representations for uncertain systems. *IEEE Transactions on Automatic Control*, 46:314 – 319, 2001.

## BIBLIOGRAPHY

---

- [33] C. L. Beck and J. Doyle. A necessary and sufficient minimality condition for uncertain systems. *IEEE Transactions on Automatic Control*, 44(10):1802–1813, 1999.
- [34] Carolyn L. Beck. *Model reduction and minimality for uncertain systems*. PhD thesis, California Institute of Technology, 1996.
- [35] J. Belikov, Ü. Kotta, and M. Tönso. Comparison of LPV and nonlinear system theory: A realization problem. *Systems and Control Letters*, 64:72 – 78, 2014.
- [36] M. Bergamasco and M. Lovera. Continuous-time predictor-based subspace identification using Laguerre filter. *IET Control Theory and Applications*, 5(7):856–867, 2011.
- [37] J. Berstel and C. Reutenauer. *Rational series and Their Languages*. Springer-Verlag, 1984.
- [38] E. Bertens, R. Fabel, M. Petreczky, D.A. van Beek, and J.E. Rooda. Supervisory control synthesis for exception handling in printers. In *Philips Conference on Applications of Control Technology*, 2009.
- [39] D. P. Bertsekas. *Dynamic programming and optimal control*, volume 1. Athena Scientific Belmont, MA, 1995.
- [40] B. Besselink. *Model reduction for nonlinear control systems : with stability preservation and error bounds*. PhD thesis, Department of Mechanical Engineering, 2012.
- [41] David Beyer. *The future of machine intelligence*. O’Reilly, 2016.
- [42] F. D. Bianchi and R. S. Sánchez-Peña. Robust identification/invalidation in an LPV framework. *International Journal of Robust and Nonlinear Control*, 2009.
- [43] P. Billingsley. *Probability and measure*. Wiley, 1986.
- [44] A. Birouche, J. Guilet, B. Mourillon, and M. Basset. Gramian based approach to model order-reduction for discrete-time switched linear systems. In *Proceedings of the 18th Mediterranean Conference on Control and Automation*, pages 1224–1229, 2010.
- [45] A. Birouche, B. Mourillon, and M. Basset. Model reduction for discrete-time switched linear time-delay systems via the  $H_\infty$  stability. *Control and Intelligent Systems*, 39(1):1–9, 2011.
- [46] A. Birouche, B. Mourillon, and M. Basset. Model order-reduction for discrete-time switched linear systems. *Int. J. Systems Science*, 43(9):1753–1763, 2012.
- [47] F. Blanchini, D. Casagrande, S. Miani, and U. Viaro. Stable LPV realization of parametric transfer functions and its application to gain-scheduling control design. *IEEE Transactions on Automatic Control*, 55(10):2271–2281, 2010.
- [48] S. Bolognani, N. Bof, D. Michelotti, R. Muraro, and L. Schenato. Identification of power distribution network topology via voltage correlation analysis. In *52nd IEEE Conference on Decision and Control*, pages 1659–1664, 2013.
- [49] X. Bombois, D. Ghosh, G. Scorletti, and J. Huillery. Lpv system identification for control using the local approach. *International Journal of Control*, 94(2):390–410, 2021.
- [50] Xavier Bombois, K Colin, P M J Van Den Hof, and H Hjalmarsson. On the informativity of direct identification experiments in dynamical networks. working paper or preprint, August 2021.
- [51] K. Boukharouba, L. Bako, and S. Lecoeuche. Identification of piecewise affine systems based on Dempster-Shafer theory. In *IFAC Symposium on System Identification, Saint Malo, France*, 2009.

- 
- [52] J. A. Boyan and T. A. W. Moore. Generalization in reinforcement learning: Safely approximating the value function. In *Advances in neural information processing systems*, pages 369–376, 1995.
- [53] P. E. Caines. Weak and strong feedback free processes. *IEEE Transactions on Automatic Control*, 21(5):737–739, 1976.
- [54] P. E. Caines and C. Chan. Feedback between stationary stochastic processes. *IEEE Transactions on Automatic Control*, 20(4):498–508, 1975.
- [55] P. E. Caines, R. Deardon, and H. P. Wynn. Conditional orthogonality and conditional stochastic realization. In Anders Rantzer and Christopher I. Byrnes, editors, *Directions in Mathematical Systems Theory and Optimization*, volume 286, pages 71–84. Springer Berlin Heidelberg, 2003.
- [56] P. E. Caines, R. Deardon, and H. P. Wynn. Bayes nets of time series: Stochastic realizations and projections. In Luc Pronzato and Anatoly Zhigljavsky, editors, *Optimal Design and Related Areas in Optimization and Statistics*, volume 28 of *Springer Optimization and Its Applications*, pages 155–166. Springer New York, 2009.
- [57] P.E. Caines. *Linear Stochastic Systems*. John Wiley and Sons, New-York, 1998.
- [58] F. Celle, J. Gauthier, and E. Milani. Existence of realizations of nonlinear analytic input-output maps. *IEEE Transactions on Automatic Control*, 31(4):378–381, 1986.
- [59] F. Celle and J.P. Gauthier. Realizations of nonlinear analytic input-output maps. *Math. Systems Theory*, 19:227–237, 1986.
- [60] F. Celle and J.P. Gauthier. Realizations of nonlinear analytic input-output maps. *Math. Systems Theory*, 19:227 – 237, 1987.
- [61] V. Cerone, D. Piga, and D. Regruto. Set-membership LPV model identification of vehicle lateral dynamics. *Automatica*, 47(8):1794–1799, 2011.
- [62] H. Chen and J.M. Maciejowski. New subspace identification method for bilinea systems. Technical Report CUED/FINFENG/TR.337, Department of Engineering, University of Cambridge, 2000.
- [63] G. Conte, C. H. Moog, and A. M. Perdon. *Algebraic Methods for Nonlinear Control Systems*. Springer-Verlag, 2007.
- [64] O.L.V. Costa, M.D. Fragoso, and R.P. Marques. *Discrete-Time Markov Jump Linear Systems*. Springer Verlag, 2005.
- [65] P. Cox, R. Tóth, and M. Petreczky. Estimation of lpv-ss models with static dependency using correlation analysis. In *1st IFAC Workshop on Linear Parameter Varying systems -LPVS 2015*, 2015.
- [66] P. Cox, R. Tóth, and M. Petreczky. Towards efficient maximum likelihood estimation of lpv-ss models. *Automatica*, 2018.
- [67] Pepijn Bastiaan Cox and Roland Tóth. Linear parameter-varying subspace identification: A unified framework. *Automatica*, 123:109296, 2021.
- [68] A. G. Dankers. *System Identification in Dynamic Networks*. PhD thesis, Delft University of Technology, 2014.
- [69] A. G. Dankers, R. Toth, P. S C Heuberger, X. Bombois, and P. Van Den Hof. Informative data and identifiability in LPV-ARX prediction-error identification. In *Proceedings of the IEEE Conference on Decision and Control*, Orlando, Florida, USA, December 2011.

## BIBLIOGRAPHY

---

- [70] O. David. fMRI connectivity, meaning and empiricism: Comments on: Roebroek et al. the identification of interacting networks in the brain using fMRI: Model selection, causality and deconvolution. *NeuroImage*, 58(2):306 – 309, 2011.
- [71] B. De Schutter. Minimal state-space realization in linear system theory: an overview. *Journal of Computational and Applied Mathematics*, 121(1 -2):331–354, 2000.
- [72] T. Defourneau and M. Petreczky. Realization theory of recurrent neural networks and rational systems. In *58th IEEE Conference on Decision and Control*, 2019.
- [73] U.B. Desai. Realization of bilinear stochastic systems. *IEEE Transactions on Automatic Control*, 31(2), 1986.
- [74] M. Dimovska and D. Materassi. Granger-causality meets causal inference in graphical models: Learning networks via non-invasive observations. In *IEEE 56th Annual Conference on Decision and Control (CDC)*, pages 5268–5273, 2017.
- [75] P. L. dos Santos, J. A. Ramos, and J. L. M. de Carvalho. Identification of LPV systems using successive approximations. In *Proc. of 47th IEEE Conference on Decision and Control*, pages 4509–4515, 2008.
- [76] P.L. dos Santos, J.A. Ramos, and J.L.M. de Carvalho. Identification of bilinear systems with white noise inputs: An iterative deterministic-stochastic subspace approach. *Control Systems Technology, IEEE Transactions on*, 17(5):1145–1153, Sept 2009.
- [77] M. Eichler. A graphical approach for evaluating effective connectivity in neural systems. *Philosophical Transactions B*, 360:953–967, 2005.
- [78] M. Eichler. Granger causality and path diagrams for multivariate time series. *Journal of Econometrics*, 137(2):334 – 353, 2007.
- [79] M. Eichler. Graphical modelling of multivariate time series. *Probability Theory and Related Fields*, 153(1):233–268, 2012.
- [80] S. Eilenberg. *Automata, Languages and Machines*. Academic Press, New York, London, 1974.
- [81] E. Elhamifar, M. Petreczky, and R. Vidal. Rank test for observability of discrete-time jump linear systems with inputs. In *American Control Conference*, 2009.
- [82] D. Eringis, J. Leth, Zh.-H. Tan, R. Wisniewski, A. Fakhrizadeh Esfahani, and Petreczky M. Pac-bayesian theory for stochastic lti systems. In *60th IEEE Conference on Decision and Control*, 2021.
- [83] L. Etienne, L. Hetel, D. Efimov, and M. Petreczky. Observer analysis and synthesis for lipschitz nonlinear systems under discrete time-varying measurements. In *IFAC World Congress*, 2017.
- [84] L. Etienne, L. Hetel, D. Efimov, and M. Petreczky. Observer synthesis under time-varying sampling for lipschitz nonlinear systems. *Automatica*, 85:433–440, 2017.
- [85] P. Faurre, M. Clerget, and F. Germain. *Opérateurs rationnels positifs: application à l’hyperstabilité et aux processus aléatoires*. Méthodes mathématiques de l’informatique. Dunod, 1979.
- [86] M. Favoreel. *Subspace methods for identification and control of linear and bilinear systems*. PhD thesis, K.U. Leuven, 1999.
- [87] W. Favoreel, B. De Moor, and P. Van Overschee. Subspace identification of bilinear systems subject to white inputs. *IEEE Transactions on Automatic Control*, 44(6):1157–1165, 1999.

- 
- [88] F. Felici, J. W. van Wingerden, and M. Verhaegen. Subspace identification of MIMO LPV systems using periodic scheduling sequence. *Automatica*, 43:1684–1697, 2007.
- [89] G. Ferrari-Trecate, M. Musellu, D. Liberati, and M. Morari. A clustering technique for the identification of piecewise-affine systems. *Automatica*, 39:205–217, 2003.
- [90] M. Fliess. Realizations of nonlinear systems and abstract transitive lie-algebras. *Bulletin of Americal Mathematical Society*, 2(3), 1980.
- [91] M. Fliess. Fonctionnelles causales non linéaires et indéterminées non commutatives. *Bull. Soc. Math. France*, (109):2 – 40, 1981.
- [92] A. E. Frazho. On stochastic bilinear systems. In *Modelling and application of stochastic processes*, pages 215–241. Kluwer Acad. Publ., Boston, MA, 1986.
- [93] K. J. Friston, L. Harrison, and W. Penny. Dynamic causal modeling. *Neuroimage*, 19(4):1273 – 1302, 2003.
- [94] H. Gao, J. Lam, and C. Wang. Model simplification for switched hybrid systems. *Systems and Control Letters*, 55:1015–1021, 2006.
- [95] H. Garnier and L. Wang. *Identification of Continuous-time Models from Sampled Data*. Advances in Industrial Control. Springer London, 2008.
- [96] J.P Gauthier and G. Bornard. Existence and uniqueness of minimal realizations in the  $c^\infty$  case. *Systems and Control Letters*, 1:395–398, 1982.
- [97] F. Gécseg and I Peák. *Algebraic theory of automata*. Akadémiai Kiadó, Budapest, 1972.
- [98] M. R. Gevers and B. Anderson. On jointly stationary feedback-free stochastic processes. *IEEE Transactions on Automatic Control*, 27(2):431–436, 1982.
- [99] D. Ghosh, X. Bombois, J. Huillery, G. Scorletti, and G. Mercère. Optimal identification experiment design for LPV systems using the local approach. *Automatica*, 87:258 – 266, 2018.
- [100] G. J. Girard, A. and Pappas. Approximate bisimulation: A bridge between computer science and control theory. *European Journal of Control*, 17(5–6):568–578, 2011.
- [101] K. Glover and J. C. Willems. Parametrizations of linear dynamical systems: canonical forms and identifiability. *IEEE Transactions on Automatic Control*, 19:640–646, 1974.
- [102] R. Goebel, A. Roebroek, D.-S. Kim, and E. Formisano. Investigating directed cortical interactions in time-resolved fMRI data using vector autoregressive modeling and Granger causality mapping. *Magnetic Resonance Imaging*, 21:1251–1261, 2003.
- [103] R. Goebel, R. G. Sanfelice, and A. R. Teel. *Hybrid Dynamical Systems: Modeling, Stability, and Robustness*. Princeton University Press, New Jersey, NJ, 2012.
- [104] J. Gonçalves, R. Howes, and S. Warnick. Dynamical structure functions for the reverse engineering of lti networks. In *46th IEEE Conference on Decision and Control*, pages 1516–1522, 2007.
- [105] I. V. Gosea, M. Petreczky, and A. C. Antoulas. Data-driven model order reduction of linear switched systems in the loewner framework. *SIAM Journal on Scientific Computing, Society for Industrial and Applied Mathematics*, 40(2).

## BIBLIOGRAPHY

---

- [106] I. V. Gosea, M. Petreczky, and A. C. Antoulas. Balanced truncation for linear switched systems. *Advances in Computational Mathematics*, 2018.
- [107] I. V. Gosea, M. Petreczky, and A. C. Antoulas. Reduced-order modeling of lpv systems in the loewner framework. In *60th IEEE Conference on Decision and Control*, 2021.
- [108] I. V. Gosea, M. Petreczky, J. Leth, R. Wisniewski, and A.C. Antoulas. Model reduction of linear hybrid systems. In *59th IEEE Conference on Decision and Control*, 2020.
- [109] I. V. Gosea, I. Pontes Duff, P. Benner, and A. C. Antoulas. *Model Order Reduction of Switched Linear Systems with Constrained Switching*, volume 36 of *IUTAM Symposium on Model Order Reduction of Coupled Systems, Stuttgart, Germany, May 22–25, 2018. IUTAM Bookseries*, pages 41 – 53. Springer, Cham, 2020.
- [110] C. W. J. Granger. Economic processes involving feedback. *Information and Control*, 6(1):28–48, 1963.
- [111] R.L. Grossman and R.G. Larson. An algebraic approach to hybrid systems. *Theoretical Computer Science*, 138:101–112, 1995.
- [112] B. Guedj. A Primer on PAC-Bayesian Learning. *arXiv preprint arXiv:1901.05353*, 2019.
- [113] B. Guepie, M. Petreczky, and S. Lecoeuque. Eddy current signatures classification by using time series: a system modeling approach. In *Annual Conference of the Prognostics and Health Management Society*, 2014.
- [114] B. Hanzon. *Identifiability, recursive identification and spaces of linear dynamical systems*. Stichting Mathematische Centrum, Amsterdam, 1989.
- [115] B. Hanzon. On the differentiable manifold of fixed order stable linear systems. *Systems and Control Letters*, 13:345 – 352, 1989.
- [116] M. Hardt, T. Ma, and B. Recht. Gradient Descent Learns Linear Dynamical Systems. *ArXiv e-prints*, September 2016.
- [117] M. Havlicek, A. Roebroeck, K. Friston, A. Gardumi, D. Ivanov, and K. Uludag. Physiologically informed dynamic causal modeling of fMRI data. *Neuroimage*, 122:355–372, 2015.
- [118] Joo P Hespanha, Payam Naghshtabrizi, and Yonggang Xu. A survey of recent results in networked control systems. *Proceedings of the IEEE*, 95(1):138–162, 2007.
- [119] Laurentiu Hetel, Christophe Fiter, Hassan Omran, Alexandre Seuret, Emilia Fridman, Jean-Pierre Richard, and Silviu-Iulian Niculescu. Recent developments on the stability of systems with aperiodic sampling: An overview. *Autom.*, 76:309–335, 2017.
- [120] K. Horváth, E. Duviella, L. Rajaoarisoa, M. Petreczky, and K. Chuquet. Model predictive control of water levels in a navigation canal affected by resonance waves. In *11th International Conference on Hydroinformatics*, 2014.
- [121] K. Horváth, L. Rajaoarisoa, E. Duviella, J. Blesa, M. Petreczky, and K. Chuquet. Enhancing inland navigation by model predictive control of water level – the cuinchy-fontinettes case. In *Transport of Water versus Transport over Water - Exploring the dynamic interplay between transport and water*, volume 58 of *Operations Research/Computer Science Interfaces Series*, 2015.
- [122] K. Horváth, L. Rajaoarisoa, E. Duviella, M. Petreczky, and K. Chuquet. Mpc control of water level in a navigation canal - the cuinchy-fontinettes case study. In *European Control Conference*, 2014.

- 
- [123] R. Howes, L. Eccleston, J. Gonçalves, G.-B. Stan, and S. Warnick. Dynamical structure analysis of sparsity and minimality heuristics for reconstruction of biochemical networks. In *47th IEEE Conference on Decision and Control*, 2008.
- [124] C. Hsiao. Autoregressive modelling and causal ordering of econometric variables. *Journal of Economic Dynamics and Control*, 4:243–259, 1982.
- [125] A. Isidori. Direct construction of minimal bilinear realizations from nonlinear input-output maps. *IEEE Transactions on Automatic Control*, pages 626–631, 1973.
- [126] A. Isidori. *Nonlinear Control Systems*. Springer Verlag, 1989.
- [127] A. Isidori, P. D’Alessandro, and A. Ruberti. Realization and structure theory of bilinear dynamical systems. *SIAM J. Control*, 12(3), 1974.
- [128] B. Jacob and H. Zwart. A review on realization theory for infinite-dimensional systems. Technical report, 2017. ArXiv 1710.08653.
- [129] Gérard Jacob and N. Oussous. *Local and Minimal Realization of Nonlinear Dynamical Systems and Lyndon Words*, pages 155–160. 12 1990.
- [130] B. Jakubczyk. Existence and uniqueness of realizations of nonlinear systems. *SIAM J. Control and Optimization*, 18(4):455 – 471, 1980.
- [131] B. Jakubczyk. Construction of formal and analytic realizations of nonlinear systems. In Isidori and Hinrichsen, editors, *Feedback Control of Linear and Nonlinear Systems*, volume 39 of *Lecture Notes in Control and Information Sciences*, pages 147 – 156, 1982.
- [132] B. Jakubczyk. Realization theory for nonlinear systems, three approaches. In *Algeb. & Geom. Methods in Nonlin. Contr. Theory*, 1986.
- [133] M. Jozsa. *Relationship between Granger non-causality and network graphs of state-space representations*. PhD thesis, University of Groningen, 2019.
- [134] M. Jozsa, M. Petreczky, and M. K. Camlibel. Causality based graph structure of stochastic linear state-space representations. In *56th IEEE Conference on Decision and Control*, pages 2442–2447, 2017.
- [135] M. Jozsa, M. Petreczky, and M. K. Camlibel. Relationship between causality of stochastic processes and zero blocks of their joint innovation transfer matrices. In *20th World Congress of the International Federation of Automatic Control*, pages 4954–4959, 2017.
- [136] M. Jozsa, M. Petreczky, and M. K. Camlibel. Relationship between Granger non-causality and network graph of state-space representations. *IEEE Transactions on Automatic Control*, 64(8):912–127, 2019.
- [137] M. Jozsa, M. Petreczky, and M. K. Camlibel. Causality and network graph in general bilinear state-space representations. *IEEE Transactions on Automatic Control*, 65(8):5623 – 2630, 2020.
- [138] A. A. Julius, M. Zavlanos, S. Boyd, and G. J. Pappas. Genetic network identification using convex programming. *Systems Biology, IET*, 3:155–166, 2009.
- [139] L. Kailasam, R. Ortner, and D. Ryabko. Improved regret bounds for undiscounted continuous reinforcement learning. In *ICML*, volume 37 of *JMLR W&CP*, pages 524–532, Lille, France, 2015.
- [140] T. Kailath. *Linear Systems*. Prentice Hall, Engelwood Cliffs, 1980.

## BIBLIOGRAPHY

---

- [141] D.S. Kaliuzhnyi-Verbovetskyi and V. Vinnikov. *Foundations of Free Noncommutative Function Theory*. Mathematical Surveys and Monographs. American Mathematical Society, 2014.
- [142] R. E. Kalman. Advanced theory of linear systems. In *Topics in Mathematical System Theory*, pages 237–339. McGraw-Hill, New York, 1969.
- [143] R. E. Kalman. Introduction to the algebraic theory of linear dynamical systems. In *Mathematical systems theory and economics, I, II (Proc. Internat. Summer School, Varenna, 1967)*, pages 41–65. Lecture Notes in Oper. Res. and Math. Econom., Vols. 11, 12. Springer, Berlin, 1969.
- [144] R. E. Kalman, P. L. Falb, and M. A. Arbib. *Topics in mathematical system theory*. McGraw-Hill Book Co., New York, 1969.
- [145] W. Kamen, Edward and M. Hafez, Khaled. Algebraic theory of linear time-varying systems. *SIAM J. Control and Optimization*, 17(4), 1979.
- [146] T. Kang, R. Moore, Y. Li, E. D. Sontag, and L. Bleris. Discriminating direct and indirect connectivities in biological networks. *National Academy of Sciences USA*, 112:12893–12898, 2015.
- [147] T. Katayama. *Subspace Methods for System Identification*. Springer-Verlag, 2005.
- [148] P. L. Kempker. *Coordination Control of Linear Systems*. PhD thesis, Amsterdam: Vrije Universiteit, 2012.
- [149] P. L. Kempker, A. C. M. Ran, and J. H. van Schuppen. Construction and minimality of coordinated linear systems. *Linear Algebra and its Applications*, 452:202–236, 2014.
- [150] A. Khalate, X. Bombois, R. Tóth, and R. Babuska. Optimal experimental design for LPV identification using a local approach. In *Proceedings of the IFAC Symposium on System Identification*, 2009.
- [151] Gerhard Kramer. *Directed information for channels with feedback*. PhD thesis, Swiss Federal Institute of Technology Zürich, 1998.
- [152] W. Kuich and A. Salomaa. *Semirings, Automata, Languages*. Springer-Verlag, 1986.
- [153] B. Kulcsár and R. Tóth. On the similarity state transformation for linear parameter-varying systems. In *Proceedings of the IFAC World Congress*, pages 4155 – 4160, 2011.
- [154] F. Lauer and G. Bloch. *Hybrid System Identification: Theory and Algorithms for Learning Switching Models*, volume 478 of *Lecture Notes in Control and Information Sciences*. Springer International Publishing, 2019.
- [155] 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.
- [156] V. Laurain, R. Tóth, W-X. Zheng, and M. Gilson. Nonparametric identification of LPV models under general noise conditions: an ls-svm based approach. pages 1761–1766.
- [157] Edward A. Lee. The past, present and future of cyber-physical systems: A focus on models. *Sensors*, 15(3):4837–4869, 2015.
- [158] L. Lee and K. Poolla. Identifiability issues for parameter varying and multidimensional linear systems. In *Proceedings of the ASME Design Engineering Technical Conference*, Sacramento, California, USA, September 1997.
- [159] L. Lee and K. Poolla. Identification of linear parameter varying systems using non linear programming. *Journal of Dynamic Systems, Measurements and Control*, 121(1), 1999.



- 
- [160] L. Lessard, B. Recht, and A. Packard. Analysis and Design of Optimization Algorithms via Integral Quadratic Constraints. *ArXiv e-prints*, August 2014.
- [161] D. Liberzon. *Switching in Systems and Control*. Birkhäuser, Boston, 2003.
- [162] A. Lindquist and G. Picci. On the stochastic realization problem. *SIAM Journal on Control and Optimization*, 17(3):365–389, 1979.
- [163] A. Lindquist and G. Picci. *Linear Stochastic Systems: A Geometric Approach to Modeling, Estimation and Identification*. Springer Berlin, 2015.
- [164] L. Ljung. *System Identification: Theory for the user (2nd Ed.)*. PTR Prentice Hall., Upper Saddle River, USA, 1999.
- [165] P. Lopes dos Santos, T. Azevedo Perdicolis, C. Novara, J. Ramos, and D. Rivera. *Linear Parameter-Varying System Identification: new developments and trends*. Advanced Series in Electrical and Computer Engineering, World Scientific, 2011.
- [166] P. Lopes dos Santos, J. A. Ramos, and J. L. M. de Carvalho. Subspace Identification of Linear Parameter Varying Systems with Innovation-Type Noise Models. In *Proc. of the 7th Portuguese Conference on Automatic Control*, Lisbon, Portugal, 2006.
- [167] Y. Ma and R. Vidal. A closed form solution to the identification of hybrid ARX models via the identification of algebraic varieties. In *Hybrid Systems: Computation and Control*, pages 449–465, 2005.
- [168] Yi Ma and R. Vidal. Identification of deterministic switched arx systems via identification of algebraic varieties. In *Hybrid Systems: Computation and Control*, volume 3414 of *Lecture Notes in Computer Science*, pages 449 – 465, 2005.
- [169] I. Markovsky, J. C. Willems, A. van Huffel, and B. De Moor. *Exact and Approximate Modeling of Linear Systems: A Behavioral Approach (Mathematical Modeling and Computation) (Mathematical Modeling and Computation)*. Society for Industrial and Applied Mathematics, USA, 2006.
- [170] J. L. Massey. Causality, feedback and directed information. In *International Symposium on Information Theory and its Applications*, pages 27–30, 1990.
- [171] T. McKelvey, A. Helmersson, and T. Ribarits. Data driven local coordinates for multivariable linear systems and their application to system identification. *Automatica*, 40:1629–1635, 2004.
- [172] M. Mejari and M. Petreczky. Consistent and computationally efficient estimation for stochastic LPV state-space models: realization based approach. In *58th IEEE Conference on Decision and Control (CDC)*, Nice, France, 2019.
- [173] M. Mejari and M. Petreczky. Realization and identification algorithm for stochastic LPV state-space models with exogenous inputs. In *3rd IFAC Workshop on Linear Parameter-Varying Systems*, Eindhoven, Netherlands, 2019.
- [174] M. Mejari and M. Petreczky. Realization and identification algorithm for stochastic LPV state-space models with exogenous inputs. Technical report, arXiv:1905.10113, 2019.
- [175] 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.
- [176] M. Milanese, J. Norton, E. Walter, H. Piet-Lahanier, and É. Walter. *Bounding Approaches to System Identification*. Springer US, 1996.

## BIBLIOGRAPHY

---

- [177] J. Mohammadpour and C. Scherer. *Control of Linear Parameter Varying Systems with Applications*. Springer, New York, 2012.
- [178] J. Mohammadpour and C. W. Scherer. *Control of Linear Parameter Varying Systems with Applications*. Springer, Heidelberg, 2012.
- [179] N. Monshizadeh, H L. Trentelman, and M K. Camlibel. A simultaneous balanced truncation approach to model reduction of switched linear systems. *IEEE Transactions on Automatic Control*, 57(12):3118–3131, 2012.
- [180] E. Mosca. *Optimal, Predictive and Adaptive Control*. Prentice Hall, 1994.
- [181] J. Nemcova, E. Pawluszewicz, and M. Petreczky. Unified framework for continuous and discrete time nash realizations (extended abstract). In *Symposium on Mathematical Theory of Networks and Systems*, 2014.
- [182] J. Nemcova and M. Petreczky. Local nash realizations. In *18th International Conference on Methods and Models in Automation and Robotics, Miedzyzdroje, Poland*, 2013.
- [183] J. Nemcova, M. Petreczky, and J. H. van Schuppen. Realization theory of nash systems. In *48th IEEE Conference on Decision and Control (CDC)*, 2009.
- [184] J. Nemcova, M. Petreczky, and J. H. van Schuppen. System reduction and system identification of rational systems. In *16th IFAC Symposium on System Identification (SYSID)*, 2012.
- [185] J. Nemcova, M. Petreczky, and J. H. van Schuppen. Realization theory of nash systems. *SIAM J. Control and Optimization*, 51:3386–3414, 2013.
- [186] J. Nemcova, M. Petreczky, and J.H. van Schuppen. An algorithm for system identification of a discrete-time polynomial system without inputs. In *17th IFAC Symposium on System Identification*, 2015.
- [187] J. Nemcova, M. Petreczky, and J.H. van Schuppen. Observability reduction algorithm for rational systems. In *55th IEEE Conference on Decision and Control*, 2016.
- [188] J. Nemcova, M. Petreczky, and J.H. van Schuppen. Rational observers of rational systems. In *55th IEEE Conference on Decision and Control*, 2016.
- [189] J. Nemcova and J.H. van Schuppen. Realization theory for rational systems: The existence of rational realizations. *SIAM J. Control Optim.*, 48(4):2840–2856, 2009.
- [190] J. Nemcova and J.H. van Schuppen. Realization theory for rational systems: Minimal rational realizations. *Acta Applicandae Mathematicae*, 110:605–626, 2010.
- [191] T. E. M. Nordling and E. W. Jacobsen. On sparsity as a criterion in reconstructing biochemical networks. In *18th IFAC World Congress*, 2011.
- [192] C. Novara. *Set membership identification of state-space LPV systems*, pages 65–93.
- [193] H. Omran, L. Hetel, M. Petreczky, J.-P. Richard, and F. Lamnabhi-Lagarigue. Dissipativity-based approach for stability analysis of nonlinear sampled-data systems with input-affine control. *Automatica*, 70:266 – 274, 2016.
- [194] N. Oussous and Michel Petitot. *Scratchpad implementation of the local minimal realization of dynamic systems*, volume 165, pages 60–75. 01 2006.
- [195] Nour Oussous. Macsyma computation of local minimal realization of dynamical systems of which generating power series are finite. *Journal of Symbolic Computation*, 12:115–126, 07 1991.

- 
- [196] N. Pambakian. LQG coordination control. Master's thesis, Delft University of Technology, 2011.
- [197] S. Paoletti, A. Juloski, G. Ferrari-Trecate, and R. Vidal. Identification of hybrid systems: A tutorial. *European Journal of Control*, 13(2-3):242 – 260, 2007.
- [198] A V. Papadopoulos and M. Prandini. Model reduction of switched affine systems. *Automatica*, 70:57–65, 2016.
- [199] R. L. M. Peeters. *System Identification Based on Riemannian Geometry: Theory and Algorithms*. PhD thesis, Free University, Amsterdam, 1994.
- [200] R. L.M. Peeters and Bernard Hanzon. Identifiability of homogeneous systems using the state isomorphism approach. *Automatica*, 41:513–529, 2005.
- [201] W.D. Penny, K.E. Stephan, A. Mechelli, and K.J. Friston. Comparing dynamic causal models. *NeuroImage*, 22(3):1157 – 1172, 2004.
- [202] M. Petreczky. *Realization Theory of Hybrid Systems*. PhD thesis, Vrije Universiteit, Amsterdam, 2006.
- [203] M. Petreczky. Realization theory of linear and bilinear switched systems: A formal power series approach: Part i. *ESAIM Control, Optimization and Calculus of Variations*, 17:410–445, 2011.
- [204] M. Petreczky. Realization theory of linear and bilinear switched systems: A formal power series approach: Part ii. *ESAIM Control, Optimization and Calculus of Variations*, 17:446–471, 2011.
- [205] M. Petreczky. Realization theory of linear hybrid systems. In *Hybrid Dynamical Systems: Observation and control*, Lecture Notes in Control and Information Sciences. Springer-Verlag, 2015.
- [206] M. Petreczky and L. Bako. On the notion of persistence of excitation for linear switched systems. In *50th IEEE Conference on Decision and Control (CDC)*, 2011.
- [207] M. Petreczky, L. Bako, and S. Lecouche. Minimality and identifiability of sarx systems. In *16th IFAC Symposium on System Identification (SYSID)*, Brussel, 2012.
- [208] M. Petreczky, L. Bako, S. Lecouche, and K.M.D. Motchon. Minimality and identifiability of sarx systems. *International Journal of Nonlinear and Robust Control*, 30:2623–2630, 2020.
- [209] M. Petreczky, L. Bako, and van J.H. Schuppen. Realization theory for discrete-time linear switched systems. *Automatica*, 49(11):3337–3344, 2013.
- [210] M. Petreczky, L. Bako, and J.H. van Schuppen. Identifiability of discrete-time linear switched systems. In *Hybrid Systems: Computation and Control*, pages 141–150. ACM, 2010.
- [211] M. Petreczky, P. Collins, D.A. van Beek, J. H. van Schuppen, and J. E. Rooda. Sampled-data control of hybrid systems with discrete inputs and outputs. In *3rd IFAC Conference on Analysis and Design of Hybrid Systems (ADHS09)*, 2009.
- [212] M. Petreczky, J. Leth, R. Pedersen, and R. Wisniewski. Model reduction for linear switched systems with autonomous switching. In *IEEE American Control Conference (ACC)*, 2021.
- [213] M. Petreczky and G. Mercere. Affine LPV systems: Realization theory, input-output equations and relationship with linear switched systems. In *51th IEEE Conference on Decision and Control (CDC)*,, pages 4511–4516, 2012.
- [214] M. Petreczky and R. Peeters. Spaces of nonlinear and hybrid systems representable by recognizable formal power series. In *Proc. MTNS2010*, pages 1051–1058, 2010.

## BIBLIOGRAPHY

---

- [215] M. Petreczky, R. Theunissen, R. Su, D.A. van Beek, J.H. van Schuppen, and J.E. Rooda. Control of input/output discrete-event systems. In *European Control Conference*, 2009.
- [216] M. Petreczky, R. Tóth, and G. Mercére. Realization theory for LPV state-space representations with affine dependence. *IEEE Transactions on Automatic Control*, 62:4667–4674.
- [217] M. Petreczky and J. H. van Schuppen. Realization theory of discrete-time linear hybrid system. In *15th IFAC Symp. on System Identification*, 2009.
- [218] M. Petreczky and J. H. van Schuppen. Observability reduction of piecewise-affine hybrid systems. In *19th International Symposium on Mathematical Theory of Networks and Systems*, 2010.
- [219] M. Petreczky and J. H. Van Schuppen. Partial-realization of linear switched systems: A formal power series approach. *Automatica*, 47(10):2177–2184, 2011.
- [220] M. Petreczky and J.H. van Schuppen. Realization theory for linear hybrid systems. *IEEE Trans. on Automatic Control*, 55:2282 – 2297, 2010.
- [221] M. Petreczky and J.H. van Schuppen. Span-reachability and observability of bilinear hybrid systems. *Automatica*, 46:501–509, 2010.
- [222] M. Petreczky and R. Vidal. Metrics and topology for nonlinear and hybrid systems. In *Hybrid Systems: Computation and Control*, volume 4416 of *LNCS*. Springer-Verlag, 2007.
- [223] M. Petreczky and R. Vidal. Realization theory of stochastic jump-markov linear systems. In *Proc. 46th IEEE Conference on Decision and Control*, 2007.
- [224] M. Petreczky and R. Vidal. Realization theory of discrete-time semi-algebraic hybrid systems. In *Hybrid Systems: Computation and Control (HSCC)*, volume *LNCS 4981*, pages 386–400, 2008.
- [225] M. Petreczky and R. Vidal. Realization theory of stochastic jump-markov linear systems. Technical Report arXiv:1412.5020, 2014.
- [226] M. Petreczky and R. Vidal. Realization theory for a class of stochastic bilinear systems. *IEEE Transactions on Automatic Control*, 63(1):69–84, 2017.
- [227] M. Petreczky, R. Wisniewski, and J. Leth. Balanced truncation for linear switched systems. *Nonlinear Analysis: Hybrid Systems*, 10:4–20, November 2013.
- [228] M. Petreczky, R. Wisniewski, and J. Leth. Moment matching for bilinear systems with nice selections. In *10th IFAC Symposium on Nonlinear Control Systems*, 2016.
- [229] M. Petreczky and S. Zhuk. Solutions of differential-algebraic equations as outputs of lti systems: Application to lq control problems. *Automatica*, 84:166–173, 2017.
- [230] Mihaly Petreczky and Ion Victor Gosea. Model reduction and realization theory of linear switched systems. In *Realization and Model Reduction of Dynamical Systems: A Festschrift in Honor of the 70th Birthday of Thanos Antoulas*. Springer International Publishing, 2022.
- [231] 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.
- [232] D. Piga and R. Tóth. LPV model order selection in an LS-SVM setting. In *52nd IEEE Conference on Decision and Control*, 2013.

- 
- [233] G. Pola and M. D. Di Benedetto. Control of cyber-physical-systems with logic specifications: A formal methods approach. *Annual Reviews in Control*, 47:178–192, 2019.
- [234] I. Pontes Duff, S. Grundel, and P. Benner. New Gramians for switched linear systems: reachability, observability, and model reduction. June 2018.
- [235] G. Popescu. Positive-definite functions on free semigroups. *Can. J. Math.*, 48(4):887 – 896, 1996.
- [236] A.S. Poznyak, K. Najim, and E. Gomez-Ramirez. *Self-Learning Control of Finite Markov Chains*. Automation and Control Engineering. Taylor & Francis, 2000.
- [237] A. C. M. Ran and J. H. van Schuppen. Coordinated linear systems. In *Coordination Control of Distributed Systems*, volume 456 of *Lecture Notes in Control and Information Sciences*, pages 113–121, 2014.
- [238] Christophe Reutenauer. The local realization of generating series of finite lie-rank. In M Fliess and M Hazewinkel, editors, *Algebraic and Geometric Methods in Nonlinear Control Theory*, pages 33–43. D.Reidel Publishing Company, 1986.
- [239] S. Rizvi, J. Mohammadpour, F. Abbasi, and R. Tóth. State-space LPV model identification using kernelized machine learning. *Automatica*, 88:38–47, 2018.
- [240] Syed Zeeshan Rizvi, Javad Mohammadpour Velni, Farshid Abbasi, Roland Tóth, and Nader Meskin. State-space LPV model identification using kernelized machine learning. *Autom.*, 88:38–47, 2018.
- [241] A. Roebroeck, E. Formisano, and R. Goebel. Reply to friston and david: After comments on: The identification of interacting networks in the brain using fmri: Model selection, causality and deconvolution. *NeuroImage*, 58(2):310 – 311, 2011.
- [242] A. Roebroeck, A. K. Seth, and P. A. Valdes-Sosa. Causal time series analysis of functional magnetic resonance imaging data. *Journal of Machine Learning Research, Proceedings Track*, 12:65–94, 2011.
- [243] J. Rosenthal. Connections between linear systems and convolutional codes. In *Codes, Systems, and Graphical Models*, The IMA Volumes in Mathematics and its Applications. Springer, New York, NY, 2001.
- [244] W. Rugh and J. S. Shamma. Research on gain scheduling. *Automatica*, 36(10):1401–1425, 2000.
- [245] W. J. Rugh. *Linear System theory*. Prentice-Hall, 1996.
- [246] D. Ryabko. Discrimination between B-processes is impossible. *Journal of Theoretical Probability*, 23(2):565–575, 2010.
- [247] D. Ryabko. Testing composite hypotheses about discrete ergodic processes. *Test*, 21(2):317–329, 2012.
- [248] H. Sandberg. *Model reduction for linear time-varying systems*. PhD thesis, Department of Automatic Control, Lund Institute of Technology, Sweden., 2004.
- [249] N. I. Sapankevych and R. Sankar. Time series prediction using support vector machines: a survey. *IEEE Computational Intelligence Magazine*, 4(2), 2009.
- [250] S. Sastry and M. Bodson. *Adaptive Control: Stability, Convergence, and Robustness*. Prentice-Hall, 1994.
- [251] H.R. Shaker and R. Wisniewski. Generalized gramian framework for model/controller order reduction of switched systems. *International Journal of Systems Science*, 42(8):1277–1291, 2011.

## BIBLIOGRAPHY

---

- [252] H.R. Shaker and R. Wisniewski. Model reduction of switched systems based on switching generalized gramians. *International Journal of Innovative Computing, Information and Control*, 8(7(B)):5025–5044, 2012.
- [253] V. Shalaeva, A. Fakhrizadeh Esfahani, P. Germain, and M. Petreczky. Improved pac-bayesian bounds for linear regression. In *34th AAAI Conference on Artificial Intelligence*, 2020.
- [254] S. Shalev-Shwartz and S. Ben-David. *Understanding machine learning: From theory to algorithms*. Cambridge university press, 2014.
- [255] Victor Solo. State-space analysis of Granger-Geweke causality measures with application to fMRI. *Neural Computation*, 28:914–949, 2016.
- [256] E. D. Sontag. *Polynomial Response Maps*, volume 13 of *Lecture Notes in Control and Information Sciences*. Springer Verlag, 1979.
- [257] E. D. Sontag. Realization theory of discrete-time nonlinear systems: Part I – the bounded case. *IEEE Trans. on Circuits and Systems*, CAS-26(4), 1979.
- [258] E.D. Sontag. Linear systems over commutative rings: A survey. *Richerche di Automatica*, 7(1), 1976.
- [259] E.D. Sontag. Recurrent neural networks: Some systems-theoretic aspects. In M. Karny, K. Warwick, and V. Kurkova, editors, *Dealing with Complexity: a Neural Network Approach*, pages 1–12, 1997.
- [260] Zh. Sun and Sh. S. Ge. *Switched linear systems : control and design*. Springer, London, 2005.
- [261] H. Sussman. Existence and uniqueness of minimal realizations of nonlinear systems. *Mathematical Systems Theory*, 10:263–284, 1977.
- [262] R. S. Sutton and A. G. Barto. *Reinforcement learning: An introduction*, volume 1. MIT press Cambridge, 1998.
- [263] R. S. Sutton, H. R. Maei, and Cs. Szepesvári. A convergent  $o(n)$  temporal-difference algorithm for off-policy learning with linear function approximation. In *Advances in neural information processing systems*, pages 1609–1616, 2009.
- [264] J.A.K. Suykens, T. Van Gestel, J. De Brabanter, B. De Moor, and J. Vandewalle. Least squares support vector machines, 2002.
- [265] M. Sznaier and C. Mazzaro. An LMI approach to the identification and (in)validation of LPV systems. In S.O.R. Moheimani, editor, *Perspectives in robust control. Lecture Notes in Control and Information Sciences*, volume 268, pages 327–346. Springer, London, 2001.
- [266] P. Tabuada. *Verification and Control of Hybrid Systems: A Symbolic Approach*. Springer-Verlag, 2009.
- [267] P. Tabuada, A. D. Ames, A. Julius, and G. J. Pappas. Approximate reduction of dynamic systems. *Systems and Control Letters*, 57(7):538–545, 2008.
- [268] M. Tanelli, D. Adagna, and M. Lovera. Identification of LPV state-space models for automatic web service systems. *IEEE Transactions on Controls Systems Technology*, 19:93–103, 2011.
- [269] D.C. Tarraf. *Control of Cyber-Physical Systems*, volume 449 of *Lecture Notes in Control and Information Sciences*. Springer, Heidelberg, 2013.
- [270] Gy. Terdik. Bilinear state space realization for polynomial stochastic systems. *Comput. Math. Appl.*, 22(7):69–83, 1991.

- 
- [271] Gy. Terdik. Bilinear stochastic models and related problems of nonlinear time series analysis. In *Lecture Notes in Statistics*, number 142. Springer, New-York, 1999.
- [272] R.J.M. Theunissen, M. Petreczky, R.R.H. Schiffelers, D.A. van Beek, and J.E. Rooda. Application of supervisory control synthesis to a patient support table of a magnetic resonance imaging scanner. *IEEE Transaction on Automation Science and Engineering*, 11:20–32, 2014.
- [273] R. Tóth. *Identification and Modeling of Linear Parameter-Varying Systems*. Springer Verlag. Lecture Notes in Control and Information Sciences 403, 2010.
- [274] R. Tóth. Modeling and identification of linear parameter-varying systems. In *Lecture Notes in Control and Information Sciences, Vol. 403*, Heidelberg, 2010. Springer.
- [275] R. Tóth, H. Abbas, and H. Werner. On the state-space realization of LPV input-output models: practical approaches. *IEEE Transactions on Control System Technology*, 20:139–153, 2012.
- [276] R. Tóth, J. C. Willems, P. S. C. Heuberger, and P. M. J. Van den Hof. The behavioral approach to linear parameter-varying systems. *IEEE Transactions on Automatic Control*, 56(7):2499–2514, 2011.
- [277] S. Vajda, K. R. Godfrey, and R. Hershcel. Similarity transformation approach to identifiability analysis of nonlinear compartmental models. *Mathematical Biosciences*, 93:217–248, 1989.
- [278] P. A. Valdes-Sosa, A. Roebroek, J. Daunizeau, and K. J. Friston. Effective connectivity: Influence, causality and biophysical modeling. *Neuroimage*, 58(2):339–361, 2011.
- [279] J. van den Hof. Structural identifiability of linear compartmental systems. *IEEE Transactions on Automatic Control*, 43:800–818, 1998.
- [280] P. M. J. Van den Hof, A. Dankers, P. S. C. Heuberger, and X. Bombois. Identification of dynamic models in complex networks with prediction error methods—basic methods for consistent module estimates. *Automatica*, 49(10):2994–3006, 2013.
- [281] A.J. van der Schaft and Nijmeijer H. *Nonlinear Dynamical Control Systems*. Springer-Verlag, 1990.
- [282] P. van Overschee and B. De Moor. *Subspace Identification for Linear Systems*. Kluwer Academic Publishers, 1996.
- [283] 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.
- [284] 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.
- [285] J. Vayssettes. and G. Mercère. A new parametrisation of matrix fraction descriptions to improve gradient-based optimisation methods. In *Proceedings of the Conference on Decision and Control*, Los Angeles, CA, USA, December 2014.
- [286] V. Verdult. *Nonlinear system identification: a state space approach*. PhD thesis, University of Twente, Twente, The Netherlands, 2002.
- [287] V. Verdult and M. Verhaegen. Identification of multivariable bilinear state space systems based on subspace techniques and separable least squares optimization. *International Journal of Control*, 74(18):1824–1836, 2001.
- [288] V. Verdult and M. Verhaegen. Subspace identification of multivariable linear parameter-varying systems. *Automatica*, 38(5):805–814, 2002.

## BIBLIOGRAPHY

---

- [289] V. Verdult and M. Verhaegen. Kernel methods for subspace identification of multivariable LPV and bilinear systems. *Automatica*, 41:1557–1565, 2005.
- [290] R. Vidal. Recursive identification of switched ARX systems. *Automatica*, 44(9):2274 – 2287, 2008.
- [291] D. Vizer and G. Mercère. An  $H_\infty$ -norm-based approach for operating point selection and LPV model identification from local experiments. *Periodica Polytechnica Electrical Engineering and Computer Science*, 58(3):121–131, 2014.
- [292] D. Vizer, G. Mercère, and O. Prot. A local approach framework for black-box and gray-box LPV system identification. In *2013 European Control Conference (ECC)*, Zurich, Switzerland, 2013.
- [293] D. Vizer, G. Mercère, O. Prot, E. Laroche, and M. Lovera. Linear fractional LPV model identification from local experiments: an  $H_\infty$ -based optimization technique. In *In IEEE Conference on Decision and Control*, Florence, Italy, December 2013.
- [294] E. Walter. Identifiability of state space models. In *Lecture Notes in Biomathematics*, volume 46. Springer-Verlag, 1982.
- [295] L.Y. Wang, G.G. Yin, J.F. Zhang, and Y. Zhao. *System Identification with Quantized Observations*. Systems & Control: Foundations & Applications. Birkhäuser Boston, 2010.
- [296] Y. Wang and E. Sontag. Algebraic differential equations and rational control systems. *SIAM Journal on Control and Optimization*, (30):1126–1149, 1992.
- [297] H. H. M. Weerts. *Identifiability and Identification Methods for Dynamic Networks*. PhD thesis, Eindhoven University of Technology, 2018.
- [298] H. H. M. Weerts, P. M. J. Van den Hof, and A. G. Dankers. Identifiability of linear dynamic networks. *Automatica*, 89:247–258, 2018.
- [299] R. Westra, M. Petreczky, and R. Peeters. Identification of piecewise linear models of complex dynamical systems. In *18th IFAC World Congress*, 2011.
- [300] J. C. Willems. Input-output and state-space representations of finite-dimensional linear time-invariant systems. *Linear algebra and its applications*, 50:581 – 608, 1983.
- [301] J. C. Willems. From time series to linear system—part i. finite dimensional linear time invariant systems. *Automatica*, 22(5):561 – 580, 1986.
- [302] J. C. Willems. From time series to linear system—part ii. exact modelling. *Automatica*, 22(6):675 – 694, 1986.
- [303] J. C. Willems. On interconnections, control, and feedback. *IEEE Transactions on Automatic Control*, 42(3):326 – 339, 1997.
- [304] J. C. Willems. The behavioral approach to open and interconnected systems. *IEEE Control Systems Magazine*, 27:46–99, 2007.
- [305] J.C. Willems. Dissipative dynamical systems part ii: Linear systems with quadratic supply rates. *Arch. Rational Mech. Anal.*, 45:352–393, 1972.
- [306] J.C. Willems and J.W. Polderman. *An Introduction to Mathematical Systems Theory: A Behavioral Approach*. Springer Verlag, New York, 1998.



- 
- [307] A. Wills and B. Ninness. On gradient-based search for multivariable system estimates. *IEEE Transactions on Automatic Control*, 53(1):298–306, 2008.
- [308] Adrian Wills and Brett Ninness. System identification of linear parameter varying state-space models. In *Linear parameter-varying system identification: new developments and trends*, pages 295–315. World Scientific, 2012.
- [309] G.D. Wood, P.J. Goddard, and K. Glover. Approximation of linear parameter-varying systems. In *Decision and Control, 1996., Proceedings of the 35th IEEE*, volume 1, pages 406–411 vol.1, dec 1996.
- [310] Y. Yuan, K. Glover, and J. Gonçalves. On minimal realisations of dynamical structure functions. *Automatica*, 55:159–164, 2015.
- [311] E. Zerz. Behavioral systems theory: A survey. *International Journal of Applied Mathematics and Computer Science*, 18(3):265 – 270, 2008.
- [312] L. Zhang, E. Boukas, and P. Shi.  $\mu$ -Dependent model reduction for uncertain discrete-time switched linear systems with average dwell time. *International Journal of Control*, 82(2):378–388, 2009.
- [313] L. Zhang, P. Shi, E. Boukas, and C. Wang. H-infinity model reduction for uncertain switched linear discrete-time systems. *Automatica*, 44(8):2944–2949, 2008.
- [314] Qinghua Zhang and Lennart Ljung. Lpv system common state basis estimation from independent local lti models. *IFAC-PapersOnLine*, 48(28):190–195, 2015. 17th IFAC Symposium on System Identification SYSID 2015.
- [315] W. Zhang, W. Liu, C. Zang, and L. Liu. Multi-agent system based integrated solution for topology identification and state estimation. *IEEE Transactions on Industrial Informatics*, 13(2):714–724, 2017.
- [316] L. Zheng-Fan, C. Chen-Xiao, and D. Wen-Yong. Stability analysis and  $H_\infty$  model reduction for switched discrete-time time-delay systems. *Mathematical Problems in Engineering*, 15, 2014.
- [317] S. Zhuk and M. Petreczky. Minimax observers for linear differential-algebraic equations. *IEEE Transactions on Automatic Control*, 62:4104–4108, 2017.