



**STATE-SPACE IDENTIFICATION OF SWITCHED
LINEAR SYSTEMS VIA SPARSE OPTIMIZATION**

Fethi BENCHERKI

MS Dissertation

Graduate School of Science
Electrical and Electronic Engineering
July, 2020

ABSTRACT

STATE-SPACE IDENTIFICATION OF SWITCHED LINEAR SYSTEMS VIA SPARSE OPTIMIZATION

Fethi Bencherki

Department of Electrical and Electronic Engineering
Anadolu University, Graduate School of Science, July, 2020

Supervisor: Assist. Prof. Dr. Semiha TÜRKAY

In this thesis, we consider the identification of linear time-varying systems from input-output measurements. The system parameters are represented in state-space form, and change in a piecewise-constant manner, only at a set of time instants to a submodel from a fixed set. The change instants and the number of changes are unknown to us. Furthermore, the number of submodels and their parameters are also assumed unknown. This thesis brings forward a methodology to identify switch linear systems given in the state-space form from input-output measurements. We do not assume that the continuous state is measured as in other works in the literature. The key step in the proposed methodology is the observed-based transformation of the state-space switched linear system (SLS) models to the switched auto-regressive with exogenous input (SARX) models. This transformation, which reduces the state-space identification problem to the SARX model estimation, is carried out by dead-beat observers under a mild restriction on the minimum dwell time. The complications induced from this transformation are dealt with successfully. Sufficient, and in some cases necessary, conditions for the identifiability of the switches and the submodels along with the persistence of excitation (PE) conditions on inputs are presented. The switches and the submodels are identified in the observer domain starting with the submodels active over long constant parameter intervals by a non-convex sparse optimization algorithm from the input-output data. A clustering approach reveals all submodels under mild assumptions. Submodels that are active on intermediate to short segments are identified by a MOESP type subspace identification algorithm from the input-output data, or by solving a discrete optimization algorithm modified from MOESP. A convex relaxation of the sparse optimization algorithm leads to the basis pursuit denoising method which is well-known in the literature of compressive sensing.

Keywords: Hybrid system; switched linear system; state-space model; identification; compressive sensing; dead-beat observer; linear time-varying; realization; subspace identification.

ÖZET

Anahtarlama Doğrusal Sistemlerin Seyreklik Optimizasyonu ile Durum-Uzay Tanılaması

Fethi Bencherki

Elektrik ve Elektronik Mühendisliği Anabilim Dalı
Anadolu Üniversitesi, Fen Bilimleri Enstitüsü, Temmuz, 2020

Danışman: Dr. Öğr. Üyesi Semiha TÜRKAY

Bu tez çalışmasında, girdi-çıkıtlı ölçümleri kullanılarak doğrusal zaman değişkenli sistemlerin tanılanması ele alınmaktadır. Sistem parametreleri durum-uzay formunda ifade edilmiştir ve bu parametreler önceden belirlenmiş bir setin alt modellerine, parçalı-sabit bir şekilde sadece zaman anları setinde değişmektedir. Değişim anları ve değişim sayısı bilinmemektedir. Buna ilave olarak, alt model sayısı ve bu alt modellerin parametrelerinin de bilinmediği varsayılmaktadır. Bu çalışma, girdi-çıkıtlı ölçümleri kullanılarak anahtarlama doğrusal sistemlerin durum-uzay formunda tanılanmasını sağlayan bir yöntem bilimi ortaya koymaktadır. Literatürde yer alan diğer çalışmalarda bulunan sürekli-zaman ölçümlerinin mevcut olduğu varsayımı bu çalışmada yapılmamıştır. Önerilen bu metodolojideki anahtar adımı durum-uzay anahtarlama doğrusal sistem (ADS) modellerinin anahtarlama harici girdili otomatik (HGOY) modellere gözleyiciler ile yapılan dönüşümler oluşturmaktadır. Durum-uzay tanıyım problemini, HGOY model kestirimine indirgeyen bu dönüşümler en küçük bekleme süresi üzerine katı olmayan kısıtlamalar koyan dead-beat gözleyiciler ile yapılmaktadır. Bu dönüşümün yarattığı teorik ve pratik güçlükler bu çalışmada başarılı bir şekilde aşılmıştır. Anahtarların ve alt modellerin sürekli uyarılma girdi koşulları ile birlikte tanılanması için yeterli, ve bazı durumlarda gerekli koşullar da sunulmuştur. Anahtarlar ve alt modellerin tanılanması gözleyici kümesinde, konveks olmayan seyreklik optimizasyon algoritması girdi-çıkıtlı verileri ile kullanılarak uzun, sabit parametre aralıklarında aktif olan alt modellerin kestirilmesiyle başlamaktadır. Kümeleme yaklaşımı, ılımlı varsayımlar altında tüm alt modelleri ortaya çıkarmaktadır. Orta ve kısa zaman aralıklarında aktif olan alt modeller ise girdi-çıkıtlı veri seti kullanılarak MOESP tipi altuzay tanıyım algoritması kullanılarak veya bu algoritmadan uyarlanmış kesikli optimizasyon problemi çözülerek elde edilmiştir. Seyrek optimizasyon algoritmasının konveks olarak gevşetilmesi, sıkıştırılmalı algılama alanında iyi bilinen taban izlemli gürültü arındırma yöntemiyle yapılmıştır.

Anahtar kelimeler: Hibrit sistem; anahtarlama doğrusal sistem; durum-uzay modeli; tanılama, sıkıştırılmalı algılama; dead-beat gözleyicisi; doğrusal zamana bağlı; gerçekleştirme; alt uzay tanıyımı.

ACKNOWLEDGMENTS

I would like to express my utmost feelings of gratitude and appreciation to my thesis supervisor Assist. Prof. Dr. Semiha TÜRKEY for her endless support, encouragement, and guidance throughout this long journey, for whom i will forever feel indebted. I also would like to thank Prof. Dr. Hüseyin AKÇAY for the useful research discussions we have shared. I also want to thank my family and friends for their emotional support.

Fethi Bencherki

July, 2020

**STATEMENT OF COMPLIANCE WITH ETHICAL PRINCIPLES
AND RULES**

I hereby truthfully declare that this thesis is an original work prepared by me; that I have behaved in accordance with the scientific ethical principles and rules throughout the stages of preparation, data collection, analysis and presentation of my work; that I have cited the sources of all the data and information that could be obtained within the scope of this study, and included these sources in the references section; and that this study has been scanned for plagiarism with “scientific plagiarism detection program” used by Anadolu University, and that “it does not have any plagiarism” whatsoever. I also declare that, if a case contrary to my declaration is detected in my work at any time, I hereby express my consent to all the ethical and legal consequences that are involved.

Fethi Bencherki

CONTENTS

	<u>Page</u>
ABSTRACT	ii
ÖZET	iii
ACKNOWLEDGMENTS	iv
STATEMENT OF COMPLIANCE WITH ETHICAL PRINCIPLES	v
TABLE OF CONTENTS	vi
LIST OF TABLES	x
LIST OF FIGURES	xi
GLOSSARY OF SYMBOLS AND ABBREVIATIONS	xiii
1. INTRODUCTION	1
1.1. Literature review and motivation of the study	2
1.1.1. Offline identification	2
1.1.2. Online identification	5
1.2. Problem formulation	6
1.3. Organization of the thesis	7
2. BACKGROUND AND UNDERLYING THEORY	9
2.1. Hybrid system state-space models	9
2.2. Recovering a common basis for the state-space submodels .	10
2.3. Deadbeat observer	11
2.4. State-space representation and Markov parameters	12

2.5.	State transformation and similarity of state-space models .	13
2.6.	Relationship between state-space models and input-output models via observer Markov parameters	14
2.7.	The system Markov parameters	16
2.8.	Process and measurement noise	17
2.9.	The observer/Kalman filter identification (OKID)	19
2.10.	Eigensystem realization algorithm (ERA)	19
2.11.	Compressive sensing	21
2.12.	Greedy algorithms	22
2.12.1.	The Orthogonal matching pursuit (OMP)	22
2.12.2.	Compressive sampling matching pursuit (CoSaMP) .	23
2.12.3.	The Block orthogonal matching pursuit (BOMP) .	23
2.12.4.	Iterative hard thresholding (IHT)	24
2.13.	Basis pursuit denoising (BPDN)	25
2.14.	Iterative reweighting	26
2.15.	Recovery guarantees	27
2.15.1.	Spark	28
2.15.2.	Null space property (NSP)	29
2.15.3.	The restricted isometric property (RIP)	30
2.15.4.	The coherence	31
2.15.5.	Block-coherence and sub-coherence	32
2.15.6.	Recovery guarantees For BOMP	32
3.	LTV-MODEL TRANSFORMATIONS AND REALIZATION . .	34
3.1.	SLS realization from generalized Hankel matrix pairs	34
3.2.	Observer-based transformation of SLS models to SARX models	41
3.3.	Computation of the generalized system Markov parameters and the observer gains	45

3.4.	Effect of process and measurement noise on observer-based transformation	48
3.5.	Example	51
3.6.	Summary	53
4.	SLS MODELS ESTIMATION FROM LTV STATE-SPACE REALIZATIONS	55
4.1.	Submodel estimation	55
4.2.	Switching sequence estimation	57
4.3.	Example	61
4.4.	Summary	64
5.	SWITCH AND SUBMODEL IDENTIFIABILITY OF SARX MODELS FROM INPUT-OUTPUT DATA	66
5.1.	Switch identifiability for MIMO SARX models	66
5.2.	The relation between segments in SLS and segments in SARX	72
5.3.	Switch identifiability conditions & further remarks for SISO SARX models	75
5.4.	Persistence of excitation of the input sequence for MIMO SARX systems	76
5.5.	Persistence of Excitation for SISO SARX systems	81
5.6.	Example I	82
5.7.	Example II	84
5.8.	Summary	87
6.	COMPRESSIVE SYSTEM IDENTIFICATION FOR SISO SLS MODELS	89
6.1.	Submodel set and switches estimation	89
6.2.	Switching sequence estimation in SARX models	92
6.3.	Basis pursuit denoising recovery algorithm	96

6.3.1. Iterative refinement	99
6.4. Summary of the proposed algorithm for identifying SLSs .	101
6.5. Example	102
6.6. Summary	106
7. CONCLUSION AND AN OUTLOOK FOR FUTURE RESEARCH	
DIRECTIONS	108
7.1. Conclusions	108
7.2. Future Research	109
A. PROOFS	110
A.1. Proof of Lemma 3.1.1	110
A.2. Proof of Lemma 3.2.1	111
A.3. Proof of Lemma 5.1.1	111
A.4. Proof of Lemma 5.1.2	112
A.5. Proof of Lemma 5.1.3	113
A.6. Proof of Lemma 5.2.1	113
A.7. Proof of Lemma 5.2.2	113
A.8. Proof of Lemma 5.2.4	114
A.9. Proof of Lemma 5.2.6	115
B. THE DBSCAN CLUSTERING	116
REFERENCES	117

LIST OF TABLES

	<u>Page</u>
Table 2.1. The OMP algorithm.	22
Table 2.2. The CoSaMP algorithm.	23
Table 2.3. The BOMP algorithm.	24
Table 2.4. The IHT algorithm.	24
Table 6.1. Monte Carlo simulation results for the three submodels of the SLS model estimated by the proposed algorithm.	106

LIST OF FIGURES

	<u>Page</u>
Figure 3.1. The switching sequences in Subsection 3.5.	52
Figure 3.2. $\ K(k)\ _F$ for switching sequence in Subsection 3.5.	52
Figure 3.3. The ℓ_1 norm of the deadbeat observer eigenvalues.	53
Figure 3.4. $\ \Phi_o(k, k - \tau)\ _F$ as a function of k for $\tau = 2, \dots, 6$ and the first switching sequence.	53
Figure 4.1. The switching sequence.	61
Figure 4.2. The ℓ_1 -norm of the eigenvalues: 'o' the LTV realization; '—' the SLS model.	61
Figure 4.3. The histogram of the clusters from the dbscan algorithm. . . .	62
Figure 4.4. The division of points between correct clusters and outliers according to dbscan algorithm.	63
Figure 4.5. The ℓ_1 -norm of the eigenvalues: 'o' the re-clustered LTV real- ization; '—' the SLS model.	63
Figure 4.7. The scheduling sequence estimate after re-clustering.	63
Figure 4.6. The submodel and the averaged re-clustered eigenvalues: 'o' the re-clustered eigenvalues; '+' the submodel.	64
Figure 4.8. The $\mathcal{H}_2$ norms of the the time-varying transfer function matrix estimate $\hat{C}(k)(zI_n - \tilde{A}(k))^{-1}\tilde{B}(k) + \hat{D}(k)$ ('-') and the true transfer function matrix estimate $C(k)(zI_n - A(k))^{-1}B(k) +$ $D(k)$ 'o'.	64
Figure 5.1. $\ (\theta(k_{i+1}) - \theta(k_{i+1} - 1))z(k_{i+1})\ _2$ on χ	83
Figure 5.2. $\bar{\sigma}\{C_{u_a y_a}(s_j)\}$ versus s_j for $s_j \in \chi$	84
Figure 5.3. $\ \Sigma_{\mathcal{R}_{s_j, s_{j+1}-1}}\ _{\ell, 0}$ (left bars) and $\ \Sigma_{\mathcal{R}_{s_j, s_{j+1}}}\ _{\ell, 0}$ (right bars) on χ . .	84

Figure 5.4.	The four smallest singular values of $\mathcal{R}_{k_i+\tau, k_{i+1}-1}$	85
Figure 5.5.	$ z^T(s_{j+1})(\theta(s_{j+1}) - \theta(s_j)) $ on χ	85
Figure 5.6.	$\bar{\sigma}\{C_{u_a y_a}(s_j)\}$ versus s_j for $s_j \in \chi$	86
Figure 5.7.	$\ \Sigma_{\mathcal{R}_{s_j, s_{j+1}-1}}\ _{\ell, 0}$ (left bars) and $\ \Sigma_{\mathcal{R}_{s_j, s_{j+1}}}\ _{\ell, 0}$ (right bars).	86
Figure 5.8.	The five SVDs of $(\mathcal{R}_{s_j, s_{j+1}-1})$ for $\tau = n = 2$ in a noiseless setup.	86
Figure 5.9.	The last four SVDs of $\mathcal{R}_{k_i+\tau, k_{i+1}-1}$ for $\tau = 2$ and 3 in a noisy setup.	87
Figure 6.1.	Flow diagram for the proposed algorithm to identify switched linear system.	101
Figure 6.2.	the histogram of DBSCAN clustering from Algorithm 5 to retrieve sub-models at one noise realization with SNR=30db.	102
Figure 6.3.	The true and the estimated system eigenvalues: the true '+', the estimated 'o' at SNR levels 40db, 30db and 20db top to bottom respectively at 10 different noise realizations.	103
Figure 6.4.	The observer system eigenvalues at an SNR of 30db for 10 different noise realizations.	103
Figure 6.5.	The estimated switching function $\hat{\chi}$ (dotted) and the true one χ (solid) for SLS model.	104
Figure 6.6.	The size of the observation window in Figure 6.5 is shortened for visualization purpose.	105
Figure 6.7.	The estimated switching function $\hat{\chi}_\theta$ for SARX.	105
Figure 6.8.	The Deadbeat observer gains $g(k)$	105
Figure 6.9.	Validation data with an SNR of 30 dB, obtained VAF = 98.28%.	107
Figure B.1.	The different types of points considered by dbscan algorithm	117

GLOSSARY OF SYMBOLS AND ABBREVIATIONS

ARMAX	Auto-Regressive Moving-Average with Exogenous Inputs
BP	Basis Pursuit
BOMP	Block Orthogonal Matching Pursuit
BPDN	Basis Pursuit Denoising
CS	Compressive Sensing
COSAMP	Compressive Sampling Matching Pursuit
DBSCAN	Density-Based Spatial Clustering of Applications with Noise
EM	Expected Maximization
ERA	Eigensystem Realization Algorithm
ERC	Exact Recovery Conditions
GPCA	General Principal Component Analysis
GTV-ARX	Generalized Time-Varying Auto-Regressive with Exogenous Inputs
GTV-ARMAX	Generalized Time-Varying Auto-Regressive Moving-Average with Exogenous Inputs
I/O	Input/Output
IHT	Iterative Hard Thresholding
LASSO	Least Absolute Shrinkage and Selection Operator
LPV	Linear Parameter-Varying
LS	Least Squares
LTV	Linear Time-Varying
MIMO	Multiple-Input, Multiple-Output
MISO	Multiple-Input, Single-Output
MFD	Matrix-Fraction Description
MMV	Multiple Measurement Vectors
MOESP	Multivariable Output Error State Space
NP	Non-Deterministic Polynomial-Time

NSP	Null space property
OE	Output Error
OKID	Observer Kalman Filter Identification
OKID/DP	OKID with Deterministic Projection
O ³ KID	The Output-Only OKID
OMP	Orthogonal Matching Pursuit
PE	Persistence of Excitation
PO-MOESP	Past Outputs Multivariable Output-Error State-space
PWA	Piecewise Affine System
PSD	Power Spectral Density
ROKID	Residual-Based OKID
RIP	Restricted Isometric Property
SLS	Switched Linear System
SARX	Switched Auto-Regressive with Exogenous Inputs
SWA	switched Affine System
SISO	Single-Input, Single-Output
SS	State Space
SVD	Singular Value Decomposition
TVERA	Time-Varying Eigensystem Realization Algorithm

1. INTRODUCTION

Hybrid systems are those dynamical systems that merge both continuous and discrete behaviour [1]. Consequently, according to the discrete state, they switch between different continuous operating modes [2]. The discrete state can be taken to be generated externally, to switch to the desired continuous dynamics arbitrarily. As a second option, the discrete state depends on the set of continuous state and inputs; this gives rise to the piecewise affine model set (PWA). This later class of Hybrid systems are capable of universally approximating non-linear dynamics with arbitrary precision: this is achieved via switching among a finite set of linear submodels, thus facilitating the non-linear system identification problem into the identification of several affine submodels [3]. Hybrid system can be as well considered a special class of LPV systems [4] with non-smooth and abrupt parameter transitions [2]. Moreover, they can be considered as time-varying (TV) systems, where the batch of data used to identify the system in an offline fashion may cover only some modes of the system and not all [5].

The identification of hybrid systems is rather a challenging problem, this indeed can be seen in the coupling between the submodel parameter estimation and the switch function estimation, as neither the former nor the later is known *a priori* [6, 7]. In recent decades, the identification of hybrid systems has attracted much interest among researchers [8]. This is the case since they appear in many different contexts and because of their universal ability of modelling dynamical systems in different engineering applications [9, 10]; namely, in chemical processes [11], robotics manufacturing processes [12, 13], air traffic systems [14, 15, 16], computer and communication networks [17, 18], power systems [19], medical applications [20], human control behavior [21], computer vision [22], and biology and environmental systems [23].

Switched linear systems mark a particular type of hybrid systems, wherein the different dynamical submodels are activated via an external switching mechanism [6]. The problem of identifying a switched linear system (SLS) can be broken to identifying the different submodels and the switching function governing the system from a set of input-output data. SLS models can be modelled in either input-output models or state-space (SS) models. Most of the existing work on SLS models or Hybrid system identification in general is exclusive to the class of input-output models, namely, the piecewise auto-regressive exogenous (PWARX) and switched autoregressive models (SARX) [24, 25, 26, 27, 28, 29, 30, 31, 32, 33, 34, 35]. An

excellent thorough survey on the developed techniques is found in [7].

Input-output models are certainly capable of describing the system [5], at least SISO, but they are hardly appropriate for describing MIMO systems due to structural identifiability issues [36]. In particular, the underlying regressor vector size is large and hence preventing the regression matrix from enjoying full rank property which gives rise to an ill-structured LS problem no matter how much exciting the input is. On the other hand, state-space models offer a unified approach to treat both SISO and MIMO systems beside their compact and convenient representation. Furthermore, the theory of observability, controllability, fault detection, observer design, system realization, and process control is well established for these models set [37]. Therefore, MIMO switched systems are more suitably represented in state-space models, wherein rank deficiency issues are lifted by virtue of minimal representations, and the problem of identifying switched MIMO systems is made more efficient while represented in state-space format [36].

The main challenges for the problem of estimating switched linear state-space models lies in the fact that the discrete state (switching mechanism), the continuous state, and the submodel parameters are all unknown to us and highly coupled [38]; beside that, compared to switched input-output models, switched state-space models mark an additional difficulty highlighted in the fact that continuous state is unknown, which explicitly entails that the regression vector is not known fully. Therefore, the partitioning of input-state space in the piecewise-affine state-space models class (PWASS) becomes a non-trivial and hard task. Another problem is that usually the identified submodels are present in different state basis, therefore can not be combined directly to fully represent the SLS system until and unless the different submodels are set to the same state basis [1, 39]

1.1. Literature review and motivation of the study

Several identification paradigms have appeared in literature to identify switched linear state-space models, see, e.g., [1, 5, 6, 20, 36, 37, 38, 39, 40, 41, 42, 43, 44, 45, 46, 47, 48]. The existing work could be divided into offline approaches and online approaches.

1.1.1. Offline identification

The problem can be tackled in a classical way, first estimating an I/O model, then converting it to state-space form. This is done by finding the state-space equivalent of each I/O submodel. Let's say the switched auto-regressive exogenous (SARX) is considered, the later equivalence is well studied and documented in literature. It was shown in [49] that every switched ARX (SARX) model of s modes is equivalent

to a switched linear state-space model, though may not be a minimal realization, with the same number of modes but not the other way around since converting observable switched state-space models with s modes to SARX models usually leads to more than s modes. Note that different state-space forms can be derived from I/O models; usually canonical forms are derived to avoid overparametrization and identifiability issues, namely, Observability canonical form, Controllability canonical form, Controller canonical form, Observer canonical form and Modal form. State transformations to convert arbitrary state-space models to those canonical forms were also established [2].

Switch detection techniques are also reported in literature, these techniques go about solving the problem by first detecting the switches, then estimating the submodels on the delimited time segments. The detection is carried on by sliding a window along the data and repeatedly checking a test criteria to extract the switching times. Change detection techniques are of recursive nature, hence more suitable for online identification. This, nonetheless, does not prevent their use for batch identification. The work in [20] follows the same strategy, and works under the assumption of stable submodels and a dwell time constraint for parameter estimation and change detection. A residual vector is continuously calculated by sliding a window over the batch of data, following that a statistical test to determine the switch instants. Submodel parameters are found by deploying a subspace algorithm segment-wise. Having the submodels from all segments in hand allows clustering the similar models in common sets then merging them to re-estimate the submodels parameters again. This process is repeated until there are no similar local models left. One drawback of this approach is that there is a slight detection delay which indicates that the switching instants can not be precisely located. However to remedy this a re-clustering mechanism of the points around the switch points(dubious points) was conducted in [20].

Another change detection technique appears in [40], it uses subspace methods. A rank test is posed on the observability subspace to detect the switches, a rank change in the observability matrix indicates a switch. This test is repeatedly checked by sliding a window over the data until all the data is scanned. This method doesn't need to impose the stability assumption but the models must involve only measurement noise ($w_k = 0$). Once the switches are obtained, a subspace algorithm is used on the delimited segments for estimating the model parameters, then merging the similar models according to a distance criteria, based on the extracted range space of the observability matrices. Note that here again a window of certain length slides over the data, and an SVD of the projected output is computed continuously

to retrieve both the observability subspace and the correspondent dimension, which indicates that the algorithm is computationally expensive. To circumvent this, the authors in [41] suggested the use of an iterative SVD update instead of the SVD of each batch of data, to ease up the computational complexity. A linear transformation to render all the identified submodels to the same state basis was also derived there.

A different way of approaching the problem was devised in [1]. Here, the algorithm starts by dividing the data into smaller non-overlapping data sets of equal length, then, a subspace algorithm is carried through each data sets, the identified state-space local models are next clustered using their Markov parameters, the robust competitive clustering algorithm (RCA), which that does not need the knowledge of the number of subsystems, is used. Later, the Markov parameters are assigned to the found clusters according to their degree of relevance. Following that, the data points responsible for generating those Markov parameters are assigned to the same cluster and the switching function can be deduced by inspecting the clustered data. The submodels identified before were based on small data sets, hence more prone to noise errors. Now that the data is clustered properly, the identification can be conducted with much bigger sets to obtain a final estimate of the submodels. This identification was again conducted via a subspace based algorithm. The problem of the submodels being in different state basis is circumvented by a state transformation that allows the models to be in the canonical form as proposed in [50], and therefore the different identified subsystems can be harmlessly combined. This algorithm however suffers when fast switching systems with high noise levels are considered. This is the case, since, when clustering the data, smaller outlier subsets populated by data belonging to mixed submodels are created [37]. The data classification obtained that way later produces unreliable model estimations. To remedy this, the authors in [37] proposed that the outlier data should not be used in the classification step, and later on reclassified using an interacting multiple model (IMM), this method proved to improve the performance and robustness for the underlying algorithm.

In [6], the algorithm therein alternates between updating a switching sequence and updating the state-space parameters via a subspace algorithm. It requires the knowledge of the number submodels along with the number of switches. By upper-bounding the number of switching among the different submodels, the problem can be casted as a binary integer programming problem that can be solved heuristically via the block-coordinate descent to obtain the switching function. The algorithm starts by randomly initializing the switching function, following that with a sub-model estimation via a subspace algorithm, then updating the switching function

given the submodels from the previous step. This process is repeated until the algorithm converges. This algorithm is favourable in the sense that the switches are detected all at once without the need for a sliding window which apparently needs length tuning which affects the performance and the accuracy of the estimate. Furthermore, the algorithm converges rapidly under less than ten iterations.

Among the first works in this context was also presented in [39] where the switching sequence is assumed to be known which simplifies the problem significantly. A subspace algorithm was used to identify the submodels parameters and a dwell time requirement was imposed according to the Hankel matrix block size that appears in the subspace algorithm formulation. Since the different submodels are identified with respect to different state basis, we can't combine them unless they are represented within the same state basis. To this end, the authors in [39] suggest that provided enough number of the submodel transitions for which the states at the transition are linearly independent, one can recover a state transformation matrix that renders all of the identified models to the same state basis. A novel contribution to solve the problem when the continuous state is known is reported in [36]. This approach relies on sparse optimization. However the assumptions of known discrete or continuous state are not practical in general. In [38], the authors managed to eliminate the unknown states off the state-space output equation on a time horizon under the assumption of pathwise observability, then employing an algorithm that alternates between data classification and parameter updates via the recursive least square method. Unfortunately the algorithm used there is far from being optimal and the effect of eliminating the states makes it impossible to recover the original switched system, and only recovering a transformed version with an increased number of discrete states.

In [42], the input-output data are projected into a high dimensional space then clustered using a GPCA (general principal component analysis) approach that segments the data into multiple subspaces. Two different embeddings were discussed and their privileges and drawbacks in terms of accuracy and speed were also compared. In [43], an approximate expected maximization (EM) based method is devised for learning the parameters of an SLS system, however this method might get stuck at local optimums. To remedy this and as a continuation to the previous approach, the work in [44] suggests a refinement that prevents local convergence, by directing the search toward better maximums of the likelihood function.

1.1.2. Online identification

Recursive methods for estimating switched affine state-space models in an online framework were also established in literature. In [45, 46], a two step online identifi-

cation approach was followed. In the first step, the Markov parameters matrices are estimated in an online fashion by employing a subspace algorithm, following that, the clustering of these matrices via a classification algorithm to determine the different local models. An online structured subspace identification method combined with switch detection strategy was reported in [5, 47], this method requires a minimum dwell time of the submodels. Finally, the work in [48] proposed a recursive approach to estimate a switched affine state-space system corrupted with unknown, non-centered and sparse error sequence via sparse optimization.

Despite of the previously listed contributions, the problem of identifying state-space SLS models is still not fully solved. Some of the presented approaches hinge on very stringent conditions, like fully known continuous state trajectory, fully known switching sequence, and very long dwell time. Others, on the other hand, solve the problem under somewhat mild restrictions, but are disfavoured due to the heavy computational load they pose. Due to this latter reasons, our main motivation is to propose a new identification scheme for the set of state-space SLS models, with a problem formulation given bellow.

1.2. Problem formulation

In this thesis, we consider a special class of linear time-varying (LTV) and single-input/single-output (SISO) systems represented by the state-space equations

$$x(k+1) = A(k)x(k) + b(k)u(k) + w(k), \quad (1.1)$$

$$y(k) = c^T(k)x(k) + d(k)u(k) + e(k) \quad (1.2)$$

where $u(k) \in \mathbb{R}$, $y(k) \in \mathbb{R}$, and $x(k) \in \mathbb{R}^n$ are respectively the input, the output, the state sequences, and c^T denotes the transpose of a given vector (matrix) c ; $w(k)$ and $e(k)$, on the other hand, are noise sequences. The state dimension n is assumed to be known and does not change with time.

Most of the results derived in this thesis are for the set of MIMO systems. Therefore, unless and only otherwise stated that it is exclusive for SISO, the formulations by default are presented in MIMO predominantly. To this end, we rewrite the MIMO version of Eqs.(1.1)-(1.2)

$$x(k+1) = A(k)x(k) + B(k)u(k) + w(k), \quad (1.3)$$

$$y(k) = C(k)x(k) + D(k)u(k) + e(k) \quad (1.4)$$

where $u(k) \in \mathbb{R}^m$, $y(k) \in \mathbb{R}^p$, and $x(k) \in \mathbb{R}^n$.

Let $\mathbb{N}$ denote the set of positive integers and φ be a switching sequence, *that*

is, a map from $\mathbb{N}$ into $\mathbb{S} = \{1, \dots, \sigma\}$ for some fixed $\sigma \in \mathbb{N}$. Suppose $\mathcal{P}_{\varphi(k_1)} = \mathcal{P}_{\varphi(k_2)}$ for all k_1, k_2 with $\varphi(k_1) = \varphi(k_2)$ where $\mathcal{P}_{\varphi(k)} = \{A(k), B(k), C(k), D(k)\}$. Thus, $\mathcal{P}_{\varphi(k)}$ has a finite range containing at most σ submodels. Let $\mathcal{P} = \{\mathcal{P}_1, \dots, \mathcal{P}_\sigma\}$. The MIMO model in Eqs.(1.3)–(1.4) with the state-space matrices changing as a function of φ is a switched linear system (SLS).

A switching sequence $\varphi(k)$ segments observation intervals, *that is*, it partitions a given interval $[1, \dots, N]$ into disjoint semi-closed intervals $[k_i, k_{i+1})$ such that $k_0 = 1$, $k_{i^*} \leq N$, and

$$\varphi(k) = \varphi(k_i), \quad k_i \leq k < k_{i+1} \quad (1.5)$$

where the sequence $[k_1, \dots, k_i, k_{i+1}, \dots, k_{i^*}]$ is the sequence of switches. To each segmentation of $[1, \dots, N]$ denoted by χ , we allocate a dwell sequence $\delta(\chi)$ and the minimum dwell time by

$$\begin{aligned} \delta(\chi) &= \{k_{i+1} - k_i : i = 0, \dots, i^* - 1\}, \\ \delta_*(\chi) &= \min \delta(\chi). \end{aligned} \quad (1.6)$$

Thus, the dwell sequence is the waiting time of the submodels and the minimum dwell time is the smallest waiting time of all submodels.

The SLS identification problem we study in this thesis is laid as follows:

Problem 1.2.1 Given a finite set of the input-output measurements $\{u(k), y(k) : k = 1, \dots, N\}$, estimate the σ submodels $\mathcal{P}_k$, $k = 1, \dots, \sigma$ and the switching sequence $\varphi(k)$, $k = 1, \dots, N$.

In the course of developing the algorithms that solve the identification problem posed above, we will impose certain restrictions on the input sequence, the submodels, and the switching sequence. The assumption of a common MacMillan degree for all submodels is made to avoid degeneracies, for example, non-square state-transition matrices.

1.3. Organization of the thesis

The following is an outline of the contents and contributions of the chapters.

Chapter 2 serves as an introduction that presents the foundations of the presented methodology in this thesis. More specifically, we review discrete-time linear time-invariant systems (LTI), system/observer Markov parameters, state-space realization methods including Observer/Kalman Filter Identification (OKID), and the Eigensystem Realization Algorithm (ERA). We as well provide a brief introduction

to compressive sensing and the different compressive sensing recovery algorithms along with recovery conditions. Chapter 3 presents a MIMO LTV state-space realization from sequences of doubly indexed Markov parameters. The observer-based transformation to convert SLS models to SARX models is presented along. An algorithm that retrieves a MIMO SLS model given its LTV realization is presented in Chapter 4. A two-step algorithm is proposed, first, by retrieving the submodels via clustering, then the switching function by making use of the previously obtained submodels. Chapter 5, on the other hand, details switch identifiability conditions for MIMO SARX models along with persistence of excitation conditions on the input signal for both MIMO SARX and SISO SARX systems separately. The compressive identification scheme to recover the SARX model from input/output data for SISO systems is laid down in chapter 6. Chapter 7 concludes the thesis and presents numerous possible future research issues. Proofs are given in Appendix A, and DBSCAN clustering is presented in Appendix B.

The contents of Chapter 3 and Chapter 5 in part and Chapter 6 were submitted to [51] for publication. The content of Chapter 4, and of Chapter 6 extended to MIMO systems are ready to be submitted for publication.

2. BACKGROUND AND UNDERLYING THEORY

This chapter is dedicated to detailing the background and definitions relevant to the study proposed in this thesis. Some of the materials reviewed here are not used in later chapters, but they are included for the sake of completeness.

2.1. Hybrid system state-space models

An affine hybrid system in the state-space form [2] is given by

$$x_{k+1} = A_{\varphi(k)}x_k + B_{\varphi(k)}u_k + \tilde{f}_{\varphi(k)} + w_k, \quad (2.1)$$

$$y_k = C_{\varphi(k)}x_k + D_{\varphi(k)}u_k + \tilde{g}_{\varphi(k)} + v_k, \quad (2.2)$$

where $x_k \in \mathbb{R}^{n_x}$, $u_k \in \mathbb{R}^{n_u}$, and $y_k \in \mathbb{R}^{n_y}$ are the continuous state, the input and the output vectors respectively, $w_k \in \mathbb{R}^{n_x}$ and $v_k \in \mathbb{R}^{n_y}$ are the process and measurement noise, and $\varphi(k) \in \{1, \dots, \sigma\}$ is the discrete state, and the matrices $A_j \in \mathbb{R}^{n_x \times n_x}$, $B_j \in \mathbb{R}^{n_x \times n_u}$, $C_j \in \mathbb{R}^{n_y \times n_x}$, $D_j \in \mathbb{R}^{n_y \times n_u}$ and vectors $\tilde{f}_{\varphi(k)}$ and $\tilde{g}_{\varphi(k)}$ indicates that sub-model $j \in \{1, \dots, \sigma\}$ is currently the active mode. In the case the terms $\tilde{f}_{\varphi(k)}$ and $\tilde{g}_{\varphi(k)}$ are discarded, the system is referred to as a linear hybrid system or more commonly a switched linear system (SLS), the addition of the affine terms makes it a switched affine system as per [49]. The state dimension n_x can be constant among the different sub-models or variant. In the case of a constant state dimension, one needs the state bases of all local state-space models to be the same, this is not the case when a varying state dimension is assumed. The discrete state $\varphi(k)$ is assumed to be arbitrary and independent of the state and input of the system and switches among finite set of modes $\varphi(k) \in \{1, \dots, \sigma\}$. In the case the discrete state depends on the state and input (regressors), the model is refer to as piecewise affine model (PWA). Precisely in this case, $\varphi(k)$ depends on a polyhedral partition of the state and input space [52], i.e, $\varphi(k) = j \iff \begin{bmatrix} x_k \\ u_k \end{bmatrix} \in \Omega_j$ where $\{\Omega_j\}_{j=1}^{\sigma}$ are convex polyhedral partitioning of the whole state-input space $\Omega \subseteq \mathbb{R}^{n_x+n_u}$. A special case arise when $\varphi(k)$ is modeled as a Markov chain with transition probabilities $p_{i,j}(k) = P(\varphi(k+1) = j / \varphi(k) = i)$ (the probability of transition from sub-model i to sub-model j), the model then becomes a jump Markov affine system.

2.2. Recovering a common basis for the state-space submodels

Let's assume now that all the different state-space submodels have been identified separately by employing an Eigensystem realization algorithm [53] in every subsection. Given that the different state-space models are defined w.r.t a similarity transformation that is different for all the sub-models, in order to combine the different sub-models, they all must be transformed to a common state basis [2]. Based on the work from [39], such task can be accomplished by considering the states just before the transitions and just after that and building the state transformations according to that.

At a transition time k_i from submodel m to n , we can have an estimate of the final state of model m , $\hat{x}_{k_i}^m$, and the initial state of model n , $\hat{x}_{k_i}^n$. The two states can be relate via

$$\hat{x}_{k_i}^m = T_{mn} \hat{x}_{k_i}^n, \quad (2.3)$$

with $k_i \in \{k_1, k_2, \dots\}$ being any of the transition times within the switching sequence. The transformation can then be recovered by

$$T_{mn} = \begin{bmatrix} \hat{x}_{k_1}^m & \hat{x}_{k_2}^m & \dots \end{bmatrix} \begin{bmatrix} \hat{x}_{k_1}^n & \hat{x}_{k_2}^n & \dots \end{bmatrix}^\dagger, \quad (2.4)$$

T_{mn} exist if the matrix $\begin{bmatrix} \hat{x}_{k_1}^n & \hat{x}_{k_2}^n & \dots \end{bmatrix}$ is full column rank n_x , which means sufficient number of independent states at the transition points is required to obtain the linear transformation. Once it is obtained, submodel n is transformed via

$$\begin{aligned} A_n &\leftarrow T_{mn} A_n T_{mn}^{-1}, \\ B_n &\leftarrow T_{mn} B_n, \\ C_n &\leftarrow C_n T_{mn}^{-1}, \\ D_n &\leftarrow D_n, \end{aligned} \quad (2.5)$$

to bring it to the state basis as model m . Another method that does not involve the computation of the linear transformation via the states is by estimating the different submodels in the canonical form, the transformation of an arbitrary state-space model to its observable canonical form has been studied in literature and the linear transformation for that was derived [50], this is case for SISO and MISO systems, but unfortunately not for MIMO systems. For observable MIMO systems where the characteristic polynomial is equal to the minimal polynomial this can be done as reported in [50]. The idea is to replace one output of the system with a linear combination of the other outputs so that a linear transformation to the

observability canonical form can be recovered [2]. For that, let's introduce a square matrix $H \in \mathbb{R}^{n_y \times n_y}$ defined as the identity matrix except one row (let's consider the first row here) equal to h^T , $h^T = \begin{bmatrix} h_1 & \dots & h_{n_y} \end{bmatrix}$ being a vector generated from a uniform distribution. If we define the observability matrix as $O = O_{n_x}(A, h^T C)$ then it is almost surely of full rank n_x , therefore the linear transformation can be taken as $T = O_{n_x}(A, h^T C)$. Before applying the linear transformation, the output transformed model is

$$\bar{y}_k = Hy_k = H(Cx_k + Du_k) = \bar{C}x_k + \bar{D}u_k. \quad (2.6)$$

Then, applying the linear transformation

$$\begin{aligned} \tilde{A} = TAT^{-1} &= \begin{bmatrix} 0 & 1 & 0 & \dots & 0 \\ 0 & 0 & 1 & \dots & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & 0 & \dots & 1 \\ -a_0 & -a_1 & -a_2 & \dots & -a_{n_x-1} \end{bmatrix}, & \tilde{B} = TB, \\ \tilde{C} = HCT^{-1} &= \begin{bmatrix} \begin{bmatrix} 1 & 0 & \dots & 0 \end{bmatrix} \\ c_2^T \\ \vdots \\ c_{n_y}^T \end{bmatrix}, & \tilde{D} = HD, \end{aligned} \quad (2.7)$$

where a_i 's are the coefficients of the characteristic polynomial of A , and c_i 's are the rows of C . This method of transformation is applied to all the submodels.

2.3. Deadbeat observer

Given an LTI system in the state-space form

$$x_{k+1} = Ax_k + Bu_k, \quad (2.8)$$

$$y_k = Cx_k + Du_k, \quad (2.9)$$

with the conventional dimensions assumption on the different matrices and vectors. Under the assumption of an observable pair (A, C) , the matrix G is said to be a dead beat observer for the system if and only if the dead beat condition is satisfied [54, 55, 56]:

$$(A + GC)^p = 0_{n_x \times n_x}. \quad (2.10)$$

p is the smallest integer satisfying $n_y p \geq n_x$, or $p \geq \frac{n_x}{n_y}$. As an output feedback, the dead beat observer places the closed loop poles at the origin, and hence the fastest observer that could be realized. Similarly to the time invariant case, a time-

varying dead beat observer is also introduced. According to [55], a linear discrete time-varying observer is said to be deadbeat if the following condition holds:

$$(A_{k+p-1} + G_{k+p-1}C_{k+p-1})(A_{k+p-2} + G_{k+p-2}C_{k+p-2}) \cdots (A_k + G_kC_k) = 0_{n_x \times n_x}, \quad (2.11)$$

for every k , and p is the smallest integer satisfying the equation. Similarly if we define the observer state transition matrix as

$$A_k^o = A_k + G_kC_k, \quad (2.12)$$

the condition can be translated to

$$\Phi_o(k+p, k) = 0. \quad (2.13)$$

2.4. State-space representation and Markov parameters

The input-output relationship of the state-space system in Eqs.(2.8-2.9) is described by:

$$y(k) = CA^k x(0) + \sum_{i=1}^k CA^{i-1}Bu(k-i) + Du(k). \quad (2.14)$$

$x(0)$ is the initial state of the system, generally assumed unknown. The parameter sequence $D, CB, CAB, CA^2B, CA^3B, \dots$ is known as the system Markov parameters or simply system pulse responses that fully characterize the system. In an observable canonical form representation of the system, these parameters appear explicitly [57]. For an I/O record of length N and under the assumption of zero initial state one can write in matrix form:

$$Y = HU, \quad (2.15)$$

with

$$\begin{aligned} Y &= \begin{bmatrix} y_0 & y_1 & \cdots & y_{N-1} \end{bmatrix} \in \mathbb{R}^{n_y \times N}, \\ H &= \begin{bmatrix} D & CB & CAB & \cdots & CA^{N-2}B \end{bmatrix} \in \mathbb{R}^{n_y \times Nn_u}, \\ U &= \begin{bmatrix} u_0 & u_1 & \cdots & u_p & \cdots & u_{N-1} \\ & u_0 & \cdots & u_{p-1} & \cdots & u_{N-2} \\ & & \ddots & \vdots & \vdots & \vdots \\ & & & u_0 & \vdots & \ddots \\ & & & & \cdots & u_0 \end{bmatrix} \in \mathbb{R}^{Nn_u \times N}. \end{aligned} \quad (2.16)$$

Note that there are $Nn_y n_u$ unknowns and only Nn_y equations, hence if $n_u > 1$ H is non-unique. If $n_u = 1$ (single input system) H can be uniquely retrieved.

In the case of an asymptotically stable system i.e $\lim_{k \rightarrow \infty} CA^k B = 0$, the input-output map can be described by a finite number of Markov parameters, for instance p of them if $CA^i B \approx 0, i \geq p$. The input-output map therefore becomes

$$y(k) = CA^k x(0) + \sum_{i=1}^p CA^{i-1} Bu(k-i) + Du(k), \quad (2.17)$$

and in matrix form

$$Y \cong H_p U_p, \quad (2.18)$$

with

$$H_p = \begin{bmatrix} D & CB & CAB & \dots & CA^{p-2}B \end{bmatrix} \in \mathbb{R}^{n_y \times pn_u},$$

$$U_p = \begin{bmatrix} u_0 & u_1 & \dots & u_{p-1} & \dots & u_{N-1} \\ & u_0 & \dots & u_{p-1} & \dots & u_{N-2} \\ & & \ddots & \vdots & \vdots & \vdots \\ & & & \vdots & \vdots & \ddots \\ & & & u_0 & \dots & u_{N-p} \end{bmatrix} \in \mathbb{R}^{pn_y \times N}. \quad (2.19)$$

Note that in this case if we select N such that $Nn_y \geq pn_y n_u$ (i.e. $N \geq pn_u$), we end up having more equations than unknowns, and hence can solve uniquely for the Markov parameters via the least square

$$H_p = YU_p^\dagger. \quad (2.20)$$

2.5. State transformation and similarity of state-space models

It is possible to find an infinite number of presentations for the same state-space model given that the I/O mapping is preserved among the different submodels. Assume T is a coordinate transformation or a mapping such that:

$$x_k = Tz_k, \quad (2.21)$$

where the matrix T is a square non-singular matrix. Substituting for x_k yields the new state-space model :

$$\begin{cases} z_{k+1} = \tilde{A}z_k + \tilde{B}u_k, \\ y_k = \tilde{C}z_k + \tilde{D}u_k, \end{cases} \quad (2.22)$$

where

$$\begin{aligned}\tilde{A} &= T^{-1}AT, \\ \tilde{B} &= T^{-1}B, \\ \tilde{C} &= CT, \\ \tilde{D} &= D,\end{aligned}\tag{2.23}$$

and hence the linear mapping

$$T : \{A, B, C, D\} \rightarrow \{T^{-1}AT, T^{-1}B, CT, D\}.\tag{2.24}$$

Notice however the Markov parameters are invariant for the different similar systems:

$$\tilde{C}\tilde{A}^k\tilde{B} = CT(T^{-1}AT)^kT^{-1}B = CA^k B.\tag{2.25}$$

2.6. Relationship between state-space models and input-output models via observer Markov parameters

It was shown that the I/O map for the state-space system depends on the initial state x_0 which is generally unknown. To remove the dependence on the initial state and fully characterize the state-space model in I/O form, we follow the strategy developed in [58, 59, 60]. Adding and subtracting My_k to the state equation:

$$\begin{aligned}x_{k+1} &= Ax_k + Bu_k + My_k - My_k \\ &= (A + MC)x_k + (B + MD)u_k - My_k \\ &= \bar{A}x_k + \bar{B}v_k.\end{aligned}\tag{2.26}$$

Denote now

$$\begin{aligned}\bar{A} &= A + MC, \\ \bar{B} &= \begin{bmatrix} B + MD & -M \end{bmatrix}, \\ v_k &= \begin{bmatrix} u_k \\ y_k \end{bmatrix}, \\ \bar{D} &= \begin{bmatrix} D & 0 \end{bmatrix},\end{aligned}\tag{2.27}$$

and hence we obtain a different form of the state-space model in Eqs.(2.8)-(2.9)

$$x_{k+1} = \bar{A}x_k + \bar{B}v_k, \quad (2.28)$$

$$y_k = Cx_k + \bar{D}v_k. \quad (2.29)$$

The gain matrix $M \in \mathbb{R}^{n_x \times n_y}$ renders the new system asymptotically stable and its existence is guaranteed if the pair (A, C) is observable. We can consider the state-space form to be an observer form if the state is an observer state and the Matrix M is the observer gain, and in the presence of measurement and process noise a Kalman filter [57]. Regardless of the nature of the system, the observer form is able to compress the system Markov parameters to a finite number of them by forcing an asymptotic behavior on the transformed system. Among the infinite number of Markov parameters, only p of them are significant and hence the I/O map

$$y_k = C\bar{A}^k x_0 + \sum_{i=1}^p C\bar{A}^{i-1}\bar{B}v_{k-i} + \bar{D}v_k. \quad (2.30)$$

given that $\bar{A}^i \equiv 0$ for $i \geq p$, then $C\bar{A}^i\bar{B} \equiv 0$. In this case, the observer is denoted as the deadbeat observer and places the system poles at the origin. The initial state dependence vanishes as well in a finite time under the deadbeat condition, i.e.,

$$y_{k+p} = \sum_{i=k}^{k+p-1} C\bar{A}^{k+p-i-1}\bar{B}v_i + \bar{D}v_{k+p}, \quad k \geq 1. \quad (2.31)$$

The sequence of parameters $\bar{D}, C\bar{B}, C\bar{A}\bar{B}, C\bar{A}^2\bar{B}, \dots$ forms the observer Markov parameters. Introducing the observer clearly helped us convert the state-space model to its equivalent I/O model. The previous I/O model depicts the output as a function of the past outputs and inputs, hence its an ARX model (Auto-regressive with exogenous inputs). To make this clear, the previous I/O map can be written as

$$y(k) = \sum_{i=1}^p \alpha_i y_{k-i} + \sum_{i=0}^p \beta_i u_{k-i}, \quad k \geq p, \quad (2.32)$$

with

$$\begin{aligned} \alpha_i &= -C(A + MC)^{i-1}M, \\ \beta_i &= C(A + MC)^{i-1}(B + MD), \\ \beta_0 &= D. \end{aligned} \quad (2.33)$$

In the presence of a Gaussian, white, zero-mean, and uncorrelated process and measurement noises with unknown covariance matrices and when choosing p sufficiently

large such that $\bar{A}^p = (A + MC)^p \approx 0$, the ARX coefficients obtained from a least square solution with enough input/output data coincide with the solution obtained when a Kalman filter with gain K is deployed, and the ARX model residuals coincide with the Kalman filter residuals [61], therefore the observer gain M is equivalent to the steady-state Kalman filter gain K in this case. The results established here lies the foundation the infamous Observer/Kalman filter identification (OKID) algorithm which identify a state-space model along with an optimal observer/Kalman filter gain from noisy I/O data assuming unknown process and measurement noise covariance matrices [60].

2.7. The system Markov parameters

Given a sequence of the observer Markov parameters, one can recover the system Markov parameters, for now let's consider

$$Y_k = CA^k B, \quad (2.34)$$

$$\bar{Y}_k = C\bar{A}^k \bar{B}. \quad (2.35)$$

The system transmission matrix, D , is given by

$$Y_{-1} = D = \bar{D}(:, 1 : n_u) = \bar{D} \begin{bmatrix} I_{n_u} \\ 0_{n_y \times n_u} \end{bmatrix} = \bar{Y}_{-1} \begin{bmatrix} I_{n_u} \\ 0_{n_y \times n_u} \end{bmatrix}. \quad (2.36)$$

The observer Markov parameters can be partitioned as

$$\begin{aligned} \bar{Y}_k &= C\bar{A}^k \bar{B} \\ &= \begin{bmatrix} C(A + MC)^k (B + MD) & -C(A + MC)^k M \end{bmatrix} \\ &= \begin{bmatrix} \bar{Y}^{(1)}(k) & \bar{Y}^{(2)}(k) \end{bmatrix}. \end{aligned} \quad (2.37)$$

Then, the system Markov parameters can be inferred from the observer Markov parameters according to the recursive formula,

$$Y_k = \bar{Y}_k^1 + \sum_{i=0}^{k-1} \bar{Y}_i^2 Y_{k-i-1} + \bar{Y}_k^2 Y_{-1}, \quad (2.38)$$

since $\bar{Y}_k \equiv 0$ for $k \geq p$ due to the deadbeat condition, we obtain

$$Y_k = \bar{Y}_k^1 + \sum_{i=0}^{p-1} \bar{Y}_i^2 Y_{k-i-1} + \bar{Y}_k^2 Y_{-1}. \quad (2.39)$$

It can be shown that the proper choice for the observer order p must satisfy the condition $n_y p \geq n_x$ as pointed in [60] to be able to later realize the actual system

of order n_x , and a maximum system order of $n_y p$ can be achieved (given that the number of block columns is strictly greater than the number of block rows of the Hankel matrix) with p observer Markov parameters. Note that $p_{\min} \geq \frac{n_x}{n_y}$, so for a multi-output system the number of observer Markov parameters required to be estimated can be significantly less than the system order.

The observer gain M can as well be inferred from the observer Markov parameters, for that let's define first the set of observer gain Markov parameters

$$Y_k^M = C A^k M. \quad (2.40)$$

Similarly to the system Markov parameters, the observer gain Markov parameters can as well be recovered from the observer Markov parameters, the following recursive formula holds

$$Y_k^M = -\bar{Y}_k^{(2)} + \sum_{i=0}^{k-1} \bar{Y}_i^{(2)} Y_{k-i-1}^M, \quad (2.41)$$

and the observer gain

$$M = (O_{k+1}^T O_{k+1})^{-1} O_{k+1}^T Y_M \quad (2.42)$$

where

$$O_{k+1} = \begin{bmatrix} C \\ CA \\ CA^2 \\ \vdots \\ CA^k \end{bmatrix} \quad Y_M = \begin{bmatrix} Y_1^M \\ Y_2^M \\ Y_3^M \\ \vdots \\ Y_k^M \end{bmatrix} = \begin{bmatrix} CM \\ CAM \\ CA^2 M \\ \vdots \\ CA^k M \end{bmatrix}. \quad (2.43)$$

O_{k+1} being the extended Observability matrix.

2.8. Process and measurement noise

Now, let's assume the presence of process and measurement noise

$$x_{k+1} = Ax_k + Bu_k + w_k, \quad (2.44)$$

$$y_k = Cx_k + Du_k + e_k. \quad (2.45)$$

Following the same previous steps by introducing an observer gain M gives the new state-space model

$$\begin{cases} x_{k+1} = \bar{A}x_k + \bar{B}v_k + \begin{bmatrix} M & I_{n_x} \end{bmatrix} \tilde{e}_k, \\ y_k = Cx_k + \bar{D}v_k + \begin{bmatrix} I_{n_y} & 0_{n_y \times n_x} \end{bmatrix} \tilde{e}_k, \end{cases} \quad (2.46)$$

where $\tilde{e}_k = \begin{bmatrix} e_k^T & w_k^T \end{bmatrix}^T$ is the augmented noise vector and the matrices $\bar{A}$, $\bar{B}$, $\bar{D}$ and vector v_k are as defined previously. The I/O map is therefore given by

$$y_k = C\bar{A}^k x_0 + \sum_{i=1}^p C\bar{A}^{i-1}\bar{B}v_{k-i} + \sum_{i=1}^p C\bar{A}^{i-1} \begin{bmatrix} M & I_{n_x} \end{bmatrix} \tilde{e}_{k-i} + \bar{D}v_k + e_k. \quad (2.47)$$

Given that $\bar{A}^i \equiv 0$ for $i \geq p$, then $C\bar{A}^i\bar{B} \equiv 0$ and $C\bar{A}^{i-1} \begin{bmatrix} M & I_{n_x} \end{bmatrix} = 0$. After the initial state vanishes, we obtain

$$y_{k+p} = \sum_{i=k}^{k+p-1} C\bar{A}^{k+p-i-1}\bar{B}v_i + \bar{D}v_{k+p} + \sum_{i=k}^{k+p-1} C\bar{A}^{k+p-i-1} \begin{bmatrix} M & I_{n_x} \end{bmatrix} \tilde{e}_i + e_{k+p}, \quad k \geq 1. \quad (2.48)$$

Again the sequence of parameters $\bar{D}, C\bar{B}, C\bar{A}\bar{B}, C\bar{A}^2\bar{B}, \dots$ consists of the observer Markov parameters, where as $\begin{bmatrix} I_{n_y} & 0_{n_y \times n_x} \end{bmatrix}, C \begin{bmatrix} M & I_{n_x} \end{bmatrix}, C\bar{A} \begin{bmatrix} M & I_{n_x} \end{bmatrix}, C\bar{A}^2 \begin{bmatrix} M & I_{n_x} \end{bmatrix}, \dots$ are the noise Markov parameters. The previous equation depicts the output as a function of the past inputs and outputs and the filtered process and measurement noises, and hence it is an ARMAX model, this can be made clear if we rewrite the model as:

$$y(k) = \sum_{i=1}^p \alpha_i y_{k-i} + \sum_{i=0}^p \beta_i u_{k-i} + \sum_{i=0}^p \gamma_i \tilde{e}_{k-i}, \quad k \geq p, \quad (2.49)$$

with

$$\begin{aligned} \alpha_i &= -C(A + MC)^{i-1}M, \\ \beta_i &= C(A + MC)^{i-1}(B + MD), \beta_0 = D, \\ \gamma_i &= C(A + MC)^{i-1} \begin{bmatrix} M & I_{n_x} \end{bmatrix}, \gamma_0 = \begin{bmatrix} I_{n_y} & 0_{n_y \times n_x} \end{bmatrix}. \end{aligned} \quad (2.50)$$

If instead, the initial state-space model was in innovation form presented in [62]

$$\begin{cases} x_{k+1} = Ax_k + Bu_k + K\varepsilon_k, \\ y_k = Cx_k + Du_k + \varepsilon_k, \end{cases} \quad (2.51)$$

where $K \in \mathbb{R}^{n_x \times n_y}$ is the Kalman gain and ε_k is the innovation, then the correspondence between the state-space model and the ARMAX model is explained by the equation

$$y(k) = \sum_{i=1}^p \alpha_i y_{k-i} + \sum_{i=0}^p \beta_i u_{k-i} + \sum_{i=1}^p \gamma_i \varepsilon_{k-i} + \varepsilon_k, \quad k \geq p, \quad (2.52)$$

with

$$\begin{aligned}\alpha_i &= -C(A + MC)^{i-1}, \\ \beta_i &= C(A + MC)^{i-1}(B + MD), \beta_0 = D, \\ \gamma_i &= C(A + MC)^{i-1}(M - K).\end{aligned}\tag{2.53}$$

M again being the deadbeat observer gain, and K the steady state Kalman filter gain. If the order p of the ARMAX model is taken to be large enough then the observer gain M tend to approach the Kalman gain K while $\bar{A}^p = (A + MC)^p \approx 0$. Given that, the coefficients γ_i vanish and the ARMAX model reverts back to being an ARX model. This basically lies the foundation to the OKID algorithm with residual whitening developed in [63].

2.9. The observer/Kalman filter identification (OKID)

This is an observer based method of identifying a linear state-space model [57]. Observer Markov parameters are estimated first from a set of input/output data. Following that, a recovery of the original system Markov parameters and finally an eigensystem realization algorithm (ERA) to obtain a minimal balanced realization of the system is presented. Given that, the gain matrix can be computed afterwards. OKID relies on the presence of an LTI optimal one step ahead state observer, such observer is achieved under the condition of zero mean and white process and measurement noises, and is known as the steady state Kalman filter as per [64]. The idea behind the OKID follows the same scheme as previously explained in the observer based transformation of the state-space equations. The observer order is normally chosen much larger than the system order to ensure that the Kalman filter reaches steady state and hence the equivalence between the predictor and system Markov parameters holds true. OKID has many variants: the standard one was reported in [60], OKID with deterministic projection (OKID/DP) in [65], a similar approach of this later algorithm is also presented in [66], the output only version of OKID O³KID in [67], and the most recent residual-based observer/Kalman filter identification (ROKID) in [68].

2.10. Eigensystem realization algorithm (ERA)

The algorithm was proposed to identify a state-space model given a set Markov parameters in [53, 69], and relies on the SVDs of the constructed Hankel matrices. The identified system is similar to the original one up to a similarity transformation. A big advantage of the ERA is that it gives a minimal balanced realization (both controllable and observable). Another method for realizing a state-space model is

to directly convert an I/O model (ARX, ARMAX, OE, etc) to state-space form, namely the conversion to the observable canonical form is straight forward since the I/O model coefficients appear explicitly in the system matrices. The drawback of this however is that it may lead to non-minimal realization [57], a system in the observability canonical form may not necessarily be controllable. A first step in the ERA is to construct the two Hankel matrices H_0 and H_1 populated by the sequence of identified system Markov parameters Y_k given in 2.34 where

$$\begin{aligned}
H_0 &= \begin{bmatrix} CB & CAB & \dots & CA^{N-1}B \\ CAB & CA^2B & \dots & CA^NB \\ \vdots & & & \vdots \\ CA^{p-1}B & \dots & & CA^{N+p-2}B \end{bmatrix} = \begin{bmatrix} Y_0 & Y_1 & \dots & Y_{N-1} \\ Y_1 & Y_2 & \dots & Y_N \\ \vdots & & \ddots & \vdots \\ Y_{p-1} & \dots & & Y_{N+p-2} \end{bmatrix} \in \mathbb{R}^{pn_y \times Nn_u}, \\
H_1 &= \begin{bmatrix} CAB & CA^2B & \dots & CA^NB \\ CA^2B & CA^3B & \dots & CA^{N+1}B \\ \vdots & & \ddots & \vdots \\ CA^pB & \dots & & CA^{N+p-1}B \end{bmatrix} = \begin{bmatrix} Y_1 & Y_2 & \dots & Y_N \\ Y_2 & Y_3 & \dots & Y_{N+1} \\ \vdots & & \ddots & \vdots \\ Y_p & \dots & & Y_{N+p-1} \end{bmatrix}. \quad (2.54)
\end{aligned}$$

Here N must be chosen such that $Nn_u \gg pn_y$ and p such that $pn_y \geq n_x$. A singular value decomposition is performed on H_0 to recover the observability and controllability matrices

$$H_0 = \begin{bmatrix} U_1 & U_2 \end{bmatrix} \begin{bmatrix} S_1 & 0 \\ 0 & S_2 \end{bmatrix} \begin{bmatrix} V_1^T \\ V_2^T \end{bmatrix} \simeq U_1 S_1 V_1^T, \quad (2.55)$$

that's the case if we discard the singular values of S_2 (the noise singular values), the observability and controllability matrix are calculated via

$$O_p = U_1 S_1^{1/2}, \quad (2.56)$$

$$C_p = S_1^{1/2} V_1^T. \quad (2.57)$$

The matrices C and B can be recovered directly from O_p and C_p as

$$C = O_p(1:n_y,:), \quad (2.58)$$

$$B = C_p(:, 1:n_u). \quad (2.59)$$

The state matrix is recovered employing the shift invariant property

$$H_1 = \begin{bmatrix} \bar{U}_1 & \bar{U}_2 \end{bmatrix} \begin{bmatrix} \bar{S}_1 & 0 \\ 0 & \bar{S}_2 \end{bmatrix} \begin{bmatrix} \bar{V}_1^T \\ \bar{V}_2^T \end{bmatrix} \simeq \bar{U}_1 \bar{S}_1 \bar{V}_1^T = O_p A C_p, \quad (2.60)$$

and hence

$$A = (O_p)^\dagger H_1 (C_p)^\dagger = S_1^{-1/2} U_1^T H_1 V_1 S_1^{-1/2}. \quad (2.61)$$

The transmission matrix is obtained directly from the estimated system Markov parameters

$$D = Y_{-1}. \quad (2.62)$$

The realized system is equivalent to the actual one up to a similarity transformation.

2.11. Compressive sensing

Classical system identification approaches are of an asymptotic nature, i.e, they are reliable when the availability of data is not an issue, however sometimes limited data records are available or the measurement process is costly, and hence the practitioner would like to infer the model from the smallest possible number of measurements, which apparently leaves us face to face with an ill-conditioned problem [70] and uniqueness issues. Motivated for this later problem, the field of compressive sensing (CS) [71, 72] proposes a paradigm under which a unique solution for under-determined problem can be recovered provided that the solution is sparse enough and the measurement matrix satisfies certain conditions. The CS problem can be seen as recovering a sparse signal $x \in \mathbb{R}^N$ satisfying $Ax = b \in \mathbb{R}^M$ where $A \in \mathbb{R}^{M \times N}$ is the measurement matrix with $M < N$ and $M \ll N$ most often. This is accomplished by special CS algorithms that ensure recovery success whenever the conditions for recovery are met.

Several CS algorithms have been proposed in literature, some of them are of ℓ_1 minimization nature since the ℓ_1 norm is a sparsity promoter [73] and provides tractable problems compared to its ℓ_0 counterpart (NP-hard). The ℓ_1 minimization problem and its variants appears in literature in different forms like the Lasso [73] and the basis pursuit (BP) [74]. Several other algorithms are of a greedy nature, like the orthogonal matching pursuit (OMP) and its extension to accommodate for block sparsity the Block Orthogonal matching pursuit (BOMP) [75, 76]. This algorithm relies on best subset selection based on greedy algorithms and it is a heuristic for solving the non-tractable problem given by

$$\begin{aligned} \min_x \quad & \frac{1}{2} \|Ax - b\|_2 \\ \text{s.t.} \quad & \|x\|_0 \leq t. \end{aligned} \quad (2.63)$$

2.12. Greedy algorithms

One approach to solving the sparse recovery problem is by iterating to approximate the support of the signal x and its corresponding components [77]. Basically at any step, one index is added to the support to render the mismatch to the measurement vector as small as possible. Such algorithms are denoted as Greedy algorithms. For many of those, remarkable recovery guarantees were successfully derived in the literature, sometimes even optimal ones.

2.12.1. The Orthogonal matching pursuit (OMP)

The orthogonal Matching Pursuit algorithm [78, 79, 80] adds at every iteration one entry to the support of $\hat{x}$, and hence takes k -iterations to recover a k -sparse signal $\hat{x}$. The support recovery is achieved by finding the column of A most correlated with the residual of measurements, those columns indices are added to the support T_i , finally the solution $\hat{x}_i$ is updated as the vector supported on T_i and which minimizes $\|y - Az\|_2$ (the least square solution) where $z \in \mathbb{R}^n$ and $\text{supp}(z) \in T_i$. The OMP steps are summarized in Table 2.1.

Table 2.1: The OMP algorithm.

The OMP algorithm for recovery of sparse signals	
Input: Compressed sensing matrix A , measurement vector y .	
Initial values: $\hat{x} = 0, r = y, T_0 = \emptyset, i = 0$.	
Iteration Step: Repeat until stopping criterion is met:	
$i := i + 1$	
$T_i \leftarrow T_{i-1} \cup \text{supp}H_1(A^T r)$	“add largest residual entry to the support”
$\hat{x}_i _{T_i} \leftarrow A_{T_i}^\dagger y$	“update the estimate of the signal”
$r \leftarrow y - A\hat{x}_i$	“update the residual of the measurements”
Output: $\hat{x}_i$.	

$H_k(x)$ with $k \in \{0, 1, \dots, n\}$ is the hard thresholding operator. $H_k : x \rightarrow H_k(x)$ associates to $x \in \mathbb{R}^n$ the vector $H_k(x) \in \mathbb{R}^n$ which is equal to x on the k -entries of x with largest magnitude and zero otherwise. The stopping criteria can either limit the number of iterations which is equal to the support of the solution $\hat{x}$ (or the sparsity level k) or ensure the error given by norm $\|y - A\hat{x}\|_p$ to be smaller than a considered tolerance. Although so simple, the algorithm suffers from a major drawback, a mistakenly added index to the support at one point (which might happen in general) stays there until the algorithm terminates [77]. A variant of the classical OMP is given in [81].

2.12.2. Compressive sampling matching pursuit (CoSaMP)

In order to tackle the previous drawback [82], the algorithm takes the expected sparsity as an additional input, it proceeds by enlarging the support like the conventional OMP, but unlike the OMP it will add at least k new entries. Subsequently, according to the support T_i of $z \in \mathbb{R}^n$, a solution is sought for $\|y - Az\|_2$, but this time only the k -largest coordinates of the minimizer are kept. The algorithm steps are summarized in Table 2.2.

Table 2.2: The CoSaMP algorithm.

The CoSaMP algorithm for recovery of sparse signals
Input: Compressed sensing matrix A , measurement vector y , sparsity level k .
Initial values: $\hat{x} = 0, r = y, T_0 = \emptyset, i = 0$.
Iteration Step: Repeat until stopping criterion is met:
$i := i + 1$
$T_i \leftarrow \text{supp}(\hat{x}_{i-1}) \cup \text{supp}H_{2k}(A^T r)$ “update the support”
$\hat{x}_i _{T_i} \leftarrow H_k(A_{T_i}^\dagger y)$ “update the estimate of the signal”
$r \leftarrow y - A\hat{x}_i$ “update the residual”
Output: $\hat{x}_i$.

2.12.3. The Block orthogonal matching pursuit (BOMP)

This is the block version of the orthogonal matching pursuit Algorithm (OMP), while instead of identifying single indices of the support, a whole block of indices is identified at each iteration. The BOMP working mechanism is pretty simple, it decides the contributing blocks by correlating the different blocks with the measurement vector and selecting the blocks with the highest correlation. When a block has been selected, its contributions is removed from the measurement vector via orthogonal projection, the process is repeated until a stopping criteria is satisfied. Conditions under-which the BOMP successfully recovers a block sparse signals x were reported in [76] (the exact recovery conditions). The BOMP steps are listed in table 2.3.

Table 2.3: The BOMP algorithm.

The BOMP algorithm for recovery of block-sparse signals
Input: Matrix A , measurements y , block size m , stopping criteria.
Initial values: $r_0 = b$, $x_0 = 0$, $T_0 = \emptyset$, $i = 0$.
repeat
1. match: $h_l = A_l^T r_i, l = 1, 2, \dots, P$.
2. identify support: $\lambda = \operatorname{argmax}_l \ h_l\ _2$.
3. update the support: $T_{i+1} = T_i \cup \lambda$.
4. update signal estimate:
$x_{i+1} = \operatorname{argmin}_{s: \operatorname{supp}(s) \subseteq T_{i+1}} \ y - As\ _2$, where $\operatorname{supp}(s)$ indicates the blocks on which s may be non-zero.
5. update residual estimate: $r_{i+1} = y - Ax_{i+1}$.
6. increase index i by 1.
until stopping criteria holds
output: $\hat{x} = x_i$.

2.12.4. Iterative hard thresholding (IHT)

This algorithm was reported in [83], it makes use of the hard thresholding operator H_k , the equation $Ax = b$ is to be transformed to $A^T Ax = A^T b$, which amounts to finding the fixed points for the mapping

$$x \rightarrow (I - A^T A)x + A^T b. \quad (2.64)$$

Hence, a classical way is to iterate the following

$$\hat{x}_i = (I - A^T A)\hat{x}_{i-1} + A^T b \quad (2.65)$$

$$= \hat{x}_{i-1} + A^T (y - A\hat{x}_{i-1}). \quad (2.66)$$

The iterative hard thresholding method is basically doing this only by employing the hard thresholding operator H_k , with the algorithm summarized in Table 2.4.

Table 2.4: The IHT algorithm.

The IHT algorithm for recovery of sparse signals
Input: Compressed sensing matrix A , measurement vector y , sparsity level k .
Initial values: $\hat{x} = 0, i = 0$.
Iteration Step: Repeat until stopping criterion is met
$i := i + 1$.
$\hat{x}_i = H_k(\hat{x}_{i-1} + A^T (y - A\hat{x}_{i-1}))$ “update the signal estimate”
Output: $\hat{x}_i$.

2.13. Basis pursuit denoising (BPDN)

The ℓ_1 minimization problem forms the backbone to the theory of compressive sensing, its geometrical properties permits recovery guarantees, plus stability and robustness extensions to accommodate for noise. It is formulated as a convex optimization problem which can be solved by using any general purpose solver. Besides that, several variants to accommodate for different settings also exist. Recall the previous under-determined problem $Ax = b$. The problem of recovering a sparse solution can be formulated by an ℓ_0 norm minimization

$$\begin{aligned} \min_x \quad & \|x\|_0 \\ \text{s.t.} \quad & b = Ax. \end{aligned} \tag{2.67}$$

However, this problem is NP-hard and hence non-tractable. Fortunately, due to some results from CS, it has been shown that under some conditions, solving the ℓ_1 minimization convex tractable version of the previous problem yields the exact same solution as mentioned in [71, 84]

$$\begin{aligned} \min_x \quad & \|x\|_1 \\ \text{s.t.} \quad & b = Ax. \end{aligned} \tag{2.68}$$

The ℓ_1 minimization problem, which is much easier to solve compared to its ℓ_0 counterpart, is known as basis pursuit (BP) [74] and can be solved via linear programming [70]. If the data is noise infected, the equality constraint can be relaxed by reformulating the problem

$$\begin{aligned} \min_x \quad & \|x\|_1 \\ \text{s.t.} \quad & \|Ax - b\|_2 \leq \varepsilon, \end{aligned} \tag{2.69}$$

where the constant ε is the noise level in data. This problem in particular is known as the basis pursuit denoising (BPDN), and can be thought of to be noise relaxed version of the BP algorithm. The BPDN problem can as well be written in the regularized form

$$\hat{x} = \arg \min_x \|Ax - b\|_2 + \lambda \|x\|_1, \tag{2.70}$$

where λ is a hyper parameter symbolizing the trade-off between the degree of fit and sparsity of the solution. The BPDN problem is closely related to LASSO (least absolute shrinkage and selection operator) problem. LASSO was developed by the statistics community, while BPDN was developed by the signal processing commu-

nity [73]. The LASSO takes the form

$$\begin{aligned} \min_x \quad & \frac{1}{2} \|Ax - b\|_2 \\ \text{s.t.} \quad & \|x\|_1 \leq t. \end{aligned} \quad (2.71)$$

This is the constrained form of the LASSO, the regularized form is given by

$$\hat{x} = \arg \min_x \frac{1}{2} \|Ax - b\|_2 + \lambda \|x\|_1. \quad (2.72)$$

For a proper choice of ε, λ and t , the solutions of the problems in Eqs.(2.69), (2.70) and (2.71) coincide. Another optimization problem that takes the form of the ℓ_1 minimization is Dantzig selector [85]. In general, the BOMP algorithm and the BPDN don't give the same solution. The BPDN problem is a general convex quadratic program and can be solved via general convex optimization solvers [86, 87] that involves quadratic programming techniques. For solvers of variety of ℓ_1 minimization problems refer to [88].

2.14. Iterative reweighting

The idea of iterative reweighting was first presented in [89], it was shown through numerical experimenting that the reweighted ℓ_1 minimization does outperform the unweighted ℓ_1 minimization in several situations [89, 90, 91, 92] and enhances the sparsity quite significantly. Iterative reweighting can be used generally to refine sparsity promoting problems involving sum of norms regularization terms. It was used in [93] for refining state estimation problem in the presence of abrupt changes using sum of norms regularization, also in [94] to enhance ARX-model segmentation via sparse optimization. The weights are basically applied at the regularization term and are updated iteratively via the usage of the current estimate, the ℓ_1 minimization problem takes the form

$$\begin{aligned} \min_x \quad & \|x\|_1 \\ \text{s.t.} \quad & Ax = b \end{aligned} \quad (2.73)$$

where as the reweighed ℓ_1 minimization is given by

$$\begin{aligned} \min_x \quad & \|W^k x\|_1 \\ \text{s.t.} \quad & Ax = b \end{aligned} \quad (2.74)$$

with

$$\begin{aligned} W^k &= \text{diag}(w^k), \\ w^k &= (w_1^k, \dots, w_n^k) \in \mathbb{R}_+^n, \end{aligned} \quad (2.75)$$

w^k is the weight vector determined by the solution of the current iterate $x^k = (x_1^k, \dots, x_n^k)^T \in \mathbb{R}^n$. The solution to the ℓ_1 minimization problem weighted with W^k yields x^{k+1} , from which w^{k+1} is computed as shown in [95]. The existing iteratively reweighted ℓ_1 minimization is based on merit function or support set, from which the weights are derived as pointed out in [96]. The function $G_\varepsilon(x) = \sum_j \log(|x_j| + \varepsilon)$ was used in [97] to design the focus algorithm and in [89] to design reweighted ℓ_1 minimization for sparse signal reconstruction. The weights in such algorithms take the famous form

$$w_j^{(i+1)} = \frac{1}{\varepsilon + |x_j^{(i)}|}. \quad (2.76)$$

Similarly iterative reweighting can be adapted to any sum of norms regularization problem

$$\min_x \|AX - b\| + \lambda \sum_j w_j \|x_j - x_{j-1}\|_{reg}, \quad (2.77)$$

where $X = \begin{bmatrix} x_1^T & x_2^T & \dots & x_n^T \end{bmatrix}^T$ and $\|\bullet\|_{reg}$ is the norm used for regularization, and could be any vector norm $(\ell_1, \dots, \ell_p)$, but we emphasize that it must be a sum of norms and not squared norms which is the Tychonov regularization [94]. The weights are updated via

$$w_j^{(i+1)} = \frac{1}{\varepsilon + \|x_j^i - x_{j-1}^i\|_{reg}}, \quad (2.78)$$

where ε is a positive parameter determining the maximum weight and to prevent division by zero in case the differences are zero, in general is chosen in the range $[10^{-2}, 10^{-4}]$. We emphasize the difference of using an ℓ_1 regularization or any other type of sum of norms regularization, like the euclidean norm for instance (ℓ_2). With ℓ_1 regularization, a time-varying model is obtained where all the components of $x(j)$ change less frequently and independently of each other. On the other hand, when using sums of norms (ℓ_2 or ℓ_F for matrices), then basically the components change infrequently but they do it together, in a grouped fashion. That's why the sum of norms regularization is denoted as the Group Lasso [98] in the statistics community.

2.15. Recovery guarantees

The compressive sensing theory guarantees a unique solution for underdetermined systems provided that model assumptions are imposed on the signal to be recovered [70]. Recovery guarantees and conditions for such underdetermined system in the context of CS were stipulated through many well established concepts, and this chapter is meant to detail them. For now, suppose we are exposed to the system of equations

$$y = Ax, \quad (2.79)$$

where $A \in \mathbb{R}^{p \times n}$ with $p \ll n$, and suppose also $x \in \mathbb{R}^n$ is an k -sparse signal, meaning only k out of its n entries are non-zero.

2.15.1. Spark

Let's first define the set of all k -sparse vectors (at most k -entries of the vector are non-zero) by

$$\Sigma_k = \{x : \|x\|_0 \leq k\}. \quad (2.80)$$

Consider also the nullity of A :

$$N(A) = \{z : Az = 0\}. \quad (2.81)$$

The set of all sparse signals can be recovered from measurements Ax with an immediate requirement that for every $x, x' \in \Sigma_k$, $Ax \neq Ax'$, otherwise the two vectors (x, x') will be non-distinguishable from measurements y .

$$Ax = Ax' \Leftrightarrow A(x - x') = 0. \quad (2.82)$$

In general, if $x, x' \in \Sigma_k$, then $(x - x') \in \Sigma_{2k}$, we conclude that all the k -sparse vectors are uniquely represented by the measurement matrix A if and only if $N(A)$ contains no vectors in Σ_{2k} [99]. This condition is characterized by the following concept from [99].

Definition 2.15.1 [100] *The spark of a given matrix A is the smallest number of columns of A that are linearly dependent.*

Theorem 2.15.1 [101] *For any vector $y \in \mathbb{R}^p$, there exist at most one signal $x \in \Sigma_k$ such that $y = Ax$ if and only if $\text{spark}(A) > 2k$.*

Since $\text{spark}(A) \in [2, p+1]$, and according to theorem (2.15.1), this introduces the additional requirement $p \geq 2k$ for uniqueness.

In general, $\text{spark}(A) \neq \text{rank}(A) + 1$. However for matrices in general positions like randomly generated matrices (all entries of A are i.i.d random variables) this holds [102], and for such matrices a sufficient recovery condition is $p \geq 2k$. For full row rank fat matrices $A \in \mathbb{R}^{p \times n}$ ($p \ll n$), any $p+1$ columns of A are linearly dependant and $\text{spark}(A) \leq p+1 = \text{rank}(A) + 1$. *Rank* is easy to compute where as *spark* requires solving a combinatorial search problem as hard as the original NP-problem since one must verify that all sets of columns of a certain size are linearly independent [99]. The following algorithm provides a procedure for the spark calculation,

Algorithm 1 Spark calculation

- ① Look for all combinations of r -columns $r = 2, \dots, p$ of the matrix A .
 - ② If for any of the combinations we get linear dependency, then save the number of vectors in that combination in a set S .
 - ③ Take the spark to be the smallest number of the previous set $\text{spark}(A) = \min(S)$.
-

2.15.2. Null space property (NSP)

The spark gives a complete characterization of sparse recovery conditions when exact sparse vectors are sought. However, when dealing with approximately sparse vectors more conditions must be imposed, in a sense to ensure that $N(A)$ does not contain too compressible vectors beside sparse vectors [99]. Suppose that $\Lambda \in \{1, 2, \dots, n\}$ is a subset of indices and let $\Lambda^c \in \{1, 2, \dots, n\} \setminus \Lambda$ be its complement. $x_\Lambda \in \mathbb{R}^n$ is the vector obtained by setting its entries indexed by Λ^c to zero, in a similar manner, A_Λ is obtained by setting the columns indexed by Λ^c to zero. Next we introduce the notion of the null space property (NSP) [103] as follows:

Definition 2.15.2 [77] *Let $A \in \mathbb{R}^{p \times n}$ and let $k \in \{1, \dots, n\}$, then A is said to have the Null Space Property (NSP) of order k if*

$$\|v_\Lambda\|_1 < \|v_{\Lambda^c}\|_1 \text{ for all } v \in \ker(A) \setminus \{0\} \text{ and all } \Lambda \in \{1, 2, \dots, n\} \text{ with } |\Lambda| \leq k. \quad (2.83)$$

Some others define the condition slightly different: $\|v_\Lambda\|_1 < \gamma \|v_{\Lambda^c}\|_1$ for some γ and hence A is said to be $NSP(k, \gamma)$.

The condition in Eq.(2.83) promotes for the fact that vectors in the null space of A should not be too concentrated on a small subset of indices, i.e, well spread and not supported on a set of small size. For example, if $v \in \mathbb{R}^n \setminus \{0\}$ is k -sparse and hence $\Lambda = \text{supp}(v)$, then $\|v_{\Lambda^c}\|_1 = 0$ which indicates from Eq.(2.83) that $\|v_\Lambda\|_1 = 0$ as well, so v can not lie in the kernel of A . Thus, if a matrix A satisfies the NSP then the only k -sparse vector in $N(A)$ is $v = 0$. A well-known relaxation of the NP-hard ℓ_0 problem given by

$$\begin{aligned} \min_x \quad & \|x\|_0 \\ \text{s.t.} \quad & Ax = b, \end{aligned} \quad (2.84)$$

takes the basis pursuit form given by

$$\begin{aligned} \min_x \quad & \|x\|_1 \\ \text{s.t.} \quad & Ax = b, \end{aligned} \quad (2.85)$$

Theorem 2.15.2 [77] *Let $A \in \mathbb{R}^{p \times n}$ and let $k \in \{1, \dots, n\}$, then every k -sparse vector x is the unique solution of the relaxed tractable basis pursuit problem (BP) given in Eq.(2.85) if and only if A has NSP of order k ($NSP(k, 1)$).*

Theorem (2.15.2) explicitly tells us that the solution to the problem in Eq.(2.84) can be obtained by solving the problem in Eq.(2.85) if A satisfies the NSP property of order k and if $y \in \mathbb{R}^p$ is such that there is a k -sparse solution to the equation $y = Ax$.

2.15.3. The restricted isometric property (RIP)

Even-though the NSP provides necessary and sufficient conditions for sparse recovery guarantees, it does not take noise into consideration, plus constructing matrices that obeys the NSP is hard in general, therefore it is recommended to consider stronger conditions to account for that. For that, a sufficient condition called the restricted isometric property was introduced by Candes and Tao in [104] and its important role was also established in CS. The following definition was borrowed from [70],

Definition 2.15.3 *A matrix A is said to satisfy the RIP of order k and isometry constant $\delta_k := \delta_k(A) \in (0, 1)$ if*

$$(1 - \delta_k) \|x\|_2^2 \leq \|Ax\|_2^2 \leq (1 + \delta_k) \|x\|_2^2 \quad (2.86)$$

holds for all k -sparse vectors $x \in \mathbb{R}^n$

If A satisfies the RIP of order $2k$ for a sufficiently small isometry constant δ_{2k} , then it is possible to recover any k -sparse signal x from the equation $y = Ax$ through a convex tractable ℓ_1 minimization [71, 74, 104]. An A satisfying the condition in Eq.(2.86) means A acts nearly isometrically when confined to only vectors from Σ_k , and therefore distance preserving. Clearly, if A has $(2k, \gamma)$ -RIP then $\text{spark}(A) \geq 2k + 1$. Similarly to the Spark, checking whether a matrix satisfies RIP is a combinatorial problem. Furthermore, a matrix satisfying RIP with sufficiently high order and a constant small enough is indeed a sufficient condition for the null space property [77].

Theorem 2.15.3 *Let $A \in \mathbb{R}^{p \times n}$ and let k be a natural number with $k \leq \frac{n}{2}$. If $\delta_{2k}(A) \leq \frac{1}{3}$ then A has the NSP of order k .*

Combining the condition given by Eq.(2.86) and theorem (2.15.3) yields the corollary below:

Corollary 2.15.3.1 [77] *Let $A \in \mathbb{R}^{p \times n}$ and let k be a natural number with $k \leq \frac{n}{2}$. If $\delta_{2k}(A) \leq \frac{1}{3}$ then every k -sparse vector x is the unique solution of the problem in Eq.(2.85) with $y = Ax$.*

Again we state that computing RIP is NP-hard problem of combinatorial but trivial randomly constructed matrices satisfy RIP with overwhelming probability, like matrices populated by i.i.d random variables with continuous distributions enjoy small restricted isometry constants and, hence satisfy the null space property.

2.15.4. The coherence

The spark, NSP, and the RIP all provide guarantees for the recovery of k -sparse signals, verifying that a matrix satisfies such conditions is of combinatorial computational complexity since each time we have to consider $\binom{n}{k}$ sub-matrices, which calls for the need for a measure which is easily verifiable. The coherence of a matrix is one such measure that can be easily computed and ensures recovery guarantees as pointed out in [99].

Definition 2.15.4 [99, 78] *The coherence of a matrix A , $\mu(A)$, is the largest absolute inner product between any two columns a_i, a_j of A*

$$\mu(A) = \max_{1 \leq i < j \leq n} \frac{|\langle a_i, a_j \rangle|}{\|a_i\|_2 \|a_j\|_2}. \quad (2.87)$$

The coherence characterizes the dependence between columns of A , for unitary matrices $\mu(A) = 0$. Successful recovery requires small $\mu(A)$ as it coincides to unitary matrices. One can show the coherence lies in the interval $\mu(A) \in \left[\sqrt{\frac{n-p}{p(n-1)}}, 1 \right]$, where the lower bound is known as the Welch bound [105]. Observe that if $n \gg p$ we get the approximation $\mu(A) \geq \frac{1}{\sqrt{p}}$. As expected, randomly generated matrices are characterized by having small coherence, furthermore the Welch bound is not attainable in case $n \gg p$. This can only be achieved if $n \leq \frac{m(m+1)}{2}$ in the real case and if $n \leq m^2$ in the complex case. The coherence can always be related to the Spark, NSP, and RIP. For example, the coherence and spark properties of a matrix can be related by employing the Gershgorin circle theorem [97], and hence the following lemma:

Lemma 2.15.1 [99] *For any matrix A ,*

$$\text{spark}(A) \geq 1 + \frac{1}{\mu(A)}. \quad (2.88)$$

Combining theorem 2.15.1 and lemma 2.15.1 gives rise to the following uniqueness guarantees:

Theorem 2.15.4 [101] *If $k < \frac{1}{2} \left(1 + \frac{1}{\mu(A)} \right)$, then for each measurement vector $y \in \mathbb{R}^p$ there exists at most one signal $x \in \sum_k$ such that $y = Ax$.*

Theorem 2.15.4 when taking the Welch bound on the coherence into consideration provides an upper bound on the level of sparsity k that guarantees uniqueness using the coherence $k = O(\sqrt{p})$. Also the coherence and the RIP are related through.

Lemma 2.15.2 [99] *If A has unit-norm columns and coherence $\mu = \mu(A)$, then A satisfies the RIP of order k with $\delta = k\mu$ for all $k < \frac{1}{\mu}$.*

2.15.5. Block-coherence and sub-coherence

When the solution to the underdetermined system manifests sparsity in a block fashion, the block coherence is considered instead as a recovery guarantee. Assume $A \in \mathbb{R}^{p \times n}$ is a concatenation of P block matrices $A_i \in \mathbb{R}^{p \times m}$ as

$$A = \begin{bmatrix} A_1 & \dots & A_i & \dots & A_P \end{bmatrix}, \quad (2.89)$$

where $n = Pm$. Assume also that there is a K -block sparse signal x that satisfies $Ax = b$, if matrix A has columns of unit norm, the block coherence of matrix A be given by [75, 76, 106]

$$\mu_{block}(A) := \max_{i,j \neq i} \|A_i^T A_j\|_2, \quad (2.90)$$

where $\|A\|_2$ is the spectral norm of A . If $m = 1$, we revert back to the conventional coherence definition [79, 107]

$$\mu(A) := \max_{i,j \neq i} \|a_i^T a_j\|_2. \quad (2.91)$$

$\{a_i\}_{i=1}^P$ being the columns of matrix A . Basically μ_{block} characterizes the inter-block properties, where as the intra-block properties are characterized via the sub-coherence [70],

$$\mu_{sub-block}(A) := \max_k \max_{i,j \neq i} \|a_{k_i}^T a_{k_j}\|_2, \quad (2.92)$$

where a_{k_i}, a_{k_j} are columns of the matrix-block A_k .

2.15.6. Recovery guarantees For BOMP

A sufficient condition that guarantees the recovery of K -block sparse signal from $Ax = b$ through BOMP was derived in [76]. The condition relies on both block coherence μ_{block} and sub-block coherence defined previously.

Theorem 2.15.5 *If x is block-sparse with K non-zero blocks of length m , then BOMP will recover x from measurements $b = Ax$ if*

$$Km < \mu_T, \quad (2.93)$$

where

$$\mu_T = \frac{1}{2} \left(\mu_{block}^{-1} + m - (m-1) \frac{\mu_{sub-block}}{\mu_{block}} \right). \quad (2.94)$$

If $m = 1$, the condition in Eq.(2.94) revert back to the recovery condition of the OMP [79], i.e., $K < \frac{1}{2} (\mu^{-1} + 1)$. Theorem (2.15.5) indicates that limited sparsity level based signals can be retrieved via BOMP, furthermore, the higher μ_T (this correlates with smaller μ_{block}) the higher the K for the retrieved signals.



3. LTV-MODEL TRANSFORMATIONS AND REALIZATION

In this chapter, in Section 3.1 we present a time-varying realization algorithm of the SLS model in Eq.(1.3-1.4), from a set of generalized Markov parameters. Section 3.2, on the other hand, discusses an observer-based transformation that allows us to obtain a time-varying ARX model starting from a time-varying state-space model. The back transformation to the LTV model starting from the ARX model is presented in Section 3.3. In concluding the theoretical part, section 3.4 discusses the effect of process and measurement noises being injected on the observer-based transformation. Section 3.5 supplements the theoretical findings of this chapter via a numerical example.

3.1. SLS realization from generalized Hankel matrix pairs

In this section, we consider the SLS model in Eqs.(1.3)–(1.6) in a noiseless setup, *that is*, we assume $w(k)$ and $e(k)$ to be zero for all k . From Eqs.(1.3)–(1.4), we calculate the system output to the initial state $x(i)$ and an input sequence as

$$y(k) = C(k)\Phi(k, i)x(i) + \sum_{j=i}^k h(k, j)u(j) \quad i \leq k \leq N, \quad (3.1)$$

given that $i \geq 1$. The *generalized system Markov parameters* and the *state transition matrix* associated with the *homogeneous part* of Eq.(1.3) are given respectively by

$$h(k, i) = \begin{cases} C(k)\Phi(k, i+1)B(i) & k > i, \\ D(k) & k = i, \\ 0 & k < i, \end{cases} \quad (3.2)$$

and

$$\Phi(k, i) = \begin{cases} A(k-1) \cdots A(i) & k > i, \\ I_n & k = i, \end{cases} \quad (3.3)$$

where $I_n \in \mathbb{R}^{n \times n}$ is the identity matrix. The input-output behavior of the SLS model in Eqs.(1.3)–(1.6) may equally be described by the generalized system Markov parameters that are presented in Eq.(3.2).

The description of an LTV system in terms of the state-space matrices is not unique. In fact, two realizations $\{A, B, C, D\}$ and $\{\tilde{A}, \tilde{B}, \tilde{C}, \tilde{D}\}$ have the same generalized system Markov parameters if they are *topologically equivalent*, *i.e.*, there

exists a bounded $T(k) \in \mathbb{R}^{n \times n}$ with a bounded inverse $T^{-1}(k)$ such that for all $k \in \mathbb{N}$,

$$\begin{aligned}\tilde{A}(k) &= T(k+1)A(k)T^{-1}(k), \\ \tilde{B}(k) &= T(k+1)B(k), \\ \tilde{C}(k) &= C(k)T^{-1}(k), \\ \tilde{D}(k) &= D(k).\end{aligned}$$

Here, the bounded matrix $T(k)$ with bounded inverse $T^{-1}(k)$ is called a *Lyapunov transformation*. As far as the input-output behavior of an LTV system is concerned, it suffices to estimate the generalized system Markov parameters from the input-output measurements. We defer this estimation problem to a later section; but, address the realization problem from generalized Hankel matrix pairs here.

Let $q, r \in \mathbb{N}$ be two given numbers. For each given triplet $(k, q, r) \in \mathbb{N}^3$ with $k > r$, we generate nested sequences of the *generalized Hankel matrices* by the formula

$$\mathcal{H}_{qr}(k) = \begin{bmatrix} h(k, k-1) & \cdots & h(k, k-r) \\ & \ddots & \vdots \\ & \cdots & h(k+q-1, k-r) \end{bmatrix}. \quad (3.4)$$

From each pair $\mathcal{H}_{qr}(k)$ and $\mathcal{H}_{qr}(k+1)$, we will extract a realization that is topologically equivalent to $\mathcal{P}_{\varphi(k)}$ assuming that the SLS model in Eqs.(1.3)–(1.4) is asymptotically stable and *uniform*. The uniformity means that $\mathcal{P}_{\varphi(k)}$ is bounded, *uniformly observable* and *uniformly controllable*. Recall that the SLS in Eqs.(1.3)–(1.4) is uniformly controllable if there exist a $\kappa_c \in \mathbb{N}$ and $\alpha_0, \alpha_1, \beta_0, \beta_1 > 0$ such that for all $k > \delta_c$,

$$G_c(k, \kappa_c) > 0, \quad (3.5)$$

$$\alpha_0 I_n \leq G_c^{-1}(k, \kappa_c) \leq \alpha_1 I_n, \quad (3.6)$$

$$\beta_0 I_n \leq \Phi^T(k, k - \kappa_c) G_c^{-1}(k, \kappa_c) \Phi(k, k - \kappa_c) \leq \beta_1 I_n, \quad (3.7)$$

where X^T denotes the transpose of a given matrix X and

$$G_c(k, \kappa_c) = \sum_{j=k-\kappa_c}^{k-1} \Phi(k, j+1) B(j) B^T(j) \Phi^T(k, j+1). \quad (3.8)$$

Likewise, the SLS in Eqs.(1.3)–(1.4) is uniformly observable if there exist a $\kappa_o \in \mathbb{N}$ and $\alpha'_0, \alpha'_1, \beta'_0, \beta'_1 > 0$ such that for all $k \in \mathbb{N}$,

$$G_o(k, \kappa_o) > 0, \quad (3.9)$$

$$\alpha'_0 I_n \leq G_o^{-1}(k, \kappa_o) \leq \alpha'_1 I_n, \quad (3.10)$$

$$\beta'_0 I_n \leq \Phi^T(k + \kappa_o, k) G_o^{-1}(k, \kappa_o) \Phi(k + \kappa_o, k) \leq \beta'_1 I_n, \quad (3.11)$$

where

$$G_o(k, \kappa_o) = \sum_{j=k}^{k+\kappa_o-1} \Phi^T(j, k) C^T(j) C(j) \Phi(j, k). \quad (3.12)$$

The controllability and the observability Gramians defined by Eq.(3.8) and Eq.(3.12) are instrumental in studying balanced truncation of switched linear systems [108]. The generalized Hankel matrix in Eq.(3.4) can be factorized as in [109],

$$\mathcal{H}_{qr}(k) = \mathcal{O}_q(k) \mathcal{R}_r(k-1), \quad k > 1, \quad (3.13)$$

where $\mathcal{O}_q(k)$ and $\mathcal{R}_r(k-1)$ are the *extended observability matrix* and the *extended controllability matrix* defined by

$$\mathcal{O}_q(k) = \begin{bmatrix} C(k) \\ C(k+1)\Phi(k+1, k) \\ \vdots \\ C(k+q-1)\Phi(k+q-1, k) \end{bmatrix} \in \mathbb{R}^{qp \times n}, \quad (3.14)$$

and

$$\mathcal{R}_r(k-1) = [B(k-1) \ \cdots \ \Phi(k, k-r+1)B(k-r)]. \quad (3.15)$$

Note from Eqs.(3.12), (3.14), and (3.9)

$$\begin{aligned} \mathcal{O}_q^T(k) \mathcal{O}_q(k) &\geq \mathcal{O}_{\kappa_o}^T(k) \mathcal{O}_{\kappa_o}(k) \\ &= G_o(k, \kappa_o) > 0, \end{aligned} \quad (3.16)$$

whenever $q \geq \kappa_o$ and from Eqs.(3.8), (3.15), and (3.5)

$$\begin{aligned} \mathcal{R}_r(k-1) \mathcal{R}_r^T(k-1) &\geq \mathcal{R}_{\kappa_c}(k-1) \mathcal{R}_{\kappa_c}^T(k-1) \\ &= G_c(k - \kappa_c, k) > 0, \end{aligned} \quad (3.17)$$

whenever $r \geq \kappa_c$. Thus, from the inequality

$$\text{rank}(AB) \leq \min \{\text{rank}(A), \text{rank}(B)\}$$

for two given matrices A and B of compatible dimensions and the chain of equalities in Eq.(3.16) and Eq.(3.17). If $q \geq \kappa_o$ and $r \geq \kappa_c$ we get for the ranks of the products

in Eq.(3.13) as

$$\text{rank}(\mathcal{O}_q(k)) = \text{rank}(\mathcal{R}_r(k-1)) = n. \quad (3.18)$$

From the Sylvester inequality in [54] applied to the product in Eq.(3.13), we obtain

$$\begin{aligned} \text{rank}(\mathcal{O}_q(k)) + \text{rank}(\mathcal{R}_r(k-1)) - n &\leq \text{rank}(\mathcal{H}_{qr}(k)) \\ &\leq \min \{ \text{rank}(\mathcal{O}_q(k)), \text{rank}(\mathcal{R}_r(k-1)) \}. \end{aligned} \quad (3.19)$$

Hence, from Eqs.(3.18) and (3.19) we have $\text{rank}(\mathcal{H}_{qr}(k)) = n$ if $q \geq \kappa_o$ and $r \geq \kappa_c$.

Let J_U and J_D be the shift matrices of row-block up and row-block down, and J_L and J_R be the column-block left and column-block right shift matrices defined by

$$J_U = \begin{bmatrix} O_{(q-1)p \times p} & I_{(q-1)p} \end{bmatrix}, \quad (3.20)$$

$$J_D = \begin{bmatrix} I_{(q-1)p} & O_{(q-1)p \times p} \end{bmatrix},$$

$$J_L = \begin{bmatrix} 0_{m \times (r-1)m} \\ I_{(r-1)m} \end{bmatrix}, \quad (3.21)$$

$$J_R = \begin{bmatrix} I_{(r-1)m} \\ 0_{m \times (r-1)m} \end{bmatrix},$$

where $O_{m \times n}$ denotes the n by m matrix of zeros. Then, from the factorization formula in Eq.(3.13) with k and $k+1$ plugged in, and Eqs.(3.20)–(3.21), we derive the following relations:

$$J_U \mathcal{O}_q(k) = (J_D \mathcal{O}_q(k+1)) A(k), \quad (3.22)$$

$$A(k) (\mathcal{R}_r(k-1) J_R) = \mathcal{R}_r(k) J_L. \quad (3.23)$$

Hence, from Eq.(3.22) or Eq.(3.23) we retrieve $A(k)$ from either of the formulas

$$\begin{aligned} A(k) &= (J_D \mathcal{O}_q(k+1))^\dagger J_U \mathcal{O}_q(k), \\ &= \mathcal{R}_r(k) J_L (\mathcal{R}_r(k-1) J_R)^\dagger, \end{aligned} \quad (3.24)$$

where $X^\dagger$ denotes the unique left or right-inverse of a given full-rank matrix X defined as $(X^T X)^{-1} X^T$ or $X^T (X X^T)^{-1}$ if $q \geq \kappa_o$ and $r \geq \kappa_c$. Notice that $\mathcal{H}_{qr}(k+1)$ is populated by discarding the first block-column and the first block-row of $\mathcal{H}_{qr}(k)$ and adding $q+r-1$ new generalized Markov parameters. If M consecutive realizations

are extracted, the number of the generalized Markov parameters required equals to $M(q+r) + qr$, since the feedthrough parameter $D(k)$ is derived from the relation $D(k) = h(k, k)$ for each k . When k is sparsely distributed, this number could be as large as $M(qr + q + r)$. From $k > r \geq \kappa_c$ and $q \geq \kappa_o$, a long waiting time may be necessary unless both κ_c and κ_o are small. Now, it remains to determine $\mathcal{O}_q(k)$ and $\mathcal{O}_q(k+1)$. We apply singular value decomposition (SVD) to $\mathcal{H}_{qr}(k)$ and $\mathcal{H}_{qr}(k+1)$

$$\begin{aligned}\mathcal{H}_{qr}(k) &= U_q(k)\Sigma(k)V_r^T(k), \\ \mathcal{H}_{qr}(k+1) &= U_q(k+1)\Sigma(k+1)V_r^T(k+1),\end{aligned}\tag{3.25}$$

and let

$$\begin{aligned}\hat{\mathcal{O}}_q(k) &= U_q(k)\Sigma^{1/2}(k), \\ \hat{\mathcal{R}}_r(k-1) &= \Sigma^{1/2}(k)V_r^T(k), \\ \hat{\mathcal{O}}_q(k+1) &= U_q(k+1)\Sigma^{1/2}(k+1), \\ \hat{\mathcal{R}}_r(k) &= \Sigma^{1/2}(k+1)V_r^T(k+1).\end{aligned}\tag{3.26}$$

The equations above provide estimates of $\mathcal{O}_q(k)$ and $\mathcal{O}_q(k+1)$ up to similarity transformations $T(k)$ and $T(k+1)$, *i.e.*, $\mathcal{O}_q(k) = \hat{\mathcal{O}}_q(k)T(k)$ and $\mathcal{O}_q(k+1) = \hat{\mathcal{O}}_q(k+1)T(k+1)$. Let

$$\hat{A}(k) = (J_D \hat{\mathcal{O}}_q(k+1))^\dagger J_U \hat{\mathcal{O}}_q(k),\tag{3.27}$$

so that

$$\hat{A}(k) = T(k+1)A(k)T^{-1}(k).\tag{3.28}$$

The estimation of $B(k)$ and $C(k)$ is in order. To this end, let

$$\begin{aligned}J_C &= \begin{bmatrix} I_p & O_{p \times (q-1)p} \end{bmatrix}, \\ J_B &= \begin{bmatrix} I_m \\ 0_{(r-1)m \times m} \end{bmatrix}.\end{aligned}\tag{3.29}$$

Then, from Eqs.(3.14) and (3.15)

$$\begin{aligned}C(k) &= J_C \mathcal{O}_q(k), \\ B(k) &= \mathcal{R}_r(k)J_B.\end{aligned}$$

As for the estimates of $C(k)$ and $B(k)$, we set

$$\begin{aligned}\hat{C}(k) &= J_C \hat{\mathcal{O}}_q(k), \\ \hat{B}(k) &= \hat{\mathcal{R}}_r(k) J_B.\end{aligned}\tag{3.30}$$

Then, from $\hat{\mathcal{R}}_r(k) = T(k+1)\mathcal{R}_r(k)$

$$\begin{aligned}\hat{C}(k) &= C(k)T^{-1}(k), \\ \hat{B}(k) &= T(k+1)B(k).\end{aligned}\tag{3.31}$$

Setting $\hat{D}(k) = h(k, k)$, we see from Eq.(3.28) and Eq.(3.31)) that $\{\hat{A}(k), \hat{B}(k), \hat{C}(k), \hat{D}(k)\}$ is topologically equivalent to $\mathcal{P}_{\varphi(k)}$. The above derivations are summarized in the form of a subspace-based realization algorithm presented as follows:

Algorithm 2 LTV state-space realization

Input: The doubly-indexed generalized system Markov parameters $h(\ell + i - 1, \ell - j), 1 \leq i \leq q, 1 \leq j \leq r, \ell = k, k + 1$.

- ① Populate $\mathcal{H}_{qr}(\ell)$ defined in Eq.(3.4).
- ② Apply SVD to $\mathcal{H}_{qr}(k)$ and $\mathcal{H}_{qr}(k + 1)$ with the diagonal entries of $\Sigma(k)$ and $\Sigma(k + 1)$ decreasingly ordered.
- ③ Estimate n by inspecting the singular values.
- ④ Estimate $\mathcal{O}_q(k)$ and $\mathcal{O}_q(k + 1)$ from Eq.(3.26).
- ⑤ With J_U and J_D in Eq.(3.20), estimate $A(k)$ from Eq.(3.27).
- ⑥ With J_C, J_B in Eq.(3.29), estimate $C(k)$ and $B(k)$ from Eq.(3.30).
- ⑦ Estimate $D(k)$ as $\hat{D}(k) = h(k, k)$.

Output: The state space quadruplet $\{\hat{A}(k), \hat{B}(k), \hat{C}(k), \hat{D}(k)\}$.

When the generalized system Markov parameters are corrupted by noise, the state dimension n in Step 3 is decided by thresholding the singular values. Note the mild requirements $q \geq \kappa_o$ and $r \geq \kappa_c$.

Observe that $\hat{A}(k)$ is similar to $A(k)$ only if $T(k) = T(k + 1)$ which is the case if $H_{q,r}(k) = H_{q,r}(k + 1)$. The difference between the eigenvalues continuously depends on the difference $T(k + 1) - T(k)$, which in turn continuously depends on $\mathcal{H}_{qr}(k) - \mathcal{H}_{qr}(k + 1)$. If q and r are large enough, the latter is small in comparison to $\mathcal{H}_{qr}(k)$. It is easy to see that a uniformly exponentially stable LTV system satisfies this requirement. This realization problem was studied in [110] from a rigid body dynamics perspective. The scheme presented above is based on a single input-output data batch. If an ensemble of input-output data are available, state-space models can be identified without constraints on time-variations of the generalized system

Markov parameters. This was followed in the works in [55, 111, 112].

It remains to resolve the issue: “how large κ_o and κ_c should be to derive a realization of the SLS model in Eqs.(1.3)–(1.4) from subsets of the generalized system Markov parameters?”. It is easy to see that the sole assumption that Eqs.(1.3)–(1.4) be an SLS model with finitely many submodels is not sufficient to bound κ_o and κ_c unless submodels are changing according to some rule. We capture the requirements on the model structure in the following assumption:

Assumption 3.1.1 *The SLS model in Eqs.(1.3)–(1.6) has σ stable submodels with Macmillan degrees n and $\delta_*(\chi) \geq 2n$.*

Under this assumption, we will show that the SLS model given by Eqs.(1.3)–(1.6) is uniform.

Lemma 3.1.1 *With $\kappa_c = \kappa_o = 2n$, let $n_1 = 2n + 1$ and $N_1 = N - 2n + 1$. The SLS model in Eqs.(1.3)–(1.6) is uniform on $[n_1, N_1]$ if it satisfies Assumption 3.1.1.*

Proof. See Appendix.

If the submodels in $\mathcal{P}$ have the observability and the controllability indices bounded above by some $\nu < n$, then Lemma 3.1.1 will hold for $\kappa_c = \kappa_o = 2\nu$ by letting $\delta_*(\chi) \geq 2\nu$ in Assumption 3.1.1. Note also that Lemma 3.1.1 holds for any σ . The first block-column and the last block-row entry of $\mathcal{H}_{q,r}(k+1)$ is equal to $h(k+q, k)$. Then, from Eq.(3.1), k must be equal or smaller than $N - q$. The last block-column and the first block-row entry of $\mathcal{H}_{q,r}(k)$ is equal to $h(k, k-r)$. Hence, $k > r$, it follows that k must satisfy the inequalities $r < k \leq N - q$. Intersecting $(r, N - q]$ with the interval in Lemma 3.1.1, we derive the following constraint set for k ,

$$I_{q,r,n} = [\max\{n_1, r + 1\}, \min\{N_1, N - q\}]. \quad (3.32)$$

The main result of this section can be stated as follows.

Proposition 3.1.1 *Consider Algorithm 2 with the generalized system Markov parameters of the SLS model in Eqs.(1.3)–(1.6) satisfying Assumption 3.1.1. Suppose $q \geq 2n$ and $r \geq 2n$, then Algorithm 2 returns a realization which is topologically equivalent to the SLS model given by Eqs.(1.3)–(1.6) on $I_{q,r,n}$ in the noiseless case.*

Recall that Algorithm 1 returns a realization $\hat{A}(k)$, $\hat{B}(k)$, $\hat{C}(k)$, $\hat{D}(k)$ that is topologically equivalent to the underlying SLS model. This realization is not similar to the submodels $\mathcal{P}_{\varphi(k)}$ pointwisely in time, yet matches the generalized Markov

parameter sequences on $I_{q,r,n}$. Matching the transfer functions to the submodels is achieved by modifying the state-space parameter estimates. We conclude this section by fixing q and r to a default value of $2n$, i.e., $q = r = 2n$. Therefore, an LTV realization from Algorithm 2 is returned on the interval

$$I_{2n,2n,n} = [2n + 1, N - 2n]. \quad (3.33)$$

3.2. Observer-based transformation of SLS models to SARX models

This method was originally devised for LTI systems, and is the backbone of observer/Kalman filter identification algorithm [58, 60]. Similarly to the LTI case but instead of constant gain matrix, a time-varying gain matrix is assumed. A TV output-feedback term $K(k)y(k)$ is added and subtracted from Eq.(1.3)

$$\begin{aligned} x(k+1) &= A(k)x(k) + B(k)u(k) + K(k)y(k) - K(k)y(k), \\ &= (A(k) + K(k)C(k))x(k) + (B(k) + K(k)D(k))u(k) - K(k)y(k) \\ &= A_o(k)x(k) + B_o(k)\zeta(k), \end{aligned}$$

with $K(k)$ a time varying gain matrix. Define

$$A_o(k) = A(k) + K(k)C(k), \quad (3.34)$$

$$B_o(k) = [B(k) + K(k)D(k) \quad -K(k)], \quad (3.35)$$

$$D_o(k) = [D(k) \quad 0], \quad (3.36)$$

$$\zeta(k) = [u^T(k) \quad y^T(k)]^T. \quad (3.37)$$

Then, the observer-based transformed state space is given by

$$x(k+1) = A_o(k)x(k) + B_o(k)\zeta(k), \quad (3.38)$$

$$y(k) = C(k)x(k) + D_o(k)\zeta(k), \quad (3.39)$$

with its response given by

$$y(k) = C(k)\Phi_o(k, i)x(i) + \sum_{j=i}^k h_o(k, j)\zeta(j) \quad 1 \leq i \leq k, \quad (3.40)$$

when that the generalized observer Markov parameters and the observer state transition matrix are defined as

$$h_o(k, i) = \begin{cases} C(k)\Phi_o(k, i+1)B_o(i) & k > i, \\ D_o(k) & k = i, \\ 0 & k < i, \end{cases} \quad (3.41)$$

and

$$\Phi_o(k, i) = \begin{cases} A_o(k-1) \cdots A_o(i) & k > i, \\ I_n & k = i. \end{cases} \quad (3.42)$$

Similarly to the LTI case, in order to eliminate the dependence on the initial state, a deadbeat observer condition is stipulated: “for all k there exist $\tau \in \mathbb{N}$ such that $\Phi_o(k, k-\tau) = 0$ with $\tau < k' \leq k \leq k''$ ”. That being assumed, the previous observer response simplifies to

$$y(k) = \sum_{j=k-\tau}^k h_o(k, j)\zeta(j), \quad k' \leq k \leq k''. \quad (3.43)$$

Equation (3.43) depicts the output as a function of past inputs and outputs, where the weights are time varying. It is a generalization of the standard auto-regressive with exogenous inputs (ARX) model, well known in the analysis of time-invariant models [60]. The model in Eq.(3.43) is referred to as a generalized time-varying ARX model (GTV-ARX) of order $\tau + 1$ in [55]. One could rewrite Eq.(3.43) as

$$y(k) = \sum_{j=k-\tau}^{k-1} h_o^{(1)}(k, j)u(j) + \sum_{j=k-\tau}^{k-1} h_o^{(2)}(k, j)y(j) + D(k)u(k) \quad k' \leq k \leq k'', \quad (3.44)$$

where

$$h_o(k, j) = \begin{bmatrix} h_o^{(1)}(k, j) & h_o^{(2)}(k, j) \end{bmatrix}, \quad (3.45)$$

with $h_o^{(1)}(k, j) \in \mathbb{R}^{p \times m}$, $h_o^{(2)}(k, j) \in \mathbb{R}^{p \times p}$.

Definition 3.2.1 *An LTV discrete-time observer is said to be a deadbeat observer on the interval $[k', k'']$ if there exists a gain sequence $K(k)$ and a smallest $\tau^* \in \mathbb{N}$ such that:*

$$\Phi_o(k, k-\tau^*) = 0, \quad \text{for all } \tau^* < k' \leq k \leq k''. \quad (3.46)$$

Unlike the LTI case where realizing a deadbeat observer requires an observable system solely [57], realizing an LTV deadbeat observer is not an easy task. This is equivalent to solving Eq.(3.44) for every admissible value of k , i.e, in matrix form

solving the linear system of equations for H_τ from

$$Y_\tau = H_\tau \zeta \quad (3.47)$$

with

$$\begin{aligned} Y_\tau &= [y^T(k') \cdots y^T(k'')]^T, \\ \zeta &= [\zeta^T(k' - \tau) \cdots \zeta^T(k') \cdots \zeta^T(k'' - \tau) \cdots \zeta^T(k'')], \\ H_\tau &= \begin{bmatrix} h_o(k', k' - \tau) & \cdots & h_o(k', k') & & 0 \\ & & & \ddots & \vdots \\ 0 & & h_o(k'', k'' - \tau) & \cdots & h_o(k'', k'') \end{bmatrix}. \end{aligned}$$

Here, Y_τ being the output vector, ζ the regressors, and H_τ is the unknown banded structured matrix of observer Markov parameters. A quick thing to notice that the number of unknowns is $(k'' - k' + 1)(\tau + 1)p(p + m)$, whereas the number of equations is equal to $(k'' - k' + 1)p$, which indicates an underlying under-determined system of equations. In [55], parallel experiments were performed to collect enough I/O data and solve the system of equations uniquely, these parameters were further used to retrieve the generalized system Markov parameters and lastly identify system matrices by employing a pointwise time-varying eigensystem realization (TVERA) [110].

Suppose the state-space system in Eqs.(1.3)–(1.4) is an LTI. That being assumed, both the gain and system matrices are constant, hence

$$h_o(k, i) = C(A + KC)^{k-i} [B + KD - K], \quad k > i, \quad (3.48)$$

then, the ARX model coefficients become constant, and the previous system of equation (3.47) reduces to having only $(\tau + 1)p(p + m)$ unknowns and $(k'' - k' + 1)p$ number of equations. If one chooses k'' such that $(k'' - k' + 1)p \geq (\tau + 1)p(p + m)$, the set of observer Markov parameters can be recovered uniquely via least squares. Moreover in the LTI case, the deadbeat observer condition reduces to

$$\tau^* = \min\{k \in \mathbb{N} : (A + KC)^k = 0\}. \quad (3.49)$$

A gain matrix $K \in \mathbb{R}^{n \times p}$ is known to exist provided the system is observable and τ satisfies $\tau \geq \tau^* = \nu$ where $n \geq \nu \geq n/p$ [58]. The gain sequence does not have to be known *a priori*; conditions for its existence are enough for the identification procedure. They can be computed via the relationship between the generalized system Markov parameters and the generalized observer Markov parameters.

Suppose now that the underlying system with state space equations (1.3)–(1.4) is a switched linear system (SLS). That being the case, the number of unknowns in Eq.(3.47) is less than the number of unknowns in the case when an arbitrary LTV system is assumed. Moreover, under some conditions on the minimum dwell time, one can recover a unique set of the generalized system Markov parameters. Due to the fact that switch locations are unknown, the least squares can not be used to estimate the unknowns. To accomplish this, we will resort to some algorithms from the compressive sensing literature [71]. We emphasize one last point here, that is the TV deadbeat observer transforms a state-space model to a time-varying ARX model with a finite order. In other words, the infinite number of generalized system Markov parameters are compressed into a small number of generalized observer Markov parameters, which are easier to estimate by solving a regression problem. So, the deadbeat observer can be viewed as a parameter compression mechanism [61].

The following result provides an upper bound on τ^* in the definition of the LTV deadbeat observer. This upper bound does not depend on σ .

Lemma 3.2.1 *Let the SLS model in Eqs.(1.3)–(1.6) be as in Assumption 3.1.1. Then, there exists a deadbeat observer and a finite gain sequence for the model in Eqs.(1.3)–(1.6) with properties: $\tau^* \leq 2n$ on $[n_1, N]$; $K(k) = K(k_i)$ for all $k \in [k_i, k_{i+1})$ and $1 \leq i < i^*$; and $\tau^* \leq n$ for all $k \in [k_i + n, k_{i+1})$ and $1 \leq i < i^*$.*

Proof. See Appendix.

From Lemma 3.2.1, we write Eq.(3.43) as

$$y(k) = \sum_{i=k-2n}^k h_o(k, i) \zeta(i), \quad n_1 \leq k \leq N. \quad (3.50)$$

In Chapter 6, we will formulate an optimization problem to estimate the switching sequence and the submodels. This problem formulation promotes sparsity by optimally selecting a gain sequence. The piecewise constant gain sequence in Lemma 3.2.1 is a feasible solution of this problem. We conclude by emphasising that the observer-based transformation derived in this section reduces state-space identification from the input-output data to the estimation of generalized observer Markov parameters by solving a regression problem.

3.3. Computation of the generalized system Markov parameters and the observer gains

In accordance with the LTI case in [58, 60], the relationship between the generalized observer Markov parameters and generalized system Markov parameters was established in [55]. Solving the previous regression problem gives us access to

$$h_o(k, j), \quad j = k - 2n, \dots, k; \quad k = n_1, \dots, N_1. \quad (3.51)$$

The system Markov parameters of the model in Eqs.(1.3)–(1.4) are defined for $k > i$ by

$$h(k, i) = C(k)\Phi(k, i+1)B(i), \quad (3.52)$$

and for $k = i$ by $h(k, k) = D(k)$ where $\Phi(k, k) = I_n$ and

$$\Phi(k, i) = A(k-1) \cdots A(i), \quad (3.53)$$

with $I_n \in \mathbb{R}^{n \times n}$ denoting the n by n identity matrix. Write each $h_o(k, i)$ as

$$h_o(k, i) = [h_o^{(1)}(k, i) \quad h_o^{(2)}(k, i)], \quad (3.54)$$

and let

$$\gamma(k, i) = h_o^{(1)}(k, i) + h_o^{(2)}(k, i)h_o^{(1)}(i, i), \quad k > i \geq 1. \quad (3.55)$$

The following recurrence formula was derived in [55] as

$$h(k, i) = \gamma(k, i) + \sum_{j=i+1}^{k-1} h_o^{(2)}(k, j)h(j, i), \quad k > i + 1. \quad (3.56)$$

For the deadbeat observers, the constraints $h_o^{(\ell)}(k, i) = 0$ for $\ell = 1, 2$ and $k - i > 2n$ must be applied in Eq.(3.55) and Eq.(3.56). We start by calculating $h(k, k-1) = \gamma(k, k-1)$ for every admissible k . Then, $h(k, k-2)$ can be calculated via:

$$\begin{aligned} h(k, k-2) &= \gamma(k, k-2) + h_o^{(2)}(k, k-1)h(k-1, k-2) \\ &= \gamma(k, k-2) + h_o^{(2)}(k, k-1)\gamma(k-1, k-2). \end{aligned}$$

and so on, until all the generalized system Markov parameters are estimated.

To estimate the observer gains $K(k)$, we propose a two-step procedure. In the

first step, we estimate the observer gain Markov parameters defined with

$$h_m(k, i) = \begin{cases} C(k)\Phi(k, i+1)K(i) & k > i+1, \\ C(k)K(k-1) & k = i+1, \\ 0 & k < i+1, \end{cases} \quad (3.57)$$

we compute them from the generalized observer Markov parameters in Eq.(3.51). For that, the following recursive formula was established in [55],

$$h_m(k, i) = -h_o^{(2)}(k, i) + \sum_{j=i+1}^{k-1} h_o^{(2)}(k, j) h_m(j, i) \quad (3.58)$$

for $k > i+1$ and initialized by $h_m(k, k-1) = -h_o^{(2)}(k, k-1)$. The second step consists of estimating $K(k-1)$ from (3.57).

Given $q \in \mathbb{N}$, concatenate the equations in (3.57) and note that

$$\begin{aligned} \begin{bmatrix} h_m(k, k-1) \\ \vdots \\ h_m(k+q-1, k-1) \end{bmatrix} &= \begin{bmatrix} C(k) \\ \vdots \\ C(k+q-1)\Phi(k+q, k) \end{bmatrix} K(k-1), \\ &= \mathcal{O}_q(k)K(k-1), \\ &= \hat{\mathcal{O}}_q(k)T(k)K(k-1). \end{aligned} \quad (3.59)$$

Compute $\hat{\mathcal{O}}_q(k)$ for the pair $(\hat{A}(k), \hat{C}(k))$ obtained from the realization algorithm outlined in Subsection 3.1 and set

$$\hat{K}(k-1) = T(k)K(k-1) = \hat{\mathcal{O}}_q^\dagger(k) \begin{bmatrix} h_m(k, k-1) \\ \vdots \\ h_m(k+q-1, k-1) \end{bmatrix}. \quad (3.60)$$

We summarize the above derivations in the form of an algorithm.

Algorithm 3 The conversion from SARX model to system Markov parameters

Input: $h_o(k, i)$, $i = 1, \dots, k$; $n_1 \leq k \leq N_1$

- ① **Initialize:** $k = n_1$
- ② **while** $i \leq k$ **do**
- ③ Calculate $\gamma(k, i)$ from Eq.(3.55) while $k - 2n \leq i < k$.
- ④ Set $h(k, k - 1) = \gamma(k, k - 1)$.
- ⑤ Estimate $h(k, i)$ from Eq.(3.56) while $k - 2n \leq i < k - 1$.
- ⑥ Set $h_m(k, k - 1) = -h_o^{(2)}(k, k - 1)$.
- ⑦ Estimate $h_m(k, i)$ from Eq.(3.58) while $k - 2n \leq i < k - 1$.
- ⑧ Estimate $K(k - 1)$ from Eq.(3.60).
- ⑨ $k \leftarrow k + 1$.
- ⑩ **end while**

Output: $h(k, i)$ and $K(k - 1)$ for $i \leq k$

Note that

$$\begin{aligned}
 \hat{A}_o(k) &= \hat{A}(k) + \hat{K}(k)\hat{C}(k) \\
 &= T(k+1)A(k)T^{-1}(k) + T(k+1)K(k)C(k)T^{-1}(k) \\
 &= T(k+1)(A(k) + K(k)C(k))T^{-1}(k),
 \end{aligned}$$

and for $k > i + 1$

$$\begin{aligned}
 \hat{\Phi}_o(k, i + 1) &= \hat{A}_o(k - 1) \cdots \hat{A}_o(i + 1) \\
 &= T(k)(A(k - 1) + K(k - 1)C(k - 1))T^{-1}(k - 1) \\
 &\quad \cdot T(k - 1)(A(k - 2) + K(k - 2)C(k - 2))T^{-1}(k - 2) \\
 &\quad \cdots (A(i + 1) + K(i + 1)C(i + 1))T^{-1}(i + 1) \\
 &= T(k)\Phi_o(k, i + 1)T^{-1}(i + 1).
 \end{aligned}$$

Also

$$\begin{aligned}
 \hat{B}_o(i) &= [\hat{B}(i) + \hat{K}(i)\hat{D}(i) - \hat{K}(i)] \\
 &= [T(i + 1)B(i) + T(i + 1)K(i)D(i) - T(i + 1)K(i)] \\
 &= T(i + 1)B_o(i),
 \end{aligned}$$

and

$$\hat{C}_o(k) = \hat{C}(k) = C(k)T^{-1}(k) = C_o(k)T^{-1}(k).$$

So, for $k > i + 1$,

$$\begin{aligned}\hat{h}_o(k, i) &= \hat{C}(k)\hat{\Phi}_o(k, i+1)\hat{B}_{ob}(i) \\ &= C(k)T^{-1}(k)T(k)\Phi_o(k, i+1)T^{-1}(i+1)T(i+1)B_o(i) \\ &= h_o(k, i).\end{aligned}$$

The last formula is also valid for $k = i + 1$, and for $k = i$, $\hat{h}_o(k, k) = h_o(k, k)$. Thus, we conclude that the quintuplet $\{\hat{A}_o(k), \hat{B}_o(k), \hat{C}_o(k), \hat{D}_o(k), \hat{K}(k)\}$ is topologically equivalent to the quintuplet $\{A_o(k), B_o(k), C_o(k), D_o(k), K(k)\}$ for every k . The main result of this section is stated as follows.

Proposition 3.3.1 *Every SLS model in Eqs.(1.3)–(1.6) satisfying Assumption 3.1.1 is in one-to-one correspondence with a time-varying ARX model of order $2n$ in the noise free case. Furthermore, Algorithm 3 recovers the generalized system Markov parameters of Eqs.(1.3)–(1.6) from the ARX model parameters.*

Lemma 3.3.1 *Algorithm 3 back transforms the ARX model in Eq.(3.43) to the LTV model given by Eqs.(1.3)–(1.4) pointwisely in time.*

As in Section 3.1, the results derived above are independent from the number of submodels; but, depend on the minimum dwell time. The ARX model is overparameterized to accommodate submodel changes at the switches. The switches are not known *a priori*. Once they are located, fewer parameters may be used. Zero-padding will only require richer inputs for parameter identifiability as opposed to parsimonius models.

3.4. Effect of process and measurement noise on observer-based transformation

In the rest of this chapter, we will investigate how the noise injected in the state-space model in Eqs.(1.3)–(1.4) is reflected in the time-varying ARX model in Eq.(3.43). The necessary modifications on the observer equations (3.38)–(3.39) are as follows:

$$x(k+1) = A_o(k)x(k) + B_o(k)\zeta(k) + [K(k) \ I_n] \tilde{e}(k), \quad (3.61)$$

$$y(k) = C(k)x(k) + D_o(k)\zeta(k) + [I_p \ 0_{p \times n}] \tilde{e}(k), \quad (3.62)$$

where $\tilde{e}(k) = [e^T(k) \ w^T(k)]^T$ and $A_o(k)$, $B_o(k)$, $D_o(k)$, and $\zeta(k)$ are as in Eqs.(3.34)–(3.37). For $k \geq i \geq 1$, the observer response is calculated as follows:

$$y(k) = C(k)\Phi_o(k, i)x(i) + \sum_{j=i}^k h_o(k, j)\zeta(j) + \sum_{j=i}^k h_e(k, j)\tilde{e}(j), \quad (3.63)$$

where $h_o(k, i)$ and $\Phi_o(k, i)$ are as in Eq.(3.41) and Eq.(3.42) and the noise Markov parameters $h_e(k, i)$ are defined by

$$h_e(k, i) = \begin{cases} C(k)\Phi_o(k, i+1) [K(i) \ I_n] & k > i, \\ [I_p \ 0_{p \times n}] & k = i, \\ 0 & k < i. \end{cases} \quad (3.64)$$

Similarly, if we stipulate dead-beat observer condition, the response for inputs $\zeta(k)$ and $\tilde{e}(k)$ given that the initial state vanishes is given by

$$y(k) = \sum_{j=k-2n}^k h_o(k, j)\zeta(j) + \sum_{j=k-2n}^k h_e(k, j)\tilde{e}(j), \quad n_1 \leq k \leq N_1. \quad (3.65)$$

Equation (3.65) can be decomposed into

$$y(k) = \sum_{j=k-2n}^{k-1} h_o^{(1)}(k, j)u(j) + \sum_{j=k-2n}^{k-1} h_o^{(2)}(k, j)y(j) + \sum_{j=k-2n}^k h_e(k, j) + D(k)u(k) \quad (3.66)$$

This is a generalization of the ARMAX (auto-regressive moving-average with exogenous input) model, where it is customized to fit the time-varying case. The model in Eq.(3.66) was derived in [55] and denoted as a generalized time-varying ARMAX model (GTV-ARMAX). Note that the moving average (MA) part is bi-linear in terms of the noise Markov parameters and the augmented noise vector, which makes it hard for direct estimation. An alternative would be to estimate the generalized observer Markov parameters $h_o(k, j)$ from the following ARX model,

$$y(k) = \sum_{j=k-2n}^k h_o(k, j)\zeta(j) + \eta(k), \quad n_1 \leq k \leq N_1, \quad (3.67)$$

where $\eta(k)$ denote the residual at time k , and

$$\hat{h}_e(k, i) = \begin{cases} \hat{C}(k)\hat{\Phi}_o(k, i+1) [\hat{K}(i) \ I_n] & k > i, \\ [I_p \ 0_{p \times n}] & k = i, \\ 0 & k < i. \end{cases} \quad (3.68)$$

Once the observer Markov parameters are obtained from the previous step, the system is realized, and the gains $K(k)$ are retrieved, an estimate of $\hat{h}_e(k, i)$ is

automatically obtained from Eq.(3.68) as follows

$$\hat{h}_e(k, i) = \hat{C}(k)\hat{\Phi}_o(k, i+1) [\hat{K}(i) \ I_n] \quad (3.69)$$

$$= [\hat{C}(k)\hat{\Phi}_o(k, i+1)\hat{K}(i) \ \hat{C}(k)\hat{\Phi}_o(k, i+1)] \quad (3.70)$$

$$= \begin{bmatrix} \hat{h}_e^{(1)}(k, i) & \hat{h}_e^{(2)}(k, i) \end{bmatrix} \quad (3.71)$$

$$= \begin{bmatrix} -h_o^{(2)}(k, j) & \hat{h}_e^{(2)}(k, i) \end{bmatrix}. \quad (3.72)$$

Note also that

$$\hat{h}_e^{(2)}(k, i)\hat{K}(i) = -h_o^{(2)}(k, j). \quad (3.73)$$

Therefore

$$\hat{h}_e^{(2)}(k, i) = -h_o^{(2)}(k, j)\hat{K}^\dagger(i), \quad (3.74)$$

where $\hat{K}^\dagger(i) = \hat{K}^T(i)(\hat{K}(i)\hat{K}^T(i))^{-1}$ is the right Moore-Penrose pseudo inverse. This holds true if and only if $n \leq p$ and $\hat{K}(i)$ is of full row rank. Another method is to realize the observer state-space model using the LTV realization in Algorithm 2. Having done that, a quadruplet $\{\hat{A}_0(k), \hat{B}_0(k), \hat{C}_0(k), \hat{D}_0(k)\}$ is obtained, and $\hat{h}_e^{(2)}(k, i)$ can easily be computed.

The residual $\eta(k)$ is given by

$$\hat{\eta}(k) = \sum_{j=k-2n}^k \hat{h}_e(k, j)\tilde{e}(j), \quad n_1 \leq k \leq N_1, \quad (3.75)$$

which can compactly be expressed as

$$Y_{\hat{\eta}} = H_e \zeta_{\tilde{e}}, \quad (3.76)$$

with

$$\begin{aligned} Y_{\hat{\eta}} &= [\hat{\eta}^T(n_1) \ \cdots \ \hat{\eta}^T(N_1)]^T, \\ \zeta_{\tilde{e}} &= [\tilde{e}^T(n_1) \ \cdots \ \tilde{e}^T(N_1)]^T, \\ H_e &= \begin{bmatrix} \hat{h}_e(n_1, 1) & \cdots & \hat{h}_e(n_1, n_1) & & 0 \\ & & & \ddots & \vdots \\ 0 & & \hat{h}_e(N_1, N_1 - 2n) & \cdots & \hat{h}_e(N_1, N_1) \end{bmatrix}. \end{aligned}$$

The vector $\zeta_{\tilde{e}}$ maybe estimated by minimizing a suitable norm of the difference $Y_{\hat{\eta}} - H_e \zeta_{\tilde{e}}$.

In this section, we have shown that the observer-based transformation converts the parameter estimation in general state-space models to ARX-model estimation with time-varying parameters. In [94], ARX-model estimation was interpreted as

state estimation in general state-space models and its connections with the Kalman filter and the recursive least-squares algorithm were noted. It is pointed out in [113] that the Kalman filter works well if the parameter variation can be described as a Gaussian white noise process.

3.5. Example

Let's define a SLS state-space model with $\sigma = 3$ and the following state-space matrices:

$$\begin{aligned} A_1 &= \begin{pmatrix} 0.15 & 0.40 & -0.65 \\ -0.75 & 0.1 & -0.35 \\ 0.20 & 0.70 & 0.20 \end{pmatrix}, B_1 = \begin{pmatrix} -0.20 & 0.45 \\ -0.06 & 0 \\ 0.22 & 0 \end{pmatrix} \\ C_1 &= \begin{pmatrix} 0 & 0.40 & 0.45 \\ -1 & -0.60 & 0.90 \end{pmatrix}, D_1 = \begin{pmatrix} 0 & -0.35 \\ -1.70 & -0.25 \end{pmatrix} \end{aligned}$$

$$\begin{aligned} A_2 &= \begin{pmatrix} 0.27 & 0.24 & -0.55 \\ 0.24 & 0.65 & 0.30 \\ -0.55 & 0.30 & 0.27 \end{pmatrix}, B_2 = \begin{pmatrix} -0.55 & 0 \\ -1.40 & 1 \\ 0.05 & -0.72 \end{pmatrix} \\ C_2 &= \begin{pmatrix} 0.70 & 1 & -0.27 \\ -0.35 & 0 & -1.10 \end{pmatrix}, D_2 = \begin{pmatrix} 2.15 & 0.25 \\ 0 & -0.36 \end{pmatrix} \end{aligned}$$

$$\begin{aligned} A_3 &= \begin{pmatrix} 0.45 & 0.02 & 0.42 \\ -0.17 & 0.53 & 0.20 \\ 0.38 & 0.26 & 0 \end{pmatrix}, B_3 = \begin{pmatrix} 0 & 0.15 \\ 0.27 & -0.46 \\ 0.07 & 0.54 \end{pmatrix} \\ C_3 &= \begin{pmatrix} 0 & 0.60 & 0.28 \\ 0 & 0.86 & 0.45 \end{pmatrix}, D_3 = \begin{pmatrix} 0 & -0.90 \\ 0 & 0.85 \end{pmatrix} \end{aligned}$$

This model was adapted from [36], where the problem of identifying a state-space SLS model for the case when the continuous state is known was treated.

In this numerical study, the model transformation obtained in Section 3.2 via the deadbeat observer will be studied in detail. Let's generate the switching sequence plotted in Figure 3.1, sampled independently from a uniform distribution of the three submodels above. The switching sequence obeys a dwell time constraints of $\delta_*(\chi) = 15$.

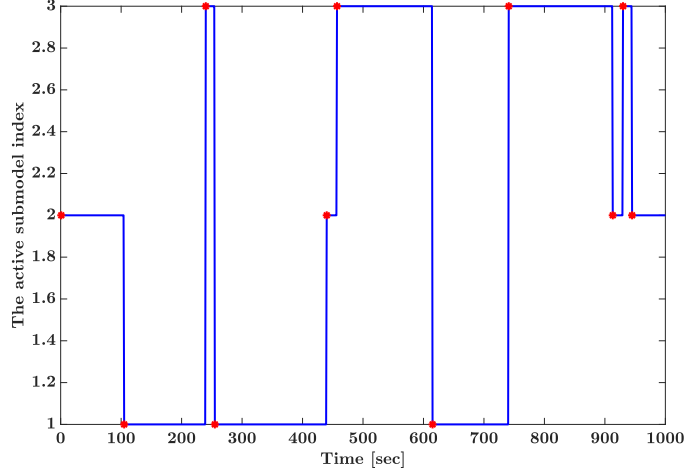


Figure 3.1: The switching sequences in Subsection 3.5.

The conversion from the SLS models to the ARX models was achieved by the pole placement technique implemented by the `place` command in MATLAB. This command takes the matrices $(A(k), C(k))$, and the desired observer poles locations $\tilde{p}$ as inputs; it returns the corresponding gain sequence $K(k)$. In our case, $\tilde{p} = [0 \ 0 \ 0]$, but since the `place` command limits the multiplicity of poles to be placed to a max of $\text{rank}(C(k)) = 2$, one of the poles of $\tilde{p}$ is set to a very small number close to zero instead of exactly zero, i.e, $\tilde{p} = [0 \ 0 \ 10^{-7}]$. This process is repeated pointwise over the desired range of k . Figure 3.2 depicts the obtained $\|K(k)\|_F$ as a function of time. For better vision a semi-log plot was used. Notice the piecewise constant nature of the gain.

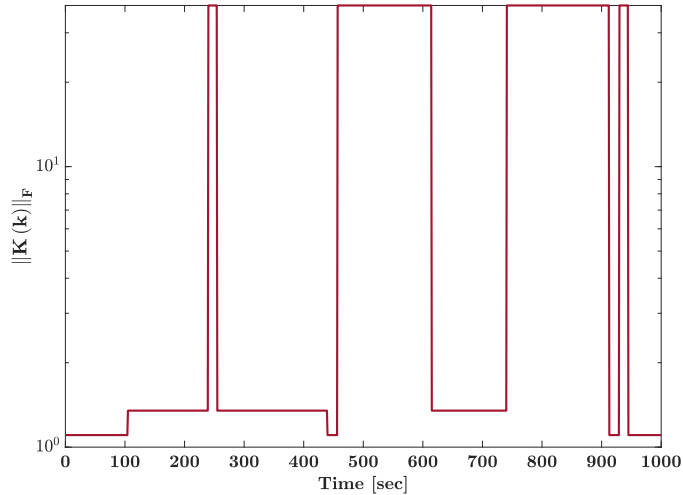


Figure 3.2: $\|K(k)\|_F$ for switching sequence in Subsection 3.5.

In Figure 3.3, the ℓ_1 norm of the deadbeat observer eigenvalues, *that is*, $\mathcal{M}(A(k) + K(k)C(k))$ is plotted with $\mathcal{M}$ defined in Eq.(4.3), as anticipated they are properly placed at the origin.

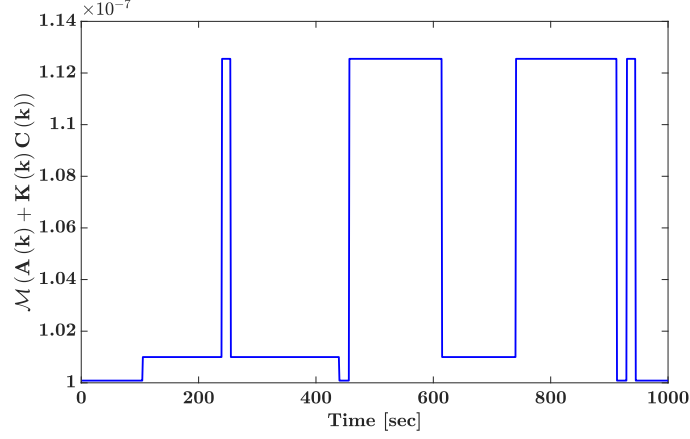


Figure 3.3: The ℓ_1 norm of the deadbeat observer eigenvalues.

In Figure 3.4, the Frobenius norm of the deadbeat observer state transition matrix defined in Eq.(3.42) is plotted versus k for several values of τ . Notice that in three steps the observer state transition matrix approximately vanishes, thus, in Definition 3.2.1 $\tau^* = 3$. Hence, the SARX order in Lemma 3.2.1 can be taken 3 on $[n_1, N]$ and only 2 in the subintervals $[k_i + 3, k_{i+1})$.

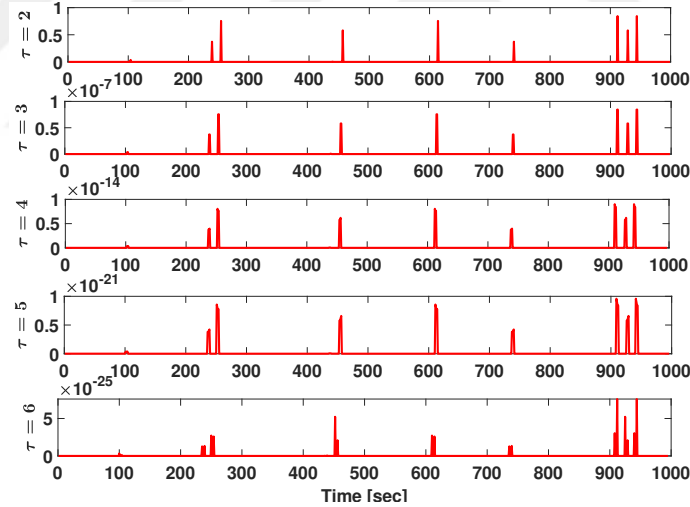


Figure 3.4: $\|\Phi_o(k, k - \tau)\|_F$ as a function of k for $\tau = 2, \dots, 6$ and the first switching sequence.

3.6. Summary

In this Chapter, we presented a time-varying realization algorithm to obtain the pointwise LTV quadruplets of a SLS model. The algorithm is driven by constructing Hankel matrix pairs populated by the SLS Markov parameters. A dwell time constraint of twice the Macmillan degree n was put forward, based on which uniformity of the SLS model was established. Beside that, The constructed Hankel pairs block-column and block-row sizes were also fixed to twice the MacMillan de-

gree. A transformation to go from LTV state-space models to TV ARX models via the injection of gain modulated outputs in the state equation appeared as well. To revert back to the LTV model from the ARX model, a recursive algorithm was put available. Along with that, a two step procedure to acquire the observer gains was suggested. The influence of injecting both the measurement noise and process onto the observer-based transformation was treated also. In the sequel, we treat the problem of recovering the SLS submodels and the switching sequence given its LTV state-space quadruplets over time.



4. SLS MODELS ESTIMATION FROM LTV STATE-SPACE REALIZATIONS

In this chapter, we address the problem of extracting a SLS model given its LTV state-space realization estimated from Algorithm 2. To start off, Section 4.1 proposes a mechanism to retrieve all the SLS submodels given that they are active at least once with long enough dwell time. Section 4.2, conversely, tackles the switch sequence estimation given the dictionary of the estimated submodels from the first step. The algorithm established in this chapter is supported via a numerical example in Section 4.3.

4.1. Submodel estimation

The extended observability matrix in Eq.(3.14) of the SLS model in Eqs.(1.3)–(1.6) is related to the extended observability matrix $\hat{\mathcal{O}}_q(k)$ of the realization delivered by Proposition 3.1.1 by

$$\hat{\mathcal{O}}_q(k+j) = \mathcal{O}_q(k+j)T^{-1}(k+j), \quad j \geq 0.$$

Hence,

$$\hat{\mathcal{T}}_q(k) = T(k)\mathcal{T}_q(k)T^{-1}(k+1) \quad (4.1)$$

with $\hat{\mathcal{T}}_q(k) = \hat{\mathcal{O}}_q^\dagger(k)\hat{\mathcal{O}}_q(k+1)$ and $\mathcal{T}_q(k) = \mathcal{O}_q^\dagger(k)\mathcal{O}_q(k+1)$. Pre-multiplying $\hat{A}(k)$ in Eq.(3.28) with $\hat{\mathcal{T}}_q(k)$, we get from Eq.(4.1)

$$\tilde{A}(k) = \hat{\mathcal{T}}_q(k)\hat{A}(k) = T(k)\mathcal{T}_q(k)A(k)T^{-1}(k).$$

The pre-multiplication of $\hat{B}(k)$ with $\hat{\mathcal{T}}_q(k)$ yields from Eq.(4.1),

$$\tilde{B}(k) = \hat{\mathcal{T}}_q(k)\hat{B}(k) = T(k)\mathcal{T}_q(k)B(k).$$

We need not operate on $\hat{C}(k)$ and $\hat{D}(k)$ since $\hat{D}(k) = D(k)$ and $\hat{C}(k) = C(k)T^{-1}(k)$. Thus, the similarity transformation $T^{-1}(k)$ maps $\tilde{\mathcal{P}}_{\varphi(k)} = \{\tilde{A}(k), \tilde{B}(k), \hat{C}(k), \hat{D}(k)\}$ to the quadruplet $\{\mathcal{T}_q(k)A(k), \mathcal{T}_q(k)B(k), C(k), D(k)\}$. Hence, if $\mathcal{T}_q(k) = I_n$, $\tilde{\mathcal{P}}_{\varphi(k)}$ will be similar to $\mathcal{P}_{\varphi(k)}$. The parameter q can have a different value than the one used in Algorithm 2 if it is not smaller than $2n$. Hence, we replace it with $2n$.

Consider the block-rows of $\mathcal{O}_{2n}(k)$ starting with $C(k)$ and

$$\begin{aligned} C(k+j-1)\Phi(k+j-1, k) &= C(k+j-1)A(k+j-2) \\ &\dots A(k) \text{ for } 2 \leq j \leq 2n. \end{aligned} \quad (4.2)$$

The calculation of $\mathcal{T}_{2n}(k)$ requires the samples $C(k+2n)$ and $C(k+j)$, $A(k+j)$ for $j = 0, \dots, 2n-1$. If $k = k_i$, $k_i \in \chi$ and $k_{i+1} - k_i > 2n$, they will be equal to $C(k_i)$ and $A(k_i)$. Therefore, $\mathcal{O}_{2n}(k) = \mathcal{O}_{2n}(k+1)$ and $\mathcal{T}_{2n}(k) = \mathcal{O}_{2n}(k)^\dagger \mathcal{O}_{2n}(k) = I_n$. As k increases, $\mathcal{T}_{2n}(k) = I_n$ for all $k \in [k_i, k_{i+1} - 2n)$ and $\tilde{\mathcal{P}}_{\varphi(k)} \sim \mathcal{P}_{\varphi(k_i)}$ where the notation ' $\sim$ ' means similar. In particular, the eigenvalues of $\tilde{A}(k)$ are constant there. Suppose $k_{i+1} - k_i \geq 5n$. Then, on a subinterval of length at least $3n$, $\tilde{\mathcal{P}}_{\varphi(k)} \sim \mathcal{P}_{\varphi(k_i)}$ and on an interval of length at most $2n$, the active submodel is indeterminate and a reclustering is necessary. Since we aim to estimate the submodels up to similarity transformations, the union of such intervals must reveal all submodels. The clustering assumption is stated below.

Assumption 4.1.1 *Every submodel of the SLS model in Eqs.(1.3)–(1.6) is visited at least once in the interval $[n_1, N - 2n]$ with a waiting time of at least $5n$.*

Denote the ℓ_1 -norm of the eigenvalues of a given matrix $X \in \mathbb{R}^{n \times n}$ by

$$\mathcal{M}(X) = \sum_{i=1}^n |\lambda_i(X)|, \quad (4.3)$$

for all $k \in [k_i, k_{i+1} - 2n)$, $\mathcal{M}(\tilde{A}(k)) = \mathcal{M}(A(k_i))$. Hence, in the noiseless case $\mathcal{P}_{\varphi(k_i)}$ may be determined by viewing the graph of $\mathcal{M}(\tilde{A}(k))$ since it is constant on at least one interval of length at least $3n$, and may have fluctuating values on intervals of length at most $2n$ when Assumption 4.1.1 holds.

Now consider an interval $[k_i, k_{i+1} - 2n)$ of length at least $3n$. Since $\tilde{\mathcal{P}}_{\varphi(k)} \sim \mathcal{P}_{\varphi(k_i)}$ for all points in this interval, there exist a canonical state-space realization denoted by $\tilde{\mathcal{P}}_{\varphi(k_i)}$. Then, the collection $\tilde{\mathcal{P}}_{\varphi(k_i)}$ with $k_i \in \chi$ left end points of such intervals recovers $\mathcal{P}$. The submodel or the scheduling sequence extraction is known as the *hybrid decoupling* stage in hybrid system identification literature.

A clustering technique based on $\mathcal{M}(\tilde{A}(k))$, for example the density-based clustering algorithm proposed in [114], may be used to extract the collection $\tilde{\mathcal{P}}_{\varphi(k_i)}$. In our numerical study, we will use the algorithm proposed in [114] and implemented by the `dbscan` command in MATLAB, for details on how the algorithm works see Appendix B. Another popular clustering method is the k -means clustering algorithm in [115] implemented by the `kmeans` command in MATLAB. Unlike the k -means clus-

tering algorithm, the density-based clustering algorithm does not need the number of clusters to be specified *a priori*. We refer the interested reader to [116] for further information on the clustering algorithms.

4.2. Switching sequence estimation

We will present two methods to estimate the switching sequence. In the first method, suppose that $C(k+j) = C(k_i)$, $0 \leq j < 2n$ and $C(k+2n) = C(k_{i+1})$ for some k which also implies that $A(k+j) = A(k_i)$, $0 \leq j < 2n$. Then,

$$\begin{aligned}\mathcal{O}_{2n}(k) &= \begin{bmatrix} C(k_i) \\ \vdots \\ C(k_i)A^{2n-1}(k_i) \end{bmatrix}, \\ \mathcal{O}_{2n}(k+1) &= \begin{bmatrix} C(k_i) \\ \vdots \\ C(k_i)A^{2n-2}(k_i) \\ C(k_{i+1})A^{2n-1}(k_i) \end{bmatrix}.\end{aligned}\tag{4.4}$$

From Eq.(3.9) and Lemma 3.1.1, $\mathcal{O}_{2n}^T(k)\mathcal{O}_{2n}(k) = G_o(k_i, 2n) > 0$ for all $k \in \mathbb{N}$. Thus, from Eq.(4.4) we get with $k = k_{i+1} - 2n$,

$$\begin{aligned}\mathcal{T}_{2n}(k)A(k) &= A(k_i) + G_o^{-1}(k_i, 2n)C^T(k_i)[A^T(k_i)]^{2n-1} \\ &\cdot [C(k_{i+1}) - C(k_i)]A^{2n}(k_i) = A(k_i) + \delta_{\mathcal{T}_{2n}}(k_i)A(k_i).\end{aligned}$$

Let

$$\mu = \min_{k_i \in \chi} |\mathcal{M}(\mathcal{T}_{2n}(k_{i+1} - 2n)A(k_i)) - \mathcal{M}(A(k_i))| \tag{4.5}$$

where the minimum is calculated over two observability pairs $\{C(k_i), A(k_i)\}$ and $\{C(k_{i+1}), A(k_{i+1})\}$ drawn from $\mathcal{P}$. There are $\sigma(\sigma-1)/2$ such pairs. Since the submodels are minimal and invariant to similarity transformations, we may assume, without loss of generality, that $\mu > 0$. Recall that a similarity transformation changes not only an observability pair, but also the corresponding controllability pair. However, any changes in the latter are not reflected in Eq.(4.5). Thus, $\mathcal{M}(\tilde{A}(k)) = \mathcal{M}(\tilde{A}(k_i))$ for all $k \in [k_i, k_{i+1} - 2n)$ and $\mathcal{M}(\tilde{A}(k)) \neq \mathcal{M}(A(k_i))$ at $k = k_{i+1} - 2n$. Translation of the change point by $2n$ is the next switch k_{i+1} .

The procedure we present next is iterative in time. Suppose that the points $k_i \leq j < \ell$ were clustered to $\varphi(k_i)$ and we want to re-cluster ℓ to either $\varphi(k_i)$ or

$\varphi(k_{i+1})$. From Eq.(3.52),

$$h(\ell, j) = \begin{cases} C(\ell) [A(k_i)]^{\ell-j-1} B(k_i), & \ell > j \\ D(\ell), & \ell = j. \end{cases} \quad (4.6)$$

Since $\tilde{\mathcal{P}}_{\varphi(k_i)}$ is similar to $\mathcal{P}_{\varphi(k_i)}$, for some $T_i \in \mathbb{R}^{n \times n}$ we have $A(k_i) = T_i^{-1} \tilde{A}(k_i) T_i$, $B(k_i) = T_i^{-1} \tilde{B}(k_i)$, and $C(k_i) = \hat{C}(k_i) T_i$. Then, Eq.(4.6) can be written as

$$h(\ell, j) = \begin{cases} \check{C} [\tilde{A}(k_i)]^{\ell-j-1} \tilde{B}(k_i), & \ell > j \\ \check{D}, & \ell = j. \end{cases} \quad (4.7)$$

where $\check{C} = C(\ell) T_i^{-1}$ and $\check{D} = D(\ell)$. Treating $\check{C}$ and $\check{D}$ as unknowns, the right-hand side of Eq.(4.7) becomes a linear estimator of $h(\ell, j)$ denoted by $\tilde{h}(\ell, j)$. Let $\|X\|_F$ denote the Frobenious norm of a given matrix X . We find $\check{C}$ and $\check{D}$ by solving the linear least-squares minimization problem

$$\min_{\check{C}, \check{D}} \sum_{j=\ell-2n}^{\ell} \|\tilde{h}(\ell, j) - h(\ell, j)\|_F^2. \quad (4.8)$$

The minimization problem in Eq.(4.8) is separable in $\check{C}$ and $\check{D}$. Thus, $\check{D} = D(\ell) = \hat{D}(\ell)$. If $D(\ell)$ is distinct for each submodel in $\mathcal{P}$, then $\check{D} = \hat{D}(\ell)$ determines also $\check{C}$ and the submodel uniquely. If $D(\ell)$ is not distinct in $\mathcal{P}$, we determine $\check{C}$ by solving the normal equations $\mathcal{H}_{1,2n}(\ell) = \check{C} \tilde{\mathcal{R}}_{2n}(\ell - 1)$ where $\tilde{\mathcal{R}}_{2n}(\ell - 1)$ is formed from the pair $\{\tilde{A}(k_i), \tilde{B}(k_i)\}$. Hence,

$$\begin{aligned} \check{C} &= \mathcal{H}_{1,2n}(\ell) \tilde{\mathcal{R}}_{2n}^\dagger(\ell - 1) \\ &= C(\ell) \mathcal{R}_{2n}(\ell - 1) \tilde{\mathcal{R}}_{2n}^\dagger(\ell - 1) = C(\ell) T_i^{-1}. \end{aligned} \quad (4.9)$$

It follows that $\mathcal{H}_{1,2n}(\ell) \tilde{\mathcal{R}}_{2n}^\dagger(\ell - 1) = C(\ell) T_i^{-1}$. The quadratic cost has a minimum value of zero if $\ell = \varphi(k_i)$ since in this case $C(\ell) T_i^{-1} = \hat{C}(k_i)$ and it is sufficient to check if $\mathcal{H}_{1,2n}(\ell) \tilde{\mathcal{R}}_{2n}^\dagger(\ell - 1) = \hat{C}(\ell)$ holds. Suppose now $\ell = \varphi(k_{i+1})$, *that is*, a change occurs. Then, $C(\ell) = C(k_{i+1}) = \hat{C}(k_{i+1}) T_{i+1}$ where T_{i+1} is the similarity transformation satisfying $A(k_{i+1}) = T_{i+1}^{-1} \tilde{A}(k_{i+1}) T_{i+1}$, and $B(k_{i+1}) = T_{i+1}^{-1} \tilde{B}(k_{i+1})$. Thus, the quadratic cost has a minimum value of zero if $\mathcal{H}_{1,2n}(\ell) \tilde{\mathcal{R}}_{2n}^\dagger(\ell - 1) = \hat{C}(\ell) T_{i+1} T_i^{-1}$. Since T_i and T_{i+1} are not known, we can only check the condition $\mathcal{H}_{1,2n}(\ell) \tilde{\mathcal{R}}_{2n}^\dagger(\ell - 1) = \hat{C}(\ell)$ which holds if and only if $T_i = T_{i+1}$. But, the latter does not hold for randomly picked similarity transformations. In summary, $\ell = \varphi(k_{i+1})$ if and only if $\|\mathcal{H}_{1,2n}(\ell) \tilde{\mathcal{R}}_{2n}^\dagger(\ell - 1) - \hat{C}(\ell)\|_F > 0$.

Suppose now that the points $\ell < j \leq k_{i+1}$ were clustered to $\varphi(k_{i+1})$ and we want

to recluster ℓ to either $\varphi(k_i)$ or $\varphi(k_{i+1})$, *that is*, we iterate backward in time:

$$h(j, \ell) = \begin{cases} C(k_{i+1}) [A(k_{i+1})]^{j-\ell-1} B(\ell), & j > \ell \\ D(\ell), & j = \ell. \end{cases} \quad (4.10)$$

The linear estimator of $h(j, \ell)$ based on Eq.(4.10) is

$$\tilde{h}(j, \ell) = \begin{cases} \hat{C}(k_{i+1}) [\tilde{A}(k_{i+1})]^{j-\ell-1} \tilde{B}, & j > \ell \\ D(\ell), & j = \ell \end{cases} \quad (4.11)$$

where the pair $\tilde{B} = T_{i+1}B(\ell)$ and $\tilde{D} = D(\ell)$ is the unique solution of the linear least-squares minimization problem

$$\min_{\tilde{B}, \tilde{D}} \sum_{j=\ell}^{\ell+2n} \|\tilde{h}(j, \ell) - h(j, \ell)\|_F^2. \quad (4.12)$$

The normal equations for $\tilde{B}$ are $\mathcal{H}_{2n,1}(\ell+1) = \hat{\mathcal{O}}_{2n}(k_{i+1})\tilde{B}$ with $\hat{\mathcal{O}}_{2n}(k_{i+1})$ formed from the pair $\{\hat{C}(k_{i+1}), \tilde{A}(k_{i+1})\}$. Since $\varphi(\ell+1) = \varphi(k_{i+1})$ and $\mathcal{P}_{\varphi(k_{i+1})} \sim \tilde{\mathcal{P}}_{\varphi(k_{i+1})}$, we use the similarity relations to derive

$$\begin{aligned} \tilde{B} &= \hat{\mathcal{O}}_{2n}^\dagger(k_{i+1})^\dagger \mathcal{H}_{2n,1}(\ell+1) \\ &= \hat{\mathcal{O}}_{2n}^\dagger(k_{i+1})^\dagger \mathcal{O}_{2n}(\ell+1) B(\ell) = T_{i+1}B(\ell). \end{aligned} \quad (4.13)$$

It follows that $\hat{\mathcal{O}}_{2n}^\dagger(k_{i+1})^\dagger \mathcal{H}_{2n,1}(\ell+1) = T_{i+1}B(\ell)$. As in the forward iteration case, we can show that $\ell = \varphi(k_i)$ if and only if $\|\hat{\mathcal{O}}_{2n}^\dagger(k_{i+1})^\dagger \mathcal{H}_{2n,1}(\ell+1) - \tilde{B}(\ell)\|_F > 0$. The backward iterations are useful in particular to recluster points to left of the switches. The forward and the backward iterations may be used together to enhance detection of the switches.

Instead of the ℓ_1 -norm of the eigenvalues, one can use a favorable statistics of her or him. As an example, first recall that the basis choice in the LTV state-space realization is made via the two SVDs in Eq.(3.25). The MATLAB performs the SVD by utilizing a specific algorithm. Repeated calling of this algorithm generates a constant $A(k)$ matrix, hence a constant realization $\mathcal{P}_{\varphi(k)}$, on each subinterval where a submodel is active. Furthermore, as the active submodel changes, computations are performed with respect to a fixed basis. In summary, the LTV realization generated has a stable pattern of the state-space parameters which can be used for clustering. Then, we vectorize the state-space matrices $\tilde{A}(k)$, $\tilde{B}(k)$, $\hat{C}(k)$, and $\hat{D}(k)$ by defining

for $k = 2n + 1, \dots, N - 2n$

$$\nu_k = (\text{vec}^T(\tilde{A}(k)) \text{ vec}^T(\tilde{B}(k)) \text{ vec}^T(\hat{C}(k)) \text{ vec}^T(\hat{D}(k))) \quad (4.14)$$

where $\text{vec}(A)$ is the column vector obtained by stacking the columns of a given matrix A one on top of another and let $\mathcal{X} = [\nu_{2n+1}^T \dots \nu_{N-2n}^T]^T$. In the second example, we plug the $\mathcal{H}_2$ -norm of the time-varying transfer function matrix $\{\tilde{A}(k), \tilde{B}(k), \hat{C}(k), \hat{D}(k)\}$ in place of ν_k .

If the generalized Markov parameters are corrupted by noise, the realizations in $\{\tilde{\mathcal{P}}_{\varphi(k)} : \varphi(k) = \varphi(k_i)\}$ must be averaged to reduce the bias of $\tilde{\mathcal{P}}_{\varphi(k_i)}$. Likewise, estimation of $\tilde{C}$ and $\tilde{D}$ via Eq.(4.9) and Eq.(4.13) filters noise out.

Identifiability, controllability, and observability concepts for the SLSs are delicate issues when the submodels have different MacMillan degrees. For example, identifiability of a SLS does not imply identifiability of its submodels [117]. The key role played by the observability in the identification of SLSs were noted in several recent works [38, 117, 118, 119]. The observability part of Lemma 3.1.1 appeared as Theorem 2 in [38]. Extension of the Ho-Kalman realization theory for the LPV state-space models was reported in [119].

We summarize the results derived in this section in the form of an estimation algorithm.

Algorithm 4 LTV realization to SLS model Conversion

Input: The state space quadruplets $\{\tilde{A}(k), \tilde{B}(k), \hat{C}(k), \hat{D}(k)\}$, $2n + 1 \leq k \leq N - 2n$ obtained from Algorithm 2.

- ① Pick one statistics among the ℓ_1 -norm of $\lambda_i(\tilde{A}(k))$, $\tilde{\mathcal{P}}_{\varphi(k)}$ in vector form, and its time-varying $\mathcal{H}_2$ norm.
- ② Estimate $\tilde{\mathcal{P}}$ by running the density-based clustering algorithm [114] as described in Subsection 4.1.
- ③ Recluster the points in the domain of $\varphi(k)$ that are not attached to $\tilde{\mathcal{P}}$ as described in Subsection 4.2.

Output: The submodels $\tilde{\mathcal{P}}_{\varphi(k)}$ and the switching sequence $\varphi(k)$, $2n+1 \leq k \leq N-2n$.

The main result of this section is stated as follows.

Proposition 4.2.1 *Consider Algorithm 4 driven by the LTV realization of the SLS model in Eqs.(1.3)–(1.6) delivered by Algorithm 2. Suppose that Assumption 4.1.1 holds. Then, $\mathcal{P}$ and φ are recovered in the noiseless case.*

4.3. Example

Submodel estimation and switching function detection from noiseless Markov parameters of the SLS model is discussed in this numerical example. Consider the same SLS as in Section 3.5. A switching signal with length $N = 1,000$ is generated randomly from a uniform distribution. This is done while adhering to a minimum dwell time condition of at least $5n = 15 \text{ secs}$. Such switching signal is depicted in Figure 4.1.

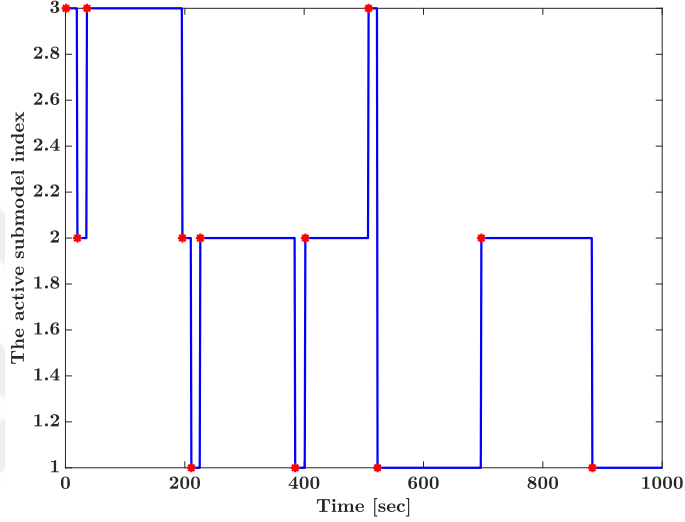


Figure 4.1: The switching sequence.

In Algorithm 2, we selected $q = r = 2n$, and obtain the LTV realizations on the interval $[2n + 1, N - 2n]$. Figure 4.2 shows $\mathcal{M}(\hat{A}(k))$ and $\mathcal{M}(A(k))$, respectively for the LTV realization and the SLS model. The mismatches occur exclusively around the switching points.

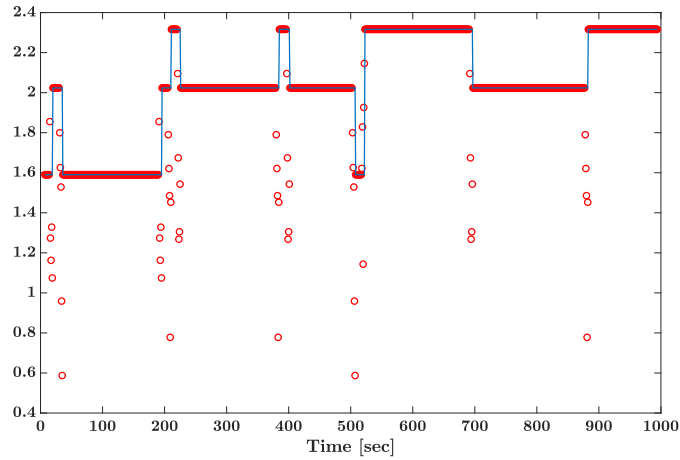


Figure 4.2: The ℓ_1 -norm of the eigenvalues: 'o' the LTV realization; '—' the SLS model.

We run the density-based clustering algorithm `dbscan` with the statistic $\mathcal{M}(\hat{A}(k))$ in our example. The two tuning parameters for `dbscan`, i.e., the threshold for neighborhood search radius `epsilon` and the minimum number of neighbors required to identify a cluster (submodel) `minpts` were chosen as 10^{-4} and 40 respectively via trial and error. The clustering algorithm gathers the dubious points into an outlier cluster labelled as -1 . The submodel clusters, on the other hand, are denoted by 1, 2, and 3. The histogram of the clusters is shown in Figure 4.3. The `dbscan` algorithm determines the number of clusters based on the tolerance values entered, in this case it indicates the presence of three clusters, similarly three sub-models, i.e., $\hat{\sigma} = 3$.

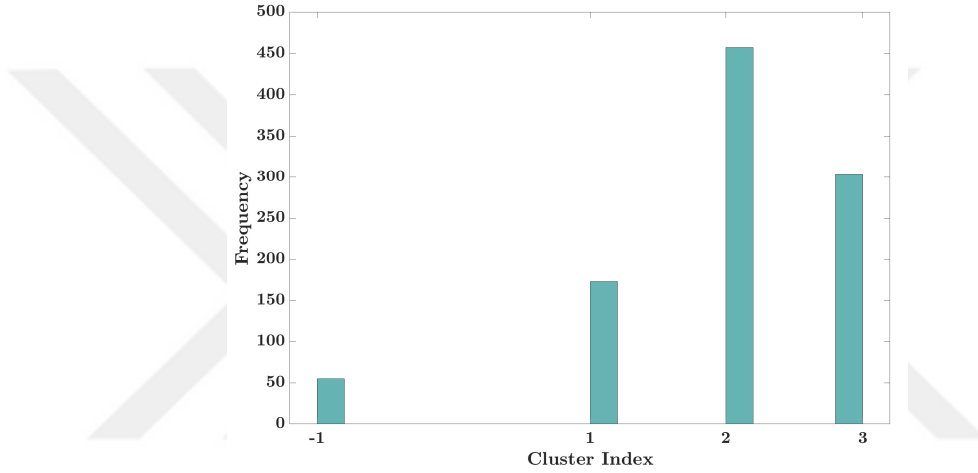


Figure 4.3: The histogram of the clusters from the `dbscan` algorithm.

Figure 4.4, on the other hand, shows how the clustering algorithm clusters the convenient points to their correct clusters, while the outliers, which are concentrated around the switching points, are all gathered in a separate cluster.

The points in the outlier cluster were redistributed into their convenient Clusters 1–3 as described in Step 3 of Algorithm 4. Figure 4.5 depicts $\mathcal{M}(\tilde{A}(k))$ and $\mathcal{M}(A(k))$, respectively for the re-clustered LTV realization and the SLS model, the transition bands merge into the neighbor constant eigenvalue intervals to yield a perfect estimate. The eigenvalues of the extracted submodels perfectly match the true submodel eigenvalues as shown in Figure 4.6. The switching sequence estimate after reclustering is plotted in Figure 4.7 with the true switching sequence. The estimate is in excellent agreement with the true sequence at every time instant. As a final confirmation, we plot the $\mathcal{H}_2$ norms of the time-varying transfer function matrix estimate $\hat{C}(k)(zI_n - \tilde{A}(k))^{-1}\tilde{B}(k) + \hat{D}(k)$ and the true transfer function matrix $C(k)(zI_n - A(k))^{-1}B(k) + D(k)$ in Figure 4.8. Recall that the $\mathcal{H}_2$ norm may also be used as a clustering statistic instead of the ℓ_1 norm of the eigenvalues.

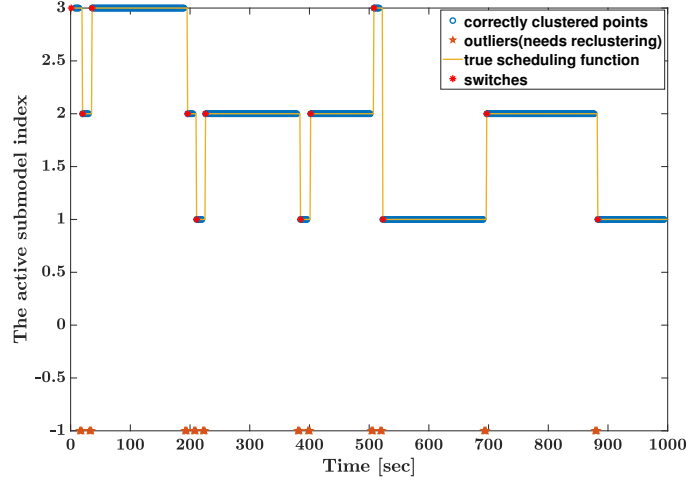


Figure 4.4: The division of points between correct clusters and outliers according to dbscan algorithm.

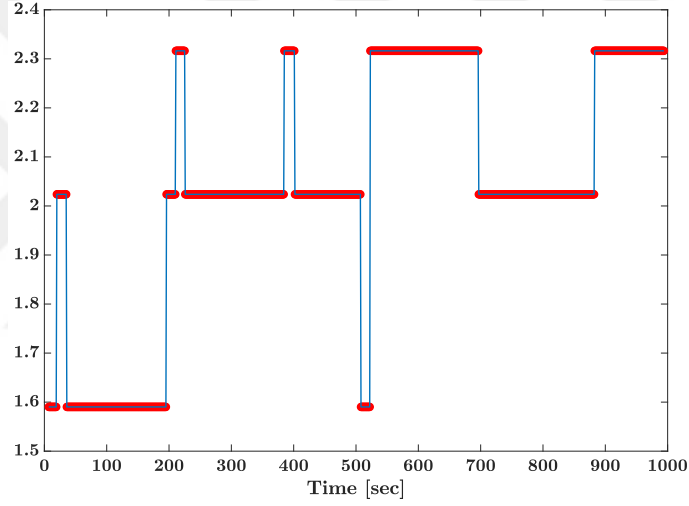


Figure 4.5: The ℓ_1 -norm of the eigenvalues: 'o' the re-clustered LTV realization; '—' the SLS model.

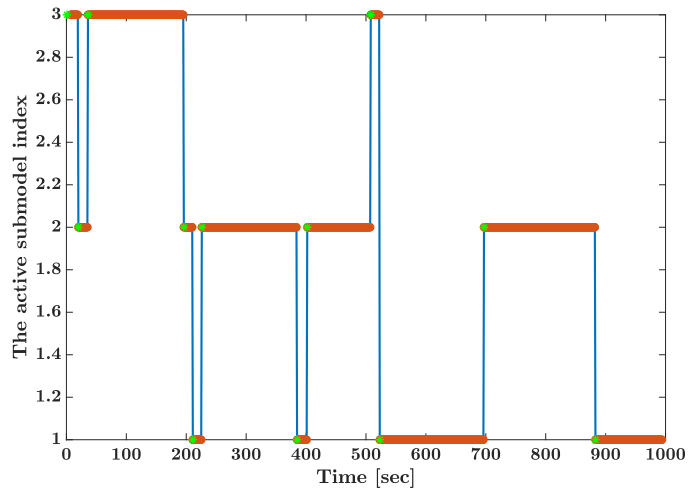


Figure 4.7: The scheduling sequence estimate after re-clustering.

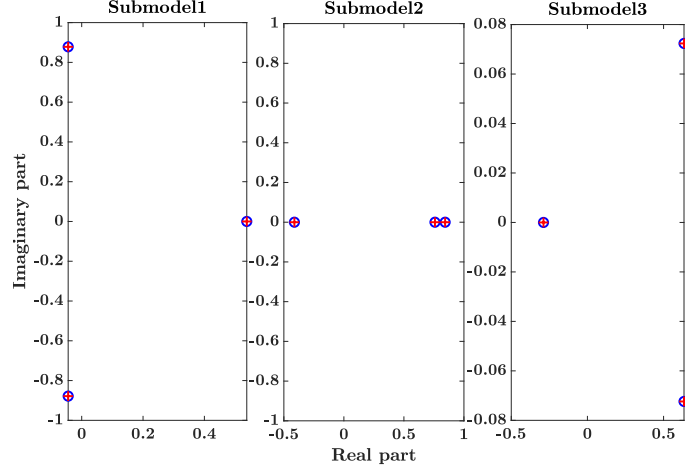


Figure 4.6: The submodel and the averaged re-clustered eigenvalues: 'o' the re-clustered eigenvalues; '+' the submodel.

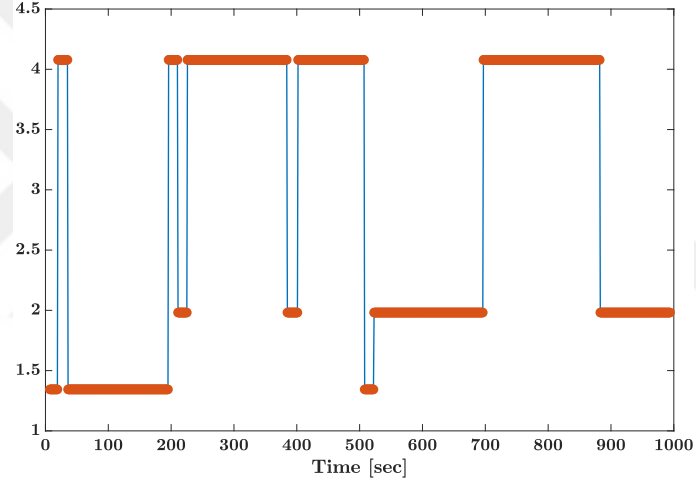


Figure 4.8: The $\mathcal{H}_2$ norms of the the time-varying transfer function matrix estimate $\hat{C}(k)(zI_n - \hat{A}(k))^{-1}\hat{B}(k) + \hat{D}(k)$ ('-') and the true transfer function matrix estimate $C(k)(zI_n - A(k))^{-1}B(k) + D(k)$ 'o'.

4.4. Summary

This chapter explored a two-stage algorithm to retrieve SLS submodels from their LTV realizations. The first step amounts to retrieving the set of submodels, based on the observation that the LTV quadruplets are similar to the SLS submodels on subintervals of long enough segments. To ensure all submodels are extractable, a dwell time constraint was imposed on the SLS submodels, to make sure that all the submodels have been visited with long enough dwell time. Submodel acquisition was accomplished via clustering a statistic of the LTV realization. Three types of statistics were suggested: the ℓ_1 -norm of the system eigenvalues, the $\mathcal{H}_2$ norm of the time-varying transfer matrix, and the vectorized state-space quadruplets. The clus-

tering was achieved via the DBSCAN Algorithm. Subsequently, the set of estimated submodels were exploited to find an estimation of the switching sequence. This was achieved by assigning the vague points to their respective generating submodel via solving discrete optimization problems, one of which hinges on the ℓ_1 -norm of the system eigenvalues statistics matching, whereas the other iterates either backward or forward in time and is based on Markov parameters matching. In the upcoming chapter, we lay down switch and submodel identifiability conditions for the set of MIMO SARX models.



5. SWITCH AND SUBMODEL IDENTIFIABILITY OF SARX MODELS FROM INPUT-OUTPUT DATA

We start this chapter by detailing switch identifiability conditions for the set of MIMO SARX models. The results reported in Section 5.1 are an extension to the ones in [120] to fit the MIMO configuration adopted in here. Next, Section 5.2 presents a set of technical lemmas that links the segments in SARX models to the segments in SLS models. Section 5.3 is concerned with the case when a SISO SARX is considered instead, remarks were made on how that is reflected on the results established in the earlier two sections. Subsequently, in Section 5.4 we tackle persistence of excitation conditions for inputs for MIMO SARX models via matrix fraction descriptions. Lastly, Section 5.5 discusses PE for the case of SISO SARX models in a straightforward and simplified manner. This chapter is concluded via two examples, one for a MIMO SLS model and the other for a SISO SLS model, both intended to illustrate the results of this chapter.

5.1. Switch identifiability for MIMO SARX models

We start by transforming the time-varying ARX model in Eq.(3.50) into a more familiar matrix equation form.

For $2n < k \leq N - 2n + 1 = N_1$, let us rewrite Eq.(3.50) from Eq.(3.36) and Eq.(3.41) as

$$y(k) = D(k)u(k) + \sum_{j=1}^{2n} h_o(k, k-j)\zeta(k-j) + e(k) \quad (5.1)$$

and stack the ARX coefficients into a compound matrix

$$\theta(k) = [D(k) \ h_o(k, k-1) \ \dots \ h_o(k, k-2n)] \in \mathbb{R}^{p \times n_z} \quad (5.2)$$

with $n_z = 2n(p+m) + m$. Thus, we can write Eq.(5.1) in the form

$$y(k) = \theta(k) z(k) + e(k), \quad 2n < k \leq N_1 \quad (5.3)$$

where the regressor matrices are defined by

$$z(k) = (u^T(k) \ \zeta^T(k-1) \ \dots \ \zeta^T(k-2n))^T \in \mathbb{R}^{n_z}. \quad (5.4)$$

Evaluate Eq.(5.3) for $2n < k \leq N_1$ and stack them to get

$$Y = Z\Theta + E \quad (5.5)$$

where

$$\begin{aligned} Y &= [y(n_1) \cdots y(N_1)]^T, \\ E &= [e(n_1) \cdots e(N_1)]^T, \\ \Theta &= [\theta(n_1) \cdots \theta(N_1)]^T, \\ Z &= \begin{bmatrix} z^T(n_1) & & 0 \\ & \ddots & \vdots \\ 0 & \cdots & z^T(N_1) \end{bmatrix}. \end{aligned} \quad (5.6)$$

The signal model in Eq.(5.5) is known as the multiple measurement vectors (MMV) model studied in [76, 106]. The frameworks presented in these works recover the MMV models that share a joint sparsity pattern by convex algorithms provided that Θ is sparsely populated and either of block-coherence, block-restricted isometry property, and block-exact recovery condition of Z is sufficiently small. The block property of Eq.(5.5) is inherited from joint parameterization of the outputs. We will first address the issue of switch identifiability in SARX models from input-output data in a noiseless setup. This study will reveal persistence of excitation conditions.

Recall that $2n$ is an upper bound on τ^* , which is not known *a priori*. If $\tau^* < 2n$, $h_o(k, k - \ell) = 0$ for all $\tau^* < \ell \leq 2n$ and as a result, $2n - \tau^*$ block column entries of $\theta(k)$ with size $\mathbb{R}^{p \times (p+m)}$ are zeros. The switches of the model in Eq.(5.3) are the time instants k satisfying

$$\delta_\theta(k) = \theta(k) - \theta(k-1) \neq 0. \quad (5.7)$$

They are generated from the SLS model in Eqs.(1.3)–(1.6) under the observer-based transformation studied in Section 3.2. Let $\varphi_\theta(k)$ and $\{s_1, \dots, s_{j^*}\} \subset [n_1, \dots, N_1]$ denote the switching sequence and its range where we set $s_0 = 2n$ so that $s_0 < s_j \leq N_1$ for all $j \leq j^*$. To the segmentation of the interval $[n_1, N_1]$ by $\varphi_\theta(k)$ denoted by χ_θ , *i.e.*, covering by semi-closed disjoint intervals as in Eq.(1.5), we allocate a dwell sequence and the minimum dwell time by

$$\begin{aligned} \delta(\chi_\theta) &= \{s_{j+1} - s_j : j = 0, \dots, j^* - 1\} \\ \delta_*(\chi_\theta) &= \min \delta(\chi_\theta). \end{aligned} \quad (5.8)$$

The submodel set for the model in Eq.(5.3) is $\mathcal{P}_\theta = \{\theta(k) : n_1 \leq k \leq N_1\}$. Let σ_θ denote the cardinality of $\mathcal{P}_\theta$. We postpone the identification of χ_θ and $\mathcal{P}_\theta$ from input-output data to later chapters but discuss the identifiability conditions here. The following definition of the switch identifiability is adapted from [120] to our MIMO setting.

Definition 5.1.1 *Consider the SARX model in Eq.(5.3). The switches in the interval $(2n, N_1]$ is causally identifiable if whenever $\varphi(k) \neq \varphi(k-1)$, it is possible to detect the change in the value of $\varphi(k)$ as soon as $y(k)$ is observed.*

A related definition is the *submodel identifiability*. Given the current regressor vector $z(k)$, two submodels with parameters $\theta_1 \in \mathbb{R}^{p \times n_z}$ and $\theta_2 \in \mathbb{R}^{p \times n_z}$ are one-step indistinguishable from $z(k)$ if $(\theta_1 - \theta_2)z(k) = 0$ [120].

Given $t, s \in \mathbb{N}$ with $n_1 \leq s \leq t \leq N_1$, we define

$$\mathcal{R}_{s,t} = [z(s) \ z(s+1) \ \cdots \ z(t)], \quad (5.9)$$

$$Y_{s,t} = [y(s) \ y(s+1) \ \cdots \ y(t)]. \quad (5.10)$$

Lemmas 5.1.1–5.1.2 and their proofs below are due to [120]. They were adapted to the MMV model and the notation in this section. In the MMV case, the generalized system Markov parameters are matrix-valued when output sequences are vector-valued. Furthermore, $D(k)$ may be nonzero. We denote the linear space spanned by the column vectors and the rank of a given matrix X by $\text{range}(X)$ and $\text{rank}(A)$.

Lemma 5.1.1 *Consider the SARX model in Eq.(5.3). If s_j and s_{j+1} are switches in $[n_1, N_1]$ and $z(s_{j+1}) \in \text{range}(\mathcal{R}_{s_j, s_{j+1}-1})$, then there exists a $\gamma \in \mathbb{R}^p$ such that $(\theta_2 - \theta_1)z(s_{j+1}) = \gamma$ for all submodel parameter pairs $\{\theta_1, \theta_2\}$ satisfying*

$$\begin{aligned} Y_{s_j, s_{j+1}-1} &= \theta_1 \mathcal{R}_{s_j, s_{j+1}-1} \\ y(s_{j+1}) &= \theta_2 z(s_{j+1}). \end{aligned} \quad (5.11)$$

Proof. See Appendix A

Lemma 5.1.1 is Lemma 1 in [120] adapted to the MIMO setting here.

Lemma 5.1.2 *Consider the SARX model in Eq.(5.3). The switches in $[n_1, N_1]$ are causally identifiable from the input-output data if and only if the following two conditions hold for all switch pairs $\{s_j, s_{j+1}\} \subset [n_1, N_1]$ and all submodel parameter*

pairs $\{\theta_1, \theta_2\} \subset \mathbb{R}^{p \times n_z}$

$$\begin{aligned} (\theta_2 - \theta_1) z(s_{j+1}) &\neq 0, \\ z(s_{j+1}) &\in \text{range}(\mathcal{R}_{s_j, s_{j+1}-1}). \end{aligned} \quad (5.12)$$

Proof. See Appendix A

Lemma 5.1.2 appeared as Theorem 2 in [120]. Changes were made not only to conform with the notation used here, but also the fact that the generalized Markov parameters in our setting are matrix-valued. Since $\mathcal{R}_{s_j, s_{j+1}-1} \in \mathbb{R}^{n_z \times \delta_j(\chi_\theta)}$, the price paid for overparameterization $\tau^* < 2n$ is increased likelihood for a switch to go undetected. The following was stated in [120].

Lemma 5.1.3 *Consider the SARX model in Eq.(5.3) with the switches s_j and s_{j+1} in $[n_1, N_1]$. Suppose that $\mathcal{R}_{s_j, s_{j+1}}$ is not full-column rank. Then, s_{j+1} is causally identifiable from the input/output data if and only if*

$$\text{rank}([Y_{s_j, s_{j+1}}^T \quad \mathcal{R}_{s_j, s_{j+1}}^T]) > \text{rank}(\mathcal{R}_{s_j, s_{j+1}}^T). \quad (5.13)$$

Proof. See Appendix A

The results derived in this section can also be stated in the reverse order by replacing $z(s_{j+1})$ and $\mathcal{R}_{s_j, s_{j+1}-1}$ with z_{s_j-1} and $\mathcal{R}_{s_{j+1}-1, s_j}$ in Eq.(5.12) to check identifiability of s_j . From the second condition in Eq.(5.12), for switch detectability we derive a necessary condition on the inputs

$$u(s_{j+1}) = \text{range}(\{u(s_{j+1}-1), u(s_{j+1}-2), \dots\}) \quad (5.14)$$

when $D(k) \neq 0$ by recursively calculating $\zeta(k)$ from Eq.(5.1) and Eq.(3.37) with $e(k) = 0$ for all k . It follows that when the SLS model in Eqs.(1.3)–(1.6) is not strictly proper, white-noise excitations cannot be used to detect the switches.

Given an input-output data set, checking the conditions in Eq.(5.12) is an NP-hard problem. An additional difficulty is that these conditions hinge on the unknown switching sequence and the submodels which are not known *a priori* to the experimenter. In the condition given by Eq.(5.12), we will consider two extreme cases depending on the rank of $\mathcal{R}_{k_j, k_{j+1}}$. To this end, first from Eq.(5.9)

$$\mathcal{R}_{s,t} \mathcal{R}_{s,t}^T = \sum_{k=s}^t z(k) z^T(k). \quad (5.15)$$

Then, we transfer Eq.(5.15) to a more familiar form by defining

$$\begin{aligned} z_r(k) &= (u_a^T(k) \ y_a^T(k))^T, \\ u_a(k) &= (u^T(k) \ u^T(k-1) \ \cdots \ u^T(k-2n))^T, \\ y_a(k) &= (y^T(k-1) \ \cdots \ y^T(k-2n))^T, \\ J_z &= \begin{bmatrix} I_m & 0 \\ 0 & \begin{bmatrix} I_{2n} \otimes \begin{bmatrix} I_m \\ 0 \end{bmatrix} & I_{2n} \otimes \begin{bmatrix} 0 \\ I_p \end{bmatrix} \end{bmatrix} \end{bmatrix}. \end{aligned}$$

where $A \otimes B$ denotes the Kronecker product of two given matrices A and B such that,

$$A \otimes B := \begin{pmatrix} a_{11}B & a_{12}B & \cdots & a_{1m}B \\ a_{21}B & a_{22}B & \cdots & a_{2m}B \\ \vdots & \vdots & \ddots & \vdots \\ a_{n1}B & a_{n2}B & \cdots & a_{nm}B \end{pmatrix}. \quad (5.16)$$

Then, $z(k) = J_z z_r(k)$ with

$$J_z^{-1} = \begin{bmatrix} I_m & 0 \\ 0 & \begin{bmatrix} I_{2n} \otimes [I_m \ 0] \\ I_{2n} \otimes [0 \ I_p] \end{bmatrix} \end{bmatrix}.$$

For the first case, assume that $\text{range}(\mathcal{R}_{s,t}) = \mathbb{R}^{n_z}$ implying $\mathcal{R}_{s,t} \mathcal{R}_{s,t}^T$ positive-definite and $t - s + 1 \geq n_z$. If this assumption is valid for all switches in $[n_1, N_1]$, then the second condition in Eq.(5.12) becomes redundant and the only necessary and sufficient condition for the switch identifiability is

$$(\theta_{s_{j+1}} - \theta_{s_j})z(s_{j+1}) \neq 0, \quad 2n < s_j < s_{j+1} \leq N_1. \quad (5.17)$$

Plugging s_j , $s_{j+1} - 1$, and $z(k) = J_z z_r(k)$ in s , t , and $z(k)$ in Eq.(5.15), we see that $\mathcal{R}_{s_j, s_{j+1}-1} \mathcal{R}_{s_j, s_{j+1}-1}^T > 0$ implies

$$\sum_{k=s_j}^{s_{j+1}-1} z_r(k) z_r^T(k) > 0, \quad 2n < s_j < s_{j+1} \leq N_1. \quad (5.18)$$

If Eqs.(1.3)–(1.4) are an LTI system, deterministic or stochastic inputs which satisfy Eq.(5.18) can be found provided that the number of terms in the sum is large enough [113], *that is*, the minimum dwell time is sufficiently large in the context of this

paper. Notice that Eq.(5.18) is equivalent to

$$R_{z_r}(s_j) = \begin{pmatrix} R_{u_a}(s_j) & R_{u_a y_a}(s_j) \\ R_{u_a y_a}^T(s_j) & R_{y_a}(s_j) \end{pmatrix} > 0 \quad (5.19)$$

where the auto/cross covariance function estimates of $u_a(k)$ and $y_a(k)$ are defined at a switch s_j by

$$R_{u_a}(s_j) = \delta_j^{-1}(\chi_\theta) \sum_{k=s_j}^{s_{j+1}-1} u_a(k) u_a^T(k), \quad (5.20)$$

$$R_{y_a}(s_j) = \delta_j^{-1}(\chi_\theta) \sum_{k=s_j}^{s_{j+1}-1} y_a(k) y_a^T(k), \quad (5.21)$$

$$R_{u_a y_a}(s_j) = \delta_j^{-1}(\chi_\theta) \sum_{k=s_j}^{s_{j+1}-1} u_a(k) y_a^T(k). \quad (5.22)$$

with $\delta_j(\chi_\theta) = s_{j+1} - s_j$. Recall the Schur's complement formula [54]:

$$\begin{bmatrix} A & B \\ B^T & D \end{bmatrix} > 0 \iff D - B^T A^{-1} B > 0. \quad (5.23)$$

Applying Eq.(5.23) to Eqs.(5.19)–(5.22), we derive the persistence of excitation condition

$$R_{y_a}(s_j) - R_{u_a y_a}^T(s_j) R_{u_a}^{-1}(s_j) R_{u_a y_a}(s_j) > 0 \quad (5.24)$$

for $2n < s_j < s_{j+1} \leq N_1$. Denote the square roots of $R_{y_a}(s_j)$ and $R_{u_a}(s_j)$ by $Q_{y_a}(s_j)$ and $Q_{u_a}(s_j)$. They satisfy $Q_{y_a}(s_j) Q_{y_a}^T(s_j) = R_{y_a}(s_j)$ and $Q_{u_a}(s_j) Q_{u_a}^T(s_j) = R_{u_a}(s_j)$ and can be obtained from the SVDs of $R_{y_a}(s_j)$ and $R_{u_a}(s_j)$. Multiplying Eq.(5.24) from the left with $Q_{y_a}^{-1}(s_j)$ and from the right with $Q_{y_a}^{-T}(s_j)$ where X^{-T} denotes $[X^T]^{-1}$ for a given nonsingular matrix X , with $C_{u_a y_a}(s_j) = Q_{u_a}^{-1}(s_j) R_{u_a y_a}(s_j) Q_{y_a}^{-T}(s_j)$ we get $I > C_{u_a y_a}^T(s_j) C_{u_a y_a}(s_j)$ which is equivalent to the spectral norm of $C_{u_a y_a}(s_j)$ bounded above by one:

$$\bar{\sigma}\{C_{u_a y_a}(s_j)\} < 1 \quad (5.25)$$

where $\bar{\sigma}(A)$ denotes the largest singular value of a given matrix A . By calculating the largest singular value of $C_{u_a y_a}(s_j)$ for each j , the condition in Eq.(5.25) can be checked if it is true. It is exactly the same PE condition for the LTI systems in system identification literature except that it applies to finite data sets.

The condition in Eq.(5.18) or Eq.(5.25) derived from the first may be difficult to enforce since it requires $s_{j+1} - s_j \geq n_z$. In fact, a much longer dwell time may be needed to guarantee $\mathcal{R}_{s_j, s_{j+1}-1} \mathcal{R}_{s_j, s_{j+1}-1}^T > 0$. The second alternative is to directly

check if the second condition in Eq.(5.12) is satisfied. If this condition is satisfied, then $\text{range}(\mathcal{R}_{s_j, s_{j+1}-1}) = \text{range}(\mathcal{R}_{s_j, s_{j+1}})$, in particular they have the same number of nonzero singular values.

Let $\lambda_A(i)$, $1 \leq i \leq n$ denote the eigenvalues of a given nonnegative definite matrix $A \in \mathbb{R}^{n \times n}$. The sparsity level of λ_A denoted by $\|\lambda_A\|_{\ell,0}$ is the number of nonzero elements in the sequence $\lambda_A(i)$, $1 \leq i \leq n$. Hence, the second condition in Eq.(5.12) is equivalent to

$$\|\Sigma_{\mathcal{R}_{s_j, s_{j+1}-1}}\|_{\ell,0} = \|\Sigma_{\mathcal{R}_{s_j, s_{j+1}}}\|_{\ell,0}. \quad (5.26)$$

The first condition in Eq.(5.12) is satisfied by assuming that the regressors are not orthogonal to the parameter changes on the set $\chi_\theta \cap [n_1, N_1]$, i.e., for all $s_j, s_{j+1} \in [n_1, N_1]$

$$z(s_{j+1}) \nperp (\theta(s_{j+1}) - \theta(s_j)). \quad (5.27)$$

This condition is not as stringent as the one in Eq.(5.25) since almost all choices of randomly generated inputs satisfy (5.27). Suppose that Eq.(5.25) or Eq.(5.26) and Eq.(5.27) hold. Then, from Lemma 5.1.2 a switch $s_j \in \chi_\theta \cap [n_1, N_1]$ is causally identifiable if and only if $\theta(s_{j+1}) - \theta(s_j) \neq 0$ as summarized in the following.

Proposition 5.1.1 *Consider the SARX model in Eq.(5.3). Assume that the input-output data satisfy Eqs.(5.26) and (5.27). Then, the switches in $\chi_\theta \cap [n_1, N_1]$ are causally identifiable from input/output data if and only if $\theta(s_{j+1}) - \theta(s_j) \neq 0$ for all pairs $\{s_j, s_{j+1}\} \subset \chi_\theta \cap [n_1, N_1]$.*

The results derived in this subsection will be illustrated via a numerical example using multisine excitations. We explored in this section the connections among excitation signals, SARX models, and switch identifiability.

5.2. The relation between segments in SLS and segments in SARX

In this section, we present several technical lemmas which link the segments in the SLS model given by Eqs.(1.3)–(1.5) to the segments in the SARX model in Eq.(5.3) and vice versa.

Lemma 5.2.1 *Consider the SLS model in Eqs.(1.3)–(1.5). Let the observer be as in Lemma 3.2.1. If $k_i, k_{i+1} \in \chi$ and $\delta_i(\chi) \geq n_1 = 2n + 1$, then $[k_i + 2n, k_{i+1}] \subseteq [s_j, s_{j+1})$ for some $s_j, s_{j+1} \in \chi_\theta$.*

Proof. See Appendix A

A converse statement is also true. But, first we need an auxiliary result that holds in the more general LTV setting without requiring the observer be as in Lemma 3.2.1.

Lemma 5.2.2 *Consider the SLS model in Eqs.(1.3)–(1.5) and the observer in Eq.(5.3). If $s_j, s_{j+1} \in \chi_\theta$, then $h(k, k-v) = h(l, l-v)$ for all $k, l \in [s_j + 2n, s_{j+1})$ and $0 \leq v < \min\{k, l\}$ and extending to $[s_j, s_{j+1})$ if $D(k) = D$ for all k . Moreover, for all k, l in $[s_j, s_{j+1})$ and $0 \leq v < \min\{k, l\}$, $h_m(k, k-v) = h_m(l, l-v)$.*

Proof. See Appendix A

Let $\mathcal{H}_{\alpha\beta}(k)$ denote the entry of $\mathcal{H}(k)$ in Eq.(3.4) with the p sized block-row and m sized block-column indices α, β . If $k \geq s_j + 2n$, since $\alpha, \beta \leq 2n$ from Lemma 5.2.2 we have

$$\begin{aligned}\mathcal{H}_{\alpha\beta}(k) &= h(k + \alpha - 1, k - \beta) \\ &= h(k + \alpha, k + 1 - \beta) = \mathcal{H}_{\alpha\beta}(k + 1).\end{aligned}$$

A requirement based on the second equality above is derived as $k < s_{j+1} - 2n$. Hence, if $s_j + 2n \leq k < s_{j+1} - 2n$, then $\mathcal{H}(k) = \mathcal{H}(k+1)$ and $D(k) = D(k+1)$. These properties are instrumental in showing that the LTV system has an LTI realization on a subinterval of $[s_j, s_{j+1})$. This result is not immediate from Lemma 5.2.2. Algorithm 2 estimates $A(k), B(k), C(k), D(k)$ in Eqs.(1.3)–(1.5) up to a topological equivalence from the pair $\mathcal{H}(k)$ and $\mathcal{H}(k+1)$. Since $T(k) = T(k+1)$, $\varphi(k) = \varphi(s_j + 2n)$ for all k in the interval $[s_j + 2n, s_{j+1} - 2n)$. Let

$$k_i = \max_{k \in \chi} \{k \leq s_j + 2n\}, \quad k_\ell = \min_{k \in \chi} \{k \geq s_{j+1} - 2n\}.$$

Suppose $\ell > i + 1$. From the definitions of k_ℓ and k_i , we get $k_i \leq s_j + 2n \leq k_{i+1} < s_{j+1} - 2n \leq k_\ell$. Since $[k_i, k_{i+1})$ and $[s_j + 2n, s_{j+1} - 2n)$ have a nonempty intersection, $\varphi(k)$ extends to $[k_i, s_{j+1} - 2n)$. Then, $k_{i+1} \notin \chi$ and we reach a contradiction. It follows that $\ell = i + 1$ and $[s_j, s_{j+1})$ is a subset of $[k_i - 2n, k_{i+1} + 2n)$.

The observer gain $K(k-1)$ is estimated from Eq.(3.60) using $\hat{C}(\ell)$, $\hat{A}(\ell)$ for $\ell = k, \dots, k+2n-2$ and $\hat{C}(k+2n-1)$. But, $C(k+2n-1)$ is estimated by Algorithm 2 from $\mathcal{H}(k+2n-1)$ and $\mathcal{H}(k+2n)$. Thus, if $s_j + 2n \leq k \leq s_{j+1} - 4n$ and $q = 2n$, the left-hand side of Eq.(3.59) does not depend on k and $\hat{K}(k-1)$ is constant. We summarize these results in the following.

Lemma 5.2.3 *Consider the SLS model in Eqs.(1.3)–(1.5) and the observer in Eq.(5.3). If $s_j, s_{j+1} \in \chi_\theta$ and $\delta_j(\chi_\theta) > 4n$, then for some $k_i, k_{i+1} \in \chi$, $[s_j +$*

$2n, s_{j+1} - 2n) \subseteq [k_i, k_{i+1})$. Moreover, if $\delta_j(\chi_\theta) \geq 6n$, then $\hat{K}(k-1)$ is constant on $[s_j + 2n, s_{j+1} - 4n]$.

Suppose s_j, s_{j+1} , and s_{j+2} are identifiable from the input-output data. A change in the parameter value at s_{j+1} is then observable. This does not mean that the parameter estimates in $[s_j, s_{j+1})$ and $[s_{j+1}, s_{j+2})$ are unbiased. All one can assert is that the switches are detectable from the input-output data. We state $\chi \subset \chi_\theta$ as a standing assumption.

Assumption 5.2.1 *Every switch of χ is also a switch of χ_θ .*

The converse statement, *i.e.*, $k \in \chi_\theta \Rightarrow k \in \chi$ is not true. Using Assumption 5.2.1, we improve Lemma 5.2.3 as follows:

Lemma 5.2.4 *Consider the SLS model in Eqs.(1.1)–(1.5) and the observer in Eq.(5.3). Suppose that Assumption 5.2.1 holds. Let $s_j, s_{j+1} \in \chi_\theta$ be with $\delta_j(\chi_\theta) > 4n$. Then, there exist $k_i, k_{i+1} \in \chi$ satisfying $0 \leq s_j - k_i \leq 2n$ and $k_{i+1} = s_{j+1}$.*

Proof. See Appendix A

We will state two assumptions which together with imply Assumption 5.2.1.

Assumption 5.2.2 *Let $B_u(k) = B(k) + K(k)D(k)$ where the gain sequence $K(k)$ is as in Lemma 3.2.1. For all $k_i \in \chi$, $\{A(k_i) + K(k_i)C(k_i), B_u(k_i), C(k_i), D(k_i)\}$ and $\{A(k_i) + K(k_i)C(k_i), -K(k_i), C(k_i), 0\}$ are minimal realizations.*

Lemma 5.2.5 *Suppose Assumptions 3.1.1 and 5.2.2 hold. Then, with $\kappa_c = \kappa_o = 2n$ and the gains in Lemma 3.2.1, Eq.(5.3) is uniform on $[n_1, N_1]$.*

Proof. This lemma is actually a corollary of Lemma 3.1.1.

Assumption 5.2.3 *At least one of the following*

$$i. \varphi(k) \neq \varphi(l) \Leftrightarrow C(k) \neq C(l) \text{ and/or } D(k) \neq D(l)$$

$$ii. \varphi(k) \neq \varphi(l) \Leftrightarrow C(k) \neq C(l)$$

holds for all $k, l \in [n_1, N_1]$.

Lemma 5.2.6 *Consider the SLS model in Eqs.(1.3)–(1.5) with the observer in Eq.(5.3). Suppose for all $1 < i \leq i^*$, $\delta_{i-1}(\chi) \geq n_1$. Then, Assumptions 5.2.2 and 5.2.3 together imply Assumption 5.2.1.*

Proof. See Appendix A

In the sequel, we address the situation where a SISO SARX is considered. The impact of that on the so far established results is broached, and some adjustments are introduced.

5.3. Switch identifiability conditions & further remarks for SISO SARX models

In the case the set of SISO systems in Eqs.(1.1-1.2) is considered, the model in Eq.(3.50) could be written as

$$y(k) = z^T(k)\theta(k) + e(k), \quad 2n < k \leq N_1, \quad (5.28)$$

with

$$\begin{aligned} \theta(k) &= (d(k) h_o(k, k-1) \cdots h_o(k, k-2n))^T \in \mathbb{R}^{n_z}, \\ z(k) &= (u(k) \zeta^T(k-1) \cdots \zeta^T(k-2n))^T \in \mathbb{R}^{n_z} \end{aligned} \quad (5.29)$$

with $n_z = 4n + 1$. Evaluate Eq.(5.28) for $n_1 \leq k \leq N_1$ and stack them to get

$$Y = Z\Theta + E \quad (5.30)$$

where

$$\begin{aligned} Y &= [y(n_1) \cdots y(N_1)]^T, \\ E &= [e(n_1) \cdots e(N_1)]^T, \\ \Theta &= [\theta^T(n_1) \cdots \theta^T(N_1)]^T, \\ Z &= \begin{bmatrix} z^T(n_1) & 0 & \\ & \ddots & \vdots \\ 0 & \cdots & z^T(N_1) \end{bmatrix}. \end{aligned} \quad (5.31)$$

In this case, switch identifiability conditions has been derived already, the previous lemmas related to switch identifiability and their proofs could be found in [120]; Proposition 5.1.1 could be stated similarly to SISO models. The PE condition at Eq.(5.25) applies as well to the SISO systems identically.

Write Eq.(5.28) as a sum of the filtered inputs and the outputs as follows

$$\begin{aligned} y(k) &= y^{(1)}(k) + y^{(2)}(k) \\ &= H_o^{(1)}(q; k)u(k) + H_o^{(2)}(q; k)y(k) \end{aligned} \quad (5.32)$$

where with $b_o^{(1)}(k) = b(k) + g(k)d(k)$ and $b_o^{(2)}(k) = -g(k)$,

$$H_o^{(1)}(z; k) = c^T(k)(zI_n - A_o(k))^{-1}b_o^{(1)}(k) + d(k), \quad (5.33)$$

$$H_o^{(2)}(z; k) = c^T(k)(zI_n - A_o(k))^{-1}b_o^{(2)}(k). \quad (5.34)$$

From Eq.(5.34), observe that the second term in Eq.(5.32) is a linear combination of the past outputs only. Assumption 5.2.2 customized for SISO SLS models:

Assumption 5.3.1 *Let $g(k)$ be as in Lemma 3.2.1. For $\ell = 1, 2$ and all $k \in [n_1, N_1]$, $H_o^{(\ell)}(z; k)$ are minimal transfer functions.*

As k varies over $[n_1, N_1]$ or χ , each collection in Eq.(5.33) and Eq.(5.34) contains σ transfer functions of MacMillan degree n . We restate lemma 5.2.5 for SISO SLS models,

Lemma 5.3.1 *Suppose Assumptions 3.1.1 and 5.3.1 hold. Then, on $[n_1, N_1]$ with $\kappa_c = \kappa_o = 2n$ and $g(k)$ in Lemma 3.2.1, $H_o^{(1)}(q; k) : u \mapsto y^{(1)}$ and $H_o^{(2)}(q; k) : y \mapsto y^{(2)}$ are uniform.*

Lemmas (5.2.1, 5.2.2, 5.2.3, 5.2.4, 5.2.6) and their proofs apply to SISO case verbatim, a slight change of notation is needed. Assumption 5.2.1 is identical to the SISO case.

Next, we address the issue of submodel identifiability from the input-output data using persistently exciting inputs. Persistence of excitation conditions for MIMO SARX systems are first laid out.

5.4. Persistence of excitation of the input sequence for MIMO SARX systems

Consider a segment $[k_i, k_{i+1})$ in χ . From Lemma 5.2.4, there exist $s_j, s_{j+1} \in \chi_\theta$ satisfying $0 \leq s_j - k_i \leq 2n$ and $s_{j+1} = k_{i+1}$. Set $\alpha_j = s_j + 2n$ and $\beta_j = s_{j+1} - 4n$. Then, $\theta(k) = \theta(\alpha_j)$ for all $k \in [\alpha_j, \beta_j]$ and from Lemma 5.2.3, the observer is LTI on $[\alpha_j, \beta_j]$. From Eq.(5.3), $\theta(\alpha_j)\mathcal{R}_{\alpha_j, \beta_j} = Y_{\alpha_j, \beta_j}$. If $\mathcal{R}_{\alpha_j, \beta_j}$ has full row-rank, we can calculate $\theta(\alpha_j)$ by minimizing the squared-sum of the prediction errors $\|Y_{\alpha_j, \beta_j} - \theta\mathcal{R}_{\alpha_j, \beta_j}\|_2^2$.

The full row-rank assumption is precisely the persistence of excitation condition on the input sequence to extract the submodel information. A persistence of excitation condition is typically imposed on the input sequence given a specific model set and a particular identification algorithm. Rewrite Eq. (3.43) for some $\tau \leq n$ as

$$y(k) + \sum_{i=1}^{\tau} h_m(i)y(k-i) = \sum_{i=0}^{\tau} h_u(i)u(k-i) \quad (5.35)$$

with $h_u(0) = D$ where for $i \geq 1$ from Eq.(3.48),

$$\begin{aligned} h_m(i) &= C(A + KC)^{i-1}K, \\ h_u(i) &= C(A + KC)^{i-1}(B + KD). \end{aligned} \quad (5.36)$$

Transform Eq.(5.35) to the frequency domain

$$\mathcal{Y}(q) = G_L(q)\mathcal{U}(q) + \mathcal{V}(q) \quad (5.37)$$

where $\mathcal{U}(q)$, $\mathcal{Y}(q)$ are the input-output q -transforms,

$$G_L(q) = \mathcal{D}_L^{-1}(q)\mathcal{N}_L(q) \quad (5.38)$$

is the left matrix-fraction description (MFD) of $G_L(q)$ with

$$\begin{aligned} \mathcal{D}_L(q) &= I_p + q^{-1}h_m(1) + \cdots + q^{-\tau}h_m(\tau) \\ \mathcal{N}_L(q) &= h_u(0) + q^{-1}h_u(1) + \cdots + q^{-\tau}h_u(\tau), \end{aligned} \quad (5.39)$$

and $\mathcal{V}(q) = \mathcal{D}_L^{-1}(q)\mathcal{E}(q)$ with $\mathcal{E}(q)$ denoting the q -transform of $e(k)$ added to account the influence of noise in Eq.(5.3). Thus, Eq.(5.35) is the one-step-ahead predictor of the MIMO ARX model in Eq.(5.3) when $\mathcal{P}$ contains only one submodel

$$\hat{y}(k|k-1) = \sum_{i=0}^{\tau} h_u(i)u(k-i) - \sum_{i=1}^{\tau} h_m(i)y(k-i). \quad (5.40)$$

To proceed further, it is necessary to make assumptions on the input sequence. It is customary to assume that $u(k)$ is ergodic or quasi-stationary when it is deterministic. Henceforth, we assume that the input sequence is either ergodic or quasi-stationary whichever suits to our purpose. We also assume that the input sequence is independent from noise.

Definition 5.4.1 *The input sequence $u(k) \in \mathbb{R}^m$, $k = 1, 2, \dots$ is persistently exciting*

of order ρ if and only if there exists a $M \in \mathbb{N}$ such that the Hankel matrix

$$\mathcal{U}_{0,\rho,M} = \begin{bmatrix} u(1) & u(2) & \cdots & u(M) \\ u(2) & u(3) & \cdots & u(M+1) \\ \vdots & & \ddots & \vdots \\ u(\rho) & u(\rho+1) & \cdots & u(M+\rho-1) \end{bmatrix} \quad (5.41)$$

has full rank ρ .

The rank condition in Definition 5.4.1 is equivalent to the positive-definiteness of the input auto-correlation matrix $\mathcal{R}_u$

$$\begin{aligned} \mathcal{R}_u &= \lim_{M \rightarrow \infty} M^{-1} \mathcal{U}_{0,\rho,M} \mathcal{U}_{0,\rho,M}^T \\ &= \begin{bmatrix} R_u(0) & R_u^T(1) & \cdots & R_u^T(\rho-1) \\ R_u(1) & R_u(0) & \cdots & R_u^T(\rho-2) \\ \vdots & & \ddots & \vdots \\ R_u(\rho-1) & R_u(\rho-2) & \cdots & R_u(0) \end{bmatrix}. \end{aligned} \quad (5.42)$$

where $R_u(\ell) = E\{u(k+\ell)u(k)^T\}$, $\ell \geq 0$. Since $A_o^i = 0$ if $i \geq \tau$, with $\mathcal{Y}_{0,\rho,M}$ defined similarly to Eq.(5.41) $y(k)$ has an auto-correlation matrix $\mathcal{R}_y = \lim_{M \rightarrow \infty} M^{-1} \mathcal{Y}_{0,\rho,M} \mathcal{Y}_{0,\rho,M}^T$. The cross-correlation matrix of $u(k)$ and $y(k)$ is defined by $\mathcal{R}_{uy} = \lim_{M \rightarrow \infty} M^{-1} \mathcal{U}_{0,\rho,M} \mathcal{Y}_{0,\rho,M}^T$.

Let $\phi_u(\omega)$, $\phi_y(\omega)$, and $\phi_{uy}(\omega)$ denote the auto-power and the cross-power spectral densities of $u(k)$ and $y(k)$. Let x be a nonzero row vector $(x_\rho \cdots x_1)$ formed by concatenating ρ row vectors $x_k \in \mathbb{R}^{1 \times m}$. Set $M_\rho(q) = \sum_{k=1}^\rho x_k q^{-k}$. Note that

$$\begin{aligned} x \mathcal{R}_u x^T &= E \left\{ x \begin{bmatrix} u(k-\rho) \\ \vdots \\ u(k-1) \end{bmatrix} [u(k-\rho) \cdots u(k-1)] x^T \right\} \\ &= E \{ (M_\rho(q)u)(M_\rho(q)u)^T \} \\ &= \int_{-\pi}^{\pi} M_\rho(e^{i\omega}) \phi_u(\omega) M_\rho^T(e^{-i\omega}) d\omega. \end{aligned}$$

Hence, $M_\rho(e^{i\omega}) \phi_u(\omega) M_\rho^T(e^{-i\omega}) \equiv 0$ iff $x \mathcal{R}_u x^T = 0$. If $\phi_u(\omega)$ is positive-definite for at least ρ distinct frequencies, then $M_\rho(e^{i\omega}) = 0$ for at least ρ distinct frequencies. But, $M_\rho(q^{-1})$ can have at most $\rho-1$ zeros on the unit circle. Hence, it vanishes identically. Then, $x = 0$ and $\mathcal{R}_u$ is positive definite.

Remembering that we are still considering one submodel only case and $\tau \leq n$, we transform $z(k)$ to another regressor $z_r(k)$ compatible with the one-step-ahead

predictor in Eq.(5.48)

$$z_r(k) = \begin{bmatrix} u_a(k) \\ y_a(k) \end{bmatrix} \quad (5.43)$$

where

$$u_a(k) = \begin{bmatrix} u(k) \\ \vdots \\ u(k-\tau) \end{bmatrix}, \quad y_a(k) = \begin{bmatrix} y(k-1) \\ \vdots \\ y(k-\tau) \end{bmatrix},$$

as $z(k) = J_z z_r(k)$ with the permutation matrix given by

$$J_z = \begin{bmatrix} I_m & 0_{m \times \tau(m+p)} \\ 0_{\tau \times m} & I_\tau \otimes [I_m \ 0_{m \times p}] \\ 0_{\tau \times p} & I_\tau \otimes [0_{p \times m} \ I_p] \end{bmatrix}$$

Then, from Eq.(5.9) with $\delta_s = t - s + 1$

$$\delta_s^{-1} \mathcal{R}_{s,t} \mathcal{R}_{s,t}^T = \delta_s^{-1} \sum_{k=s}^t z(k) z^T(k) = J_z \delta_s^{-1} \sum_{k=s}^t z_r(k) z_r^T(k) J_z^T$$

which converges to as $\delta_s \rightarrow \infty$,

$$\mathcal{R}_z = J_z \mathcal{R}_{z_r} J_z^T = J_z \begin{pmatrix} \mathcal{R}_{u_a} & \mathcal{R}_{u_a y_a} \\ \mathcal{R}_{u_r y_a}^T & \mathcal{R}_{y_a} \end{pmatrix} J_z^T.$$

Since J_z is a permutation matrix, it is invertible. Therefore, $\mathcal{R}_z$ is positive-definite if and only if $\mathcal{R}_{z_r}$ is positive-definite.

Let x and w be two row vectors $(x_\tau \cdots x_0)$ and $(w_\tau \cdots w_1)$ constructed by concatenating τ row vectors $x_k \in \mathbb{R}^{1 \times m}$ and $w_k \in \mathbb{R}^{1 \times p}$. Define two polynomial matrices in q^{-1} by setting $M_u(q) = \sum_{k=0}^\tau x_k q^{-k}$ and $M_w(q) = \sum_{k=1}^\tau w_k q^{-k}$. Then,

$$\begin{aligned} [x \ w] \mathcal{R}_{z_r} [x \ w]^T &= \mathbb{E} \left\{ [x \ w] \begin{bmatrix} u_a \\ y_a \end{bmatrix} \begin{bmatrix} u_a^T & y_a^T \end{bmatrix} \begin{bmatrix} x^T \\ w^T \end{bmatrix} \right\} \\ &= \int_{-\pi}^{\pi} [M_u(e^{i\omega}) \ M_w(e^{i\omega})] \begin{bmatrix} \phi_u & \phi_{uy} \\ \phi_{yu} & \phi_y \end{bmatrix} \begin{bmatrix} M_u^T(e^{-i\omega}) \\ M_w^T(e^{-i\omega}) \end{bmatrix} d\omega \\ &= \int_{-\pi}^{\pi} \{ M_u(e^{i\omega}) \phi_u(\omega) M_u^T(e^{-i\omega}) + M_u(e^{i\omega}) \phi_{uy}(\omega) \\ &\quad \cdot M_w^T(e^{-i\omega}) + M_w(e^{i\omega}) \phi_{yu}(\omega) M_u^T(e^{-i\omega}) + M_w(e^{i\omega}) \\ &\quad \cdot \phi_y(\omega) M_w^T(e^{-i\omega}) \} d\omega. \end{aligned}$$

Note that $\phi_{uy}^T(-\omega) = \phi_{yu}(\omega)$ and $\phi_{yu}(\omega) = G_L(e^{i\omega})\phi_u(\omega)$ from the independence of $u(k)$ and $v(k)$. Again from the independence, $\phi_y(\omega) = G_L(e^{i\omega})\phi_u(\omega)G_L^T(e^{-i\omega}) + \phi_v(\omega)$. Now, $[x \ w]\mathcal{R}_{z_r}[x \ w]^T$ can be written as

$$\begin{aligned} [x \ w]\mathcal{R}_{z_r}[x \ w]^T &= \int_{-\pi}^{\pi} (M_w(e^{i\omega})G_L(e^{i\omega}) + M_u(e^{i\omega}))\phi_u(\omega) \\ &\quad \cdot (M_w(e^{-i\omega})G_L(e^{-i\omega}) + M_u(e^{-i\omega}))^T d\omega \\ &\quad + \int_{-\pi}^{\pi} M_w(e^{i\omega})\phi_v(\omega)M_w^T(e^{-i\omega})d\omega. \end{aligned}$$

$[x \ w]\mathcal{R}_{z_r}[x \ w]^T = 0$ iff both integrands are zero above. If $\phi_v(\omega)$ is positive-definite for at least τ distinct frequencies, then $M_w(e^{i\omega}) = 0$ for at least τ distinct frequencies. But, $M_w(q^{-1})$ can have at most $\tau - 1$ zeros on the unit circle. Hence, it vanishes identically. The first term on the right-hand side is reduced to $\int_{-\pi}^{\pi} M_u(e^{i\omega})\phi_u(\omega)M_u(e^{-i\omega})d\omega = 0$. Thus, if $u(k)$ is persistently exciting of order $\tau + 1$, then $M_u(e^{i\omega})$ must identically vanish. It follows that $[x \ w] = 0$ and $\mathcal{R}_{z_r}$ is positive-definite.

Lastly, we check if $\phi_v(\omega) > 0$ for at least τ distinct frequencies. Recall that $\mathcal{V} = \mathcal{D}_L^{-1}\mathcal{E}$. Let $\phi_e(\omega)$ denote the auto-PSD of $e(k)$. Then, $\phi_v(\omega) = \mathcal{D}_L^{-1}(e^{i\omega})\phi_e(\omega)\mathcal{D}_L^{-T}(e^{-i\omega})$. Thus, $\phi_v(\omega) > 0$ for at least τ distinct frequencies if and only if $\phi_e(\omega) > 0$ for at least τ distinct frequencies.

Suppose now $v(k) = 0$ for all k . Starting from the left MFD in Eq.(5.39), that is not necessarily irreducible, find a right co-prime factorization $G_L(q) = N_R(q)D_R^{-1}(q)$ [54]. The quadratic form above can then be written as

$$\begin{aligned} [x \ w]\mathcal{R}_{z_r}[x \ w]^T &= \int_{-\pi}^{\pi} (M_w(e^{i\omega})N_R(e^{i\omega}) + M_u(e^{i\omega})D_R(e^{i\omega})) \\ &\quad \cdot D_R^{-1}(e^{i\omega})\phi_u(\omega)D_R^{-T}(e^{-i\omega}) \\ &\quad \cdot (M_w(e^{-i\omega})N_R(e^{-i\omega}) + M_u(e^{-i\omega})D_R(e^{-i\omega}))^T d\omega. \end{aligned}$$

Since $G_L(q)$ is proper, the column degree of $N_R(q)$ is bounded above by $\deg(D_R(q))$ which is bounded above by $\deg(D_L(q)) = \deg \det(D_L(q)) \leq p\tau$. Hence, $M_w(q^{-1})N_R(q^{-1}) + M_u(q^{-1})D_R(q^{-1})$ has a row degree bounded above by $\tau + p\tau$. Suppose $[x \ w]\mathcal{R}_{z_r}[x \ w]^T = 0$ and $u(k)$ is PE of order ρ . This means that $\phi_u(\omega)$ and hence $D_R^{-1}(e^{i\omega})\phi_u(\omega)D_R^{-T}(e^{-i\omega})$ are positive-definite for at least ρ distinct frequencies. But, N_R and D_R are right-coprime. Thus, $[M_u(e^{i\omega}) \ M_w(e^{i\omega})]$ vanishes at least at ρ distinct frequencies. So does the Bezout $M_wN_R + M_uD_R$. If $\rho > (p+1)n$, then both $M_wN_R + M_uD_R$ vanishes identically. From the right-coprimeness of N_R and D_R , M_w and M_u vanish identically. Hence, $[x \ w] = 0$ and $\mathcal{R}_{z_r}$ is positive-definite.

We state identifiability conditions in asymptotic form for submodels with sufficiently

long dwell times in the following.

Assumption 5.4.1 *There exists a $\delta_u \in \mathbb{N}$ such that for all segments with $\delta_i(\chi) \geq \delta_u$, $e(k) \neq 0$, $u(k)$ is PE of order $n+1$, and noise is PE of at least order n , then $\mathcal{R}_{s_j, s_{j+1}-1}$ is full-rank for all $\tau \leq n$ and if $\delta_i(\chi) \geq \delta_u$, $e(k) = 0$, $u(k)$ is PE of order $(p+1)n$, and Eq.(5.39) is an MFD of Eq.(5.35) for some $\nu \leq \tau \leq n$, then $\mathcal{R}_{s_j, s_{j+1}-1}$ has rank $\deg(D_R(q))$.*

Recall that $\mathcal{R}_{s_j, s_{j+1}-1}$ has $\bar{n}_z = \tau(p+m) + m$ rows. Its rank can be trapped to an interval by reducing τ from n by 1 at each step. The length of this uncertainty interval is $p+m$. If Eq.(5.39) is a coprime factorization, no iterations are needed. In the presence of noise, as in Eq.(5.3) overparameterized models can be used. Next, we derive similar conditions of PE for the single-input single-output setup.

5.5. Persistence of Excitation for SISO SARX systems

For SISO systems, Rewrite Eq.(3.43) for some $\tau \leq n$ as

$$y(k) + \sum_{l=1}^{\tau} \bar{\beta}_l y(k-l) = \bar{d} + \sum_{l=1}^{\tau} \bar{\alpha}_l u(k-l) + e(k) \quad (5.44)$$

by adding noise $e(k)$ and removing time dependence in the parameters. Set for $l = 1, \dots, \tau$

$$\bar{\alpha}_l = c^T (A + gc^T)^{l-1} (b + gd), \quad (5.45)$$

$$\bar{\beta}_l = c^T (A + gc^T)^{l-1} g. \quad (5.46)$$

Transform Eq.(5.44) into frequency domain

$$\begin{aligned} \mathcal{Y}(q^{-1}) &= \frac{\mathcal{N}(q^{-1})}{\mathcal{D}(q^{-1})} \mathcal{U}(q^{-1}) + \mathcal{V}(q^{-1}) \\ &= \frac{\bar{d} + \bar{\alpha}_1 q^{-1} + \dots + \bar{\alpha}_\tau q^{-\tau}}{1 + \bar{\beta}_1 q^{-1} + \dots + \bar{\beta}_\tau q^{-\tau}} \mathcal{U}(q) + \mathcal{V}(q) \end{aligned} \quad (5.47)$$

where $\mathcal{V}(q^{-1}) = \mathcal{E}(q^{-1})/\mathcal{D}(q^{-1})$, and the one-step-ahead predictor of Eq.(5.3) on $[\alpha_j, \beta_j]$ is given by

$$\hat{y}(k|k-1) = \sum_{l=0}^{\tau} \bar{\alpha}_l u(k-l) - \sum_{l=1}^{\tau} \bar{\beta}_l y(k-l). \quad (5.48)$$

with $\bar{\alpha}_0 = \bar{d}$.

It is a well-known fact [62] that when $u(k)$ and $y(k)$, with $u(k)$ independent from $e(k)$, and with $e(k)$ persistently exciting of any order, the parameters $\bar{\alpha}_l$ and

$\bar{\beta}_l$ in Eq.(5.48) can be uniquely determined by minimizing $\|Y_{\alpha_j, \beta_j} - \mathcal{R}_{\alpha_j, \beta_j}^T \theta\|_2^2$ if $u(k)$ is persistently exciting (PE) of order $\tau + 1$. In the noiseless case, $\mathcal{N}(q^{-1})$ and $\mathcal{D}(q^{-1})$ must be co-prime and $u(k)$ must be PE of order $2\tau + 1$. An ergodic or quasi-stationary sequence $u(k)$ is PE of order ρ if its power spectrum is nonzero at least at ρ distinct frequencies. Since $\delta_j(\chi_\theta) \geq \delta_i(\chi) - 2n$, $\beta_j - \alpha_j \geq \delta_i(\chi) - 8n$. Recall also the chain of inequalities $\delta_j + 2n \geq \delta_i(\chi) \geq \delta_j(\chi_\theta)$.

Under Assumption 5.3.1, $H_o^{(1)}(z; k)$ and $H_o^{(2)}(z; k)$ in Eqs.(5.33)–(5.34) are n th order transfer functions in two sets with cardinality σ . Then, $\tau = n$ in Eq.(5.44). The input-output equation for $\mathcal{P}_{\varphi(k_i)}$ is given by Eq.(5.32) from which the predictors in Eq.(5.48) are derived. Since there are only 2σ transfer functions $H_o^{(1)}(z; k)$ and $H_o^{(2)}(z; k)$ with MacMillan degree n , the parameters $\bar{\alpha}_l, \bar{\beta}_l$, $l = 1, \dots, n$ are identifiable for every chosen interval $[\alpha_j, \beta_j]$ provided that $u(k)$ is PE of order $2n + 1$ and $\beta_j - \alpha_j \geq \delta_u^*$ for all j and some $\delta_u^* > 0$. Now, set $\delta_u = \delta_u^* + 8n$. Then, either $\delta_i(\chi) \geq \delta_u$ or more conservatively $\delta_j(\chi_\theta) \geq \delta_u$ guarantees that $\beta_j - \alpha_j \geq \delta_u^*$. Hence, $\mathcal{R}_{\alpha_j, \beta_j}$ has full row-rank. Passing to $[s_j, s_{j+1})$, we see that $\mathcal{R}_{s_j, s_{j+1}-1}$ has full row-rank. We summarize this property as a requirement.

Assumption 5.5.1 *There exists a $\delta_u \in \mathbb{N}$ such that for all segments with $\delta_i(\chi) \geq \delta_u$, $\text{rank}(\mathcal{R}_{s_j, s_{j+1}-1}) = n_1$ where s_j and s_{j+1} are as in Lemma 5.2.4.*

Although we assumed $\tau = n$ to guarantee $\chi \subset \chi_\theta$ in Lemma 5.3.1, $\tau < n$ is a possible case, in particular, when k_i is not an identifiable switch from the input-output data. When $\mathcal{R}_{s_j, s_{j+1}-1}$ is rank deficient, $\theta(k)$ can be reparameterized. The minimal model order may be determined from $\|\Sigma_{\mathcal{R}_{s_j, s_{j+1}-1}}\|_0 = 2\tau + 1$. There must be a sufficient number of segments in χ satisfying Assumption 5.5.1 to recover all submodels in $\mathcal{P}$. We state this condition as a standing assumption.

Assumption 5.5.2 *Every submodel in $\mathcal{P}$ is active in at least one segment with a dwell time of at least δ_u .*

The identifiability results derived in this chapter will be illustrated next via a numerical example using multi-sine excitations in the next section.

5.6. Example I

The same SLS model from Section 3.5 is studied here again. We check the switch identifiability conditions derived in Section 5.1 for the ARX order $\tau = 2n = 6$. A switching sequence of length $N = 1,000$ is generated, and the dwell time satisfies $\delta_*(\chi) = 31$. We excited the SLS system with a multisine input consisting of five

superposed harmonics with frequencies, phases, and amplitudes randomly selected from uniform distributions. We verified the switch identifiability in a noiseless setup since identifiability of a switch is not lost if perturbations are small enough while we checked the identifiability of submodels by injecting noise with small signal-to-noise ratio. The latter approach, besides being more realistic, has eliminated delicate parametrizations issues in designing PE inputs for noiseless identification experiments. The conversion from SLS to SARX was achieved in the same way as in Subsection 3.5. By calculating $\|(\theta(k_{i+1}) - \theta(k_{i+1} - 1))z(k_{i+1})\|_2$, we see from Figure 5.1 that Eq.(5.27) is satisfied for the switch set χ . Recall from Lemma 5.2.6 that $\chi \subset \chi_\theta$.

Next, we verify the strong PE condition at Eq.(5.25), Figure 5.2 shows that this condition does not hold at any switching point in the switch set χ . Since Eq.(5.25) does not hold, we directly check the condition in Eq.(5.26). We perform two SVDs: one for $\mathcal{R}_{k_i, k_{i+1}-1}$ and one for $\mathcal{R}_{k_i, k_{i+1}}$ by noting that $\mathcal{R}_{k_i, k_{i+1}} = [\mathcal{R}_{k_i, k_{i+1}-1} \ z(k_{i+1})]$, note that the number of significant singular values of $\mathcal{R}_{k_i, k_{i+1}-1}$ and $\mathcal{R}_{k_i, k_{i+1}}$ are the same. The augmentation of $z(k_{i+1})$ into $\mathcal{R}_{k_i, k_{i+1}-1}$ does not lead to an increase in the number of significant singular values of $\mathcal{R}_{k_i, k_{i+1}-1}$ as shown in Figure 5.3 and therefore no increase in the rank, hence, the condition in Eq.(5.26) is satisfied. We conclude that by using the multisine excitations the switch set χ is identifiable.

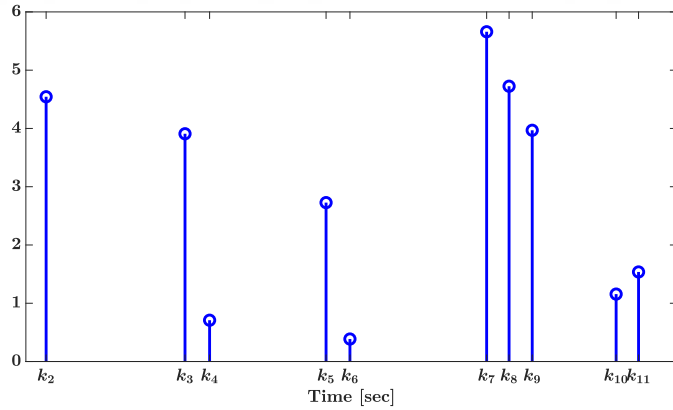


Figure 5.1: $\|(\theta(k_{i+1}) - \theta(k_{i+1} - 1))z(k_{i+1})\|_2$ on χ .

Next, submodels identifiability is checked for observer orders $\tau = 2, 3$. We inject an additive Gaussian noise to the outputs to achieve an SNR of 40db and verify that the matrix $\mathcal{R}_{k_i+\tau, k_{i+1}-1}$ is full rank by inspecting the number of non-zero singular values of $\mathcal{R}_{k_i+\tau, k_{i+1}-1}$ and checking whether it is equal to its row size n_z . Figure 5.4 depicts its four smallest singular values which appear to be nonzero, hence indicates no drop in rank for $\mathcal{R}_{k_i+\tau, k_{i+1}-1}$.

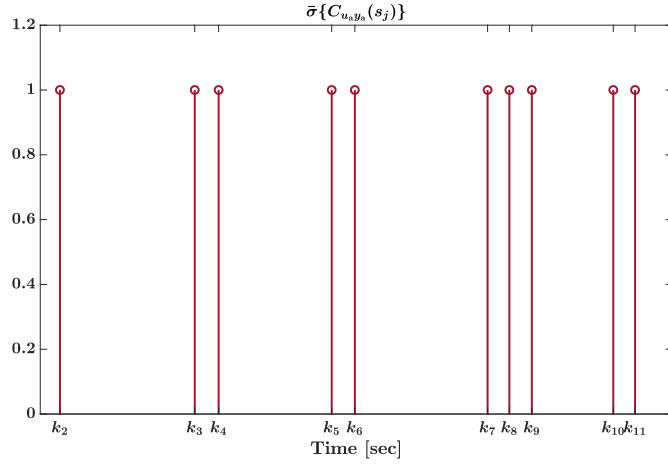


Figure 5.2: $\bar{\sigma}\{C_{u_a y_a}(s_j)\}$ versus s_j for $s_j \in \chi$.

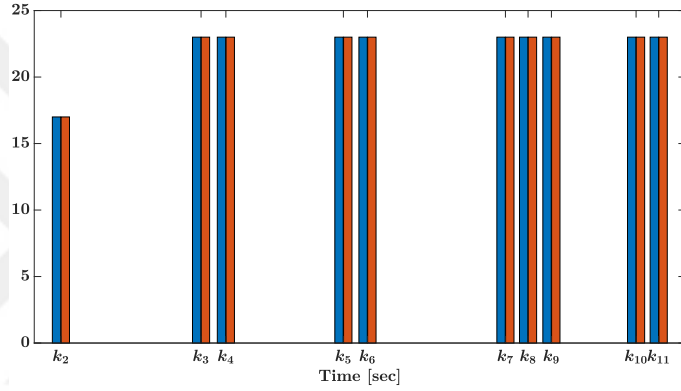


Figure 5.3: $\|\Sigma_{\mathcal{R}_{s_j, s_{j+1}-1}}\|_{\ell,0}$ (left bars) and $\|\Sigma_{\mathcal{R}_{s_j, s_{j+1}}}\|_{\ell,0}$ (right bars) on χ .

5.7. Example II

In this example, a SISO SLS system with three submodels ($\sigma = 3$) of order $n = 2$ is considered:

$$\begin{aligned} A_1 &= \begin{pmatrix} 0 & 0.8 \\ -0.8 & 0.5 \end{pmatrix}, b_1 = \begin{pmatrix} 1.4 \\ -2 \end{pmatrix}, c_1^T = -(1 \ 1.5), d_1 = 1; \\ A_2 &= \begin{pmatrix} -0.9 & -1.25 \\ 1.8 & 1.5 \end{pmatrix}, b_2 = \begin{pmatrix} -1 \\ 0.5 \end{pmatrix}, c_2^T = (-1 \ 1), d_2 = -1.5; \\ A_3 &= \begin{pmatrix} 0.8 & 0 \\ 0 & -0.3 \end{pmatrix}, b_3 = \begin{pmatrix} 1 \\ 2 \end{pmatrix}, c_3^T = (2 \ 3), d_3 = 2. \end{aligned}$$

Similarly to Section 5.6, we check switch and submodel identifiability conditions here. Switch identifiability are checked for observer order $\tau = 2n = 4$. A switch sequence of length $N = 1,000$ and dwell time $\delta_*(\chi) = 19$ was sampled from a uniform

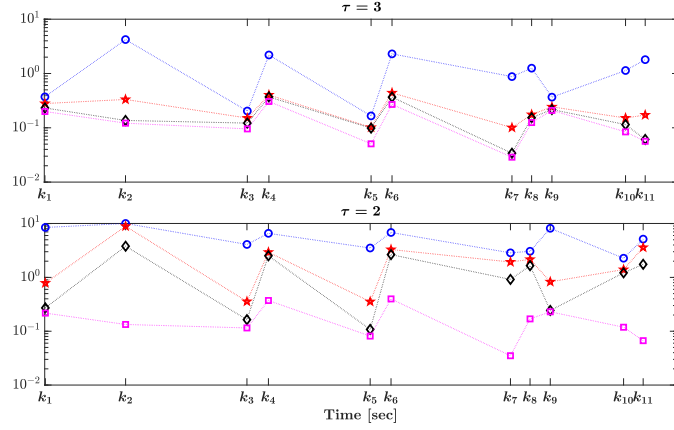


Figure 5.4: The four smallest singular values of $\mathcal{R}_{k_i + \tau, k_{i+1} - 1}$.

distribution. The system was excited with multisine input generated the same way as in Section 5.6. The conversion to the SARX model is done via the MATLAB command `acker`. First, switch identifiability was checked in a noiseless setup. The Condition in Eq.(5.27) was checked by calculating $|z^T(s_{j+1})(\theta(s_{j+1}) - \theta(s_j))|$ and plotting it versus χ . Figure 5.5 shows that this condition is satisfied. Figure 5.6, on the other hand, is for verifying the condition in Eq.(5.25). We can clearly see that it is not satisfied on χ . Nonetheless, when checking the alternative condition in Eq.(5.26) in Figure 5.7, it holds true.

Now let's check the submodel identifiability in a noisy setup for observer orders $\tau = 2$ and 3 for the same gain sequence. We inject a Gaussian noise to the output measurements to achieve a SNR of 40db. As expected, an inspection of the last 4 singular values plotted in Figure 5.9 reveals that $\mathcal{R}_{k_i + \tau, k_{i+1} - 1}$ has full rank as requested. The condition in Assumption 5.5.1 is also verified in a noiseless setup. We plot all of the $2n + 1$ singular values of $(\mathcal{R}_{s_j, s_{j+1} - 1})$ to verify the matrix is of full rank. Figure 5.8 confirms that this requirement is satisfied, since none of its singular values are insignificant.

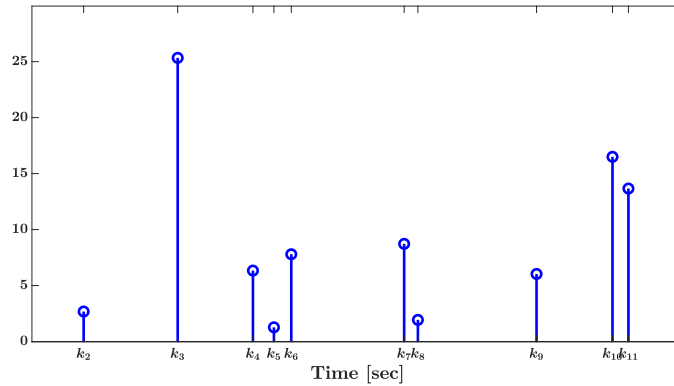


Figure 5.5: $|z^T(s_{j+1})(\theta(s_{j+1}) - \theta(s_j))|$ on χ .

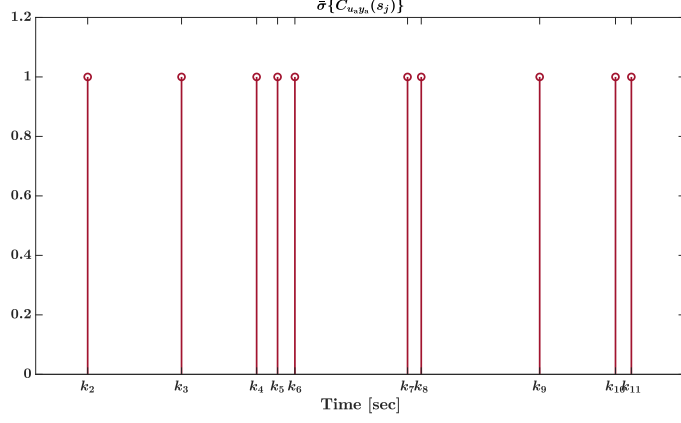


Figure 5.6: $\bar{\sigma}\{C_{u_a y_a}(s_j)\}$ versus s_j for $s_j \in \chi$.

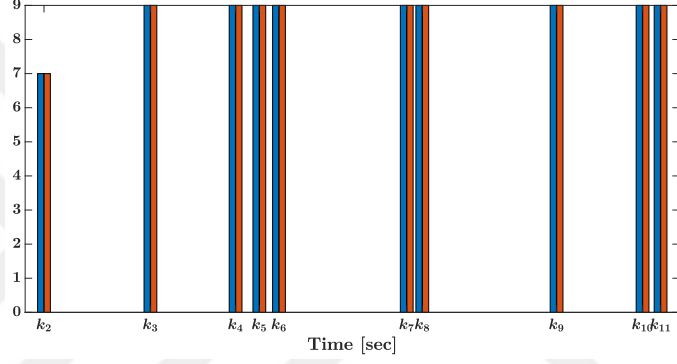


Figure 5.7: $\|\Sigma_{\mathcal{R}_{s_j, s_{j+1}-1}}\|_{\ell_0}$ (left bars) and $\|\Sigma_{\mathcal{R}_{s_j, s_{j+1}}}\|_{\ell_0}$ (right bars).

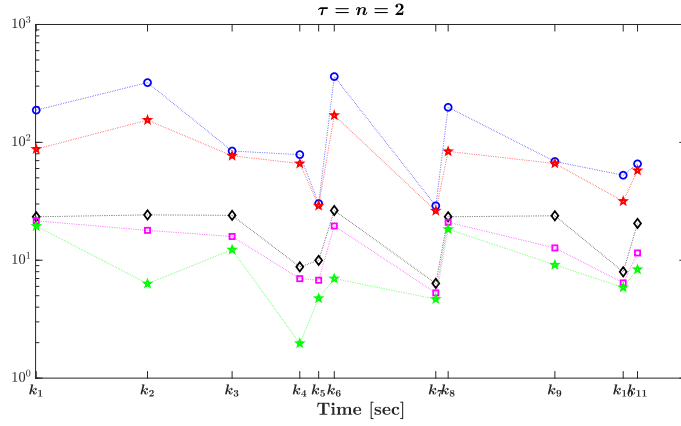


Figure 5.8: The five SVDs of $(\mathcal{R}_{s_j, s_{j+1}-1})$ for $\tau = n = 2$ in a noiseless setup.

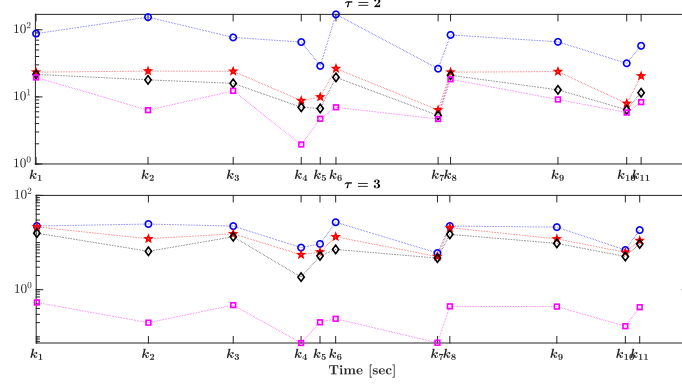


Figure 5.9: The last four SVDs of $\mathcal{R}_{k_i+\tau, k_{i+1}-1}$ for $\tau = 2$ and 3 in a noisy setup.

5.8. Summary

We started this chapter by putting forward switch identifiability conditions. In particular, two conditions were stipulated for a switch to not go undetectable. Firstly, the regressors at the switches not being orthogonal to the parameter changes, is noted to be not so stringent. The second condition requires the regressor at the switches to belong to the range space of previous regressors. An attempt was made to lift the second condition completely, but unfortunately it amounted to a much more stringent condition. Nonetheless, by reverting back to the old condition, which is not stringent to fulfill as shown in the example section, switch detection conditions are considered not very strict and hold true in general. Subsequently to that, a complete section was devoted to clarify the relation between segments in SLS models and those in SARX models. There it was established that the SLS switch set is a subset of that of the SARX. Beside that, given a segment with a certain dwell time, the right-end switch point in a SARX coincides with that of the SLS, where as the left-end switch is shifted with amount less than $2n$. It was also observed that given some segments in SARX with long enough dwell time, the SLS Markov parameters retrieved from subintervals of those segments are constant and hence give rise to LTI realizations. Furthermore, we have shown that on an even smaller subintervals of the previously referred segments, the estimated observer gains are as well constant. These two observations are indeed key to the submodel estimation attempt in later sections. Right after that, a small discussion on switch identifiability for the SISO case was broached; few modifications along with comments were made. Then, persistence of excitation conditions for MIMO SARX were discussed by following matrix fraction descriptions (MFD) approach and co-prime factorizations. In a similar manner, PE conditions in the SISO case were also specified. Basically, in a noisy setup, given that the inputs are persistently exciting of order $\tau + 1$, the noise being persistently exciting of any order, and the input and noise being independent, then

the parameters of the ARX model are retrieved uniquely from the data gathered over flat subintervals of certain long enough SARX segments. In a noiseless configuration, on the other hand, one needs inputs to be PE of order $2\tau + 1$. On top of that, the transfer functions numerator and denominator polynomials have to be co-prime. The theoretical part of this chapter was concluded by stipulating a PE requirement that is of high importance regarding submodels recovery. A standing assumption that obliges all the submodels to appear at least once with a certain length δ_u was put to ensure successful recovery of submodels. In the upcoming chapter, we will explore how the underlying sparsity assumption can be exploited into retrieving the submodel and switching sequence informations from the input-output data.



6. COMPRESSIVE SYSTEM IDENTIFICATION FOR SISO SLS MODELS

In this chapter, a compressive identification scheme is devised for SLS estimation. In particular, Section 6.1 brings forward a NP-hard sparse optimization problem to estimate the set of submodels along with the set of switches for both SLS and its observer-based transformed SARX model from input-output data. Section 6.2, alternatively, exploits the directory of submodels obtained from the previous step to detect which submodel is active on not long-enough segments, for which the parameter estimates are biased. This active submodel decision was conducted via the MOESP subspace algorithm or a less stringent and simplified active index selection variant of MOESP. Section 6.3 presents a relaxation of the previous non-convex problem, which when supplemented with reweighting was demonstrated to be useful through a numerical example in Section 6.5.

6.1. Submodel set and switches estimation

Results in this chapter are derived exclusively for the SISO model. Therefore, the SARX system in Eq.(5.28) is considered, however extension to MIMO can be made in a straight forward manner.

Let us reconsider the model in Eq.(5.30). The parameters in Θ are renewed simultaneously along its rows. Between s_j and s_{j+1} , $\theta(k)$ does not change. In fact, this observation formed the basis of Proposition 5.1.1. Note that $\theta(k)$ being constant for all $s_j \leq k < s_{j+1}$ implies that the corresponding $n_z(s_{j+1} - s_j)$ rows of Θ remain unaltered, yet parameter renewals may take place in any columns of $\theta(s_{j+1})$ as can be verified by checking Eqs.(3.41)–(3.42) for several cases. Since parameters may change along several rows of Θ at each switch, we need to count them once to utilize sparsity. We transform Eq.(5.5) into a form in which the unknowns are nonzero entries of a block-sparse matrix. Write the first-order differences in Eq.(5.7) as

$$\Theta = \underbrace{\begin{bmatrix} I & 0 \\ & \ddots & \vdots \\ I & \cdots & I \end{bmatrix}}_{\mathcal{L}} \underbrace{\begin{bmatrix} \delta_\theta(n_1) \\ \vdots \\ \delta_\theta(N_1) \end{bmatrix}}_{\delta_\Theta} + \begin{bmatrix} I \\ \vdots \\ I \end{bmatrix} \theta(2n). \quad (6.1)$$

Set $E = 0$ and substitute Eq.(6.1) in Eq.(5.30) to derive

$$Y = Z \underbrace{[\mathcal{L} \ (1_{N_1-2n} \otimes I)]}_{\mathcal{A}} \underbrace{[\delta_{\bar{\Theta}}^T \ \theta^T(2n)]}_{\bar{\Theta}}^T. \quad (6.2)$$

where 1_M denotes the column vector of ones in $\mathbb{R}^M$. If we set $N_2 = N_1 - 2n$, then

$$Y = Z [\mathcal{L} \ (1_{N_2} \otimes I)] [\delta_{\bar{\Theta}}^T \ \theta^T(2n)]^T = \mathcal{A}\bar{\Theta}. \quad (6.3)$$

Let $\mathfrak{I}(\|\delta_{\theta}(k)\|_2 > 0)$ denote the indicator function defined to take a value of 1 when $\|\delta_{\theta}(k)\|_2 > 0$ and a value of 0 otherwise. The ℓ_0 quasi-norm of the sequence $\delta_{\theta}(k)$ is defined by

$$\|\delta_{\theta}\|_{0,\mathfrak{I}} = \sum_{k=n_1}^{N_1} \mathfrak{I}(\|\delta_{\theta}(k)\|_2 > 0). \quad (6.4)$$

Since k is a switch if and only if $\delta(k) \neq 0$, $\|\delta_{\theta}\|_{0,\mathfrak{I}}$ is the number of the switches for the model in Eq.(5.3) in the interval $[n_1, N_1]$. In the noiseless case, all identifiable switches are found by solving the nonconvex sparse optimization problem:

$$\begin{aligned} \min_{\bar{\Theta}} \quad & \|\delta_{\theta}\|_{0,\mathfrak{I}} \\ \text{s.t.} \quad & Y = \mathcal{A}\bar{\Theta}. \end{aligned} \quad (6.5)$$

Let $\bar{\Theta}^{\sharp} = \begin{bmatrix} \delta_{\bar{\Theta}^{\sharp}} \\ \theta^{\sharp}(2n) \end{bmatrix}$ be a solution of Eq.(6.5) and denote its error by $\bar{\Theta}^{\star} = \bar{\Theta}^{\sharp} - \bar{\Theta}$. For $n_1 \leq k \leq N_1$, from

$$\begin{aligned} \delta_{\theta^{\sharp}}(k) - \delta_{\theta}(k) &= \theta^{\sharp}(k) - \theta^{\sharp}(k-1) - \theta(k) + \theta(k-1) \\ &= \theta^{\star}(k) - \theta^{\star}(k-1) = \delta_{\theta^{\star}}(k), \end{aligned} \quad (6.6)$$

Partition $\bar{\Theta}^{\star}$ similarly to $\bar{\Theta}$. From $\mathcal{A}\bar{\Theta}^{\star} = 0$ We derive $\mathcal{A}[\delta_{\bar{\Theta}^{\star}}^T \ \Delta_{2n}^T]^T = 0$ with $\Delta_{2n} = \theta^{\sharp}(2n) - \theta(2n)$. Then, from Eq.(5.31) and Eq.(6.1)

$$\begin{bmatrix} z^T(n_1) & 0 \\ \vdots & \ddots \\ z^T(N_1) & \cdots & z^T(N_1) \end{bmatrix} \delta_{\bar{\Theta}^{\star}} = - \begin{bmatrix} z^T(n_1) \\ \vdots \\ z^T(N_1) \end{bmatrix} \Delta_{2n}.$$

Now, from the above equality and Eq.(6.6) for $n_1 \leq k \leq N_1$,

$$z^T(k) \sum_{j=n_1}^k \delta_{\theta^{\star}}(j) = z^T(k) (\theta^{\star}(k) - \Delta_{2n}) = -z^T(k) \Delta_{2n}. \quad (6.7)$$

It follows that $z^T(k)\theta^*(k) = 0$ for all $n_1 \leq k \leq N_1$.

Let us consider a segment $[s_j, s_{j+1})$ in χ_θ where s_j and s_{j+1} are identifiable switches from the input-output data. Then, $\delta_{\theta(k)} = 0$ for all $k \in (s_j, s_{j+1})$. Since $\Theta^\#$ is optimal and Θ is feasible, $\delta_{\theta^\#}(k) = 0$ for all $k \in (s_j, s_{j+1})$. Hence, for all $k \in (s_j, s_{j+1})$, $\delta_{\theta^*}(k) = 0$ and $\theta^*(k) = \theta^*(s_{j+1} - 1)$, i.e, it is constant here. From Lemma 5.2.4, $0 \leq s_j - k_i \leq 2n$ and $k_{i+1} = s_{j+1}$ for some $1 \leq i < i^*$. If $\delta_j(\chi_\theta) \geq \delta_u$, from Assumption 5.5.1 $\mathcal{R}_{s_j, s_{j+1}-1}$ has full rank. Pick $\gamma_j = (\alpha_j + \beta_j)/2$. Then, $\theta^*(\gamma_j - 2n) = 0$. Hence, $\theta^\#(\gamma_j) = \theta(\gamma_j)$ on $[\alpha_j, \beta_j]$ and extending to the entire interval $[s_j, s_{j+1})$. Replace $\theta^\#(\gamma_j)$ with the average value of $\theta^\#(k)$ on $[s_j, s_{j+1})$. Since $d(k)$ is constant on $[s_j, s_{j+1})$ we can run Algorithm 3 then Algorithm 2 with $\theta^\#(k)$ first on $[s_j, s_{j+1})$ to estimate the system Markov parameters and the observer gain Markov parameters, then on the interval $[s_j, s_{j+1} - 2n)$ to estimate the state-space matrices, and finally on $[s_j, s_{j+1} - 4n]$ to estimate the observer gains. These intervals validate calculations since the latter are most often redundant due to time-invariance. The analysis above shows that the presented scheme recovers $\mathcal{P}_{\varphi(k_i)}$ and χ if $\delta_j(\chi_\theta) \geq \delta_u$ and $\delta_*(\chi) > 4n$ since every switch in χ is the right-end point of a segment in χ_θ satisfying $\delta_j(\chi_\theta) > 2n$.

Recovering all submodels in $\mathcal{P}$ can be cast as a problem of clustering. A statistics based on $\hat{A}_{\varphi(k_i)}, \hat{b}_{\varphi(k_i)}, \hat{c}_{\varphi(k_i)}^T, \hat{d}_{\varphi(k_i)}$ may be chosen as the ℓ_1 -norm of the eigenvalues of $\hat{A}_{\varphi(k_i)}$, i.e., $\mathcal{M}(\hat{A}_{\varphi(k_i)}) = \sum_{i=1}^n |\lambda_i(\hat{A}_{\varphi(k_i)})|$ or the time-varying $\mathcal{H}_2$ norm of $\hat{\mathcal{P}}_{\varphi(k)}$. Since $\mathcal{M}(\hat{A}_{\varphi(k_q)}) = \mathcal{M}(\hat{A}_{\varphi(k_r)})$ whenever $\varphi(k_q) = \varphi(k_r)$, in the noiseless case, $\mathcal{P}_{\varphi(k_i)}$ may be determined by viewing the graph of $\mathcal{M}(\hat{A}_{\varphi(k_i)})$ over the left-end points of the segments in χ_θ with lengths at least δ_u . By running one of the clustering algorithms in [114, 115], we recover a collection of submodels in $\mathcal{P}$. It remains to show that this collection exhausts all submodels in $\mathcal{P}$; but, this follows from Assumption 5.5.2. We summarize the results derived above in the form of an identification algorithm.

Algorithm 5 Estimation of $\mathcal{P}$ and the switches in χ

Input: $u(k)$ and $y(k)$ for $n_1 \leq k \leq N$.

- ① Solve the problem in Eq.(6.5) for $\bar{\Theta}^\#$.
- ② Choose segments with $\delta_j(\chi_\theta) \geq \delta_u$.
- ③ Estimate $h(k, i)$ from Steps 1-4 in Algorithm 3 using $\theta^\#(k)$.
- ④ Estimate submodels from Algorithm 2.
- ⑤ Choose a statistics either $\mathcal{M}(\hat{A}_{\varphi(k_i)})$ or $\|\hat{\mathcal{P}}_{\varphi(k_i)}\|_2$.
- ⑥ Estimate $\mathcal{P}$ by running [114] over the segments in Step 2.
- ⑦ Declare $s_{j+1} \in \chi_\theta$ a switch in χ if $\delta_j(\chi_\theta) \geq n_1$

Output: $\hat{\mathcal{P}}$ and $\hat{\chi}$.

The gain $g(k_i)$ are found in closed-form by forcing n -roots of the characteristic polynomial of $A_o(k_i) = A(k_i) + g(k_i)c^T(k_i)$ to zero. Since $g(k_i) \in \mathbb{R}^n$, it is uniquely

determined if $A(k_i)$ has no eigenvalues at zero and $A(k_i)$ and $c^T(k_i)$ is an observable pair. If $g(k_i)$ is unique at every switch and Assumption 5.5.2 holds, then the gain sequence in Lemma 3.2.1 can have precisely σ distinct elements. If, on the other hand, $A(k_i)$ has at least one eigenvalue at zero, starting from a diagonal form, one can drive all nonzero eigenvalues to zero by selecting a gain from a subspace. As the same submodel appears at other switches, arbitrary choices can be made from this subspace. This means that the same submodel may have different observer Markov parameters on different segments.

Despite many possibilities exist for choosing the observer gains and the observer Markov parameters, the ensuing analysis above has shown that $\mathcal{P}$ may be retrieved entirely by minimizing a sparsity promoting norm in Eq.(6.5). We achieved this objective in several stages. First, we noted that the gain sequence in Lemma 3.2.1 is in the feasible parameter set of Eq.(6.5), and has the same switching sequence as the SLS model in Eqs.(1.1)–(1.5). Hence, any solution of Eq.(6.5) must be as sparse as or sparser than χ within a tolerance level. Then, we searched for constant parameter intervals of sufficient lengths in χ_θ . Lemma 5.2.4 demonstrated that these intervals are also long intervals in χ and the tolerance level was bounded by $2n$. In our calculations, we utilized the transformations between the system Markov parameters and the observer Markov parameters via the LTV realization theory.

The sparse optimization problem aims to flatten χ_θ as much possible as by choosing a suitable gain sequence. This process may render some switches unidentifiable from the input-output data. This is especially true at a switch k_i where $A(k_i)$ has at least one eigenvalue at zero. The minimality condition in Assumption 5.3.1 must then be applied to the subspace spanned by the gains at k_i .

The submodels in $\mathcal{P}$ are recovered from Algorithms (3,2) starting with the solution of Eq.(6.5). Algorithms (3,2) are LTV realization algorithms based on Markov parameter matching on the entire time axis $[n_1, N_1]$. The identified submodels are topologically equivalent to the SLS system in Eqs.(1.1)–(1.5) and they can be used to generate Markov parameters on subsets of $[n_1, N_1]$. We will exploit this feature to estimate the switching function from Markov parameters in the sequel. It is not necessary to bring the identified submodels into a common basis as in other works in the literature [41].

6.2. Switching sequence estimation in SARX models

Recall that if Lemma 5.2.4 holds, then a segment $[k_i, k_{i+1})$ in χ with $\delta_i(\chi) > 2n$ is mapped by the observer to the union of a segment $[s_j, s_{j+1})$ in χ_θ for some $j \in \mathbb{N}$

satisfying the inequalities $s_{j+1} = k_{i+1}$ and $0 \leq s_j - k_i \leq 2n$ and a partitioning of $[k_i, s_j)$ in χ_θ . The switches in $[k_i, s_j)$ may not even be identifiable from the input-output data. In addition, the solution of Eq.(6.5) over the segments in this interval are not necessarily equal to the true parameters. Lack of identifiability, however, may be overcome by resorting to the submodel directory. This situation is easily handled by noting that for all $k_i \leq k < s_j$, $\mathcal{P}_{\varphi(k)} = \mathcal{P}_{\varphi(k_i)}$ where k_i and $\mathcal{P}_{\varphi(k_i)}$ are identified for $[s_j, s_{j+1})$ by the scheme presented in Subsection 6.1.

The remaining problem is to determine which submodel is active on a given segment $[s_j, s_{j+1})$ with $\delta_j(\chi_\theta) < \delta_u$. The scheme in Subsection 6.1 cannot be applied to the present problem since $\theta^\#(k) = \theta(k)$ not necessarily applies to all $k \in [s_j, s_{j+1})$. Recall that adjustment of the gains is part of the optimization problem in Eq.(6.5) and the gains estimated for other intervals cannot be used for this interval. The estimation procedure we present here does not use the observer equation (5.3); but, the input-output equations (1.1)–(1.2).

Given $[s_j, s_{j+1})$ with $\delta_j(\chi_\theta) < \delta_u$, pick $q \in \mathbb{N}$ satisfying $q > n$ and form two Hankel matrices from the input-output data

$$\begin{aligned} \mathcal{U}_{s_j|q} &= \begin{bmatrix} u(s_j) & & u(s_{j+1}-q) \\ & \ddots & \vdots \\ u(s_j+q-1) & \cdots & u(s_{j+1}-1) \end{bmatrix}, \\ \mathcal{Y}_{s_j|q} &= \begin{bmatrix} y(s_j) & & y(s_{j+1}-q) \\ & \ddots & \vdots \\ y(s_j+q-1) & \cdots & y(s_{j+1}-1) \end{bmatrix}, \end{aligned} \quad (6.8)$$

and generate a Toeplitz matrix from A_l, b_l, c_l^T, d_l

$$\Omega_l = \begin{bmatrix} d_l & & \\ & \ddots & \vdots \\ c_l^T A_l^{q-2} b_l & \cdots & d_l \end{bmatrix}, \quad l \in \mathbb{S}.$$

When $\varphi(k_i) = l$ where $\varphi(k_i)$ is the index of the submodel active on $[s_j, s_{j+1})$ with $0 \leq s_j - k_i \leq 2n$, $s_{j+1} = k_{i+1}$, the following so-called data equation in subspace identification

$$\mathcal{Y}_{s_j|q} = \mathcal{O}_q \mathcal{X}_{s_j} + \Omega_{\varphi(k_i)} \mathcal{U}_{s_j|q} \quad (6.9)$$

can be derived with $\mathcal{X}$ denoting the compound state matrix

$$\mathcal{X}_{s_j} = [x(s_j) \quad \cdots \quad x(s_{j+1}-q)]. \quad (6.10)$$

Let the LQ decomposition of $[\mathcal{U}_{s_j|q}^T \ \mathcal{Y}_{s_j|q}^T]^T$ be given by

$$\begin{bmatrix} \mathcal{U}_{s_j|q} \\ \mathcal{Y}_{s_j|q} \end{bmatrix} = \begin{bmatrix} \mathcal{L}_{11} & 0 \\ \mathcal{L}_{21} & \mathcal{L}_{22} \end{bmatrix} \begin{bmatrix} \mathcal{Q}_1^T \\ \mathcal{Q}_2^T \end{bmatrix} \quad (6.11)$$

where $\mathcal{L}_{11}$ and $\mathcal{L}_{22}$ are lower triangular and $\mathcal{Q}_1$ and $\mathcal{Q}_2$ are orthogonal matrices with dimensions compatible with the row sizes of $\mathcal{U}_{s_j|q}$ and $\mathcal{Y}_{s_j|q}$. The multivariable output error state space (MOESP) method [121] estimates the observability range space by first performing the LQ decomposition in Eq.(6.11) and then applying the SVD to $\mathcal{L}_{22}$

$$\mathcal{L}_{22} = [U_1 \ U_2] \begin{bmatrix} \Sigma_1 & 0 \\ 0 & 0 \end{bmatrix} \begin{bmatrix} V_1^T \\ V_2^T \end{bmatrix} = U_1 \Sigma_1 V_1^T \quad (6.12)$$

where $\Sigma_1 \in \mathbb{R}^{n \times n}$. The extended observability matrix estimate is given by

$$\hat{\mathcal{O}}_q = U_1 \Sigma_1^{1/2}. \quad (6.13)$$

Then, substituting $q = 2n$, we estimate $A_{\varphi(k_i)}$ from Eq.(6.13)

$$\hat{A}_{\varphi(k_i)} = (J_D \hat{\mathcal{O}}_{2n})^\dagger J_U \hat{\mathcal{O}}_{2n} \quad (6.14)$$

and calculate the statistics $\mathcal{M}(\hat{A}_{\varphi(k_i)})$. We find the active model index by comparing $\mathcal{M}(\hat{A}_{\varphi(k_i)})$ with the statics of the submodels in $\hat{\mathcal{P}}$. The MOESP algorithm retrieves the observability pair $A_{\varphi(k_i)}, c_{\varphi(k_i)}^T$ up to a similarity transformation if the following three conditions

$$\text{rank}(\mathcal{X}_{s_j}) = n, \quad (6.15)$$

$$\text{rank}(\mathcal{U}_{s_j|q}) = q, \quad (6.16)$$

$$\text{span}(\mathcal{X}_{s_j}) \cap \text{span}(\mathcal{U}_{s_j|q}) = \{0\} \quad (6.17)$$

where $\text{span}(X)$ is the space spanned by the row vectors of a given matrix X are satisfied. The first condition is the controllability of the submodel active on $[s_j, s_{j+1})$ while the second is satisfied by selecting PE inputs of order q . The last condition requires data collected in open-loop experiment. A pseudo-code implementing this method is outlined below.

Algorithm 6 Estimation of $\varphi(k)$ by MOESP

Input: $u(k), y(k)$, $n_1 \leq k \leq N$ and $\mathcal{P}$.

- ① **while** $\delta_j(\chi_\theta) < \delta_u$ **do**
- ② Compute $\mathcal{U}_{s_j|q}$ and $\mathcal{Y}_{s_j|q}$ in Eq.(6.8).
- ③ Perform the LQ decomposition in Eq.(6.11).
- ④ Apply SVD to $\mathcal{L}_{22}$.
- ⑤ Compute $\hat{A}_{\varphi(k_i)}$ from Eqs.(6.13)–(6.14).
- ⑥ Compute $\mathcal{M}(\hat{A}_{\varphi(k_i)})$.
- ⑦ If $\mathcal{M}(\hat{A}_{\varphi(k_i)}) = \mathcal{M}(A_l)$, $l \in \mathbb{S}$, set $\varphi(k_i) = l$.
- ⑧ **end while**

Output: $\varphi(k_i)$.

An easier way is first to calculate $\Upsilon(l) = \mathcal{Y}_{s_j|q} - \Omega_l \mathcal{U}_{s_j|q}$ for each $l \in \mathbb{S}$, then apply SVD to $\Upsilon(l)$, next estimate $\hat{A}_{\varphi(k_i)}$ from Eqs.(6.13)–(6.14), and finally compute $\mathcal{M}(\hat{A}_{\varphi(k_i)})$. If l is right choice, then $\mathcal{M}(\hat{A}_l) = \mathcal{M}(\hat{A}_{\varphi(k_i)})$. Based on these observations, we propose a modified method to determine $\varphi(k_i)$. The following pseudo-code implements this modification.

Algorithm 7 A simplified index estimation method

Input: $u(k), y(k)$, $n_1 \leq k \leq N$ and $\mathcal{P}$.

- ① **while** $l \leq \sigma$ **do**
- ② Compute $\Upsilon(l) = \mathcal{Y}_{s_j|q} - \Omega_l \mathcal{U}_{s_j|q}$.
- ③ Apply SVD to $\Upsilon(l)$.
- ④ Compute $\hat{\mathcal{O}}_q$.
- ⑤ Compute $\hat{A}_{\varphi(k_i)}$.
- ⑥ Compute $\mathcal{M}(\hat{A}_{\varphi(k_i)})$.
- ⑦ If $\mathcal{M}(\hat{A}_{\varphi(k_i)}) = \mathcal{M}(A_l)$, set $l = \varphi(k_i)$ and stop.
- ⑧ **end while**

Output: $\varphi(k_i)$.

The identifiability conditions for the SLS model in Eqs.(1.1)–(1.2) with the MOESP algorithm are summarized in the following. Recall that they are used to determine which submodel is active on a given segment $[s_j, s_{j+1})$ satisfying $\delta_j(\chi_\theta) < \delta_u$.

Assumption 6.2.1 *For some $\delta^m > 6n$ and all j , $\delta_j(\chi_\theta) \geq \delta^m$ and Eqs.(6.15)–(6.17) are satisfied.*

Assumption 5.5.2 and Assumption 6.2.1 appear to be completely different. In fact, the former was stated for the ARX models whereas the latter was put forward for the state-space models. The ARX models are normally identified by least-squares

methods whereas state-space models are usually identified by subspace methods which rely on orthogonal/oblique projections and SVD. Conceivably, δ^m may be larger than δ_u . In this case, one may opt Algorithm 7 since it does not require the stringent PE conditions in Assumption 6.2.1. The main result of this section is stated as follows.

Theorem 6.2.1 *Consider the noise-free SLS model in Eqs.(1.1)–(1.5). Assume that the input-output data, χ , and $\mathcal{P}$ satisfy Assumptions 3.1.1 (strengthened to $\delta_*(\chi) > 6n$), 5.3.1–5.5.2, and 6.2.1. Then, Algorithm 5 followed by Algorithm 6 or Algorithm 7 recovers $\mathcal{P}$ and χ .*

6.3. Basis pursuit denoising recovery algorithm

set

$$X = \delta_\Theta = \begin{bmatrix} \delta_\theta(n_1) \\ \vdots \\ \delta_\theta(N_1) \end{bmatrix}, \quad (6.18)$$

consider the following robust reformulation of Eq.(6.5)

$$\begin{aligned} \min_{\bar{\Theta}} \quad & \|\delta_\theta\|_{0,\gamma} \\ \text{s.t.} \quad & \|Y - \mathcal{A}\bar{\Theta}\|_2 \leq \varepsilon \end{aligned} \quad (6.19)$$

to accommodate inaccuracies in the measurements. From the block-rows of X , Θ is recovered as $\theta(2n + k) = \sum_{j=1}^k X_j + \theta(2n)$. Since N is finite, $\delta_*(\chi) > 2n$, and χ has a finite number of switches in the noiseless case, the Assumptions in Theorem 6.2.1 still holds for χ if $\sup_k \|e(k)\|_2$ is sufficiently small. The rest of the assumptions hold as well when noise is small. The solution of Eq.(6.19) is sparse since $\bar{\Theta}^\sharp$ that solves Eq.(6.5) is a feasible solution to Eq.(6.19). Then, Algorithm 5 with the first step replaced by Eq.(6.19) recovers parameters of the SARX model in Eq.(5.3) on the interval $[n_1, N_1]$ if $\varepsilon \leq \varepsilon_0$ for some $\varepsilon_0 > 0$.

A convex relaxation of the ℓ_0 minimization problem in Eq.(6.19) through the basis pursuit denoising method (BPDN) leads to segmentation problems studied in [94, 34, 122, 123, 124]. In addition to this convexification, the orthogonal matching pursuit (OMP) algorithm [125] is another widely used algorithm due to its efficiency in computation. Both algorithms are suitable to recover sparse signals that have a block-sparse structure.

Recovery guarantees for these algorithms have been proposed in the compressive sensing/approximation literature. Three well-known recovery conditions are the restricted isometry property (RIP) [104], the exact recovery condition (ERC) [126],

and the mutual coherence [107]. A related concept is the spark used to find the sparsest vectors in kernels of matrices [127, 128]. The mutual coherence is easiest to calculate, but most conservative compared to others, which are usually hard or even impossible to calculate. The block-coherence measure was introduced in [76], and an uncertainty relation for block-sparse signals was derived based on this measure.

Convex relaxations of Eq.(6.5) are based on the recovery conditions for $\mathcal{A}$. Since the columns of $Z\mathcal{L}$ are derived from the first column by truncation, recovery guarantees for the measurement models $Y = \mathcal{A}\bar{\Theta}$ and $Y = Z\Theta$ are hardly useful for uniqueness, stability, and robustness studies. Usually one takes a leap of faith and believes that a suitable recovery condition holds. In fact, they hold with overwhelming probability, at least for randomly generated measurement matrices [129, 130]. The recovery conditions depend on the measurement matrix. Conceivably, models different than $Y = \mathcal{A}\bar{\Theta}$ or $Y = Z\Theta$ may yield less conservative recovery conditions. Nevertheless, via them, we were able to derive necessary and sufficient conditions for the switch and submodel identifiability of the model in Eq.(1.1)–(1.6) from the input/output data.

Now, we propose an efficient convex optimization problem instead of Eq.(6.5). Exact recovery will depend on recovery the conditions. Similarly to the ℓ_0 quasi-norm, we define the mixed ℓ_2/ℓ_1 norm over the interval $[n_1, N_1]$ by

$$\|X\|_{2,\mathcal{I}} = \sum_{k=1}^{N_2} \|X_k\|_2. \quad (6.20)$$

The algorithm we suggest is then

$$\begin{aligned} \min_{\bar{\Theta}} \quad & \|X\|_{2,\mathcal{I}} \\ \text{s.t.} \quad & Y = \mathcal{A}\bar{\Theta}. \end{aligned} \quad (6.21)$$

Treating the situation in which the observations are noisy, *i.e.*, $E \neq 0$ in Eq.(5.5), suppose $\|E\|_2 \leq \epsilon$. In order to recover $\bar{\Theta}$, we use the modified algorithm

$$\begin{aligned} \min_{\bar{\Theta}} \quad & \|X\|_{2,\mathcal{I}} \\ \text{s.t.} \quad & \|Y - \mathcal{A}\bar{\Theta}\|_2 \leq \epsilon. \end{aligned} \quad (6.22)$$

The constraint $\|Y - \mathcal{A}\bar{\Theta}\|_2 \leq \epsilon$ in Eq.(6.22) may be augmented to the objective function as a regularization term, *that is*, $\|X\|_{2,\mathcal{I}} + \lambda \|Y - \mathcal{A}\bar{\Theta}\|_2^2$, $\lambda > 0$. By changing λ , a family of pareto-optimal solutions are obtained. This approach is useful, in particular, when ϵ is not known. This mixed norm optimization problem can readily be solved by the alternative direction method of multipliers (ADMM) algorithm

[131]. A highly efficient and specialized version of the ADMM algorithm is applied in [132] to trend filtering.

A robust recovery result for Eq.(6.22) including best block sparse approximations of the true parameter matrix in addition to noise was derived, See, Theorem 2 in [106]. An identification procedure which allows one to extract submodel parameters one after another was presented in [133]. It is based on an ℓ_1 relaxation of a combinatorial ℓ_0 optimization problem and the measurement model in this work is different than ours.

The set of feasible solutions to Eq.(6.19), or likewise its convex relaxation in Eq.(6.22) is indeed a large set. In the underlying setup, gains play the role of shaping the feasible set of solutions. For instance, $g(k)$ presented in Lemma 3.2.1 suggests that gains changes solely at the switch points. This, then, suggests that feasible set of parameters could be shrinked by introducing new constraints to the optimization problem without of-course disturbing the underlying assumption of sparsity on the block differences. For now, let

$$\Lambda_i = \begin{bmatrix} h_o^T(n_1, n_1 - i) \\ \vdots \\ h_o^T(N_1, N_1 - i) \end{bmatrix} = \begin{bmatrix} \Lambda_i^1 \\ \vdots \\ \Lambda_i^{N_2} \end{bmatrix}, \quad i = 1, \dots, 2n, \quad (6.23)$$

we propose the following two variants of the constrained BPDN:

$$\begin{aligned} \min_{\bar{\Theta}} \quad & \|X\|_{2,\mathcal{I}} \\ \text{s.t.} \quad & \|Y - \mathcal{A}\bar{\Theta}\|_2 \leq \varepsilon \\ & \sum_{i=1}^{2n} \|\Lambda_i\|_2 \leq \rho. \end{aligned} \quad (6.24)$$

$$\begin{aligned} \min_{\bar{\Theta}} \quad & \|X\|_{2,\mathcal{I}} \\ \text{s.t.} \quad & \|Y - \mathcal{A}\bar{\Theta}\|_2 \leq \varepsilon \\ & \sum_{i=1}^{2n} \sum_{j=1}^{N_2} \|\Lambda_i^j\|_2 \leq \rho, \end{aligned} \quad (6.25)$$

which can both be written in a regularized format by appending the two constraints to the objective as follows

$$\min_{\bar{\Theta}} \|Y - \mathcal{A}\bar{\Theta}\|_2 + \lambda \|X\|_{2,\mathcal{I}} + \gamma \sum_{i=1}^{2n} \|\Lambda_i\|_2. \quad (6.26)$$

$$\min_{\bar{\Theta}} \|Y - \mathcal{A}\bar{\Theta}\|_2 + \lambda \|X\|_{2,\mathcal{I}} + \gamma \sum_{i=1}^{2n} \sum_{j=1}^{N_2} \|\Lambda_i^j\|_2, \quad (6.27)$$

with optimal values $\lambda, \gamma \geq 0$ to be selected by gridding. The problem in Eq.(6.26) penalizes block of parameters, hence block sparsity is sought, where all entries are either zero or not, together in a coupled fashion. The problem in Eq.(6.27), on the other hand, penalizes single entries to the block, which permits more freedom in selection, i.e, some entries to the blocks might not be zero while others are. The problem in Eq.(6.26) is used when a fixed SARX order is sought throughout the entire time horizon, whereas Eq.(6.27) permits a time-varying order by the proper selection of sparsity. The problem in Eq.(6.26) is more computationally appealing than the problem in Eq.(6.27), this becomes even more pronounced when the re-weighting mechanism is introduced in the sequel. This, however, is paid by sacrificing the flexibility in the parametrization. To fix the SARX order at $\tau^* = n$ throughout entire time, Eq.(6.26) is to be opted. However, this forces the gains to be time-varying around the switches to accommodate for the insufficiency in the parameters required to properly model the system in that region. The gains in this case exhibit a time-varying behaviour around switches and piecewise constant elsewhere. Contrarily, if one desires to have a time varying order, say $\tau^* = n$ in the regions $[k_i + n, k_{i+1})$ and more than n around switches, (6.27) is recommended instead. This latter scenario arises when the gains witness a change only at the switch points, in accordance with SLS submodels, as stipulated in Lemma 3.2.1. These results highlight how the choice of observer gains can be thought of to be a design problem, accomplished by enforcing a pattern of sparsity on the parameters, which also shows the flexibility of our devised approach to the problem.

By enforcing sparsity on the parameter vector after a certain optimal order τ^* point-wisely in time, we are basically seeking a sharp decay in the parameters, a criteria which is delivered by a deadbeat observer. Furthermore, if the dynamic matrix has no eigenvalues originally placed at the origin, which indicates the uniqueness of the optimal deadbeat sequence, then the optimization problem induced gain sequence is a unique optimal deadbeat sequence throughout time, and hence piecewise constant.

6.3.1. Iterative refinement

The constrained ℓ_1 relaxations in Eqs.(6.24),(6.25) of the problem in Eq.(6.19) are the tightest convex relaxation of the objective. A better heuristic, the iterative weighted ℓ_1 relaxation, was proposed in [89, 134]. The iterative refinement procedure when applied to problem defined in Eq.(6.26) takes the following form

Algorithm 8 The constrained BPDN 1

Input: $\lambda \geq 0, \gamma \geq 0$, and maximum number of iterations l_{max} , initialize the weights $w_k^{(0)} = 1, v_i^{(0)} = 1$ for all k, i .

① **while** $l \leq l_{max}$ **do**

② Solve the problem:

$$\min_{\bar{\Theta}} \|Y - \mathcal{A}\bar{\Theta}\|_2 + \lambda \sum_{k=1}^{N_2} w_k^{(l)} \|X_k\|_2 + \gamma \sum_{i=1}^{2n} v_i^{(l)} \|\Lambda_i\|_2$$

③ Update weights by setting:

$$\begin{aligned} w_k^{(l+1)} &= (\mu + \|X_k\|_2)^{-1} \\ v_i^{(l+1)} &= (\mu + \|\Lambda_i\|_2)^{-1} \end{aligned}$$

for some fixed $\mu > 0$ and all $k = 1, \dots, N_2, i = 1, \dots, 2n$.

④ **end while**

⑤ Estimate $\theta(k)$ in the range $[n_1, N_1]$ as $\theta(2n+k) = \sum_{j=1}^k X_j + \theta(2n)$

Output: $\hat{\theta}(k), k = 2n, \dots, N_1$.

On the other hand, when applied to the problem in Eq.(6.27)

Algorithm 9 The constrained BPDN 2

Input: $\lambda \geq 0, \gamma \geq 0$, and maximum number of iterations l_{max} , initialize the weights $w_k^{(0)} = 1, v_{i,j}^{(0)} = 1$ for all k, i, j .

① **while** $l \leq l_{max}$ **do**

② Solve the problem:

$$\min_{\bar{\Theta}} \|Y - \mathcal{A}\bar{\Theta}\|_2 + \lambda \sum_{k=1}^{N_2} w_k^{(l)} \|X_k\|_2 + \gamma \sum_{i=1}^{2n} \sum_{j=1}^{N_2} v_{i,j}^{(l)} \|\Lambda_i^j\|_2$$

③ Update weights by setting:

$$\begin{aligned} w_k^{(l+1)} &= (\mu + \|X_k\|_2)^{-1} \\ v_{i,j}^{(l+1)} &= (\mu + \|\Lambda_i^j\|_2)^{-1} \end{aligned}$$

for some fixed $\mu > 0$ and all $k = 1, \dots, N_2, i = 1, \dots, 2n$, and $j = 1, \dots, N_2$.

④ **end while**

⑤ Estimate $\theta(k)$ in the range $[n_1, N_1]$ as $\theta(2n+k) = \sum_{j=1}^k X_j + \theta(2n)$

Output: $\hat{\theta}(k), k = 2n, \dots, N_1$.

Note how the second constraint in Algorithm 8 has only $2n$ weights, whereas in Algorithm 9 it has $2nN_2$. This highlight our previous claim on computational complexity. Given the final solution from either algorithms, a final refinement step

is conducted. The problem is solved again after adjusting the weights manually; the weights that are below some threshold are set to zero, whereas the weights above that threshold are set to a very large number, say 10^5 , this will help improve the fit over the constants segments. Many convex optimization standard solvers could be used to implement the presented algorithms. In this work, CVX package is used [86], which converts the problem to a second-order cone program (SOCP) and calls a standard interior-point cone solver.

6.4. Summary of the proposed algorithm for identifying SLSs

Figure 6.1 presents a flowchart to implement the proposed SLS Algorithm. The writings beside the curved thick arrows show which algorithm or procedure used to achieve that step, the curved boxes display the output at that step.

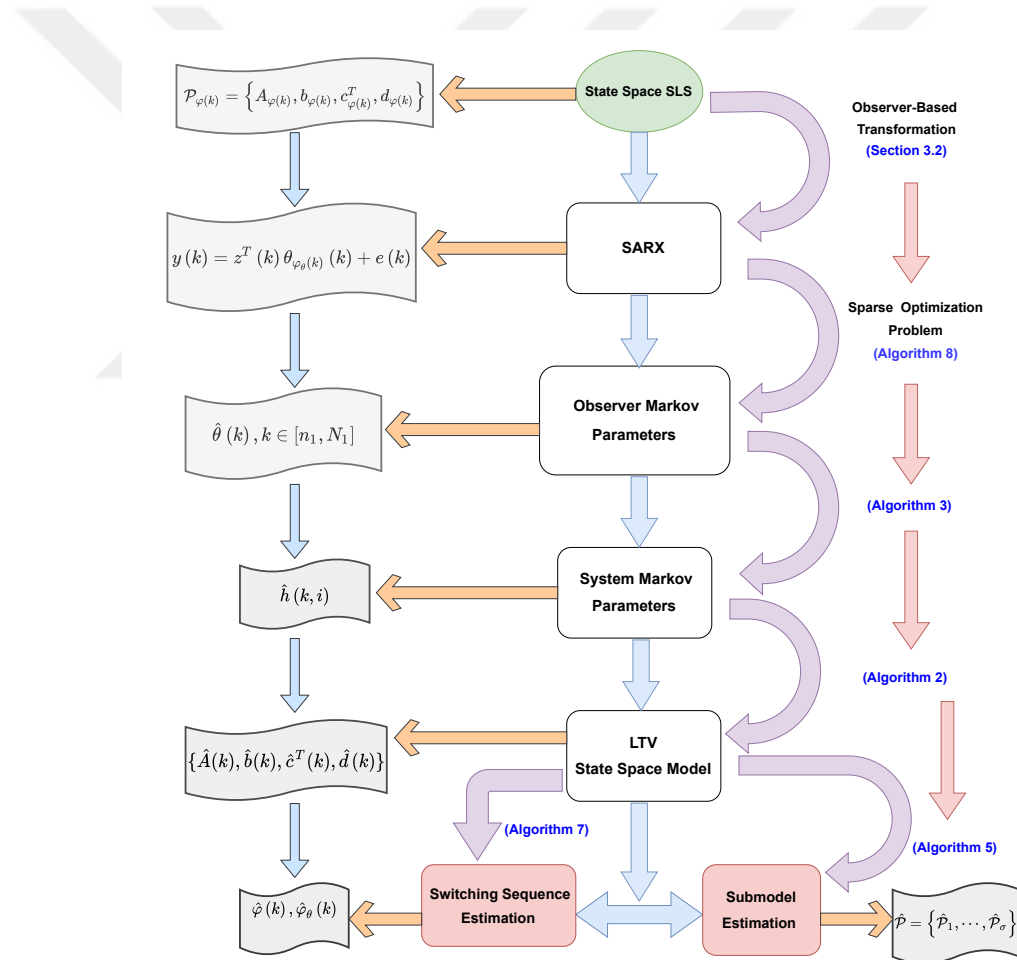


Figure 6.1: Flow diagram for the proposed algorithm to identify switched linear system.

6.5. Example

Consider the same SLS as in Section 5.7. The initial state is taken $x(0) = [1, 1]^T$, and a switching function was sampled from a uniform distribution. To test the capabilities of the derived algorithm, fast dynamic switching is adopted by considering a dwell time of $\delta_*(\chi) = 27$. This system is excited with a multisine input consisting of five superposed harmonics generated as in Section 5.6, we generate $N = 2,000$ data pairs (u_k, y_k) to be used in the identification. The output was subsequently contaminated with Gaussian additive noise to achieve different SNR levels of 40db, 30db, and 20db.

As a first step, Algorithm 8 is implemented via CVX package [86]. The hyperparameters λ and γ are selected via gridding. We set $\mu = 10^{-5}$ and $l_{max} = 5$ in Algorithm 8; Prior to this, a refinement step was done to enhance the fit. Once θ is obtained, segments over which $\theta = \theta^\sharp$, i.e., $\delta_j(\chi_\theta) \geq \delta_u$ are used to estimate $h(k, i)$ by Algorithm 3. In this example the first three segments are long enough, therefore they were used to retrieve the SLS submodels. After selecting long segments that contain submodels information, Algorithm 2 with default values $q = r = 2n = 4$ is used to obtain the LTV submodels. Given the LTV model, the statistics $\mathcal{M}(\hat{A}(k))$ is computed, and the DBSCAN clustering is done to retrieve the submodels. Figure 6.2 depicts the histogram resulted from clustering at an SNR of 30db. We can clearly see that the correct number of submodels was identified, i.e., $\hat{\sigma} = 3$.

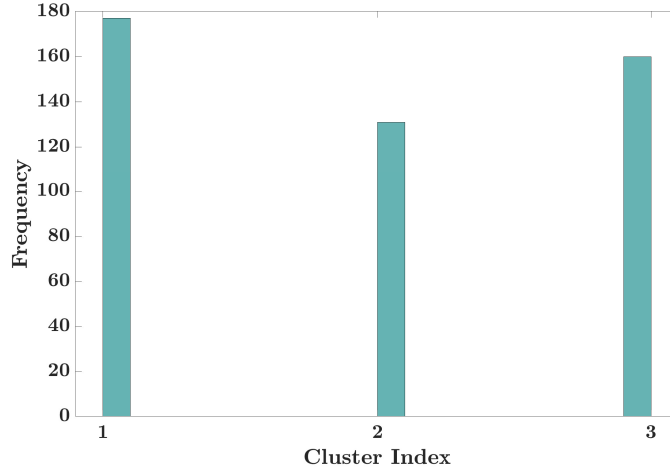


Figure 6.2: the histogram of DBSCAN clustering from Algorithm 5 to retrieve sub-models at one noise realization with SNR=30db.

Figure 6.3 shows the estimated and true eigenvalues of SLS submodels at three different SNR levels. Notice the strong consistency of the estimates even at high noise levels.

Given the system submodels, the observer gains are calculated from Eq.(3.60),

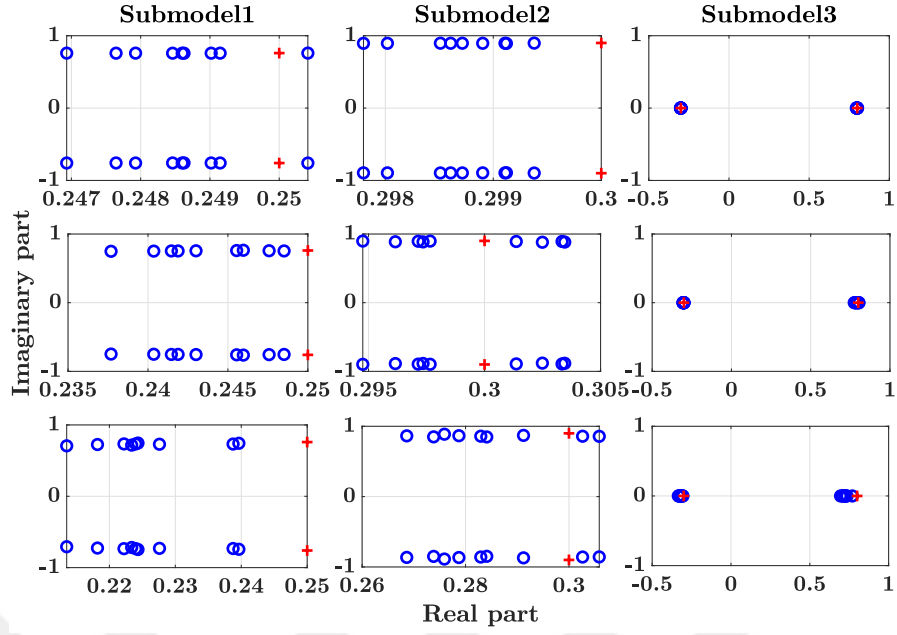


Figure 6.3: The true and the estimated system eigenvalues: the true '+', the estimated 'o' at SNR levels 40db, 30db and 20db top to bottom respectively at 10 different noise realizations.

and following that the 3 deadbeat observer submodels are calculated. Figure 6.4 shows the eigenvalues of $A_o(k)$ at an SNR of 30db, Note that they are properly placed at the origin in only two steps $\tau^* = n = 2$, which clearly shows that the underlying observer is indeed an optimal deadbeat observer.

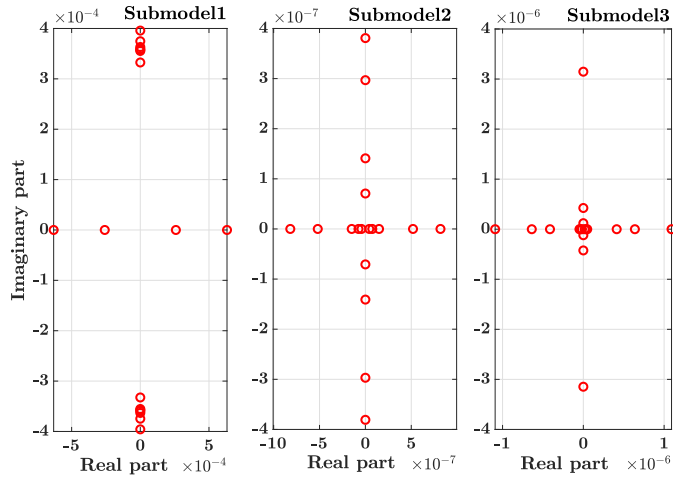


Figure 6.4: The observer system eigenvalues at an SNR of 30db for 10 different noise realizations.

By employing Algorithm 7, both the SLS and SARX switching functions are estimated in the range $[n_1, N_1]$ excluding the long segments. Figure 6.5 depicts the estimated SLS switching function χ plotted along with the true one at an SNR of 30db for one noise realization, there we can see the exact match to the true one except

at few points zoomed in Figure 6.6 for clarity. Figure 6.7, on the other hand, depicts the estimated SARX switching function $\hat{\chi}_\theta$. We can differentiate two set of points, the first set marks the points that properly belong to one of the SARX submodes, whereas the second set marks the points generated from mixed submodels, with length of at most $2n$, they were assigned an index value of 0. Figure 6.8 depicts the deadbeat observer gains components at an SNR of $30db$ for one noise realization over a time interval of $[n_1, N_1]$. The estimated SARX submodels at one noise instant at an SNR of $30db$ over the first three segments (long segments) are given by

$$\theta_1 = \begin{bmatrix} 0.968 \\ 1.108 \\ 0.501 \\ 4.611 \\ -0.636 \\ -2.921 \times 10^{-19} \\ -2.350 \times 10^{-18} \\ -8.216 \times 10^{-19} \\ -3.351 \times 10^{-18} \end{bmatrix}, \theta_2 = \begin{bmatrix} -1.531 \\ 2.394 \\ 0.607 \\ -3.524 \\ -0.885 \\ -2.719 \times 10^{-19} \\ -8.017 \times 10^{-18} \\ -2.321 \times 10^{-18} \\ 1.439 \times 10^{-18} \end{bmatrix}, \theta_3 = \begin{bmatrix} 1.98 \\ 7.038 \\ 0.476 \\ -4.496 \\ 0.239 \\ 7.668 \times 10^{-19} \\ -6.869 \times 10^{-19} \\ -2.134 \times 10^{-19} \\ 2.084 \times 10^{-18} \end{bmatrix}$$

Note that the last two block rows of the SARX submodels are practically zero, which clearly highlight that this part is originally overparametrized and that $\tau^* = n = 2$ in the region $[k_i + 2, k_{i+1})$ for all i as pointed out by Lemma 3.2.1. This also shows that overparametrization is taken care of by the algorithm successfully.

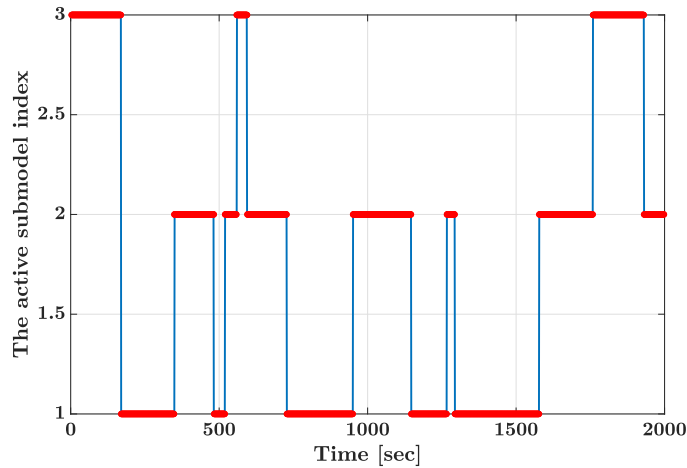


Figure 6.5: The estimated switching function $\hat{\chi}$ (dotted) and the true one χ (solid) for SLS model.

We end this numerical example by studying the average performance of the proposed algorithm when it is subjected to random measurement noise. We perform Monte Carlo simulations employing relative normed errors of the three estimated

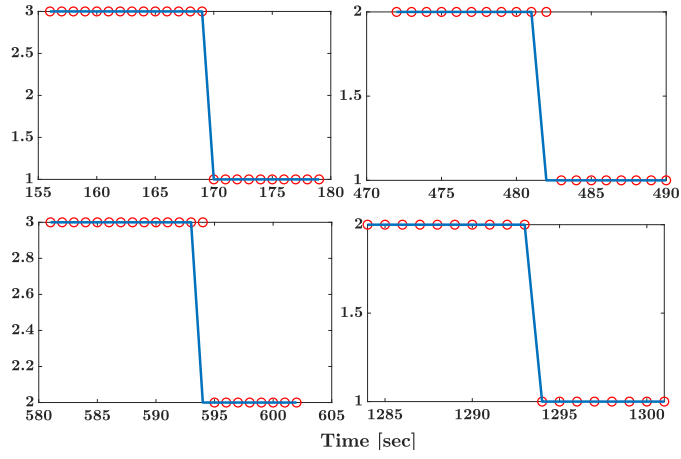


Figure 6.6: The size of the observation window in Figure 6.5 is shortened for visualization purpose.

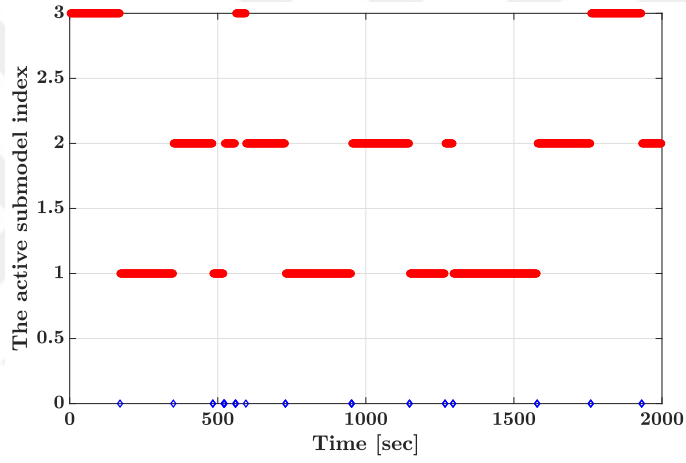


Figure 6.7: The estimated switching function $\hat{\chi}_\theta$ for SARX.

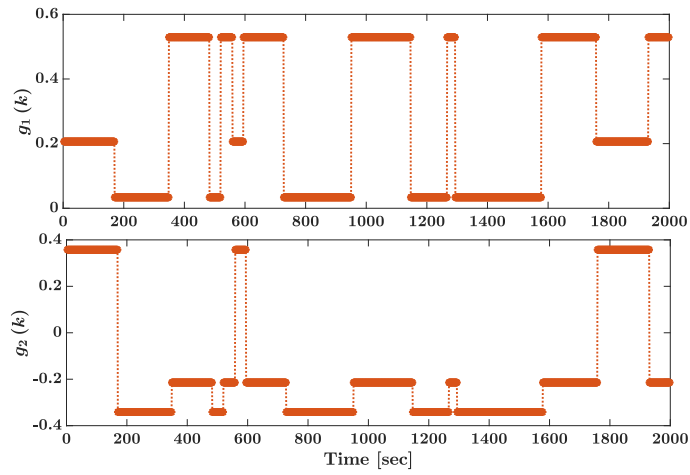


Figure 6.8: The Deadbeat observer gains $g(k)$.

state space quadruplets $\{\hat{A}_i, \hat{b}_i, \hat{c}_i^T, \hat{d}_i\}$, $i = 1, \dots, \sigma$ as an evaluation criteria. Table 6.1 displays the results obtained over 10 noise realizations; this error criteria was used

i	40db	30db	20db
1	0.0023	0.0121	0.0536
2	0.0084	0.0253	0.0677
3	0.0014	0.0058	0.0276

Table 6.1: Monte Carlo simulation results for the three submodels of the SLS model estimated by the proposed algorithm.

in [36] and is defined by

$$\frac{\|M_i - \hat{M}_i\|_F}{\|M_i\|_F}, \quad (6.28)$$

with

$$\hat{M}_i = \begin{bmatrix} \hat{A}_i & \hat{b}_i \\ \hat{c}_i^T & \hat{d}_i \end{bmatrix}. \quad (6.29)$$

Notice that the algorithm still gives good estimates even at relatively high noise levels. Furthermore, the estimated model capability of constructing the output given the true input signal was tested on a new data set different than that used for identification. This was achieved via calculating the variance-accounted-for (VAF) defined by:

$$VAF = \max \left\{ 1 - \frac{\text{var}(y(k) - \hat{y}(k))}{\text{var}(y(k))}, 0 \right\} \times 100\%. \quad (6.30)$$

The estimated SLS can not be used for output prediction right away. The submodels have been identified in a separate manner, hence they are given up to a similarity transformation which is different for each submodel. This highlights the need to recover a common state basis among all submodels. This issue could be resolved by reverting to canonical representations to maintain control of the state basis in which the system matrices are, for instance the observability canonical form. Recall that the similarity transformation to transform submodel i given by the state space quadruplet $\{A_i, b_i, c_i^T, d_i\}$ to the observability canonical form is given by $T = \mathcal{O}(A_i, c_i^T)$, provided that the pair (A_i, c_i^T) is observable. After applying the transformation, the submodels could be combined harmlessly. Figure 6.9 depicts the true and estimated output on the validation data set; only the first 500 data points were plotted for visualisation purpose. Note the high VAF value, which indicates the superiority of the estimate.

6.6. Summary

In this chapter, sparsity was exploited in the formulation of a non-convex sparse optimization problem through which the submodels, SLS switch set, and SARX

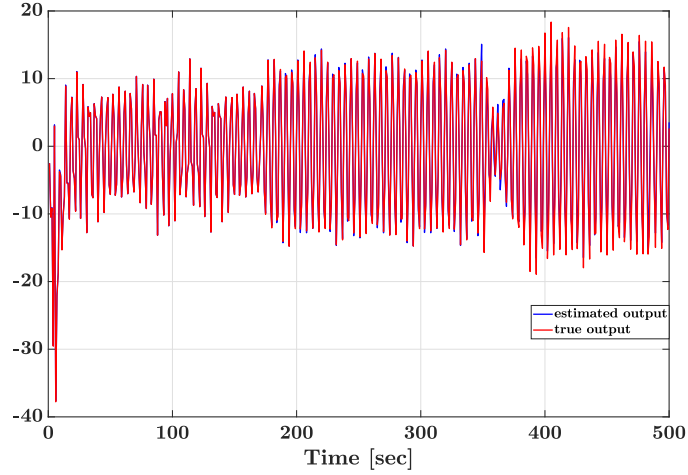


Figure 6.9: Validation data with an SNR of 30 dB, obtained VAF = 98.28%.

switch set are all recovered. It was noted that if one wishes to extract the submodels, long enough segments of the obtained SARX model with length greater than some threshold δ_u are targeted. The application of Algorithm 3 to recover the system parameters, then Algorithm 2 to obtain a pointwise LTV realization over the targeted segments, followed next by a DBSCAN clustering of a statistics of the obtained realizations proves to reveal all SLS submodels. Along with that, the switch sequences were also acquired. To extract the rest of the switching function by finding the submodel responsible for generating the input-output data in short to intermediate segments, a MOESP algorithm was imported here, it only makes use of the input-output data. However, due to the stringent PE requirement that MOESP might pose, we proposed a less stringent variant, a simple index determining scheme that exploits the dictionary of submodels obtained previously. A convex-relaxation of the previous problem was proposed. Along with that, to reduce the feasible set of solutions which is shaped by the gains, we explored the addition of new constraints to the relaxed version of the sparsity problem, two constraints were put available. This gave rise to two distinct constrained basis pursuit denoising problems, which when coupled with the reweighting mechanism yields the reweighted constrained BPDN, a better heuristic that works iteratively. In the sequel we conclude this thesis by giving final remarks and suggested extensions to the work presented throughout the chapters.

7. CONCLUSION AND AN OUTLOOK FOR FUTURE RESEARCH DIRECTIONS

7.1. Conclusions

This thesis witnessed significant and novel contributions to the literature. In chapter 3, a state-space LTV realization algorithm from a sequence of doubly-indexed generalized Markov parameters was presented. This algorithm is applicable to arbitrary LTV systems, yet it was used in this thesis to obtain a LTV realization of a SLS model. Next, a novel approach to transform MIMO SLS models to the input/output SARX models was presented. This was achieved by modulated outputs. The modulation terms, i.e., the observer gains, were assumed to be deadbeat gains, so that the dependence on the initial state vanishes and the number of Markov parameters required to represent the system are compressed. To our best knowledge, this approach was utilized on LTI systems and arbitrary LTV systems; but not on hybrid systems. Furthermore, we presented a recursive mechanism to revert back from the observer domain to the system domain; “the obtention of the system Markov parameters starting from the observer Markov parameters”. Along with that, the observer gain calculations were also performed. The effect of the process and measurement noise injection on the observer-based transformation was also discussed in detail. Chapter 4 presented an algorithm that retrieves the SLS submodels and its switching function from its LTV realization. There, it has been shown that under the assumption that all submodels are visited at least once with long enough waiting time, taken to be $5n$ here, they can be retrieved successfully. The fact that the LTV quadruplets are in similarity transformation with the SLS submodels over subintervals of long segments was exploited in submodel extraction. Submodel retrieval was achieved via clustering the vectorized state-space quadruplets, or a statistic of the quadruplets, i.e., the ℓ_1 norm of dynamic matrix eigenvalues or the $\mathcal{H}_2$ of the time-varying transfer function. Next, under the knowledge of submodels, one can always retrieve the switching function by clustering the $2n$ dubious points resulting from long segments and the points resulting from short segments. Two discrete optimization solving based approaches were proposed. The first is with respect to the system eigenvalues while the second one is with respect to Markov parameters matching. Both the forward in time and backward in time versions for the Markov parameters matching approach were proposed.

Chapter 5 is concerned with the switch identifiability conditions relying heav-

ily on the results derived in [120]. They were extended to the MIMO setting. Two conditions are necessary for a switch to not go undetectable. The first requires a non-vanishing dot product between the parameter changes and the regressor at the switches, while the second is for the regressor at the switches to be in the range space spanned by the previous regressors. These two conditions were noted to be mild by means of a numerical example employing superposed harmonics as inputs. Next, segments in SLS models were related to those in SARX models. The fact that the SLS switch set is a subset of the SARX switch set was also established. Furthermore, the existence of LTI realizations of the SLS submodels and gains within subintervals of long SARX segments were as well emphasized. Persistence of excitation conditions were derived for the SISO SARX models, and following matrix fraction decomposition approach techniques for MIMO SARX systems as well. Ultimately, a PE condition on segments length was put to ensure submodels recovery.

In Chapter 6, the SLS submodels and switch sequence informations were extracted over long segments obtained from solving a non-convex sparse optimization problem from the input/output data. For detecting submodels which are active over intermediate to short segments from the input output data, we utilized MOESP subspace algorithm. A further simplified and less stringent MOESP version that employs the previously identified submodels to decide the active submodel index is also presented. A relaxation to the previous non-convex problem was pursued. This relaxation revealed the BPDN algorithm, well known in compressive sensing literature. New constraints were added to constrain the feasible set of solutions. This ultimately gives rise to two constrained BPDN algorithms which can be implemented via the CVX package.

7.2. Future Research

Several extensions of the current work are proposed here:

- Unknown submodel order case is worth investigating.
- Extension of the presented algorithm to output-only data.
- Extension to the PWASS model structure.
- Concentration of measure (COM) inequalities derivations for SARX and SLS model structures. The relation to COM of structured Gaussian random variables is also a possible future work.
- Applying the developed identification algorithm to real life problems, i.e., structural health conditions monitoring, fault detection, road profile modelling, etc.

Appendix A.

PROOFS

A.1. Proof of Lemma 3.1.1

Let $k \in \mathbb{N}$ be a given number bounded by k' from the below and by k'' from the above. We will fix k' and k'' later. Consider the interval $[k - 2n, k - 1]$. Since $\delta_*(\chi) \geq 2n$, there exists $t_k \in \mathbb{N}$ such that $T_k = [t_k, t_k + n - 1] \subset [k - 2n, k - 1]$ and $\mathcal{P}_{\varphi(j)} = \mathcal{P}_{\varphi(t_k)}$ for all $j \in T_k$. It suffices to let $k' = 2n + 1$. Now, from Eq.(3.8)

$$\begin{aligned}
 G_c(k, 2n) &= \sum_{j=k-2n}^{k-1} \Phi(k, j+1) B(j) B^T(j) \Phi^T(k, j+1) \\
 &\geq \sum_{j=t_k}^{t_k+n-1} \Phi(k, j+1) B(j) B^T(j) \Phi^T(k, j+1) \\
 &= \sum_{j=t_k}^{t_k+n-1} A_{\varphi(t_k)}^{k-j-1} B_{\varphi(t_k)} B_{\varphi(t_k)}^T [A_{\varphi(t_k)}^T]^{k-j-1} \\
 &= [A_{\varphi(t_k)}]^{k-t_k-n} \sum_{j=0}^{n-1} A_{\varphi(t_k)}^j B_{\varphi(t_k)} B_{\varphi(t_k)}^T \\
 &\quad \cdot [A_{\varphi(t_k)}^T]^j [A_{\varphi(t_k)}^T]^{k-t_k-n}.
 \end{aligned}$$

Since $\mathcal{P}_{\varphi(t_k)}$ is stable, the first term on the right-hand side of the last equality is nonsingular. The center term is positive-definite since the minimality of submodel $\mathcal{P}_{\varphi(t_k)}$ implies its controllability. It follows that $G_c(k, 2n) > 0$ and $2n$ is a feasible value for κ_c .

The proof that $G_o(k, 2n) > 0$ and $2n$ is a feasible value for κ_o is similar. The constraint $k + 2n - 1 \leq N$ suggests picking $k'' = N - 2n + 1$. Choose $s_k \in \mathbb{N}$ such that $\mathcal{P}_{\varphi(j)} = \mathcal{P}_{\varphi(s_k)}$ for all $j \in S_k$ and $S_k = [s_k, s_k + n - 1] \subset [k, k + 2n - 1]$.

It is easy to see that $G_c(k, 2n)$, $G_o(k, 2n)$, $\Phi(k, k - 2n)$, and $\Phi(k + 2n, 2n)$ are all bounded above. Thus, Eqs.(3.5)–(3.7) and Eqs.(3.9)–(3.11) hold with suitable constants. It follows that the SLS model in Eqs.(1.3)–(1.6) is uniform on the interval $[2n + 1, N - 2n + 1]$ with $\kappa_c = \kappa_o = 2n$ since σ is finite.

A.2. Proof of Lemma 3.2.1

For each $1 \leq k \leq N$, note that

$$\min \{j \in \mathbb{N} : (A_{\varphi(k)} + K(k)C_{\varphi(k)})^j = 0\} \leq n. \quad (\text{A.1})$$

Choose the sequence $K(k)$ so that $K(k_1) = K(k_2)$ whenever $\varphi(k_1) = \varphi(k_2)$. Thus, $K(k)$ is a sequence of at most σ matrices denoted by $K_{\varphi(k)}$. Given $k \geq 1$, note that

$$\Phi_o(k, k - 2n) = \Phi_o(k, t_k + n)\Phi_o(t_k + n, t_k)\Phi_o(t_k, k - 2n) \quad (\text{A.2})$$

for some interval $T_k = [t_k, t_k + n] \subset [k - 2n, k]$ such that $\mathcal{P}_{\varphi(j)} = \mathcal{P}_{\varphi(t_k)}$ for all $j \in T_k$. This is possible since $\delta_*(\chi) \geq 2n$. Now from Eq.(A.1),

$$\Phi_o(t_k + n, t_k) = (A_{\varphi(t_k)} + K_{\varphi(t_k)}C_{\varphi(t_k)})^n = 0. \quad (\text{A.3})$$

Hence from Eqs.(A.2)–(A.3), $\Phi_o(k, k - 2n) = 0$ for all $k' \leq k \leq k_{i^*}$ with $k' = 2n + 1$ since $k_{i^*-1} - k_{i^*} \geq 2n$. Set $k'' = N - 2n$. If $k'' \leq k_{i^*}$, then $\Phi_o(k, k - 2n) = 0$ for all $k' \leq k \leq k''$. Suppose $k'' > k_{i^*}$ and consider the case $k_{i^*} < k \leq k''$. If $k - n < k_{i^*}$, set $t_k = k - 2n$ in Eq.(A.2) and Eq.(A.3) and note that $\varphi(j) = \varphi(k_{i^*-1})$ for all $j \in T_k$. If $k - n > k_{i^*}$, set $t_k = k - n$ in Eq.(A.2) and Eq.(A.3) and note that $\varphi(j) = \varphi(k_{i^*})$ for all $j \in T_k$. It follows that Definition 3.2.1 is satisfied for the sequence $K_{\varphi(k)}$, $\tau = 2n$, $k' = 2n + 1$, and $k'' = N - 2n$. Hence $\tau^* \leq 2n$.

If $k \in [k_i + n, k_{i+1})$, then

$$\Phi_o(k, k - n) = \prod_{j=1}^n (A_{\varphi(k_i)} + K(k - j)C_{\varphi(k_i)}). \quad (\text{A.4})$$

Fix the observer gains as $K(k - j) = K_{\varphi(k_i)}$, $j = 1, \dots, n$. Then, from Eq.(A.1) $\Phi_o(k, k - n) = 0$.

A.3. Proof of Lemma 5.1.1

Since $z(s_{j+1}) \in \text{range}(\mathcal{R}_{s_j, s_{j+1}-1})$, $\mathcal{R}_{s_j, s_{j+1}-1}v = z(s_{j+1})$ for some $v \neq 0$. Consider two pairs $\{\theta_1, \theta_2\}$ and $\{\tilde{\theta}_1, \tilde{\theta}_2\}$ satisfying Eq.(5.11). Then,

$$\begin{aligned} & \{\theta_2 - \theta_1\} z(s_{j+1}) - \{\tilde{\theta}_2 - \tilde{\theta}_1\} z(s_{j+1}) \\ &= \theta_2 z(s_{j+1}) - \tilde{\theta}_2 z(s_{j+1}) + \tilde{\theta}_1 z(s_{j+1}) - \theta_1 z(s_{j+1}) \\ &= \tilde{\theta}_1 \mathcal{R}_{s_j, s_{j+1}-1}v - \theta_1 \mathcal{R}_{s_j, s_{j+1}-1}v \\ &= Y_{s_j, s_{j+1}-1}v - Y_{s_j, s_{j+1}-1}v = 0. \end{aligned}$$

Thus, $(\theta_2 - \theta_1)z(s_{j+1})$ is a constant vector denoted by γ .

A.4. Proof of Lemma 5.1.2

We first prove the necessity of the conditions. Clearly, the first condition in Eq.(5.12) is necessary for a switch to be causally identifiable. To show that the second condition in Eq.(5.12) is also necessary, suppose that it fails. Let $\mathcal{R}_{s_j, s_{j+1}-1}^\perp$ denote a basis for the orthogonal complement of $\mathcal{R}_{s_j, s_{j+1}-1}$. Then, $v^T = z^T(s_{j+1})\mathcal{R}_{s_j, s_{j+1}-1}^\perp \neq 0$. Define

$$\hat{\theta} = \theta_1 + \|v\|_2^{-2}(y(s_{j+1}) - \theta_1 z(s_{j+1}))(\mathcal{R}_{s_j, s_{j+1}-1}^\perp v)^T. \quad (\text{A.5})$$

Multiplication of Eq.(A.5) from right by $z(s_{j+1})$ yields

$$\begin{aligned} \hat{\theta} z(s_{j+1}) &= \theta_1 z(s_{j+1}) + \|v\|_2^{-2}(y(s_{j+1}) - \theta_1 z(s_{j+1}))\|v\|_2^2 \\ &= y(s_{j+1}). \end{aligned} \quad (\text{A.6})$$

Multiplication of Eq.(A.5) from right by $\mathcal{R}_{s_j, s_{j+1}-1}$ yields

$$\begin{aligned} \hat{\theta} \mathcal{R}_{s_j, s_{j+1}-1} &= \theta_1 \mathcal{R}_{s_j, s_{j+1}-1} + \|v\|_2^{-2}(y(s_{j+1}) - \theta_1 z(s_{j+1})) \\ &\quad \cdot (\mathcal{R}_{s_j, s_{j+1}-1}^T \mathcal{R}_{s_j, s_{j+1}-1}^\perp v)^T \\ &= \theta_1 \mathcal{R}_{s_j, s_{j+1}-1} = Y_{s_j, s_{j+1}-1}. \end{aligned} \quad (\text{A.7})$$

Thus, from Eqs.(A.6)–(A.7) the model

$$\hat{\theta} z(k) = y(k), \quad s_j \leq k \leq s_{j+1} \quad (\text{A.8})$$

explains the data and the switch is not causally identifiable from the input/output data.

Now, we prove the sufficiency of the conditions. Suppose $z(s_{j+1}) \in \text{range}(\mathcal{R}_{s_j, s_{j+1}-1})$ and $(\theta_2 - \theta_1)z(s_{j+1}) \neq 0$. Then, from Lemma 5.1.1 any pair $\{\theta_1, \theta_2\}$ satisfying Eq.(5.11) satisfies also for a fixed γ , $(\theta_2 - \theta_1)z(s_{j+1}) = \gamma \neq 0$. Write Eq.(5.11) as

$$\begin{aligned} \theta_1 z(k) &= y(k), \quad s_j \leq k < s_{j+1}, \\ \theta_2 z(s_{j+1}) &= y(s_{j+1}). \end{aligned}$$

Since $\theta_1 \neq \theta_2$, there can not be a single parameter as in Eq.(A.8) explaining the data

in $\{s_j, s_{j+1}\}$. Hence, the switch is causally identifiable from the input/output data.

A.5. Proof of Lemma 5.1.3

The left-hand of the Eq.(5.13) is larger than or equal to the right-hand side since $\text{rank}([A \ B]) \geq \text{rank}(A)$ for any two matrices A and B with the same number of rows. The inequality in Eq.(5.13) is strict. If not, $Y_{s_j, s_{j+1}} = \theta \mathcal{R}_{s_j, s_{j+1}}$ for some θ and Eq.(A.8) holds. Hence, Eq.(5.12) is violated.

A.6. Proof of Lemma 5.2.1

For $k_i \leq k < k_{i+1}$ and $1 < \ell \leq 2n$, note that

$$h_o(k, k - \ell) = C(k) \prod_{s=1}^{\ell-1} A_o(k - s) B_o(k - \ell). \quad (\text{A.9})$$

If $k \geq k_i + 2n$, then for all $1 < \ell \leq 2n$

$$h_o(k, k - \ell) = C(k_i) A_o^{\ell-1}(k_i) B_o(k_i). \quad (\text{A.10})$$

where we used $K(k) = K(k_i)$ for all $k \in [k_i, k_{i+1})$. Equation (A.10) holds for $\ell = 1$ and $h_o(k, k) = [D(k_i) \ 0]$. Hence,

$$\theta(k) = \theta(k_i + 2n), \quad \text{for all } k \in [k_i + 2n, k_{i+1}).$$

Take s_j as the greatest lower bound on $k_i + 2n$ and s_{j+1} as the lowest upper bound on k_{i+1} in χ_θ .

A.7. Proof of Lemma 5.2.2

Let k, l be in $[s_j + 2n, s_{j+1})$. Then, $h_o(k, k - v) = h_o(l, l - v)$ and $h_o(k - v, k - v) = h_o(l - v, l - v)$ for all $v = 0, \dots, 2n$. In Eq.(3.55), for $1 \leq t < k$ substitute $k - t$ in place of i in $\gamma(k, i)$:

$$\begin{aligned} \gamma(k, k - t) &= h_o^{(1)}(k, k - t) + h_o^{(2)}(k, k - t) h_o^{(1)}(k - t, k - t) \\ &= h_o^{(1)}(l, l - t) + h_o^{(2)}(l, l - t) h_o^{(1)}(l - t, l - t) \\ &= \gamma(l, l - t) \end{aligned}$$

if $t \leq 2n$. If $t > 2n$, then $h_o^{(1)}(k, k - t) = h_o^{(2)}(k, k - t) = 0$ for all k and we again have $\gamma(k, k - t) = \gamma(l, l - t) = 0$. Hence on substituting $t = 1$, we get

$$h(k, k - 1) = \gamma(k, k - 1) = \gamma(l, l - 1) = h(l, l - 1).$$

If $D(k) = D$ for all k , then $h_o^{(1)}(k-t, k-t) = d$ and we may let $k, l \in [s_j, s_{j+1})$ without changing the above conclusion.

For $t > 1$, we proceed by induction. Consider the second term on the right-hand side of Eq.(3.56). Denote it by J . Substitute $k-t$ in place of i in J and change the variable j inside the summand to μ . Then, change μ to $\xi = \mu + t - k$. As a result,

$$\begin{aligned} J &= \sum_{\mu=k-t+1}^{k-1} h_o^{(2)}(k, \mu) h(\mu, k-t), \quad t \geq 2 \\ &= \sum_{\xi=1}^{t-1} h_o^{(2)}(k, k-t+\xi) h(k-t+\xi, k-t) \\ &= \sum_{\xi=1}^{t-1} h_o^{(2)}(l, l-t+\xi) h(k-t+\xi, k-t). \end{aligned}$$

Thus, J is a linear combination of the system Markov parameters $h(k-1, k-t), \dots, h(k-t+1, k-t)$. Put $t = 2$. Then, $h(k-1, k-2) = \gamma(k-1, k-2) = h(l, l-1)$ and $h(k, k-2)$ becomes a linear combination of $\gamma(l, l-2)$ and $h(l, l-1)$. Put $t = 3$. Then, $h(k-1, k-3)$ and $h(k-2, k-3)$ are the only terms in the linear combination. Hence, $h(k, k-3)$ is a linear combination of $\gamma(l, l-3), \gamma(l, l-2), \gamma(l, l-1)$. Now, assume that $h(k, 2)$ is a linear combination of $\gamma(l, 2), \dots, \gamma(l, l-1)$. Put $t = k-1$. Then, J becomes a linear combination of $\gamma(l, 2), \dots, \gamma(l, l-1)$. As a result, $h(k, 1)$ is linear combination of $\gamma(l, l-\mu)$ for $\mu = 1, \dots, k-1$, completing the induction process. Hence, $h(k, k-t) = h(l, l-t)$ for all t satisfying $0 \leq t < \min\{k, l\}$.

For the second part, observe that the recurrence formulas in Eq.(3.58) and Eq.(3.56) are identical if $h_m(k, i)$ and $-h_o^{(2)}(k, i)$ are plugged in places of $h(k, i)$ and $\gamma(k, i)$ in Eq.(3.56). Since $\gamma(k, i)$ is not calculated, the case $D(k) = D$ applies. Hence, for all v in $[0, 2n]$ and $k, l \in [s_j, s_{j+1})$ we have $h_m(k, k-v) = h_m(l, l-v)$.

A.8. Proof of Lemma 5.2.4

From Lemma 5.2.3, $[s_j + 2n, s_{j+1} - 2n) \subset [k_i, k_{i+1})$. Suppose $s_j - k_i > 2n$. Recall from Lemma 5.2.1 that for some s_ℓ and $s_{\ell+1}$ in χ_θ , $[k_i + 2n, k_{i+1}) \subseteq [s_\ell, s_{\ell+1})$. Then, $j > l$ from the chain of inequalities $s_\ell \leq k_i + 2n < s_j$ and we get $\delta_j(\chi_\theta) \leq 2n$ from $s_j \geq s_{\ell+1} \geq k_{i+1} \geq s_{j+1} - 2n$. We reach a contradiction. It follows that $s_j - k_i \leq 2n$. The case $s_j < k_i$ is not possible. If it were possible, $k_i = s_\ell$ for some $\ell > j$ since $k_i \in \chi_\theta$. But, $k_i \leq s_j + 2n$ or $s_\ell - s_j \leq 2n$. Hence, $\delta_j(\chi_\theta) \leq 2n$. Contradiction. Thus, $s_j - k_i \geq 0$. Suppose $s_{j+1} > k_{i+1}$. From Assumption 5.2.1, $k_{i+1} \in \chi_\theta$. Then, $s_t = k_{i+1}$ for some $t \leq j$. Since $s_{j+1} - 2n \leq k_{i+1}$, we then have $\delta_j(\chi_\theta) \leq 2n$. We reach a contradiction. Hence, $s_{j+1} \leq k_{i+1}$. From $[k_i + 2n, k_{i+1}) \subseteq [s_\ell, s_{\ell+1})$, we then have $s_{j+1} \leq k_{i+1} \leq s_{\ell+1}$. Hence $j \leq l$. The case $l > j$ is not possible because $s_\ell \leq k_i + 2n \leq s_j + 2n$ would imply

$\delta_j(\chi_\theta) \leq 2n$. Thus, $j = \ell$ and $s_{j+1} = k_{i+1}$.

A.9. Proof of Lemma 5.2.6

Similarly to Eq.(3.4) define the Hankel matrices

$$\mathcal{H}_o^{(1)}(k) = \begin{bmatrix} h_o^{(1)}(k, k-1) & \cdots & h_o^{(1)}(k, k-2n) \\ & \ddots & \vdots \\ & \cdots & h_o^{(1)}(k+2n-1, k-2n) \end{bmatrix}.$$

Introduce the extended observability/controllability matrices

$$\begin{aligned} \mathcal{O}_{2n}^{(1)}(k) &= [C^T(k) \cdots \Phi_o^T(k+2n, k) C^T(k+2n-1)]^T, \\ \mathcal{R}_{2n}^{(1)}(k-1) &= [B_u(k-1) \cdots \Phi_o(k, k-2n+1) B_u(k-2n)] \end{aligned}$$

for $2n < k \leq N_1$ and factorize $\mathcal{H}_o^{(1)}(k)$ as

$$\mathcal{H}_o^{(1)}(k) = \mathcal{O}_{2n}^{(1)}(k) \mathcal{R}_{2n}^{(1)}(k-1).$$

From Lemma 5.2.5, $\text{rank}(\mathcal{H}_o^{(1)}(k)) = n$ for all k . Write

$$\begin{aligned} \theta^{(1)}(k_i) &= [D(k_i) \ C(k_i) \mathcal{R}_{2n}^{(1)}(k_i-1)], \\ \theta^{(1)}(k_i-1) &= [D(k_{i-1}) \ C(k_{i-1}) \mathcal{R}_{2n}^{(1)}(k_i-2)]. \end{aligned}$$

Since $\delta_{i-1}(\chi) > 2n+1$, $\mathcal{R}_{2n}^{(1)}(k_i-1) = \mathcal{R}_{2n}^{(1)}(k_i-2)$ and from Eq.(5.2.5), $\mathcal{R}_{2n}^{(1)}(k_i-1)$ has full rank. Suppose $k_i \in \chi$. Then, $\varphi(k_i-1) \neq \varphi(k_i)$ and $[C(k_i) \ D(k_i)] \neq [C(k_i-1) \ D(k_i-1)]$ from Assumption 5.2.3. Therefore, $\theta^{(1)}(k_i-1) \neq \theta^{(1)}(k_i)$ if $d(k_i) \neq d(k_i-1)$. If $c(k_i) \neq c^T(k_i-1)$, the same conclusion is drawn since $\mathcal{R}_{2n}^{(1)}(k_i-1)$ and $\mathcal{R}_{2n}^{(1)}(k_i-2)$ are equal and have full rank. It follows that $\theta(k_i-1) \neq \theta(k_i)$ and $k_i \in \chi_\theta$. Suppose ii. holds in Assumption 5.2.3, construct $\mathcal{H}_o^{(2)}(k)$ similarly to $\mathcal{H}_o^{(1)}(k)$ from the Markov parameters $h_o^{(2)}(k, i)$. Factorization of $\mathcal{H}_o^{(2)}(k)$ shows that $\theta^{(2)}(k_i-1) \neq \theta^{(2)}(k_i)$ if $C(k_i) \neq C(k_{i-1})$. Then, $\theta(k_i-1) \neq \theta(k_i)$. and $k_i \in \chi_\theta$.

Appendix B.

THE DBSCAN CLUSTERING

The Density-based spatial clustering of applications with noise (DBSCAN) [114] is a well established unsupervised data clustering algorithm. It groups together into clusters points that are lying close in high-density regions, leaving the points that lie alone in low-density regions as outliers. Unlike k -means clustering, The DBSCAN does not need the number of clusters to be specified beforehand.

The DBSCAN algorithm takes epsilon and minpts as inputs. DBSCAN starts by jumping onto a random point; this point is considered as a core point candidate. We evaluate with the epsilon if there are minpts-1 Points within the epsilon radius of the candidate core-points. If we got exact minpts point inside this epsilon, a cluster can be formed. Then an **expandcluster** function is called, its job is to expand the cluster, all the points within the radius of the epsilon will do their search again with the same epsilon radius, so the cluster will keep expanding. This step is repeated further and stops until the cluster cannot be expanded anymore. We then jump to another point which is not clustered yet, if the point has less than minpts points within its radius, it will be flagged as an outlier. This process is repeated until all points are clustered into their convenient clusters or flagged as noise. For a point to be assigned to a cluster, it must satisfy the condition that its epsilon neighborhood contains at least a minimum number of neighbors (minpts), or, the point can lie within the epsilon neighborhood of another point that satisfies the epsilon and minpts condition. The DBSCAN algorithm identifies three kinds of points:

1. Core point: a point in a cluster that has at least minpts neighbors in its epsilon neighborhood.
2. Border point: a point in a cluster that has fewer than minpts neighbors in its epsilon neighborhood.
3. Noise point: an outlier that does not belong to any cluster.

Figure B.1 gives a schematic explanation of the different points considered by the algorithm. The DBSCAN algorithm is available as built-in function in MATLAB through the command:

```
[idx,corepts]=dbscan(X,epsilon,minpts)
```

where X is a matrix of observations, observations being the different rows. The output idx is the vector of labels assigning each observation to its specific cluster, $corepts$ is a vector labelling the observations as core-points by assigning one or not by assigning zero.

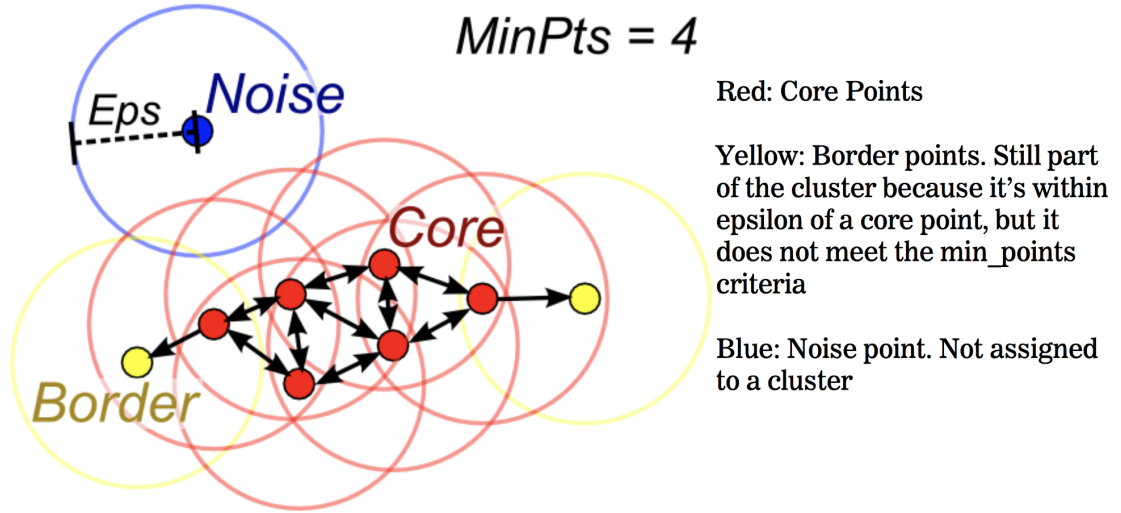


Figure B.1: The different types of points considered by `dbscan` algorithm

The DBSCAN clustering algorithm steps are as follows:

Algorithm 10 The DBSCAN Algorithm

Input: N objects to be clustered and global parameters ϵ , $\minpts$.

- ① Arbitrary select a point P .
- ② Retrieve all points density-reachable from P wrt ϵ and $\minpts$.
- ③ If P is a core point, a cluster is formed.
- ④ If P is a border point, no points are density-reachable from P and DBSCAN visits the next point of the data base.
- ⑤ Continue the process until all points have been processed.

Output: Clusters of objects.

REFERENCES

- [1] R. V. Lopes, G. A. Borges, and J. Y. Ishihara, “New algorithm for identification of discrete-time switched linear systems,” in *2013 American Control Conference*, pp. 6219–6224, IEEE, 2013.
- [2] F. Lauer and G. Bloch, “Hybrid system identification,” in *Hybrid System Identification*, pp. 77–101, Springer, 2019.
- [3] R. Rui, T. Ardeshiri, and A. Bazanella, “Identification of piecewise affine state-space models via expectation maximization,” in *2016 IEEE Conference on Computer Aided Control System Design (CACSD)*, pp. 1066–1071, IEEE, 2016.
- [4] R. Tóth, *Modeling and identification of linear parameter-varying systems*, vol. 403. Springer, 2010.
- [5] L. Bako, G. Mercère, and S. Lecoeuche, “On-line structured identification of switching systems with possibly varying orders,” in *2007 European Control Conference (ECC)*, pp. 4065–4072, IEEE, 2007.
- [6] M. G. Sefidmazgi, M. M. Kordmahalleh, A. Homaifar, A. Karimoddini, and E. Tunstel, “A bounded switching approach for identification of switched mimo systems,” in *2016 IEEE International Conference on Systems, Man, and Cybernetics (SMC)*, pp. 004743–004748, IEEE, 2016.
- [7] S. Paoletti, A. L. Juloski, G. Ferrari-Trecate, and R. Vidal, “Identification of hybrid systems a tutorial,” *European journal of control*, vol. 13, no. 2-3, pp. 242–260, 2007.
- [8] A. J. Van Der Schaft and J. M. Schumacher, *An introduction to hybrid dynamical systems*, vol. 251. Springer London, 2000.
- [9] M. S. Branicky, “Introduction to hybrid systems,” in *Handbook of networked and embedded control systems*, pp. 91–116, Springer, 2005.
- [10] J. Lunze and F. Lamnabhi-Lagarrigue, *Handbook of hybrid systems control: theory, tools, applications*. Cambridge University Press, 2009.
- [11] B. Lennartson, M. Tittus, B. Egardt, and S. Pettersson, “Hybrid systems in process control,” *IEEE control systems magazine*, vol. 16, no. 5, pp. 45–56, 1996.
- [12] R. Carloni, R. G. Sanfelice, A. R. Teel, and C. Melchiorri, “A hybrid control strategy for robust contact detection and force regulation,” in *2007 American Control Conference*, pp. 1461–1466, IEEE, 2007.

- [13] T. Schlegl, M. Buss, and G. Schmidt, "A hybrid systems approach toward modeling and dynamical simulation of dextrous manipulation," *IEEE/ASME transactions on mechatronics*, vol. 8, no. 3, pp. 352–361, 2003.
- [14] X. Jin and B. Huang, "Robust identification of piecewise/switching autoregressive exogenous process," *AIChE journal*, vol. 56, no. 7, pp. 1829–1844, 2010.
- [15] W. Glover and J. Lygeros, "A stochastic hybrid model for air traffic control simulation," in *International Workshop on Hybrid Systems: Computation and Control*, pp. 372–386, Springer, 2004.
- [16] M. Prandini and J. Hu, "Application of reachability analysis for stochastic hybrid systems to aircraft conflict prediction," in *2008 47th IEEE Conference on Decision and Control*, pp. 4036–4041, IEEE, 2008.
- [17] J. Lee, S. Bohacek, J. P. Hespanha, and K. Obraczka, "Modeling communication networks with hybrid systems," *IEEE/ACM Transactions on Networking*, vol. 15, no. 3, pp. 630–643, 2007.
- [18] A. Germani, C. Manes, and P. Palumbo, "Simultaneous system identification and channel estimation: a hybrid system approach," in *2007 46th IEEE Conference on Decision and Control*, pp. 1764–1769, IEEE, 2007.
- [19] I. A. Hiskens and M. Pai, "Hybrid systems view of power system modelling," in *2000 IEEE International Symposium on Circuits and Systems. Emerging Technologies for the 21st Century. Proceedings (IEEE Cat No. 00CH36353)*, vol. 2, pp. 228–231, IEEE, 2000.
- [20] K. M. Pekpe, G. Mourot, K. Gasso, and J. Ragot, "Identification of switching systems using change detection technique in the subspace framework," in *2004 43rd IEEE Conference on Decision and Control (CDC)(IEEE Cat. No. 04CH37601)*, vol. 4, pp. 3720–3725, IEEE, 2004.
- [21] R. Murray-Smith, "Modelling human control behaviour with context-dependent markov-switching multiple models," *IFAC Proceedings Volumes*, vol. 31, no. 26, pp. 461–466, 1998.
- [22] R. Vidal and Y. Ma, "A unified algebraic approach to 2-d and 3-d motion segmentation and estimation," *Journal of Mathematical Imaging and Vision*, vol. 25, no. 3, pp. 403–421, 2006.
- [23] E. Cinquemani, A. Miliadis-Argeitis, and J. Lygeros, "Identification of genetic regulatory networks: A stochastic hybrid approach," *IFAC Proceedings Volumes*, vol. 41, no. 2, pp. 301–306, 2008.
- [24] L. Bako and R. Vidal, "Algebraic identification of mimo sarx models," in *International Workshop on Hybrid Systems: Computation and Control*, pp. 43–57, Springer, 2008.

- [25] A. Bemporad, A. Garulli, S. Paoletti, and A. Vicino, “A bounded-error approach to piecewise affine system identification,” *IEEE Transactions on Automatic Control*, vol. 50, no. 10, pp. 1567–1580, 2005.
- [26] A. Bemporad, J. Roll, and L. Ljung, “Identification of hybrid systems via mixed-integer programming,” in *Proceedings of the 40th IEEE Conference on Decision and Control (Cat. No. 01CH37228)*, vol. 1, pp. 786–792, IEEE, 2001.
- [27] G. Ferrari-Trecate, M. Muselli, D. Liberati, and M. Morari, “A clustering technique for the identification of piecewise affine systems,” *Automatica*, vol. 39, no. 2, pp. 205–217, 2003.
- [28] A. L. Juloski, S. Weiland, and W. Heemels, “A bayesian approach to identification of hybrid systems,” *IEEE Transactions on Automatic Control*, vol. 50, no. 10, pp. 1520–1533, 2005.
- [29] Z. Lassoued and K. Abderrahim, “An experimental validation of a novel clustering approach to pwarx identification,” *Engineering Applications of Artificial Intelligence*, vol. 28, pp. 201–209, 2014.
- [30] N. Ozay, C. Lagoa, and M. Sznaiier, “Set membership identification of switched linear systems with known number of subsystems,” *Automatica*, vol. 51, pp. 180–191, 2015.
- [31] R. Vidal, “Recursive identification of switched arx systems,” *Automatica*, vol. 44, no. 9, pp. 2274–2287, 2008.
- [32] R. Vidal, S. Soatto, Y. Ma, and S. Sastry, “An algebraic geometric approach to the identification of a class of linear hybrid systems,” in *42nd IEEE International Conference on Decision and Control (IEEE Cat. No. 03CH37475)*, vol. 1, pp. 167–172, IEEE, 2003.
- [33] Y. Ma and R. Vidal, “Identification of deterministic switched arx systems via identification of algebraic varieties,” in *International Workshop on Hybrid Systems: Computation and Control*, pp. 449–465, Springer, 2005.
- [34] H. Ohlsson and L. Ljung, “Identification of switched linear regression models using sum-of-norms regularization,” *Automatica*, vol. 49, no. 4, pp. 1045–1050, 2013.
- [35] A. Hartmann, J. M. Lemos, R. S. Costa, J. Xavier, and S. Vinga, “Identification of switched arx models via convex optimization and expectation maximization,” *Journal of Process Control*, vol. 28, pp. 9–16, 2015.
- [36] L. Bako, F. Lauer, G. Bloch, *et al.*, “Identification of mimo switched state-space models,” in *2013 American Control Conference*, pp. 71–76, IEEE, 2013.
- [37] R. V. Lopes, J. Y. Ishihara, and G. A. Borges, “Identification of state-space switched linear systems using clustering and hybrid filtering,” *Journal of the Brazilian Society of Mechanical Sciences and Engineering*, vol. 39, no. 2, pp. 565–573, 2017.

- [38] L. Bako, G. Mercère, R. Vidal, and S. Lecoeuche, "Identification of switched linear state space models without minimum dwell time," *IFAC Proceedings Volumes*, vol. 42, no. 10, pp. 569–574, 2009.
- [39] V. Verdult and M. Verhaegen, "Subspace identification of piecewise linear systems," in *2004 43rd IEEE Conference on Decision and Control (CDC)(IEEE Cat. No. 04CH37601)*, vol. 4, pp. 3838–3843, IEEE, 2004.
- [40] J. Borges, V. Verdult, M. Verhaegen, and M. A. Botto, "A switching detection method based on projected subspace classification," in *Proceedings of the 44th IEEE Conference on Decision and Control*, pp. 344–349, IEEE, 2005.
- [41] J. Borges, V. Verdult, and M. Verhaegen, "Iterative subspace identification of piecewise linear systems," *IFAC Proceedings Volumes*, vol. 39, no. 1, pp. 368–373, 2006.
- [42] K. Huang, A. Wagner, and Y. Ma, "Identification of hybrid linear time-invariant systems via subspace embedding and segmentation (ses)," in *2004 43rd IEEE Conference on Decision and Control (CDC)(IEEE Cat. No. 04CH37601)*, vol. 3, pp. 3227–3234, IEEE, 2004.
- [43] L. Blackmore, S. Gil, S. Chung, and B. Williams, "Model learning for switching linear systems with autonomous mode transitions," in *2007 46th IEEE Conference on Decision and Control*, pp. 4648–4655, IEEE, 2007.
- [44] S. Gil and B. Williams, "Beyond local optimality: An improved approach to hybrid model learning," in *Proceedings of the 48th IEEE Conference on Decision and Control (CDC) held jointly with 2009 28th Chinese Control Conference*, pp. 3938–3945, IEEE, 2009.
- [45] K. M. Pekpe and S. Lecoeuche, "Online classification of switching models based on subspace framework," in *Analysis and Design of Hybrid Systems 2006*, pp. 154–159, Elsevier, 2006.
- [46] K. M. Pekpe and S. Lecoeuche, "Online clustering of switching models based on a subspace framework," *Nonlinear Analysis: Hybrid Systems*, vol. 2, no. 3, pp. 735–749, 2008.
- [47] L. Bako, G. Mercère, and S. Lecoeuche, "On-line structured subspace identification with application to switched linear systems," *International Journal of Control*, vol. 82, no. 8, pp. 1496–1515, 2009.
- [48] D. Chen, L. Bako, and S. Lecoeuche, "A recursive sparse learning method: Application to jump markov linear systems," *IFAC Proceedings Volumes*, vol. 44, no. 1, pp. 3198–3203, 2011.
- [49] S. Weiland, A. L. Juloski, and B. Vet, "On the equivalence of switched affine models and switched arx models," in *Proceedings of the 45th IEEE Conference on Decision and Control*, pp. 2614–2618, IEEE, 2006.

- [50] G. Mercère and L. Bako, “Parameterization and identification of multivariable state-space systems: A canonical approach,” *Automatica*, vol. 47, no. 8, pp. 1547–1555, 2011.
- [51] F. Bencherki, S. Türkay, and H. Akçay, “State-space identification of switched linear systems by sparse optimization,” *submitted to Automatica with reference number 20-0995*, (2020).
- [52] A. Garulli, S. Paoletti, and A. Vicino, “A survey on switched and piecewise affine system identification,” *IFAC Proceedings Volumes*, vol. 45, no. 16, pp. 344–355, 2012.
- [53] J.-N. Juang and R. S. Pappa, “An eigensystem realization algorithm for modal parameter identification and model reduction,” *Journal of guidance, control, and dynamics*, vol. 8, no. 5, pp. 620–627, 1985.
- [54] T. Kailath, *Linear systems*, vol. 156. Prentice-Hall Englewood Cliffs, NJ, 1980.
- [55] M. Majji, J.-N. Juang, and J. L. Junkins, “Observer/kalman-filter time-varying system identification,” *Journal of guidance, control, and dynamics*, vol. 33, no. 3, pp. 887–900, 2010.
- [56] J.-N. Juang, *Applied system identification*. Prentice-Hall, Inc., 1994.
- [57] H. Celik, “Identification and prediction of discrete-time bilinear state-space models: Interaction matrices and superstates,” 2010.
- [58] M. Phan and R. Longman, “Relationship between state-space and input-output models via observer markov parameters,” *WIT Transactions on The Built Environment*, vol. 22, 1970.
- [59] M. Phan, L. G. Horta, J.-N. Juang, and R. W. Longman, “Linear system identification via an asymptotically stable observer,” *Journal of Optimization Theory and Applications*, vol. 79, no. 1, pp. 59–86, 1993.
- [60] J.-N. Juang, M. Phan, L. G. Horta, and R. W. Longman, “Identification of observer/kalman filter markov parameters-theory and experiments,” *Journal of Guidance, Control, and Dynamics*, vol. 16, no. 2, pp. 320–329, 1993.
- [61] M. Q. Phan, F. Vicario, R. W. Longman, and R. Betti, “An all-interaction matrix approach to linear and bilinear system identification,” in *Modeling, Simulation and Optimization of Complex Processes HPSC 2015*, pp. 165–177, Springer, 2017.
- [62] M. Verhaegen and V. Verdult, *Filtering and system identification: a least squares approach*. Cambridge university press, 2007.
- [63] M. Phan, L. G. Horta, J.-N. Juang, and R. W. Longman, “Improvement of observer/kalman filter identification (okid) by residual whitening,” 1995.

- [64] F. Vicario, M. Q. Phan, R. W. Longman, and R. Betti, “Generalized framework of okid for linear state-space model identification,” in *Modeling, Simulation and Optimization of Complex Processes HPSC 2015*, pp. 249–260, Springer, 2017.
- [65] F. Vicario, M. Q. Phan, R. Betti, and R. W. Longman, “Okid via output residuals: a converter from stochastic to deterministic system identification,” *Journal of Guidance, Control, and Dynamics*, vol. 40, no. 12, pp. 3226–3238, 2017.
- [66] R. Doraiswami and L. Cheded, “Robust kalman filter-based least squares identification of a multivariable system,” *IET Control Theory & Applications*, vol. 12, no. 8, pp. 1064–1074, 2018.
- [67] F. Vicario, M. Q. Phan, R. Betti, and R. W. Longman, “Output-only observer/kalman filter identification (o3kid),” *Structural Control and Health Monitoring*, vol. 22, no. 5, pp. 847–872, 2015.
- [68] A. Alenany, D. Westwick, and H. Shang, “A modified observer/kalman filter identification (okid) algorithm employing output residuals,” *International Journal of Dynamics and Control*, vol. 7, no. 2, pp. 636–643, 2019.
- [69] J.-N. Juang, J. Cooper, and J. Wright, “An eigensystem realization algorithm using data correlations (era/dc) for modal parameter identification,” 1987.
- [70] B. M. Sanandaji, *Compressive system identification (CSI): Theory and applications of exploiting sparsity in the analysis of high-dimensional dynamical systems*. Colorado School of Mines, 2013.
- [71] D. L. Donoho *et al.*, “Compressed sensing,” *IEEE Transactions on information theory*, vol. 52, no. 4, pp. 1289–1306, 2006.
- [72] Y. C. Eldar and G. Kutyniok, *Compressed sensing: theory and applications*. Cambridge university press, 2012.
- [73] R. Tibshirani, “Regression shrinkage and selection via the lasso,” *Journal of the Royal Statistical Society: Series B (Methodological)*, vol. 58, no. 1, pp. 267–288, 1996.
- [74] S. Chen, D. Donoho, and M. Saunders, “Atomic decomposition by basis pursuit: Sia m journal on scientific computing, 20, 33–61,” 1998.
- [75] Y. C. Eldar and M. Mishali, “Block sparsity and sampling over a union of subspaces,” in *2009 16th International Conference on Digital Signal Processing*, pp. 1–8, IEEE, 2009.
- [76] Y. C. Eldar, P. Kuppinger, and H. Bolcskei, “Block-sparse signals: Uncertainty relations and efficient recovery,” *IEEE Transactions on Signal Processing*, vol. 58, no. 6, pp. 3042–3054, 2010.

- [77] H. Boche, R. Calderbank, G. Kutyniok, and J. Vybíral, “A survey of compressed sensing,” in *Compressed sensing and its applications*, pp. 1–39, Springer, 2015.
- [78] S. G. Mallat and Z. Zhang, “Matching pursuits with time-frequency dictionaries,” *IEEE Transactions on signal processing*, vol. 41, no. 12, pp. 3397–3415, 1993.
- [79] J. A. Tropp, “Greed is good: Algorithmic results for sparse approximation,” *IEEE Transactions on Information theory*, vol. 50, no. 10, pp. 2231–2242, 2004.
- [80] J. A. Tropp, A. C. Gilbert, J. S. Martin, J. Tropp, A. Gilbert, and M. Strauss, “Algorithms for simultaneous sparse approximation,” in *EURASIP J. App. Signal Processing*, Citeseer, 2006.
- [81] D. L. Donoho, Y. Tsaig, I. Drori, and J.-L. Starck, “Sparse solution of underdetermined systems of linear equations by stagewise orthogonal matching pursuit,” *IEEE transactions on Information Theory*, vol. 58, no. 2, pp. 1094–1121, 2012.
- [82] D. Needell and J. A. Tropp, “Cosamp: Iterative signal recovery from incomplete and inaccurate samples,” *Applied and computational harmonic analysis*, vol. 26, no. 3, pp. 301–321, 2009.
- [83] T. Blumensath and M. E. Davies, “Iterative hard thresholding for compressed sensing,” *Applied and computational harmonic analysis*, vol. 27, no. 3, pp. 265–274, 2009.
- [84] E. J. Candès, J. Romberg, and T. Tao, “Robust uncertainty principles: Exact signal reconstruction from highly incomplete frequency information,” *IEEE Transactions on information theory*, vol. 52, no. 2, pp. 489–509, 2006.
- [85] P. J. Bickel, Y. Ritov, A. B. Tsybakov, *et al.*, “Simultaneous analysis of lasso and dantzig selector,” *The Annals of Statistics*, vol. 37, no. 4, pp. 1705–1732, 2009.
- [86] M. Grant and S. Boyd, “Cvx: Matlab software for disciplined convex programming, version 2.1 (2014),” URL: <http://cvxr.com/cvx>, 2016.
- [87] P. Haessig, “Yalmip: Optimization made easy!,” 2017.
- [88] S. R. Becker, E. J. Candès, and M. C. Grant, “Templates for convex cone problems with applications to sparse signal recovery,” *Mathematical programming computation*, vol. 3, no. 3, p. 165, 2011.
- [89] E. J. Candes, M. B. Wakin, and S. P. Boyd, “Enhancing sparsity by reweighted ℓ_1 minimization,” *Journal of Fourier analysis and applications*, vol. 14, no. 5-6, pp. 877–905, 2008.

- [90] X. Chen and W. Zhou, “Convergence of reweighted ℓ_1 minimization algorithms and unique solution of truncated lp minimization,” *Department of Applied Mathematics, The Hong Kong Polytechnic University*, 2010.
- [91] I. Daubechies, R. DeVore, M. Fornasier, and C. S. Güntürk, “Iteratively reweighted least squares minimization for sparse recovery,” *Communications on Pure and Applied Mathematics: A Journal Issued by the Courant Institute of Mathematical Sciences*, vol. 63, no. 1, pp. 1–38, 2010.
- [92] M.-J. Lai and J. Wang, “An unconstrained ℓ_q minimization with $0 < q \leq 1$ for sparse solution of underdetermined linear systems,” *SIAM Journal on Optimization*, vol. 21, no. 1, pp. 82–101, 2011.
- [93] H. Ohlsson, F. Gustafsson, L. Ljung, and S. Boyd, “State smoothing by sum-of-norms regularization,” in *49th IEEE Conference on Decision and Control (CDC)*, pp. 2880–2885, IEEE, 2010.
- [94] H. Ohlsson, L. Ljung, and S. Boyd, “Segmentation of arx-models using sum-of-norms regularization,” *Automatica*, vol. 46, no. 6, pp. 1107–1111, 2010.
- [95] Y.-B. Zhao and D. Li, “Rewighted ℓ_1 minimization for sparse solutions to underdetermined linear systems,” *SIAM Journal on Optimization*, vol. 22, no. 3, pp. 1065–1088, 2012.
- [96] Z. Xie and J. Hu, “Rewighted ℓ_1 minimization for sparse solutions to underdetermined linear systems,” in *2013 6th International Congress on Image and Signal Processing (CISP)*, vol. 3, pp. 1660–1664, IEEE, 2013.
- [97] I. F. Gorodnitsky and B. D. Rao, “Sparse signal reconstruction from limited data using focuss: A re-weighted minimum norm algorithm,” *IEEE Transactions on signal processing*, vol. 45, no. 3, pp. 600–616, 1997.
- [98] M. Yuan and Y. Lin, “Model selection and estimation in regression with grouped variables,” *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, vol. 68, no. 1, pp. 49–67, 2006.
- [99] M. A. Davenport, M. F. Duarte, Y. C. Eldar, and G. Kutyniok, “Introduction to compressed sensing,” *preprint*, vol. 93, no. 1, p. 2, 2011.
- [100] Y. C. Eldar, *Sampling theory: Beyond bandlimited systems*. Cambridge University Press, 2015.
- [101] D. L. Donoho and M. Elad, “Optimally sparse representation in general (nonorthogonal) dictionaries via ℓ_1 minimization,” *Proceedings of the National Academy of Sciences*, vol. 100, no. 5, pp. 2197–2202, 2003.
- [102] X. Feng and Z. Zhang, “The rank of a random matrix,” *Applied mathematics and computation*, vol. 185, no. 1, pp. 689–694, 2007.
- [103] A. Cohen, W. Dahmen, and R. DeVore, “Compressed sensing and best k -term approximation,” *Journal of the American mathematical society*, vol. 22, no. 1, pp. 211–231, 2009.

- [104] E. J. Candes and T. Tao, “Decoding via linear programming,” *IEEE Trans. Inform. Theory*.
- [105] T. Strohmer and R. Heath, “Grassmannian frames with applications to coding and communication,” *arXiv preprint math/0301135*, 2003.
- [106] Y. C. Eldar and M. Mishali, “Robust recovery of signals from a structured union of subspaces,” *IEEE Transactions on Information Theory*, vol. 55, no. 11, pp. 5302–5316, 2009.
- [107] D. L. Donoho and X. Huo, “Uncertainty principles and ideal atomic decomposition,” *IEEE transactions on information theory*, vol. 47, no. 7, pp. 2845–2862, 2001.
- [108] A. Birouche, J. Guillet, B. Mourllion, and M. Basset, “Gramian based approach to model order-reduction for discrete-time switched linear systems,” in *18th Mediterranean Conference on Control and Automation, MED’10*, pp. 1224–1229, IEEE, 2010.
- [109] S. Shokoohi and L. M. Silverman, “Identification and model reduction of time-varying discrete-time systems,” *Automatica*, vol. 23, no. 4, pp. 509–521, 1987.
- [110] M. Majji, J.-N. Juang, and J. L. Junkins, “Time-varying eigensystem realization algorithm,” *Journal of guidance, control, and dynamics*, vol. 33, no. 1, pp. 13–28, 2010.
- [111] M. Verhaegen and X. Yu, “A class of subspace model identification algorithms to identify periodically and arbitrarily time-varying systems,” *Automatica*, vol. 31, no. 2, pp. 201–216, 1995.
- [112] K. Liu, “Identification of linear time-varying systems,” *Journal of Sound and Vibration*, vol. 206, no. 4, pp. 487–505, 1997.
- [113] L. Ljung, “System identification—theory for the user, 2nd edn (upper saddle river, nj: Ptr prentice hall),” *Feasible Mind Uploading*, vol. 101, 1999.
- [114] M. Ester, H.-P. Kriegel, J. Sander, X. Xu, *et al.*, “A density-based algorithm for discovering clusters in large spatial databases with noise,” in *Kdd*, vol. 96, pp. 226–231, 1996.
- [115] D. Arthur and S. Vassilvitskii, “k-means++: the advantages of careful seeding, p 1027–1035,” in *SODA’07: proceedings of the eighteenth annual ACM-SIAM symposium on discrete algorithms. Society for Industrial and Applied Mathematics, Philadelphia, PA*, 2007.
- [116] T. Hastie, R. Tibshirani, and J. Friedman, *The elements of statistical learning: data mining, inference, and prediction*. Springer Science & Business Media, 2009.
- [117] M. Petreczky, L. Bako, and J. H. van Schuppen, “Identifiability of discrete-time linear switched systems,” in *Proceedings of the 13th ACM international conference on Hybrid systems: computation and control*, pp. 141–150, 2010.

- [118] R. Vidal, A. Chiuso, and S. Soatto, “Observability and identifiability of jump linear systems,” in *Proceedings of the 41st IEEE Conference on Decision and Control, 2002.*, vol. 4, pp. 3614–3619, IEEE, 2002.
- [119] M. Petreczky, R. Tóth, and G. Mercere, “Realization theory for lpv state-space representations with affine dependence,” *IEEE Transactions on Automatic Control*, vol. 62, no. 9, pp. 4667–4674, 2016.
- [120] N. Ozay, M. Sznaier, C. M. Lagoa, and O. I. Camps, “A sparsification approach to set membership identification of switched affine systems,” *IEEE Transactions on Automatic Control*, vol. 57, no. 3, pp. 634–648, 2011.
- [121] M. VERHAEGEN, “Subspace model identification part 1. the outoput-error state-space model identification class of algorithms; part2. analysis of the elementary output-error state-space model identification algorithms,” *Int. J. Control*, vol. 56, no. 5, pp. 1211–1241, 1992.
- [122] L. I. Rudin, S. Osher, and E. Fatemi, “Nonlinear total variation based noise removal algorithms,” *Physica D: nonlinear phenomena*, vol. 60, no. 1-4, pp. 259–268, 1992.
- [123] S.-J. Kim, K. Koh, S. Boyd, and D. Gorinevsky, “ ℓ_1 trend filtering,” *SIAM review*, vol. 51, no. 2, pp. 339–360, 2009.
- [124] R. J. Tibshirani *et al.*, “Adaptive piecewise polynomial estimation via trend filtering,” *The Annals of Statistics*, vol. 42, no. 1, pp. 285–323, 2014.
- [125] J. A. Tropp and A. C. Gilbert, “Signal recovery from random measurements via orthogonal matching pursuit,” *IEEE Transactions on information theory*, vol. 53, no. 12, pp. 4655–4666, 2007.
- [126] J. A. Tropp, “Just relax: Convex programming methods for identifying sparse signals in noise,” *IEEE transactions on information theory*, vol. 52, no. 3, pp. 1030–1051, 2006.
- [127] R. Gribonval and M. Nielsen, “Sparse representations in unions of bases,” *IEEE transactions on Information theory*, vol. 49, no. 12, pp. 3320–3325, 2003.
- [128] A. M. Bruckstein, D. L. Donoho, and M. Elad, “From sparse solutions of systems of equations to sparse modeling of signals and images,” *SIAM review*, vol. 51, no. 1, pp. 34–81, 2009.
- [129] S. Foucart and H. Rauhut, “A mathematical introduction to compressive sensing,” in *Applied and Numerical Harmonic Analysis*, 2013.
- [130] B. M. Sanandaji, T. L. Vincent, K. Poolla, and M. B. Wakin, “A tutorial on recovery conditions for compressive system identification of sparse channels,” in *2012 IEEE 51st IEEE Conference on Decision and Control (CDC)*, pp. 6277–6283, IEEE, 2012.

- [131] S. Boyd, N. Parikh, E. Chu, B. Peleato, J. Eckstein, *et al.*, “Distributed optimization and statistical learning via the alternating direction method of multipliers,” *Foundations and Trends® in Machine learning*, vol. 3, no. 1, pp. 1–122, 2011.
- [132] A. Ramdas and R. J. Tibshirani, “Fast and flexible admm algorithms for trend filtering,” *Journal of Computational and Graphical Statistics*, vol. 25, no. 3, pp. 839–858, 2016.
- [133] L. Bako, “Identification of switched linear systems via sparse optimization,” *Automatica*, vol. 47, no. 4, pp. 668–677, 2011.
- [134] K. Mohan and M. Fazel, “Iterative reweighted algorithms for matrix rank minimization,” *The Journal of Machine Learning Research*, vol. 13, no. 1, pp. 3441–3473, 2012.

