UNIVERSITY OF CALIFORNIA SAN DIEGO

Low-Rank Approximations for Estimation of Multivariable Dynamics with Application to Energy Systems

A dissertation submitted in partial satisfaction of the
requirements for the degree
Doctor of Philosophy

in

Engineering Sciences (Mechanical Engineering)

by

Yangsheng Hu

Committee in charge:

Professor Raymond A. de Callafon, Chair
Professor Robert R. Bitmead
Professor Philip E. Gill
Professor Sonia Martínez Diaz
Professor William M. McEneaney
Professor Jason R. Schweinsberg

2021

The dissertation of Yangsheng Hu is approved, and it is acceptable in quality and form for publication on microfilm and electronically.

University of California San Diego

2021

DEDICATION

To my family.

EPIGRAPH

There is more wisdom in your body

than in your deepest philosophy.

—Friedrich Nietzsche

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

ACKNOWLEDGEMENTS

I have received a lot of help and encouragement from different people during studying at UCSD. First, I would like to extend the maximum amount of appreciation to my advisor, Raymond A. de Callafon. Raymond is insightful and diligent professionally. He is also very kind, humourous, and understanding in our daily life. I am sincerely grateful that he offered me a precious opportunity to pursue my Ph.D. in a field I am excited about and gave me invaluable guidance along the way. I would also like to express my sincere appreciation to my dissertation committee of Robert R. Bitmead, Philip E. Gill, Sonia Martínez, William M. McEneaney, and Jason R. Schweinsberg for their constructive support.

I was attending several terrific classes offered by Raymond A. de Callafon, Sonia Martínez, Miroslav Krstic, Robert R. Bitmead, William M. McEneaney, Jason R. Schweinsberg, Maurício de Oliveira, Philip E. Gill, and Steven Swanson. My grateful thanks go to them for sharing their knowledge and experience from the perspectives of both theory and application. It is a privilege to benefit from them, who are both outstanding researchers and teachers.

I would like to thank Maksim Subbotin and Viktor Rill at Robert Bosch LLC for offering me the internship opportunity. It was a wonderful and memorable experience in the Bay Area and I really enjoyed staying in a place full of innovation and creativity.

I would like to thank my labmates, Alicia Dautt-Silva, Amir Valibeygi, Li Tan, Sai Akhil Reddy Konakalla, Xin Zhao, and Yunfeng Jiang, whose friendliness, talent, and insight impressed and inspired me a lot. It was also a great pleasure to meet a lot of great friends at UCSD, Dan Li, Heng Qiao, Jiajie Shi, Mingsong Jiang, Muhan Zhao, Naiqing Zhang,

Pengcheng Cao, Xiaobo Bi, Zhichao Li, Zhouyi Liao and many other dear friends for making my Ph.D. journey enjoyable. My best wishes to all of them.

My deepest thanks go to my parents for their love, patience, trust, and support throughout all these years. I dedicate the dissertation to them.

Chapter 2, in full, is a reprint of the material as it appears in *Covariance Based Realization Algorithm for identification of linear time periodic system*, Y. Hu and R.A. de Callafon, 2018 Annual American Control Conference (ACC), Wisconsin Center, Milwaukee, USA, Jun. 27–29, 2018, pp. 1102–107. The dissertation author was the primary investigator and author of this paper.

Chapter 3, in full, is a reprint of the material as it appears in *Variance reduction in Covariance Based Realization Algorithm with application to closed-loop data*, Y. Hu, Y. Jiang and R.A. de Callafon, Automatica, vol. 113, pp. 108683, 2020. A preliminary version of the work appeared as *Optimal weighting for covariance based realization algorithm*, Y. Hu and R.A. de Callafon, 56th Annual Conference on Decision and Control (CDC), Melbourne, VIC, Australia, Dec. 12–15, 2017, pp. 5274–5279. The dissertation author was the primary investigator and author of the papers.

Chapter 4, in full, has been submitted for publication in IEEE Transactions on Automatic Control. It may appear as *Noniterative tensor network-based estimation for MIMO Volterra system identification*, Y. Hu, L. Tan and R.A. de Callafon. The dissertation author was the primary investigator and author of the paper.

Chapter 5, in part, is a reprint of the material as it appears in *Persistent excitation condition for MIMO Volterra system identification with Gaussian distributed input signals*, Y. Hu, L. Tan and R.A. de Callafon, 58th Annual Conference on Decision and Control

(CDC), Nice, France, Dec. 11–13, 2019, pp. 1752–1757. The corresponding simulation example is from part of *Noniterative tensor network-based estimation for MIMO Volterra system identification*, Y. Hu, L. Tan and R.A. de Callafon, which has been submitted for publication in IEEE Transactions on Automatic Control. The dissertation author was the primary investigator and author of the papers.

Chapter 6, in part, has been submitted for publication in IEEE Transactions on Power Systems. It may appear as *Microgrid dynamic modeling with power flow covariance data*, Y. Hu and R.A. de Callafon. The material of Chapter 6.4, in full, is a reprint of part of the material as it appears in *Covariance based estimation for reduced order models of microgrid power flow dynamics*, Y. Hu, S.A.R. Konakalla and R.A. de Callafon, 18th Symposium on System Identification (SYSID 2018), Stockholm, Sweden, Jul. 9–11, 2018, pp. 903–908. The dissertation author was the primary investigator and author of the papers.

Chapter 7, in full, has been submitted for publication in IEEE Transactions on Control Systems Technology. It may appear as *Tensor network based MIMO Volterra model for lithium-ion batteries*, Y. Hu, R.A. de Callafon, N. Tian and H. Fang. The dissertation author was the primary investigator and author of the paper.

VITA

2013	Bachelor of Engineering in Automation, Xidian University, China
2016	Master of Science in Control Science and Engineering, University of Science and Technology of China
2019	Controls Research Intern, Robert Bosch LLC, USA
2021	Doctor of Philosophy in Engineering Sciences (Mechanical Engineering), University of California San Diego

PUBLICATIONS

Y. Hu, R.A. de Callafon, N. Tian and H. Fang, “Modeling of lithium-ion batteries via tensor network based Volterra model,” *Modeling, Estimation and Control Conference (MECC 2021)*, under review.

L. Tan, Y. Hu, S. Tan, R.A. de Callafon and I. Altintas “Wildfire perimeter detection via rank-based principal component analysis,” *IEEE 60th Annual Conference on Decision and Control (CDC)*, under review.

Y. Hu and R.A. de Callafon, “Microgrid dynamic modeling with power flow covariance data,” *IEEE Transactions on Power Systems*, under review.

Y. Hu, R.A. de Callafon, N. Tian and H. Fang, “Tensor network based MIMO Volterra model for lithium-ion batteries,” *IEEE Transactions on Control Systems Technology*, under review.

Y. Hu, L. Tan and R.A. de Callafon, “Noniterative tensor network-based estimation for MIMO Volterra system identification,” *IEEE Transactions on Automatic Control*, under review.

Y. Hu, Y. Jiang and R.A. de Callafon, “ Variance reduction in Covariance Based Realization Algorithm with application to closed-loop data,” *Automatica*, vol. 113, pp.108683, 2020.

Y. Hu, L. Tan and R.A. de Callafon, “Persistent excitation condition for MIMO Volterra system identification with Gaussian distributed input signals,” in *Proceedings of IEEE 58th Annual Conference on Decision and Control (CDC)*, Nice, France, Dec. 11–13, 2019, pp. 1752–1757.

Y. Hu, S.A.R. Konakalla and R.A. de Callafon, “Covariance based estimation for reduced order models of microgrid power flow dynamics,” in *Proceedings of IFAC 18th Symposium on System Identification (SYSID 2018)*, Stockholm, Sweden, Jul. 9–11, 2018, pp. 903–908.

Y. Hu and R.A. de Callafon, “Covariance Based Realization Algorithm for identification of linear time periodic system,” in *Proceedings of 2018 Annual American Control Conference (ACC)*, Wisconsin Center, Milwaukee, USA, Jun. 27–29, 2018, pp. 1102–1107.

Y. Hu and R.A. de Callafon, “Optimal weighting for covariance based realization algorithm,” in *Proceedings of IEEE 56th Annual Conference on Decision and Control (CDC)*, Melbourne, VIC, Australia, Dec. 12–15, 2017, pp. 5274–5279.

ABSTRACT OF THE DISSERTATION

Low-Rank Approximations for Estimation of Multivariable Dynamics with Application to Energy Systems

by

Yangsheng Hu

Doctor of Philosophy in Engineering Sciences (Mechanical Engineering)

University of California San Diego, 2021

Professor Raymond A. de Callafon, Chair

System identification is a powerful tool for estimation of the systematic and causal dynamic relations by learning from measurements of input/output data. It provides an effective alternative when first-principles modeling is intractable. It plays an important role in model-based controller design especially during the stage of software-in-the-loop (SIL) and has been widely exploited in a wide range of real-world applications. This dissertation presents a study of theories and applications on multi-input multi-output (MIMO) system identification for estimation of multivariable dynamics between input/output data. The objective is to finally obtain a simplified dynamic model by emphasizing low-rank information during estimation with the aid of singular value decomposition (SVD).

In this dissertation, we study the Covariance Based Realization Algorithm (CoBRA) for estimation of linear time-invariant and time-periodic dynamics and the tensor network (TN) based algorithm for estimation of nonlinear dynamics described by Volterra series. The CoBRA, one branch of subspace methods by using covariance data, is able to focus on the estimation of a low-order deterministic model from noisy data by exploiting the low-rank feature of the data matrix. We propose an optimal implementation of the CoBRA and investigate its efficacy in a closed-loop setting compared with other subspace methods. A MIMO Volterra model is powerful to approximate nonlinear dynamics on the basis of input/output observations. We cope with the curse of dimensionality during model formulation and present TN-based noniterative algorithms for MIMO Volterra system identification. The proposed algorithms show numerical advantages over iterative algorithms.

This dissertation also studies two real-world applications in energy systems which involve estimation of multivariable dynamics. First, we investigate the microgrid dynamic modeling with power flow covariance data. We propose a model structure which consists of a MIMO linear part concatenated by a static nonlinear part to account for the power loss in transmission lines. Second, we study the modeling for lithium-ion batteries to accurately predict the output terminal voltage. We develop a TN-based Volterra double-capacitor (VDC) model, which is capable of predicting both static and dynamic nonlinearities simultaneously in a more accurate way than other equivalent circuit models (ECMs).

Chapter 1

Introduction

1.1 System Identification

System identification is a technique building mathematical models of dynamic systems from input/output measurement data [1, 2, 3, 4]. It serves as a powerful alternative when first-principles modeling is intractable. On the other hand, it is impossible to model every single detail of a system due to the complexity or limited knowledge. This is one limitation of using first-principles modeling. However, data-driven modeling using system identification can capture important features observed from the data and facilitate the modeling process by careful selections of certain model structures. With the proliferation of various sensors of high precision and low cost, system identification becomes more and more attractive nowadays. As a core subject of control systems theory, system identification is also crucial for the model-based controller design.

A large amount of papers on system identification have been published over the last 60 years. The theory of system identification started in the mid-1960s when two important papers were published. The first one was contributed by K.J. Åström and B. Torsten in [5], in which the maximum likelihood (ML) was extended to estimate AutoRegressive Moving Average with eXogenous inputs (ARMAX) models. Many relevant identification techniques which are now labeled as prediction error methods (PEM) and instrumental variable (IV) methods have been developed since then. These techniques have been summarized and analyzed comprehensively in [1, 2]. The second one was contributed by B.L. Ho and R.E. Kálmán in [6], in which the deterministic state-space realization problem was solved for the first time by constructing a Hankel matrix by using impulse responses. This work provided insights into identification of a state-space model which more naturally accounts for a multi-input multi-output (MIMO) system than a transfer function used in PEM. A new effort in system identification and digital signal processing by referring to the QR decomposition and the singular value decomposition (SVD) emerged in the mid-1980s. These realization theory-based techniques have been intensively studied and led to a development of the so-called subspace (identification) methods. It should be noted that the origin of subspace methods may date back to multivariable statistical analysis especially to the principal component analysis (PCA) due to H. Hotelling in the 1930s [7, 8, 9]. The first comprehensive monograph on subspace methods for linear system identification was published by P. Van Overschee and B.L. De Moor in [10]. T. Katayama advanced subspace methods by incorporating stochastic realization theory [11]. Another development of subspace methods is the emergence of predictor-based subspace identification (PBSID) [12], which is similar to estimating a high-order ARX structure. An advantage of subspace methods over PEM especially in MIMO system identification is that there is no need to refer to nonlinear optimization algorithms which often lead to local optima.

A lot of effort has been put into the linear system identification mostly based on classical statistical methods. However, nonlinear system identification still remains as an important open issue and available techniques that have been demonstrated effective in practice can be found in [13, 14]. In addition, system identification theory is being pushed forward in recent years by interacting with other fields such as statistics, machine learning, etc and it brings up many interesting topics [15, 16, 17].

1.2 Motivation and Problem Formulation

Estimation of multivariable dynamics may be required when multiple inputs and even multiple outputs are involved in an application problem. System identification, serving as the interface between the real-world applications and the mathematical world of control theory and model abstractions, is usually an necessity for dynamic modeling and successful applications. In addition, different techniques of system identification are selected based on the character of the models to be estimated: linear, nonlinear, time-invariant, parametric, MIMO, etc. The problem of estimating a reliable model becomes more difficult and complicated especially when only short and noisy data is available and/or the model structure is too rich with too many parameters to be estimated which leads to overfitting problem. Also, one expects a simplified model capturing the major dynamics between the input/output data for the sake of numerical robustness and controller design. Thus, how to perform low-rank approximations during system identification becomes crucial for estimation of multivariable dynamics.

The real-world systems are in general nonlinear by nature but can be approximated by a linear time-invariant model around a certain equilibrium. This observation lays the foundation for the intensive research in system identification by using linear time-invariant models. For the estimation of linear multivariable dynamics, subspace methods are preferred due to its numerical advantage over PEM. It is worth investigating the properties and implementation of subspace methods in the case of noise-contaminated data. The Covariance Based Realization Algorithm (CoBRA), as a branch of subspace methods, allows one to extract the low-rank deterministic information from the noisy data by using covariance pre-processing. The corresponding estimates are consistent, but the optimal implementation of the CoBRA is still an open issue.

Although linear system identification is widely used, the need for nonlinear system identification extends far beyond the control application field. Nonlinear models are instrumental in achieving a basic understanding of how a system works such as brain activity modeling and chemical reactions. Since every system that is not linear is nonlinear, nonlinear system identification is an expansive topic and it is impossible to give a full overview of all aspects of this field. A user guide for the state-of-the-art nonlinear system identification can be found in [14]. Among these techniques, we are interested in the identification of Volterra models, which are always bounded-input bounded-output stable. In addition, a MIMO Volterra model of high degree and long memory length is able to capture complicated coupled nonlinear multivariable dynamics. However, a normal representation of such a model suffers from the curse of dimensionality and requires a prohibitively large storage to save an enormous amount of parameters. This observation motivates us to break the curse of dimensionality with the aid of a tensor network (TN) technique. However, developing consistent and numerically robust TN-based algorithms which are able to estimate a model with low TN-ranks from short and noisy data is crucial and also an open issue for Volterra system identification.

System identification has attracted a lot of attention due to its practical significance and data-driven property which makes it widely applicable to many different fields. On the other hand, applying the system identification techniques to real-world applications will provide insights into the practical efficacy and inspire further improvements. In this dissertation, we will investigate two real-world applications in energy systems where system identification is applied for estimation of multivariable dynamics. We only consider two types of energy systems: microgrid power delivery system and battery energy storage system. The first application is the estimation of microgrid power flow dynamics via the CoBRA. Feedback control of transient effects in power flow through a microgrid requires a better understanding of the dynamic aspects of distributed energy resources (DERs) located within the microgrid. The second application is the estimation of battery dynamics via TN-based Volterra system identification. Rechargeable batteries have been widely used nowadays due to the proliferation of portable consumer electronics, electric vehicles, etc and advanced battery management systems require proper battery models. The battery dynamics is nonlinear by nature and a careful modeling will facilitate utilizing the full potential from a battery without violating constraints.

1.3 Main Contributions

In this dissertation, we aim to push forward the theories on the CoBRA and MIMO Volterra system identification and apply them to the real-world applications. Our main contributions are summarized as follows:

- **A detailed investigation of CoBRA**: The CoBRA, a branch of subspace methods using covariance data, has been demonstrated to be effective in identifying a low-order model from experimental data contaminated with unknown and possibly high-order spectral characteristics. We propose to modify the basic CoBRA to adapt to different settings. First, we extend the CoBRA to identify an LTP system by referring to its generalized time-lifted model. We develop a novel method to compute a topologically equivalent realization for the linear time-periodic (LTP) system and a measure of the estimation quality. Second, we investigate the statistical properties of the CoBRA in detail and develop an optimal implementation of the CoBRA in terms of minimum variance. The optimal CoBRA is then incorporated into a two-stage closed-loop system identification and shows superior performance to other competitive methods.

- **Development of TN-based algorithms for MIMO Volterra system identification**: Volterra model serves as one of the powerful alternatives to approximate the nonlinear dynamics on the basis of input/output observations. Few literature exploits the Volterra models of high degree or even the MIMO case to capture complicated coupled nonlinear behaviors within data. This is due to the exponentially growing number of kernel coefficients as the degree increases, also known as the curse of dimensionality. We propose novel TN-based algorithms for MIMO Volterra system identification since the TN structure can break this curse by trading storage for computation. We propose two noniterative TN-based algorithms with/without regularization and both can automatically seek a low-rank representation of a MIMO Volterra model to overcome overfitting. The proposed algorithms show advantages over the Alternating Linear Scheme (ALS) and the Modified ALS (MALS) which are iterative and do not guarantee a convergence to an appropriate solution. In addition, by using the properties of cumulant, we derive the persistent excitation condition for MIMO Volterra system identification with Gaussian distributed input signals.

- **Microgrid dynamic modeling with power flow covariance data**: Feedback control of transient effects in power flow through a microgrid requires a better understanding of the dynamic aspects of DERs located within the microgrid. We present a data-driven approach to modeling DER power flow dynamics within a microgrid by using the discrete-time dependent covariance between active and reactive power flow. First, we consider a simplified case with the microgrid excited around one power equilibrium and demonstrate that the CoBRA works well in a microgrid setting. Then, we consider a more realistic case taking into account nonlinearities such as ramp limit in the DER inputs and power loss due to the admittance of the microgrid network. It is shown via a simulation and an experimental study that simulated and measured power flow transient effects can be matched with high accuracy.

- **Lithium-ion battery modeling for terminal voltage prediction**: Accurate battery modeling is fundamental for battery management system to function well and extract the full potential from a battery without violating constraints. It has been demonstrated that a (nonlinear) double-capacitor (NDC) model can accurately describe the operation of a lithium-ion cell by using an electrical circuit structure. We propose a TN-based Volterra double-capacitor (VDC) model to further improve the prediction performance of the NDC model. The whole modeling can be regarded as a combination of equivalent circuit model (ECM) and supervised learning. We propose a bond core sweeping algorithm to estimate low-rank TN-cores. The experimental results show that the VDC model outperforms other ECMs in battery modeling.

1.4 Organization

The remaining part of the dissertation is organized into three parts. Part I performs an in-depth study in the CoBRA, including Chapters 2,3. Part II investigates the MIMO Volterra system identification, including Chapters 4,5. Part III concerns the real-world applications to energy systems by using the CoBRA and MIMO Volterra system identification, including Chapters 6,7.

In Chapter 2, we extend the CoBRA to perform the parameter estimation for MIMO LTP systems in the time domain. A generalized time-lifted state-space model for an LTP system is developed so that the CoBRA can be applied to get a low-rank deterministic linear time-invariant model. We formulate a topologically equivalent realization for the original LTP system by using the estimated time-lifted model.

In Chapter 3, we conduct an in-depth study of the statistical behavior of the noise effects on the CoBRA. We propose an approach to reducing the variance of an estimate obtained by the CoBRA via the choice of optimal row and column weighting matrices. We adopt a two stage technique with the estimate on an intermediate instrument for closed-loop implementation of the CoBRA.

In Chapter 4, we investigate the MIMO Volterra system identification in the TN representation, which alleviates the high storage cost due to the curse of dimensionality. We propose noniterative TN-based algorithms with two tuning factors to solve either a linear or ridge regression to produce low-rank estimates and the simulation results show advantages over the iterative algorithms proposed in other literature.

In Chapter 5, we derive a persistent excitation condition for the parameter estimation in MIMO Volterra system identification in the case of zero mean, Gaussian distributed (not necessarily white) input signals. We assume a symmetric structure of kernels without loss of generality. We reformulate and simplify the persistent excitation problem by performing a moments to cumulants conversion.

In Chapter 6, we use the CoBRA to identify a low-order dynamic model during microgrid dynamic modeling with power flow covariance data. We consider the nonlinearity caused by line power loss by estimating the admittance matrix of the microgrid network. The resulting model consists of a linear dynamic part and a static nonlinear part and shows high prediction accuracy in a simulation and an experimental study.

In Chapter 7, we use the MIMO Volterra system identification to improve the prediction performance in battery modeling. We adopt the Volterra model to build a mathematical mapping from features, which are obtained from a linear double-capacitor model, to the output terminal voltage. We implement the TN-based algorithm to compute a low-rank representation of the Volterra model.

In Chapter 8, we summarize and draw concluding remarks for this dissertation. We will also look into the future of relevant research.

Part I

Covariance Based Realization Algorithm

Chapter 2

Identification of Linear Time Periodic Systems via CoBRA

This chapter focuses on the parameter estimation of multi-input multi-output (MIMO), linear time-periodic (LTP) systems in the time domain. Discrete-time state-space models are selected for the description of LTP systems and the Covariance Based Realization Algorithm (CoBRA), one branch of subspace methods, is used for the identification of the generalized time-lifted state-space model for an LTP system. The use of the CoBRA for estimation is motivated by the robustness against output noise with (high-order) noise dynamics and a focus on the estimation of low-rank deterministic model for the LTP system. In addition to the use of the CoBRA, a novel method is proposed to compute a topologically equivalent realization for the original LTP system. The method includes two different but theoretically equivalent approaches to calculating the state matrices, the estimate difference between which can be also used as a measure for the estimation quality.

2.1 Introduction

System identification for linear time-invariant (LTI) models is well established in recent decades [2, 3]. System identification is about building a mathematical model with low order, followed by a procedure to estimate a finite number of parameters, from experimental input and output data. However, there are many cases where the time-invariant assumption cannot be met due to time-varying intrinsic nature of the data generating system [18, 19]. Among these time-varying cases, a variety of systems can be modeled appropriately through LTP models.

If the physical system contains periodic motion (e.g. electrical motors, fans, wind turbines, human locomotion, etc.), then a nonlinear time-invariant system can be formulated around the periodic stable trajectory. Alternatively, a system may be subjected to multirate sampling and LTP models are often encountered [20, 21, 22, 23]. LTP models can also capture the features that are not presented in the LTI approximation. For example, the effect of frequency-coupling dynamics can be captured by an LTP model in the applications that use grid connected DC/AC inverters [24]. As in the LTI case, there is a large number of possible situations where the parameters of the LTP model for a real system can not be determined directly from first principles. That's when system identification or experiment based modeling can be used as a powerful alternative regardless of the physical structure.

However, relatively little attention of system identification has been paid to LTP models. A comprehensive summary of different descriptions of LTP models can be found in [25]. Most experimental identification methods for LTP models are conducted in the frequency domain based on the idea of harmonic transfer function (HTF) in the exponentially modulated periodic (EMP) regime [20, 26]. Very few results are performed in the time domain and they are typically applicable to the general time-varying case [27]. Although a large amount of data is required, it is more straightforward to perform the time domain identification. This chapter investigates a novel method to identify a state-space representation of an LTP model in the time domain while reducing the requirement of storage space.

A (MIMO) LTP model can be reformulated as an LTI time-lifted state-space form with augmented input and output signals [25]. Once an LTI time-lifted representation is used, identification methods for (MIMO) LTI systems can be incorporated. Popular techniques are subspace methods [3, 10, 11]. One advantage of a subspace method is the lack of nonlinear optimization and facilitated by the robust numerical computation of QR factorization and singular value decomposition (SVD). However, the storage requirements and numerical computations become intensive since a large number of data points may be needed in the time domain.

The CoBRA, a branch of subspace methods, provides a solution to this problem [28, 29, 30]. The CoBRA enables the data compression by using covariance functions. In addition, correlation with an instrumental variable avoids the estimation of possibly high order noise model and allows a focus on the estimation of potential low-order deterministic dynamics.

The contribution of this chapter is to formulate a CoBRA that enables the identification of discrete-time LTP models. This is done by proposing a novel method to compute a topologically equivalent realization in the time domain. Two different but theoretically equivalent approaches to calculating the state matrices are given in the proposed method. The extent of the closeness between the two estimated state matrices can be used as an additional measure for the estimation quality.

The remaining part of the chapter is organized as follows: Section 2.2 provides insights into the potential time-invariant property of LTP models and proposes a generalized time-lifted state-space reformulation. The procedure of incorporating the CoBRA into the identification of LTP models is presented in Section 2.3. Pole location constraints based on the concept of monodromy matrix are also included. In Section 2.4, a novel method for a topologically equivalent realization of the original LTP model is proposed. Section 2.5 illustrates the efficiency of the proposed method by a simulation and Section 2.6 summarizes this chapter.

2.2 Representation of Discrete-Time LTP Systems

2.2.1 Time-Invariant Relation in Discrete-Time LTP Systems

Consider the following state-space model for a discrete-time LTP system:

$$\begin{cases} x\left(t+1\right) = A\left(t\right)x\left(t\right) + B\left(t\right)u\left(t\right) \\ y\left(t\right) = C\left(t\right)x\left(t\right) + D\left(t\right)u\left(t\right) \end{cases} \tag{2.1}$$

where $A(t) \in \mathbb{R}^{n\times n}, B(t) \in \mathbb{R}^{n\times p}, C(t) \in \mathbb{R}^{m\times n}, D(t) \in \mathbb{R}^{m\times p}$ are periodic matrices of period T, and $t \in \mathbb{Z}$. $u(t)$, $y(t)$, and $x(t)$ are input, output, and state vectors, respectively. Note that it is assumed here that the dimension of $x(t)$ is fixed to be n. In order to determine the stability of the above system, we should refer to the corresponding monodromy matrix [31]:

$$\Psi\left(l\right) = \Phi\left(l+T, l\right),\ l \in \{1, 2, \cdots, T\} \tag{2.2}$$

where $\Phi(t,l)$ is state-transition matrix defined by

$$\Phi\left(t,l\right)=\begin{cases} I, & \text{if } t=l \\ A\left(t-1\right)\cdots A\left(l+1\right)A\left(l\right), & \text{if } t>l \end{cases} \tag{2.3}$$

It is obvious that $\Psi(l)$ is a serial concatenation of $A(t)$ over a single period and its eigenvalues are called characteristic multipliers, independent of l [31]. The system is stable if and only if its monodromy matrix is stable, i.e., all the eigenvalues of $\Psi(l)$ are lying in the unit disk in the complex plane. From now on, we assume that the LTP system to be identified is stable.

Since it is not straightforward to see the potential invariant relationship from (2.1), we first resort to its equivalent input-output representation via the convolution sum

$$y\left(t\right)=\sum_{i=0}^{\infty}M_i\left(t\right)u\left(t-i\right) \tag{2.4}$$

where $M_i(t)\in\mathbb{R}^{m\times p},\ i=0,1,2\cdots$ are periodic Markov coefficients of period T over t. The relationship between (2.1) and (2.4) can be described by

$$\begin{aligned} M_0\left(t\right)&=D\left(t\right) \\ M_{kT+s}\left(t\right)&=C\left(t\right)\Psi(t)^k\Phi\left(t,t-s+1\right)B\left(t-s\right) \end{aligned} \tag{2.5}$$

where $k=0,1,\cdots$ and $s\in\{1,2,\cdots,T\}$. Using the periodicity of $M_i(t)$, we can rewrite the output $y\left(kT+s\right)$ of (2.4) at time instant $t=kT+s$ according to

$$\begin{aligned} y\left(kT+s\right)&=\sum_{i=0}^{T-1}\underbrace{\left[\sum_{j=0}^{\infty}M_{jT+i}\left(s\right)u\left(\left(k-j\right)T+s-i\right)\right]}_{H_{i,s}(z)u_{s-i}(kT)} \\ &=\left[H_{0,s}\left(z\right)\ H_{1,s}\left(z\right)\ \cdots\ H_{T-1,s}\left(z\right)\right]\begin{bmatrix} u_s\left(kT\right) \\ u_{s-1}\left(kT\right) \\ \vdots \\ u_{s-T+1}\left(kT\right)\end{bmatrix} \end{aligned} \tag{2.6}$$

where $u_l\left(kT\right) = u\left(kT+l\right)$, $H_{i,s}(z)$ is a time-invariant transfer function for any fixed i and s, and z is a one step ahead shift operator over k. For a given s, we can obtain a time-invariant relation as shown in (2.6). It is obvious that the system is also with period $T_c = cT, c \in \{1, 2, \cdots\}$. The reason for introducing T_c is that the proposed method may need to construct the time-lifted model using more than one period of the signals. Also, the symbol η is used from now instead of s to represent the more general case. Thus, we can further define an augmented signal in a generalized sense

$$u_{(\eta)}\left(k\right) = \left[u_{\eta}(kT_c)' \cdots u_{\eta+T_c-1}(kT_c)'\right]' \tag{2.7}$$

and $y_{(\eta)}(k)$ is defined in a similar way. Note that x' represents the transpose of a vector (or matrix) x in this chapter. We are not using x^{T} instead as in other chapters to avoid the confusion between the transpose and period T in this chapter. Without too much effort, we have the following time-invariant relation from (2.6)

$$y_{(\eta)}\left(k\right) = G_{(\eta)}\left(z\right) u_{(\eta)}\left(k\right) \tag{2.8}$$

where $G_{(\eta)}(z)$ is a fixed block transfer function from $u_{(\eta)}(k)$ to $y_{(\eta)}(k)$ for a given $\eta \in \{1, 2, \cdots, T_c\}$. At this point, we have deduced the potential time-invariant relation contained in the given LTP system.

2.2.2 Generalized Time-Lifted State-Space Reformulation

Since a realization algorithm is used in this chapter, a state-space form is preferred. Although (2.8) provides us with a time-invariant framework between the augmented signals $\{u_{(\eta)}(t), y_{(\eta)}(t)\}$, we would not go further into the specific representation of $G_{(\eta)}(z)$. Instead, we denote $\eta = dT + l,\ l \in \{1, 2, \cdots, T\}$, where d is some integer to make this equation hold, and a state-space form of the relation in (2.8), which is a generalization of the time-lifted state-space model in [25], and given by

$$\begin{cases} x_\eta\left(k+1\right) = A_{(\eta)} x_\eta\left(k\right) + B_{(\eta)} u_{(\eta)}\left(k\right) \\ \quad y_{(\eta)}\left(k\right) = C_{(\eta)} x_\eta\left(k\right) + D_{(\eta)} u_{(\eta)}\left(k\right) \end{cases} \tag{2.9}$$

where $x_\eta\left(k\right) = x\left(kT_c + \eta\right)$, and matrices $A_{(\eta)} \in \mathbb{R}^{n\times n}$, $B_{(\eta)} \in \mathbb{R}^{n\times pT_c}$, $C_{(\eta)} \in \mathbb{R}^{mT_c\times n}$, and $D_{(\eta)} \in \mathbb{R}^{mT_c\times pT_c}$ are functions of $\{A(t), B(t), C(t), D(t)\}$ in (2.1) defined as follows:

$$A_{(\eta)} = \Phi\left(l + T_c, l\right) = \left[\Psi\left(l\right)\right]^c \tag{2.10}$$

$$\begin{aligned} B_{(\eta)} = \Big[& \Phi\left(l + T_c, l+1\right) B\left(l\right) \cdots \\ & \Phi\left(l + T_c, l + T_c - 1\right) B\left(l + T_c - 2\right) \quad B\left(l + T_c - 1\right) \Big] \\ = \Big[& \left[\Psi\left(l\right)\right]^{c-1} \Phi\left(l + T, l+1\right) B\left(l\right) \cdots \\ & \Phi\left(l + T, l + T - 1\right) B\left(l + T - 2\right) \quad B\left(l + T - 1\right) \Big] \end{aligned} \tag{2.11}$$

$$\begin{aligned} C_{(\eta)} = \Big[& C(l)' \quad \Phi(l+1, l)' C(l+1)' \\ & \cdots \Phi(l + T_c - 1, l)' C(l + T_c - 1)' \Big]' \\ = \Big[& C(l)' \quad \Phi(l+1, l)' C(l+1)' \\ & \cdots \left[\Psi(l)'\right]^{c-1} \Phi(l + T - 1, l)' C(l + T - 1)' \Big]' \end{aligned} \tag{2.12}$$

$$D_{(\eta)} = \left\{ \left(D_{(\eta)}\right)_{ij} \right\}, \quad i, j = 1, 2, \cdots, T_c \tag{2.13}$$

where $\left(D_{(\eta)}\right)_{ij}$ is given by

$$\left(D_{(\eta)}\right)_{ij} = \begin{cases} 0, & \text{if } i < j \\ D\left(l+i-1\right), & \text{if } i = j \\ C\left(l+i-1\right)\Phi\left(l+i-1, l+j\right)B\left(l+j-1\right), & \text{if } i > j \end{cases} \tag{2.14}$$

Remark 1. *Note that $B_{(\eta)}$ is the reversed extended controllability matrix and $C_{(\eta)}$ is the extended observability matrix. $D_{(\eta)}$ can be nonzero even if the original LTP system in (2.1) has one-step time delay, i.e., $D(l) = 0$. Although the time-lifted state-space model in (2.9) is with the augmented signals $\{u_{(\eta)}(t), y_{(\eta)}(t)\}$, the order of the reformulated model is still the same as the original one.*

2.3 CoBRA for LTP Systems

Given the augmented signals $\{u_{(\eta)}(t), y_{(\eta)}(t)\}$, the CoBRA can be used to identify the unknown matrices in (2.9) up to a similarity transformation. It can reduce the storage requirements and allow pole location constraints during parameter estimation. Only basic notations and procedures are outlined here and more details on CoBRA method can be found in [28, 29, 30].

2.3.1 Realization by CoBRA

Usually only the noise contaminated measured output $z(t) = y(t) + v(t)$ is available, where $v(t)$ is a quasi-stationary zero mean (probably colored) Gaussian noise with unknown spectrum. Although it may not cause an ambiguity, it should still be mentioned that z here is not an operator as in Section 2.2. For the time-lifted version in (2.9), we have

$z_{(\eta)}(k) = y_{(\eta)}(k) + v_{(\eta)}(k)$. We may assume that $v(t)$ is the output of a linear (probably time-invariant or periodic) system with a zero mean Gaussian white noise $e(t)$ as the input. Similarly as in (2.8), the following time-invariant relation holds:

$$v_{(\eta)}(k) = H_{(\eta)}(z)\, e_{(\eta)}(k) \tag{2.15}$$

where $H_{(\eta)}(z)$ is a fixed block transfer function from $e_{(\eta)}(k)$ to $v_{(\eta)}(k)$ for a given $\eta = \{1, 2, \cdots, T_c\}$. The augmented $e_{(\eta)}(k)$ is still Gaussian white noise although its dimension increases. Thus, the augmented $v_{(\eta)}(k)$ is still a quasi-stationary zero mean Gaussian noise but with a different spectrum. In many cases, we care more about the deterministic part of a system than the noise dynamics. For this reason, we introduce an instrumental variable $\xi(t)$ satisfying:

- $\xi(t) \in \mathbb{R}^{pT_c \times 1}$ is correlated with $u_{(\eta)}(t)$.
- $\xi(t) \in \mathbb{R}^{pT_c \times 1}$ is uncorrelated with $v_{(\eta)}(t)$.

In that case, we have $R_{z_{(\eta)}\xi} = R_{y_{(\eta)}\xi} + R_{v_{(\eta)}\xi} = R_{y_{(\eta)}\xi}$ since the stochastic part $R_{v_{(\eta)}\xi} = 0$ in the ideal case. The time-lifted state-space form in (2.9) can be rewritten as:

$$\begin{cases} R_{x_\eta\xi}(\tau+1) = A_{(\eta)} R_{x_\eta\xi}(\tau) + B_{(\eta)} R_{u_{(\eta)}\xi}(\tau) \\ R_{z_{(\eta)}\xi}(\tau) = C_{(\eta)} R_{x_\eta\xi}(\tau) + D_{(\eta)} R_{u_{(\eta)}\xi}(\tau) \end{cases} \tag{2.16}$$

where $\tau \in \{-\bar{\tau}, -\bar{\tau}+1, \cdots, 0, 1, \cdots, \bar{\tau}\}$ and the initial condition is $R_{x_\eta\xi}(-\bar{\tau}) = R_{\bar{\tau}}$. The values of these covariance functions can be used to create the block Hankel matrices

$$\begin{cases} \mathbf{R}^{(-\bar{\tau})}_{z_{(\eta)}\xi} = \text{Hankel}\left(R_{z_{(\eta)}\xi}(\tau), r, -\bar{\tau}, \bar{\tau}-1\right) \\ \mathbf{R}^{(-\bar{\tau}+1)}_{z_{(\eta)}\xi} = \text{Hankel}\left(R_{z_{(\eta)}\xi}(\tau), r, -\bar{\tau}+1, \bar{\tau}\right) \\ \mathbf{R}_{u_{(\eta)}\xi} = \text{Hankel}\left(R_{u_{(\eta)}\xi}(\tau), r+1, -\bar{\tau}, \bar{\tau}\right) \end{cases} \tag{2.17}$$

where

$$\operatorname{Hankel}\left(g\left(\tau\right),r,\tau_1,\tau_2\right)=\begin{bmatrix} g\left(\tau_1\right) & g\left(\tau_1+1\right)\cdots & g\left(\tau_2-r+1\right)\\ g\left(\tau_1+1\right) & g\left(\tau_1+2\right)\cdots & g\left(\tau_2-r+2\right)\\ \vdots & \vdots & \vdots\\ g\left(\tau_1+r-1\right) & g\left(\tau_1+r\right)\cdots & g\left(\tau_2\right)\end{bmatrix} \tag{2.18}$$

and r is the number of block rows. Note that the Hankel matrix of the input signal should be of full row rank. These Hankel matrices are related by the data matrix equations according to

$$\mathbf{R}_{z_{(\eta)}\xi}^{(-\bar{\tau})}=\mathrm{O}_{\bar{\tau}}\mathbf{R}_{x_\eta\xi}+\mathrm{T}\mathbf{R}_{u_{(\eta)}\xi} \tag{2.19}$$

$$\mathbf{R}_{z_{(\eta)}\xi}^{(-\bar{\tau}+1)}=\mathrm{O}_{\bar{\tau}}A_{(\eta)}\mathbf{R}_{x_\eta\xi}+\vec{\mathrm{T}}\mathbf{R}_{u_{(\eta)}\xi} \tag{2.20}$$

where $\mathbf{R}_{x_\eta\xi}=\begin{bmatrix}R_{x_\eta\xi}\left(-\bar{\tau}\right) & R_{x_\eta\xi}\left(-\bar{\tau}+1\right) & \cdots & R_{x_\eta\xi}\left(\bar{\tau}-r\right)\end{bmatrix}$, T and $\vec{\mathrm{T}}$ are Toeplitz matrices of unknown Markov parameters, $\mathbf{R}_{z_{(\eta)}\xi}^{(-\bar{\tau}+1)}$ is the one-step time-shifted version of $\mathbf{R}_{z_{(\eta)}\xi}^{(-\bar{\tau})}$ in (2.17), and $\mathrm{O}_{\bar{\tau}}$ is the extended observability matrix

$$\mathrm{O}_{\bar{\tau}}=\begin{bmatrix}C_{(\eta)}{}' & \left(C_{(\eta)}A_{(\eta)}\right)' & \cdots & \left(C_{(\eta)}A_{(\eta)}^{\bar{\tau}-1}\right)'\end{bmatrix}' \tag{2.21}$$

Now we may introduce an orthogonal projection

$$\Pi_{\mathbf{R}'_{u_{(\eta)}\xi}}^{\perp}=\mathbf{I}-\mathbf{R}'_{u_{(\eta)}\xi}\left[\mathbf{R}_{u_{(\eta)}\xi}\mathbf{R}'_{u_{(\eta)}\xi}\right]^{-1}\mathbf{R}_{u_{(\eta)}\xi} \tag{2.22}$$

and (2.19) and (2.20) can be rewritten into

$$\mathbf{R}_{z_{(\eta)}\xi}^{(-\bar{\tau})}\Pi_{\mathbf{R}'_{u_{(\eta)}\xi}}^{\perp}=\mathrm{O}_{\bar{\tau}}\mathbf{R}_{x_\eta\xi}\Pi_{\mathbf{R}'_{u_{(\eta)}\xi}}^{\perp} \tag{2.23}$$

$$\mathbf{R}_{z_{(\eta)}\xi}^{(-\bar{\tau}+1)}\Pi_{\mathbf{R}'_{u_{(\eta)}\xi}}^{\perp}=\mathrm{O}_{\bar{\tau}}A_{(\eta)}\mathbf{R}_{x_\eta\xi}\Pi_{\mathbf{R}'_{u_{(\eta)}\xi}}^{\perp} \tag{2.24}$$

From the expression, the estimation of matrix $A_{(\eta)}$ can be done by solving the optimization problem

$$\min_{A_{(\eta)}} \left\| \mathrm{O}_{\bar{\tau}} A_{(\eta)} \mathbf{R}_{x_\eta \xi} \Pi^{\perp}_{\mathbf{R}'_{u_{(\eta)}\xi}} - \mathbf{R}^{(-\bar{\tau}+1)}_{z_{(\eta)}\xi} \Pi^{\perp}_{\mathbf{R}'_{u_{(\eta)}\xi}} \right\|_F \tag{2.25}$$

whose solution is explicit and given by

$$\bar{A}_{(\eta)} = (\mathrm{O}_{\bar{\tau}})^{\ddagger} \mathbf{R}^{(-\bar{\tau}+1)}_{z_{(\eta)}\xi} \Pi^{\perp}_{\mathbf{R}'_{u_{(\eta)}\xi}} \left(\mathbf{R}_{x_\eta \xi} \Pi^{\perp}_{\mathbf{R}'_{u_{(\eta)}\xi}} \right)^{\dagger} \tag{2.26}$$

Note that $\mathrm{O}_{\bar{\tau}}$ and $\mathbf{R}_{x_\eta \xi} \Pi^{\perp}_{\mathbf{R}'_{u_{(\eta)}\xi}}$ can be estimated through the singular value decomposition of (2.23) via

$$\begin{aligned} \mathbf{R}^{(-\bar{\tau})}_{z_{(\eta)}\xi} \Pi^{\perp}_{\mathbf{R}'_{u_{(\eta)}\xi}} &= \begin{bmatrix} U_n & U_s \end{bmatrix} \begin{bmatrix} \Sigma_n & 0 \\ 0 & \Sigma_s \end{bmatrix} \begin{bmatrix} {V_n}' \\ {V_s}' \end{bmatrix} \\ &\approx \underbrace{U_n \Sigma_n^{1/2}}_{\mathrm{O}_{\bar{\tau}}} \underbrace{\Sigma_n^{1/2} {V_n}'}_{\mathbf{R}_{x_\eta \xi} \Pi^{\perp}_{\mathbf{R}'_{u_{(\eta)}\xi}}} \end{aligned} \tag{2.27}$$

where n is the selected order for the time-lifted system in (2.9) so that Σ_n contains the dominant singular values in comparison to Σ_s. An estimate $\bar{C}_{(\eta)}$ for matrix $C_{(\eta)}$ can be achieved by selecting the first mT_c rows of $\mathrm{O}_{\bar{\tau}}$ computed in (2.27).

Estimation of matrices $B_{(\eta)}, D_{(\eta)}$ and the initial condition $R_{\bar{\tau}}$ can then be done via a linear regression optimization as in common subspace methods. Given matrices $\bar{A}_{(\eta)}$ and $\bar{C}_{(\eta)}$, these parameters are related via

$$R_{z_{(\eta)}\xi}(\tau) = \bar{C}_{(\eta)} \bar{A}^{\tau+\bar{\tau}}_{(\eta)} R_{\bar{\tau}} + \sum_{i=1}^{\tau+\bar{\tau}} \bar{C}_{(\eta)} \bar{A}^{i-1}_{(\eta)} B_{(\eta)} R_{u_{(\eta)}\xi}(\tau - i) + D_{(\eta)} R_{u_{(\eta)}\xi}(\tau) \tag{2.28}$$

and this relation can be transformed into a linear regression form by using the vectorization operator vec$(\cdot)$:

$$\operatorname{vec}\left(R_{z_{(\eta)}\xi}\left(\tau\right)\right) = \underbrace{\begin{bmatrix} \varphi_0\left(\tau\right) \; \varphi_B\left(\tau\right) \; \varphi_D\left(\tau\right) \end{bmatrix}}_{\varphi(\tau)} \begin{bmatrix} \operatorname{vec}\left(R_{\bar{\tau}}\right) \\ \operatorname{vec}\left(B_{(\eta)}\right) \\ \operatorname{vec}\left(D_{(\eta)}\right) \end{bmatrix} \tag{2.29}$$

where

$$\begin{cases} \varphi_0\left(\tau\right) = \mathbf{I}_{pT_c} \otimes \bar{C}_{(\eta)} \bar{A}_{(\eta)}^{\tau+\bar{\tau}} \\ \varphi_B\left(\tau\right) = \sum\limits_{i=1}^{\tau+\bar{\tau}} R'_{u_{(\eta)}\xi}\left(\tau - i\right) \otimes \bar{C}_{(\eta)} \bar{A}_{(\eta)}^{i-1} \\ \varphi_D\left(\tau\right) = R'_{u_{(\eta)}\xi}\left(\tau\right) \otimes \mathbf{I}_{mT_c} \end{cases} \tag{2.30}$$

Finally, denote

$$\mathbf{Y} = \begin{bmatrix} \operatorname{vec}\left(R_{z_{(\eta)}\xi}\left(-\bar{\tau}\right)\right) \\ \operatorname{vec}\left(R_{z_{(\eta)}\xi}\left(-\bar{\tau}+1\right)\right) \\ \vdots \\ \operatorname{vec}\left(R_{z_{(\eta)}\xi}\left(\bar{\tau}\right)\right) \end{bmatrix}, \mathbf{X} = \begin{bmatrix} \varphi\left(-\bar{\tau}\right) \\ \varphi\left(-\bar{\tau}+1\right) \\ \vdots \\ \varphi\left(\bar{\tau}\right) \end{bmatrix}$$

then the estimates of $\bar{R}_{\bar{\tau}}$, $\bar{B}_{(\eta)}$, $\bar{D}_{(\eta)}$ can be obtained by solving the following optimization problem:

$$\min_{R_{\bar{\tau}}, B_{(\eta)}, D_{(\eta)}} \left\| \mathbf{Y} - \mathbf{X} \begin{bmatrix} \operatorname{vec}\left(R_{\bar{\tau}}\right) \\ \operatorname{vec}\left(B_{(\eta)}\right) \\ \operatorname{vec}\left(D_{(\eta)}\right) \end{bmatrix} \right\|_2 \tag{2.31}$$

2.3.2 Pole Location Constraints

It is obvious from (2.9) that matrix $\bar{A}_{(\eta)}$ is a multiple of the monodromy matrix $\Psi\left(l\right)$, whose eigenvalues are fixed and independent of l. Thus, pole location constraints can be incorporated into the estimation of $A_{(\eta)}$ during solving (2.25). Stability constraints are mostly used to avoid unstable estimates due to computational errors. It can be transformed into the following relaxed convex optimization problem:

$$\begin{aligned} \min_{Q,P} \quad & \left\| \mathrm{O}_{\bar{\tau}} Q - \mathbf{R}_{z_{(\eta)}\xi}^{(-\bar{\tau}+1)} \Pi^{\perp}_{\mathbf{R}'_{u_{(\eta)}\xi}} \left(\mathbf{R}_{x_{\eta}\xi} \Pi^{\perp}_{\mathbf{R}'_{u_{(\eta)}\xi}} \right)^{\dagger} P \right\|_F \\ \text{s.t.} \quad & a \otimes P + b \otimes Q + b' \otimes Q' \geq 0, \\ & P = P' > 0, \\ & \operatorname{trace}(P) = n \end{aligned} \tag{2.32}$$

where $a = 1 - \delta, \delta \in [0, 1]$ and $b = \begin{bmatrix} 0 & 1 \\ 0 & 0 \end{bmatrix}$. The estimate of $A_{(\eta)}$ can be found from the optimal $\{\bar{Q}, \bar{P}\}$ via $\bar{A}_{(\eta)} = \bar{Q}\bar{P}^{-1}$ and the eigenvalues of $\bar{A}_{(\eta)}$ are restricted inside the disk with radius $(1 - \delta)$ in the complex plain. More details and other kinds of constraints can be found in [32].

2.4 Topologically Equivalent Realization

An estimated state-space model of (2.9) up to a similarity transformation can be obtained from Section 2.3. The next step is to extract the underlying LTP model described in (2.1) from the estimated time-lifted model of (2.9).

2.4.1 Definition of Topological Equivalence

For the LTI case, only an equivalent realization up to a similarity transformation can be achieved by subspace identification methods. Similarly, we would like to find an equivalent form for (2.1).

Definition 1. *[33] The matrices $\bar{A}(t)$, $\bar{B}(t)$, $\bar{C}(t)$, and $\bar{D}(t)$ are said to be topologically* *equivalent to matrices $A(t)$, $B(t)$, $C(t)$, and $D(t)$ in (2.1) if the following conditions*

$$\begin{cases} \bar{A}\left(t\right)=L\left(t+1\right)A\left(t\right)L^{-1}\left(t\right) \\ \bar{B}\left(t\right)=L\left(t+1\right)B\left(t\right) \\ \bar{C}\left(t\right)=C\left(t\right)L^{-1}\left(t\right) \\ \bar{D}\left(t\right)=D\left(t\right) \end{cases} \tag{2.33}$$

are satisfied, where the matrices $L(t)$ and $L^{-1}\left(t\right)$ in (2.33) *are bounded for all $t\in\mathbb{Z}$.*

The following relation can be also derived using (2.33):

$$\bar{\Phi}\left(t_2,t_1\right)=L\left(t_2\right)\Phi\left(t_2,t_1\right)L^{-1}\left(t_1\right) \tag{2.34}$$

where $\bar{\Phi}\left(t_2,t_1\right)$ is the state-transition matrix corresponding to $\bar{A}(t)$.

If $\{\bar{A}(t),\bar{B}(t),\bar{C}(t),\bar{D}(t)\}$ are periodic matrices of period T, matrix $L(t)$ should also be periodic, i.e., $L\left(kT+l\right)=L\left(l\right)$. Thus, the following equations holds:

$$\begin{cases} \bar{A}\left(1\right)=L\left(2\right)A\left(1\right)L^{-1}\left(1\right) \\ \qquad\vdots \\ \bar{A}\left(T-1\right)=L\left(T\right)A\left(T-1\right)L^{-1}\left(T-1\right) \\ \bar{A}\left(T\right)=L\left(1\right)A\left(T\right)L^{-1}\left(T\right) \\ \bar{A}\left(kT+l\right)=\bar{A}\left(l\right) \end{cases} \tag{2.35}$$

$$\bar{\Psi}\left(l\right)=L\left(l\right)\Psi\left(l\right)L^{-1}\left(l\right) \tag{2.36}$$

and similar properties can be obtained for $\bar{B}(t)$ and $\bar{C}(t)$.

Remark 2. *Given the same input signal and zero initial state, a system with a topologically equivalent form of (2.1) will produce the same output.*

2.4.2 Identification Framework

Assumption 1. *The pair $\{A(t), B(t)\}$ in (2.1) is uniformly controllable [27]. In terms of (2.11), it means there exists a positive integer $c = \delta$ such that $B_{(\eta)}$ is of full row rank n for any $l \in \{1, 2, \cdots, T\}$. Denote the smallest δ as δ_1.*

Assumption 2. *The pair $\{A(t), C(t)\}$ in (2.1) is uniformly observable [27]. In terms of (2.12), it means there exists a positive integer $c = \delta$ such that $C_{(\eta)}$ is of full column rank n for any $l \in \{1, 2, \cdots, T\}$. Denote the smallest δ as δ_2.*

Let Assumption 1,2 hold for the following discussions. Thus, select c such that $c > \max\{\delta_1, \delta_2\}$. Suppose we've already got the realizations of the time-lifted state-space models in (2.9) for $\eta \in \{1, 2, \cdots, T_c\}$. In other words, $\{\bar{A}_{(\eta)}, \bar{B}_{(\eta)}, \bar{C}_{(\eta)}, \bar{D}_{(\eta)}\}$, which are the similarity transformations of the matrices shown in (2.10)–(2.13), can be achieved. This can be done by using LTI system identification methods such as CoBRA method in Section 2.3. Denote these relations as

$$\begin{cases} \bar{A}_{(\eta)} = L\left(l\right) A_{(\eta)} L^{-1}\left(l\right) \\ \bar{B}_{(\eta)} = L\left(l\right) B_{(\eta)} \\ \bar{C}_{(\eta)} = C_{(\eta)} L^{-1}\left(l\right) \\ \bar{D}_{(\eta)} = D_{(\eta)} \end{cases} \tag{2.37}$$

It is enough to choose $\eta = 1, 2, \cdots, T$, in which case $\eta = l$.

Result 1. *Given $\{\bar{A}_{(\eta)}, \bar{B}_{(\eta)}, \bar{C}_{(\eta)}, \bar{D}_{(\eta)}\}$, a topologically equivalent realization $\{\bar{A}(t),\ \bar{B}(t),\ \bar{C}(t), \bar{D}(t)\}$ of (2.1) is given by*

$$\bar{B}\left(l\right) = \bar{B}_{(l+1)}\left(:, p\left(T_c - 1\right) + 1 : pT_c\right) \tag{2.38}$$

$$\bar{C}\left(l\right) = \bar{C}_{(l)}\left(1 : m, :\right) \tag{2.39}$$

$$\bar{A}(l)\,\bar{B}_{(l)}(:, p+1 : pT_c) = \bar{B}_{(l+1)}(:, 1 : p(T_c - 1)) \tag{2.40}$$

$$\bar{C}_{(l+1)}(1 : m(T_c - 1), :)\,\bar{A}(l) = \bar{C}_{(l)}(m+1 : mT_c, :) \tag{2.41}$$

and $\bar{D}(l)$ can be obtained directly from (2.14). Note that the period of these matrices is T.

Proof. Under Assumption 1,2, the result can be proved using the periodicity and relations in (2.10)–(2.13) and (2.37). □

Remark 3. *The interesting point is that there are two approaches to compute matrix $\bar{A}(l), l \in \{1, 2, \cdots, T\}$ as shown in (2.40) and (2.41), denoted as $\bar{A}^{\alpha}(l)$ and $\bar{A}^{\beta}(l)$ respectively. Theoretically speaking, it is sufficient to use only one of them, i.e., $\bar{A}(l) = \bar{A}^{\alpha}(l) = \bar{A}^{\beta}(l)$. However, the advantage of this point is that it can be used to testify if the estimated model is right. Taking computational errors into account, the achieved model is not applicable if $d(\bar{A}^{\alpha}(l), \bar{A}^{\beta}(l)) > \epsilon$, where $d(\cdot, \cdot)$ is a metric and ϵ is the allowance of error.*

2.5 Simulation Results

The efficiency of the proposed CoBRA method for identification of an LTP system is illustrated by means of a simulation example. Consider an LTP system described by a state-space model in (2.1) with the following parameters:

$$\begin{cases} A\left(1\right) = \begin{bmatrix} 1.55 & -0.6 \\ 1 & 0 \end{bmatrix}, A\left(2\right) = \begin{bmatrix} 1.78 & -0.93 \\ 1 & 0 \end{bmatrix} \\ B\left(1\right) = B\left(2\right) = \begin{bmatrix} 1 \\ 0 \end{bmatrix} \\ C\left(1\right) = \begin{bmatrix} 1 & -0.9 \end{bmatrix}, C\left(2\right) = \begin{bmatrix} 1 & -0.85 \end{bmatrix} \\ D\left(1\right) = D\left(2\right) = 0 \end{cases}$$

where the period is $T = 2$. The measured output is $z(t) = y(t) + v(t)$, where $v(t)$ is a colored noise described by

$$v\left(t\right) = \frac{q^{-1}}{1 + 0.3q^{-1}} e\left(t\right)$$

in which $e(t)$ is zero mean Gaussian white noise.

For excitation a PRBS signal is used as an input signal with the amplitude changing randomly between -1 and 1. For the noise $v(t)$ perturbing the output, the variance of $e(t)$ is chosen to be 0.3^2. Furthermore, c is chosen as $c = 2$ so that $T_c = 2T$ for the time-lifted state-space model in (2.9), which is a four-input four-output LTI system from the augmented input $u_{(\eta)}(k)$ to the output $y_{(\eta)}(k)$. We use the CoBRA in Section 2.3 to get an estimate $\{\bar{A}_{(\eta)}, \bar{B}_{(\eta)}, \bar{C}_{(\eta)}, \bar{D}_{(\eta)}\}$ for $\eta = 1, 2, \cdots, T$ in (2.9), where the instrumental variable is chosen as $\xi(t) = u_{(\eta)}(k)$.

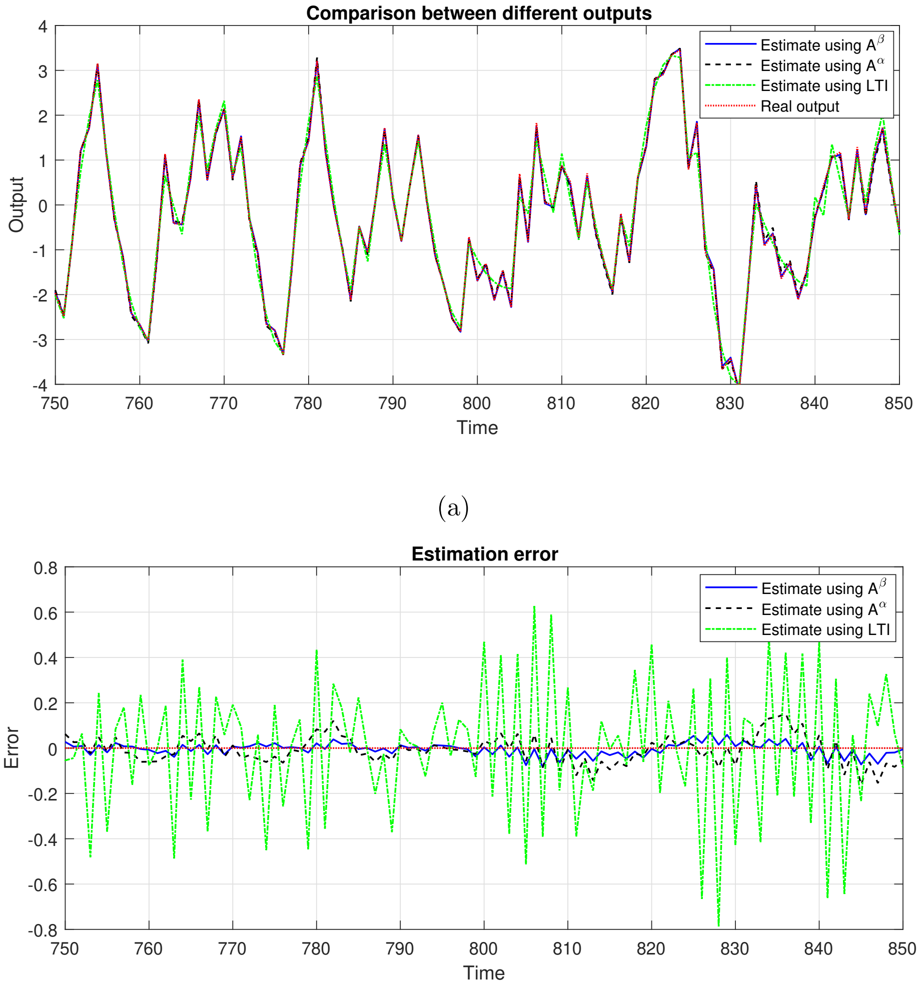
The values of the covariance functions are computed by using their sample counterparts with data length $N = 1500$. Note that the original data length is $4N = 6000$. Other parameters are set as $\bar{\tau} = 50$, $r = 30$ in (2.17), and $n = 2$ in (2.27). Pole location constraints are not implemented in this simulation. Finally, a topologically equivalent realization $\{\bar{A}(l), \bar{B}(l), \bar{C}(l), \bar{D}(l)\}$ with $l \in \{1, 2, \cdots, T\}$ can be obtained by using Result 1. It can be also observed that the parameters of the given LTP system change mildly. Thus, it is reasonable to consider an LTI model for identification. This can be done directly by using the CoBRA and included here for comparison purposes.

From our experience, choosing $\bar{A}(l) = \bar{A}^{\beta}(l)$ computed in (2.41) performs better. For the purpose of a good demonstration, a single simulation experiment with the above setting is conducted. The computed estimates $\bar{A}(l)$ are shown as below:

$$\begin{cases} \bar{A}^{\alpha}(1) = \begin{bmatrix} 0.7073 & -0.1857 \\ -0.2679 & -0.9537 \end{bmatrix} \\ \bar{A}^{\beta}(1) = \begin{bmatrix} 0.7054 & -0.1685 \\ -0.2665 & -0.9525 \end{bmatrix} \end{cases} \tag{2.42}$$

$$\begin{cases} \bar{A}^{\alpha}(2) = \begin{bmatrix} 0.7417 & 0.3651 \\ 0.3241 & -0.8709 \end{bmatrix} \\ \bar{A}^{\beta}(2) = \begin{bmatrix} 0.7423 & 0.3630 \\ 0.3193 & -0.8864 \end{bmatrix} \end{cases} \tag{2.43}$$

It can be observed that $\bar{A}^{\alpha}(l) \approx \bar{A}^{\beta}(l)$, $l \in 1, 2, \cdots, T$. Thus, we conclude that the original periodic dynamics are captured and the estimated results are acceptable. The corresponding simulated outputs and output errors are illustrated in Fig. 2.1. The accuracy of both approaches is acceptable, with the method in (2.41) leading to slightly smaller output errors. Note that, without taking the periodic dynamics into account, the (average) variance of the output error from an LTI model is larger than the variance of the output error obtained from the estimated LTP models.



(a)

(b)

Figure 2.1: The simulated outputs of the estimated topologically equivalent LTP models via CoBRA. The output error from an LTI model estimated via CoBRA is also illustrated with a larger output error variance.

Theoretically, $\bar{A}^{\alpha}(l)$ should be equal to $\bar{A}^{\beta}(l)$. In practice, noise and computational errors exist, and estimation of $A(l)$ using (2.41) is recommended. Its accuracy just depends on the estimate $\bar{C}_{(\eta)}$. However, estimating $A(l)$ through the method in (2.40) is based on the estimate $\bar{B}_{(\eta)}$, which is further based on the estimate $\{\bar{A}_{(\eta)}, \bar{C}_{(\eta)}\}$ as shown in (2.29). Thus, computational errors may be accumulated.

2.6 Conclusion

In this chapter, a novel time domain parameters estimation method for LTP models has been proposed based on the CoBRA. The method is based on time-lifted signals, but these signals may be of a high dimension, so a large amount of data points may be required. The CoBRA reduces the storage requirements and emphasizes the identification of the deterministic part of the system. Two approaches to estimating the state matrices of an LTP system are introduced and provide us with a way of determining if the estimate captures the periodic dynamics of the LTP system. The proposed method is efficient in terms of the simulated outputs as illustrated in the simulation example.

This chapter is based on the following paper that was published:

Y. Hu and R.A. de Callafon, "Covariance Based Realization Algorithm for identification of linear time periodic system," in *Proceedings of 2018 Annual American Control Conference (ACC)*, Wisconsin Center, Milwaukee, USA, Jun. 27–29, 2018, pp. 1102–1107.

Chapter 3

Closed-Loop System Identification via CoBRA

The Covariance Based Realization Algorithm (CoBRA), a branch of subspace methods, enables the estimation of multivariable models using a large number of data points due to the use of finite size covariance matrices. The covariance pre-processing allows the CoBRA to ignore any (high-order) noise dynamics and focus on the estimation of the low-order deterministic model. However, subspace methods and the CoBRA in particular are not maximum-likelihood methods. In this chapter, an in-depth study on the statistical behavior of the noise effects is conducted. A method to reduce the variance of an estimate obtained by the CoBRA is proposed via the choice of optimal row and column weighting matrices. For closed-loop implementation of the CoBRA, a two stage technique is adopted with the estimate on an intermediate instrument. At the first stage, a least-length perturbation of a scalar accuracy function is used to obtain an analytic solution for the optimal weighting matrices. The instruments produced by the first stage are then used for the identification at the second stage to extract the plant model.

3.1 Introduction

In this chapter, we focus on the closed-loop identification of multi-input multi-output (MIMO) discrete-time LTI systems. Popular techniques for the identification of MIMO systems are subspace methods, which are well-known to have a computational advantage over prediction-error methods (PEM) [10, 11]. It should be noted that subspace methods will identify a state-space model up to within a similarity transformation. In a closed-loop setting, the correlation of the input signals with the process and measurement noise signals is a well-known issue that hampers consistent parameter estimation when noise model estimation is omitted. Solutions to circumvent this issue have been traditionally classified as direct, indirect, and joint input-output methods [34]. A comprehensive summary of recent direct subspace methods can be found in [12]. Those methods are based on a predictor form and assume that the bias is negligible for an appropriate past window size, making the methods equivalent to identification of a vector-ARX structure. Direct subspace methods have less restrictions on the controller, but they are seeking an innovation model, which may be of high-order since the noise dynamics are included. Joint input-output methods based on Orthogonal Decomposition (ORT) and Canonical Correlation Analysis (CCA) are proposed in [11] and followed by a model order reduction in a closed-loop setting.

In many cases, one is more interested in the estimation of a low-order deterministic model. Thus, an alternative approach is based on the introduction of an instrument, which is correlated with the closed-loop data, but uncorrelated with the noise on the data. The use of an instrument allows one to ignore noise dynamics and focus on the deterministic model identification. With the aid of the instrumental variable (IV), the CoBRA has been demonstrated to be efficient in identifying low-order deterministic models [28, 35, 29]. Furthermore, the use of covariance functions allows pre-processing of the MIMO closed-loop data and restricts the column number of Hankel data matrices to a level $O(1)$ instead of $O(N)$ as in traditional subspace methods, where N is the data length. In this sense, the CoBRA can cope with a large amount of data without extensive storage requirement and enable consistent model identification in the presence of colored closed-loop noise. Note that the use of covariance functions for MIMO system identification is not new and has been investigated in other works [36, 37, 38], but they refrain from using an IV to reduce the noise effects.

In this chapter, the CoBRA is incorporated into a two stage indirect method in a closed-loop setting. At the first stage, the CoBRA is applied to perform a joint input-output identification, which is actually an open-loop identification of the whole closed-loop system. At the second stage, the CoBRA is applied again to extract the plant model from the estimated noise-free input and output signals, which are the instruments produced by the closed-loop model from the first stage.

It is obvious that the estimation accuracy of the first stage plays a very important role in identifying an accurate model of the plant. Tuning the IV can be a reasonable approach to achieving better performance. In [39], the optimal extended instrumental variable (EIV) estimation was proposed to obtain an optimal estimation in terms of parameter covariance. In [40], the optimal refined instrumental variable (RIV) estimation, which is actually equivalent to the optimal EIV, was described from the aspect of maximum likelihood optimization.

Both optimal EIV and RIV are tackling Box-Jenkins-like models. In [41], a combined instrumental variable-weighting subspace fitting (IV-WSF) approach was proposed to estimate a deterministic transfer function model based on an IV matrix (also named instrumental-product matrix). The parameters were estimated through optimally extracting the range or null space of the Sylvester matrix within the IV matrix.

In the CoBRA framework, the covariance function of a sampled signal and IV can be seen as an IV matrix. However, the CoBRA stacks different IV matrices to construct Hankel matrices within a state-space model and extracts the range space of the extended observability matrix for parameter estimation. In addition, for MIMO state-space models, an optimal column weighting matrix was proposed for MOESP-like (Multivariable Output-Error state SPace identification) subspace methods in [42] by using a similar idea from [39, 43].

This chapter analyzes a two stage CoBRA for closed-loop MIMO system identification. Under mild assumptions, a statistical analysis of the noise effect reduction by the IV is formulated on the basis of the properties of a sum of stochastic variables. An approach is then provided to reduce the variance of an estimate obtained by the CoBRA via the choice of optimal row and column weighting matrices. For the closed-loop implementation of the CoBRA, a two stage procedure is proposed with an estimate obtained from an intermediate instrument. At the first stage, a least-length perturbation of a scalar accuracy function is used to obtain an analytic solution for the optimal weighting matrices. The resulting instruments produced by the first stage are used for the identification at the second stage to formulate a plant model.

The remaining part is organized as follows. Section 3.2 presents the objective of the closed-loop identification and an introduction of the two stage technique and CoBRA. A statistical analysis of a stochastic vector signal with bounded covariance is given in Section 3.3. In Section 3.4, an optimization problem is formulated that can be used to tune the IV to achieve an optimal variance reduction. A least-length perturbation is also introduced to achieve an analytic solution. The two stage CoBRA is summarized in Section 3.5, where in Section 3.6, a closed-loop simulation example is given to demonstrate the efficiency of the proposed method. Section 3.7 summarizes this chapter.

3.2 CoBRA in a Closed-Loop Setting

3.2.1 Closed-Loop Identification of MIMO Systems

Consider a LTI minimal system $G(q^{-1})$ described by a state-space model:

$$\begin{cases} \mathbf{x}\left(t+1\right)= A\mathbf{x}\left(t\right)+B\mathbf{u}\left(t\right) \\ \quad \mathbf{y}\left(t\right)= C\mathbf{x}\left(t\right)+D\mathbf{u}\left(t\right)+\mathbf{v}\left(t\right) \end{cases} \tag{3.1}$$

where $A\in\mathbb{R}^{n\times n}$, $B\in\mathbb{R}^{n\times p}$, $C\in\mathbb{R}^{m\times n}$, $D\in\mathbb{R}^{m\times p}$, and $\mathbf{v}\left(t\right)$ is a zero mean stochastic noise signal with unknown spectral properties, but assumed to be uncorrelated with the input. Note that if we are only interested in modeling the deterministic part of the system, any process noise on the state equation can be ignored here since it can be incorporated into $\mathbf{v}\left(t\right)$. The noisy output can also be represented as $\mathbf{y}\left(t\right)=\bar{\mathbf{y}}\left(t\right)+\mathbf{v}\left(t\right)$, where $\bar{\mathbf{y}}\left(t\right)$ is the noise-free part.

In a closed-loop setting as shown in Fig. 3.1, the input can be represented as

$$\mathbf{u}\left(t\right)=\mathbf{r}_1\left(t\right)-C\left(q^{-1}\right)\left[\mathbf{y}\left(t\right)-\mathbf{r}_2\left(t\right)\right]=\bar{\mathbf{u}}\left(t\right)+\mathbf{w}\left(t\right) \tag{3.2}$$

where the controller $C\left(q^{-1}\right)$ is assumed to be LTI, $\{\mathbf{r}_1(t),\mathbf{r}_2(t)\}$ are the reference signals and

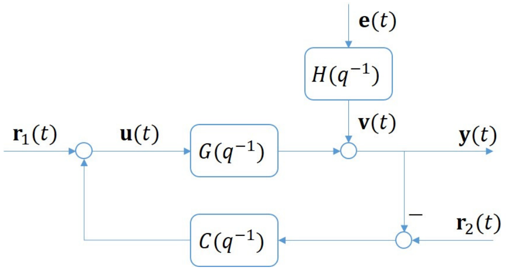


Figure 3.1: Block diagram of a closed-loop system.

$\bar{\mathbf{u}}\,(t)$ and $\mathbf{w}\,(t)$ are respectively the deterministic and stochastic part of $\mathbf{u}(t)$. Obviously, the noise part $\mathbf{w}(t)$ of the input is correlated with $\mathbf{v}(t)$ in the output, which is a common issue in a closed-loop identification setting.

The objective is to identify, from the experimental data $\{\mathbf{r}_1(t), \mathbf{r}_2(t), \mathbf{u}(t), \mathbf{y}(t)\}$, the deterministic parameters $\{A, B, C, D\}$ in (3.1) up to within a similarity transformation, regardless of the spectral properties of the noise $\mathbf{v}(t)$.

To cope with the correlation of $\{\mathbf{u}\,(t)\,, \mathbf{y}\,(t)\}$ with $\mathbf{v}(t)$ during a closed-loop identification, a two stage technique is used in this chapter. The first stage is to identify the deterministic model from the reference signals $\{\mathbf{r}_1\,(t)\,, \mathbf{r}_2\,(t)\}$ to the noisy joint input-output signals $\{\mathbf{u}\,(t)\,, \mathbf{y}\,(t)\}$. This model can be used to produce an estimate of the noise-free version $\{\bar{\mathbf{u}}\,(t)\,, \bar{\mathbf{y}}\,(t)\}$, known as instruments. The second stage is to identify the deterministic part of the system described in (3.1) by using $\{\bar{\mathbf{u}}\,(t)\,, \bar{\mathbf{y}}\,(t)\}$ from the first stage.

3.2.2 Realization through CoBRA

CoBRA can serve as a powerful tool to extract the underlying low-order deterministic state-space model. In order to handle deterministic and stochastic signals simultaneously, a general (cross) covariance function of two zero mean signals $\mathbf{s}_1(t)$ and $\mathbf{s}_2(t)$ is denoted as:

$$\begin{aligned} R_{s_1 s_2}(\tau) &= \bar{E}\left\{\mathbf{s}_1(t)\,\mathbf{s}_2(t-\tau)^{\mathrm{T}}\right\} \\ &= \lim_{N\to+\infty} \tfrac{1}{N} \sum_{t=0}^{N-1} E\left\{\mathbf{s}_1(t)\,\mathbf{s}_2(t-\tau)^{\mathrm{T}}\right\} \end{aligned} \tag{3.3}$$

whose sample counterpart is

$$\hat{R}_{s_1 s_2}(\tau) = \frac{1}{N} \sum_{t=0}^{N-1} \mathbf{s}_1(t)\,\mathbf{s}_2(t-\tau)^{\mathrm{T}} \tag{3.4}$$

Note that $\bar{E}\{\cdot\}$ is called the extended expectation associated with the normal expectation operator $E\{\cdot\}$. Set $\mathbf{s}_i(t) = 0$ for $t \notin [0, N-1]$. A bounded IV $\xi(t)$ is introduced to eliminate the noise effect and satisfying

(a) $\xi(t) \in \mathbb{R}^{p\times 1}$ is correlated with $\{\mathbf{u}(t), \mathbf{y}(t)\}$,

(b) $\xi(t) \in \mathbb{R}^{p\times 1}$ is uncorrelated with $\mathbf{v}(t)$.

Note that $R_{v\xi}(\tau) = \mathbf{0}$ and $R_{y\xi}(\tau) = R_{\bar{y}\xi}(\tau)$ in the ideal case if $\mathbf{v}(t)$ is assumed to be of zero mean. This means that computing the covariance functions could be used to eliminate the noise effect. However, the sample covariance function in (3.4) is used in practice when finite number N of sampled experimental data is available. Thus, the noise term will not vanish. Without too much effort, we can obtain the following matrix equation:

$$\hat{\mathbf{R}}_{y\xi} = \mathrm{O}_r \hat{\mathbf{R}}_{x\xi} + \Psi \hat{\mathbf{R}}_{u\xi} + \hat{\mathbf{R}}_{v\xi} \tag{3.5}$$

where

$$\begin{aligned}
&\mathrm{O}_r = \begin{bmatrix} C^{\mathrm{T}} & (CA)^{\mathrm{T}} & \cdots & (CA^{r-1})^{\mathrm{T}} \end{bmatrix}^{\mathrm{T}} \\
&\hat{\mathbf{R}}_{x\xi} = \text{Hankel}\left(\hat{R}_{x\xi}(\tau), 1, \tau_1, \tau_2 - r + 1\right) \\
&\hat{\mathbf{R}}_{f\xi} = \text{Hankel}\left(\hat{R}_{f\xi}(\tau), r, \tau_1, \tau_2\right), f \in \{y, u, v\} \\
&\text{Hankel}(g(\tau), \tau_0, \tau_1, \tau_2) = \begin{bmatrix} g(\tau_1) & g(\tau_1+1) & \cdots & g(\tau_2-\tau_0+1) \\ g(\tau_1+1) & g(\tau_1+2) & \cdots & g(\tau_2-\tau_0+2) \\ \vdots & \vdots & & \vdots \\ g(\tau_1+\tau_0-1) & g(\tau_1+\tau_0) & \cdots & g(\tau_2) \end{bmatrix}
\end{aligned}$$

and Ψ is a Toeplitz matrix with unknown Markov parameters. A high-quality estimation of the extended observability matrix O_r is the most important part in subspace methods. Note that r, satisfying $r > n$, is the number of block rows, and τ_1, τ_2 are selected so that $-\bar{\tau} \leq \tau_1 < \tau_2 \leq \bar{\tau}$ for some finite $\bar{\tau} > 0$ and the Hankel matrix of the input signal are of full row rank. Then, the following matrix equation can be obtained by introducing an orthogonal projection:

$$\hat{\mathbf{R}}_{y\xi}\Pi^{\perp}_{\hat{\mathbf{R}}^{\mathrm{T}}_{u\xi}} = \mathrm{O}_r\hat{\mathbf{R}}_{x\xi}\Pi^{\perp}_{\hat{\mathbf{R}}^{\mathrm{T}}_{u\xi}} + \hat{\mathbf{R}}_{v\xi}\Pi^{\perp}_{\hat{\mathbf{R}}^{\mathrm{T}}_{u\xi}} \tag{3.6}$$

where $\Pi^{\perp}_{\hat{\mathbf{R}}^{\mathrm{T}}_{u\xi}} = \mathbf{I} - \hat{\mathbf{R}}^{\mathrm{T}}_{u\xi}\left[\hat{\mathbf{R}}_{u\xi}\hat{\mathbf{R}}^{\mathrm{T}}_{u\xi}\right]^{-1}\hat{\mathbf{R}}_{u\xi}$ and the Toeplitz matrix Ψ with unknown system information is removed. Note that $\hat{\mathbf{R}}_{x\xi}\Pi^{\perp}_{\hat{\mathbf{R}}^{\mathrm{T}}_{u\xi}}$ is also assumed to be with full row rank equal to the system order n.

The remaining procedures are similar to classical (MOESP [44]) subspace methods: the first step is to perform SVD on $\hat{\mathbf{R}}_{y\xi}\Pi^{\perp}_{\hat{\mathbf{R}}^{\mathrm{T}}_{u\xi}}$ and then extract the range space of the extended observability matrix O_r, which leads to $\hat{\mathrm{O}}_{\mathrm{r}} = \mathrm{O}_r\mathbf{T}^{-1}$ with $\mathbf{T}$ being a nonsingular similarity transformation matrix and the estimate of parameters $\{\hat{A}, \hat{C}\}$ through the shift-invariance structure of $\hat{\mathrm{O}}_r$; the second step is to estimate parameters $\{\hat{B}, \hat{D}\}$ (and the initial state) via a linear regression optimization by using the estimated $\{\hat{A}, \hat{C}\}$ from the first step. Note that the estimated $\{\hat{A}, \hat{B}, \hat{C}, \hat{D}\}$ are related to the original $\{A, B, C, D\}$ in (3.1) by a similarity transformation $\mathbf{T}$ and they are actually equivalent from the perspective of input-output description. This is a common feature for subspace methods.

In the ideal case of a perfect covariance estimate using (3.3), consistent parameter estimation can be achieved. However, the noise effect of the residue $\hat{\mathbf{R}}_{v\xi}\Pi^{\perp}_{\hat{\mathbf{R}}^{\mathrm{T}}_{u\xi}}$ may not be eliminated even if the noise signals are ideally stationary, independent and identically distributed (i.i.d.), and of zero mean. This is due to the use of the sample counterpart in (3.4) and the limited number of data points in practice. Note that (3.4) is computing the average of the terms under summation. For an i.i.d. stochastic signal, we can resort to the Law of Large Numbers for the properties of its average over the time line. Under the structure of CoBRA, it's interesting and necessary to investigate the properties of the average of the product of a more general noise signal and IV.

3.3 A Statistical Analysis of A Stochastic Vector Signal with Bounded Covariance

Since we are computing the covariance functions using (3.4) in practice, the noise term $\hat{\mathbf{R}}_{v\xi}$ in (3.6) is actually composed of the average of the product of $\mathbf{v}(t)$ and $\xi(t-\tau)$ over a given time interval. The properties of this average for a finite data length N will provide us an insight into how to design or tune an IV appropriately to reduce the noise effect contained in $\hat{\mathbf{R}}_{v\xi}\Pi^{\perp}_{\hat{\mathbf{R}}^{\mathrm{T}}_{u\xi}}$ as much as possible. It leads to deriving an optimal CoBRA in

Section 3.4.

Proposition 1. *Let* $\left\{\mathbf{n}'(t) \in \mathbb{R}^{m'}, t \in \mathbb{N}, m' \in \mathbb{N}\right\}$ *be an arbitrary stochastic signal and* $S_N^i = \sum_{t=i}^{N} \mathbf{n}'(t)$ *a partial sum. Denote* $S_N = S_N^1$*. If there exist* $c > 0, \delta < \frac{3}{2}$ *such that the variance of any* S_N^i *satisfies*

$$\operatorname{Tr}\left\{\operatorname{Var}\left\{S_N^i\right\}\right\} \leq c(N-i+1)^{\delta} \tag{3.7}$$

then

$$\frac{S_N}{N} \to \mu_N, \quad w.p.1, \quad \text{as} \quad N \to +\infty \tag{3.8}$$

where $\mu_N = \frac{1}{N}\sum_{t=1}^{N} E\left\{\mathbf{n}'(t)\right\}$ *and* $w.p.1$ *means with probability 1.*

Proof. One can subtract μ_N from $\frac{S_N}{N}$ beforehand. Thus, we can assume that $\mu_N = \mathbf{0}$ in the proof without loss of generality. Also, if one can prove that

$$\left(\frac{S_N}{N}\right)_k \to 0, \quad w.p.1, \quad \text{as} \quad N \to +\infty \tag{3.9}$$

for all $k \in \{1, 2, \cdots, m'\}$, then we can prove the result in (3.8). The idea here is similar to the proof of Theorem 2B.1 in [2]. To apply Borel-Cantelli lemma to prove the convergence of $\left\{\left(\frac{S_N}{N}\right)_k\right\}$, one could first sparsify the sequence by considering the subsequence $\left\{\left(\frac{S_{N^2}}{N^2}\right)_k\right\}$. For $\forall \varepsilon > 0$, the following relation can be obtained via Chebyshev's inequality and condition (3.7)

$$\mathrm{P}\left\{\left|\left(\frac{S_{N^2}}{N^2}\right)_k\right| > \varepsilon\right\} \leq \frac{1}{\varepsilon^2}\operatorname{Var}\left\{\left(\frac{S_{N^2}}{N^2}\right)_k\right\} \leq \frac{c}{\varepsilon^2}\frac{N^{2\delta}}{N^4}$$

where $\mathrm{P}\{\cdot\}$ denotes the probability of a given statement. Thus,

$$\sum_{N=1}^{+\infty} \mathrm{P}\left\{\left|\left(\frac{S_{N^2}}{N^2}\right)_k\right| > \varepsilon\right\} \leq \frac{c}{\varepsilon^2}\sum_{N=1}^{+\infty}\frac{1}{N^{4-2\delta}} < +\infty$$

for $\delta < \frac{3}{2}$, which, via Borel-Cantelli's lemma, further implies that

$$\left(\frac{S_{N^2}}{N^2} \right)_k \to 0, \quad w.p.1, \quad \text{as} \quad N \to +\infty \tag{3.10}$$

Now, we come back to the original sequence. Obviously,

$$\begin{aligned} \sup_{N^2 \leq i \leq (N+1)^2} \left| \left(\tfrac{S_i}{i} \right)_k \right| &= \left| \left(\tfrac{S_{\hat{N}}}{\hat{N}} \right)_k \right| \\ &\leq \tfrac{1}{\hat{N}} \left| \left(S_{N^2} \right)_k \right| + \tfrac{1}{\hat{N}} \left| \left(S_{(N+1)^2}^{N^2+1} \right)_k \right| \end{aligned}$$

where $\hat{N}$ is the solution to the first equality. In addition, the first term on the right of the inequality approaches 0 w.p.1 in terms of (3.10). With the same technique as above, one can easily prove that the second term also approaches 0 w.p.1. Thus,

$$\sup_{N^2 \leq i \leq (N+1)^2} \left| \left(\frac{S_i}{i} \right)_k \right| \to 0, \quad w.p.1, \quad \text{as} \quad N \to +\infty \tag{3.11}$$

which proves (3.9). □

Note that only a sufficient condition is given and it is not required for $\mathbf{n}'(t)$ to be stationary or white. It only assumes that the variance of the sum does not increase too fast. In other words, the covariance between $\mathbf{n}'(t_1)$ and $\mathbf{n}'(t_2)$ is bounded and should go to zero not too slowly as $|t_1 - t_2| \to +\infty$. This assumption is mild but abstract. If we further assume that the stochastic signal $\mathbf{n}'(t)$ is stationary, the variance of $\frac{S_N}{N}$ can be written in a closed form.

Corollary 1. *Let $\left\{ \mathbf{n}'(t) \in \mathbb{R}^{m'}, t \in \mathbb{N}, m' \in \mathbb{N} \right\}$ be a stationary stochastic signal described by $\mathbf{n}'(t) = H(q^{-1}) \mathbf{e}(t)$, where $\mathbf{e}(t) \in \mathbb{R}^{m'}$ is a white noise with zero mean and variance $\mathbf{I}$, and $H(q^{-1})$ is a stable LTI filter with distinct poles $\{a_k, k = 1, 2, \cdots, m_v\}$. Then,*

$$\frac{S_N}{N} \to \mathbf{0}, \quad w.p.1, \quad \text{as} \quad N \to +\infty \tag{3.12}$$

and the variance $\mathrm{Var}\left\{\frac{S_N}{N}\right\}$ *satisfies*

$$\begin{aligned} & \mathrm{Tr}\left\{\mathrm{Var}\left\{\frac{S_N}{N}\right\}\right\} \\ = & \frac{1}{N}\left(\gamma_0 - 2\sum_{k=1}^{m_v}\frac{\gamma_k}{1-a_k^{-1}}\right) + \frac{2}{N^2}\sum_{k=1}^{m_v}\frac{\gamma_k}{1-a_k^{-1}}\frac{1-a_k^N}{1-a_k} \\ = & O\left(\frac{1}{N}\right) \end{aligned} \tag{3.13}$$

where $\{\gamma_k, k = 0, 1, \cdots, m_v\}$ *are some fixed constants related to a given* $H\left(q^{-1}\right)$*.*

Proof. It is obvious that

$$\mathbf{n}'(t) = D\mathbf{e}(t) + \sum_{l=1}^{+\infty} CA^{l-1}B\mathbf{e}(t-l)$$

where $\{A, B, C, D\}$ is a state-space realization of $H\left(q^{-1}\right)$ and assume without loss of generality that A is in a diagonal form. To reduce the number of unnecessary denotations, we use the same representation as in (3.1) for the system matrices of $H\left(q^{-1}\right)$ without causing confusion. Since $H\left(q^{-1}\right)$ is a stable filter, the variance of $\mathbf{n}'(t)$ should be finite, i.e., $\mathrm{Tr}\left\{\mathrm{Var}\left\{\mathbf{n}'(t)\right\}\right\} = \gamma_0$ for some constant γ_0. Denote $\mathrm{Cov}\{\cdot,\cdot\}$ as the covariance function of two stochastic vector signals. For any $\tau > 0$, the covariance matrix is as follows:

$$\mathrm{Cov}\left\{\mathbf{n}'(t), \mathbf{n}'(t-\tau)\right\} = CA^{\tau-1}BD^{\mathrm{T}} + \sum_{l=0}^{+\infty} CA^{\tau+l}BB^{\mathrm{T}}A^{l}C^{\mathrm{T}}.$$

Assume that the eigenvalues of A satisfy the condition $|a_1| < |a_2| < \cdots < |a_{m_v}|$. This assumption will not affect the form of the final result when conjugate complex poles are taken into account. Since B, C, and D are fixed, the components of $CA^{\tau-1}BD^{\mathrm{T}}$ will be the linear combination of $\left\{a_k^{\tau-1}, k = 1, 2, \cdots, m_v\right\}$. Similarly, the components of $CA^{\tau+l}BB^{\mathrm{T}}A^{l}C^{\mathrm{T}}$ will be the linear combination of $\left\{a_k^{\tau+l}a_h^{l}, k, h = 1, 2, \cdots, m_v\right\}$, where $a_k^{\tau+l}a_h^{l} = a_k^{\tau}(a_k a_h)^{l}$. Thus, there exist certain constants $\alpha_k, \beta_{k,h}$ such that

$$\begin{aligned}
&\mathrm{Tr}\left\{\mathrm{Cov}\left\{\mathbf{n}'\left(t\right),\mathbf{n}'\left(t-\tau\right)\right\}\right\}\\
=\;& \mathrm{Tr}\left\{CA^{\tau-1}BD^{\mathrm{T}}\right\}+\sum_{l=0}^{+\infty}\mathrm{Tr}\left\{CA^{\tau+l}BB^{\mathrm{T}}A^{l}C^{\mathrm{T}}\right\}\\
=\;& \sum_{k=1}^{m_v}\alpha_k a_k^{\tau-1}+\sum_{l=0}^{+\infty}\sum_{k=1}^{m_v}\sum_{h=1}^{m_v}\beta_{k,h}a_k^{\tau}(a_k a_h)^l\\
=\;& \sum_{k=1}^{m_v}\underbrace{\left(\frac{\alpha_k}{a_k}+\sum_{h=1}^{m_v}\frac{\beta_{k,h}}{1-a_k a_h}\right)}_{\gamma_k}a_k^{\tau}
\end{aligned}$$

Note that $|a_k| < 1$ for a stable filter and

$$\begin{aligned}
&\mathrm{Tr}\left\{\mathrm{Var}\left\{\tfrac{S_N}{N}\right\}\right\}\\
=\;& \mathrm{Tr}\left\{\frac{\sum\limits_{i=1}^{N}\sum\limits_{j=1}^{N}\mathrm{Cov}\{\mathbf{n}'(i),\mathbf{n}'(j)\}}{N^2}\right\}\\
=\;& \frac{\sum\limits_{i=1}^{N}\sum\limits_{j=1}^{N}\mathrm{Tr}\{\mathrm{Cov}\{\mathbf{n}'(i),\mathbf{n}'(j)\}\}}{N^2}\\
=\;& \tfrac{1}{N^2}\left(N\gamma_0+2\sum_{i=1}^{N}\sum_{j=1}^{i-1}\sum_{k=1}^{m_v}\gamma_k a_k^{i-j}\right)\\
=\;& \tfrac{1}{N^2}\left(N\gamma_0+2\sum_{k=1}^{m_v}\gamma_k\sum_{i=1}^{N}a_k^{i}\sum_{j=1}^{i-1}a_k^{-j}\right)
\end{aligned}\qquad(3.14)$$

which can prove (3.13) after a few calculations. Since $\mathbf{n}'\left(t\right)$ is stationary and of zero mean, (3.13) also implies that (3.7) holds, which proves (3.12). □

In order to cope with the noise part $\hat{\mathbf{R}}_{v\xi}\Pi^{\perp}_{\hat{\mathbf{R}}^{\mathrm{T}}_{u\xi}}$, it is necessary to investigate the properties of the cross covariance function between a stochastic signal and a bounded deterministic signal.

Proposition 2. *Let* $\left\{\varphi'\left(t\right)\in\mathbb{R}^{p'}, t\in\mathbb{N}, p'\in\mathbb{N}\right\}$ *be a bounded deterministic signal such that* $\left\|\varphi'\left(t\right)\right\|_2 \le c < \infty$ *and* $\varphi'(t)$ *is uncorrelated with the* $\mathbf{n}'\left(t\right)$ *in Corollary 1. Denote* $S_{\varphi',N}\left(\tau\right) = \sum\limits_{i=1}^{N}\mathbf{n}_{\varphi'}\left(i,\tau\right)$*, where* $\mathbf{n}_{\varphi'}\left(i,\tau\right) = \mathrm{vec}\left\{\mathbf{n}'\left(i\right)\varphi'(i-\tau)^{\mathrm{T}}\right\}$*. Then,*

$$\frac{S_{\varphi',N}\left(\tau\right)}{N}\rightarrow\mathbf{0},\quad w.p.1,\quad as\ \ N\rightarrow+\infty \qquad (3.15)$$

and the variance $\mathrm{Var}\left\{\frac{S_{\varphi',N}(\tau)}{N}\right\}$ *satisfies*

$$\begin{aligned} & \mathrm{Tr}\left\{\mathrm{Var}\left\{\frac{S_{\varphi',N}(\tau)}{N}\right\}\right\} \\ = \;& \sum_{l=1}^{p'} \frac{1}{N^2}\left(\sum_{i=1}^{N} \mathrm{w}_l^{i,i,\tau}\gamma_0 + 2\sum_{i=1}^{N}\sum_{j=1}^{i-1} \mathrm{w}_l^{i,j,\tau} \sum_{k=1}^{m_v} \gamma_k a_k^{i-j}\right) \\ = \;& O\left(\frac{1}{N}\right) \end{aligned} \tag{3.16}$$

where $\mathrm{w}_l^{i,j,\tau} = \varphi_l'(i-\tau)\,\varphi_l'(j-\tau)$ *is bounded weighting and* $\{\gamma_k, k = 0, 1, \cdots, m_v\}$ *are some fixed constants as in Corollary 1.*

Proof. The covariance function of $\mathbf{n}_{\varphi_l'}(t,\tau)$ is

$$\begin{aligned} & \mathrm{Cov}\left\{\mathbf{n}_{\varphi_l'}(i,\tau), \mathbf{n}_{\varphi_l'}(j,\tau)\right\} \\ = \;& E\left\{\mathbf{n}'(i)\,\varphi_l'(i-\tau)\,\varphi_l'(j-\tau)\,\mathbf{n}'(j)^{\mathrm{T}}\right\} \\ = \;& \mathrm{w}_l^{i,j,\tau}\mathrm{Cov}\left\{\mathbf{n}'(i), \mathbf{n}'(j)\right\} \end{aligned}$$

and we have

$$\begin{aligned} & \mathrm{Tr}\left\{\mathrm{Var}\left\{\frac{S_{\varphi',N}(\tau)}{N}\right\}\right\} \\ = \;& \sum_{l=1}^{p'} \mathrm{Tr}\left\{\frac{\sum_{i=1}^{N}\sum_{j=1}^{N} \mathrm{w}_l^{i,j,\tau}\mathrm{Cov}\{\mathbf{n}'(i),\mathbf{n}'(j)\}}{N^2}\right\} \end{aligned} \tag{3.17}$$

Similarly as in (3.14), the proof of (3.16) is straightforward. Note that the weighting $\mathrm{w}_l^{i,j,\tau}$ is bounded, and $\mathbf{n}'(t)$ is stationary and of zero mean. Thus, (3.16) also implies that the corresponding version of (3.7) holds, which proves (3.15). □

Remark 4. *Note that the newly constructed signal* $\mathbf{n}_{\varphi'}(t,\tau)$ *is not stationary since it may have different values of covariance functions at different time instants. In fact, similar results as in Proposition 2 can be obtained for a non-stationary* $\mathbf{n}'(t)$ *as long as it satisfies (3.7).*

Compared with (3.14) for a fixed N, the variance in (3.16) is altered due to the introduction of weighting. The value of the weighting $\mathrm{w}_l^{i,j,\tau}$ depends on the lth component of

$\varphi'(t)$. Obviously, for a fixed length N, the value of (3.16) arrives at its minimum if $\varphi'(t) = \mathbf{0}$. However, we need to add some constraints on $\varphi'(t)$ for specific purposes throughout the procedure of tuning $\varphi'(t)$ in practice.

3.4 Variance Reduced Open-Loop Identification

As explained in Section 3.2.2, the estimation of the extended observability matrix O_r plays an important role in the first step of the CoBRA. Thus, a good estimate of O_r expects to lead to a good parameter estimation of the system model. In order to quantify the accuracy of the estimation, the covariance matrix of the parameter estimates is usually considered [45]. Equivalently, the covariance matrix of the estimation errors could also be used, where the trace function of the covariance matrix is used for a comparison of the accuracy.

We can seek possible approaches to designing or tuning the IV $\xi(t)$ based on a scalar accuracy function to achieve an optimal parameter estimation. Note that we are seeking an optimal estimate of O_r in terms of a minimum variance in this chapter, rather than directly considering the system matrices or transfer functions. To consider a more general case, we post-multiply (3.6) by a positive definite matrix $\mathbf{W}_c$, pre-multiply it by a positive definite orthogonal matrix $\mathbf{W}_r$, and perform SVD on the left hand side

$$\mathbf{W}_r \hat{\mathbf{R}}_{y\xi} \Pi^{\perp}_{\hat{\mathbf{R}}^{\mathrm{T}}_{u\xi}} \mathbf{W}_c = \begin{bmatrix} \hat{Q}_1 & \hat{Q}_2 \end{bmatrix} \begin{bmatrix} \hat{S}_1 & \mathbf{0} \\ \mathbf{0} & \hat{S}_2 \end{bmatrix} \begin{bmatrix} \hat{V}_1^{\mathrm{T}} \\ \hat{V}_2^{\mathrm{T}} \end{bmatrix} \tag{3.18}$$

where $\mathrm{diag}\{\hat{S}_1, \hat{S}_2\}$ contains the singular values distributed in a non-increasing order. $\hat{S}_1$ represents the set of n dominant singular values corresponding to the n principal left singular vectors. Note that $\hat{S}_2$ will go to zero if the noise effect is eliminated as the data length N goes to infinity. The reason for making $\mathbf{W}_r$ an orthogonal matrix will be explained in Section 3.4.1.

It can be observed from (3.6) that the residue term will impact the quality of parameter estimation. From Proposition 2, the component $\hat{\mathbf{R}}_{v\xi}$ of the residue in (3.6) will vanish when the data length N goes to infinity as long as the noise signal $\mathbf{v}(t)$ satisfy the corresponding conditions. As mentioned in Remark 4, the residue term will also vanish (or approach a constant) if the noise signals satisfy the conditions listed in Proposition 1, where the stochastic signal is not necessary to be stationary. Thus, one approach to eliminating the noise effect with a mild assumption on the noise signals is to let the data length N go to infinity.

However, limited data is more practical. In this case, we can assume that the noise signals within a certain period are stationary and satisfy the conditions in Corollary 1. Given the data with a fixed length N, we are seeking an estimate with a minimum noise effect. A scalar accuracy function needs to be defined to quantify the quality of the estimate. Obviously, this function should be a function of the residue term and arrive at its minimum when the residue term vanishes.

3.4.1 Optimal Variance Reduction in Weighted CoBRA

The following asymptotic relations can be derived from (3.6) and (3.18) by using Proposition 2:

$$\mathbf{W}_r\left(\hat{\mathbf{R}}_{y\xi}\Pi^{\perp}_{\hat{\mathbf{R}}^{\mathrm{T}}_{u\xi}} - \mathrm{O}_r\hat{\mathbf{R}}_{x\xi}\Pi^{\perp}_{\hat{\mathbf{R}}^{\mathrm{T}}_{u\xi}}\right)\mathbf{W}_c \to \mathbf{0},\ w.p.1$$
$$\mathbf{W}_r\hat{\mathbf{R}}_{y\xi}\Pi^{\perp}_{\hat{\mathbf{R}}^{\mathrm{T}}_{u\xi}}\mathbf{W}_c - \hat{Q}_1\hat{S}_1\hat{V}_1^{\mathrm{T}} \to \mathbf{0},\ w.p.1$$

as N goes to infinity. To distinguish the asymptotic equivalence from the exactly holding equivalence, such relations are denoted with a symbol $\overset{\mathrm{A}}{=}$:

$$\mathbf{W}_r\hat{\mathbf{R}}_{y\xi}\Pi^{\perp}_{\hat{\mathbf{R}}^{\mathrm{T}}_{u\xi}}\mathbf{W}_c \overset{\mathrm{A}}{=} \mathbf{W}_r\mathrm{O}_r\hat{\mathbf{R}}_{x\xi}\Pi^{\perp}_{\hat{\mathbf{R}}^{\mathrm{T}}_{u\xi}}\mathbf{W}_c \overset{\mathrm{A}}{=} \hat{Q}_1\hat{S}_1\hat{V}_1^{\mathrm{T}} \tag{3.19}$$

In addition, there exists a nonsingular matrix $\mathbf{T}_{r,c}$ related to $\mathbf{W}_r$ and $\mathbf{W}_c$ such that

$$\mathbf{W}_r\mathrm{O}_r\hat{\mathbf{R}}_{x\xi}\Pi^{\perp}_{\hat{\mathbf{R}}^{\mathrm{T}}_{u\xi}}\mathbf{W}_c = \left(\mathbf{W}_r\mathrm{O}_r\mathbf{T}^{-1}_{r,c}\right)\left(\mathbf{T}_{r,c}\Delta\mathbf{W}_c\right)$$

and

$$\mathbf{W}_r\mathrm{O}_r\mathbf{T}^{-1}_{r,c} \overset{\mathrm{A}}{=} \hat{Q}_1,\ \mathbf{T}_{r,c}\Delta\mathbf{W}_c \overset{\mathrm{A}}{=} \hat{S}_1\hat{V}_1^{\mathrm{T}} \tag{3.20}$$

where $\Delta = \hat{\mathbf{R}}_{x\xi}\Pi^{\perp}_{\hat{\mathbf{R}}^{\mathrm{T}}_{u\xi}}$. For subspace methods, one is only interested in the extraction of the range space of O_r, spanned by its column vectors and denoted as range(O_r). Note that $\mathrm{range}\left(\mathrm{O}_r\right) = \mathrm{range}\left(\mathrm{O}_r\mathbf{T}^{-1}_{r,c}\right)$ and $\mathrm{O}_r\mathbf{T}^{-1}_{r,c} \overset{\mathrm{A}}{=} \mathbf{W}_r^{-1}\hat{Q}_1$ from (3.20). Thus, O_r is estimated by $\hat{\mathrm{O}}_r = \mathbf{W}_r^{-1}\hat{Q}_1$ with range($\hat{\mathrm{O}}_r$) $\overset{\mathrm{A}}{=}$ range(O_r).

The estimation error of $\hat{\mathrm{O}}_r$ can be defined as the part lying in the orthogonal complement of the subspace $\mathrm{range}\left(\mathrm{O}_r\right)$:

$$\varepsilon = \Pi^{\perp}_{\mathrm{O}_r}\hat{\mathrm{O}}_r = \Pi^{\perp}_{\mathrm{O}_r}\mathbf{W}_r^{-1}\hat{Q}_1 \tag{3.21}$$

where $\Pi^{\perp}_{\mathrm{O}_r} = \mathbf{I} - \mathrm{O}_r\left(\mathrm{O}_r^{\mathrm{T}}\mathrm{O}_r\right)^{-1}\mathrm{O}_r^{\mathrm{T}}$. The matrix factorization for $\Pi^{\perp}_{\mathrm{O}_r}$ suggested in [42, 46] to handle the singularity will not affect the final results in this chapter because of the cyclic property of trace operator $\mathrm{Tr}\,(AB) = \mathrm{Tr}\,(BA)$ and the property of the product $\Pi_{\mathrm{O}_r}^{\perp}\Pi_{\mathrm{O}_r}^{\perp} = \Pi_{\mathrm{O}_r}^{\perp}$ of the matrix projection.

Remark 5. *Optimally extracting the range space is considered to be finding the corresponding basis vectors in the directions that produce unbiased range space with the smallest variance. $\mathbf{W}_r$ and $\mathbf{W}_c$ are actually applied to tune the angles of the basis vectors. To make (3.21) reasonable, all possible estimated basis vectors should be with the same scaling. In this sense, $\mathbf{W}_r$ should be orthogonal performing only rotation operations. $\hat{Q}_1$, which is related to $\mathbf{W}_r$ and $\mathbf{W}_c$, has already been scaled via SVD.*

The noise term in (3.6) can be rewritten such that

$$\hat{\mathbf{R}}_{v\xi} = \underbrace{\frac{1}{N}\sum_{t=0}^{N-1} \mathbf{n}\,(t)\,\varphi(t-\tau_1)^{\mathrm{T}}}_{M_{\varphi,N}(\tau_1)} \tag{3.22}$$

where

$$\mathbf{n}\,(t) = \begin{bmatrix} \mathbf{v}\,(t) \\ \mathbf{v}\,(t+1) \\ \vdots \\ \mathbf{v}\,(t+r-1) \end{bmatrix} \tag{3.23}$$

$$\varphi\,(t) = \begin{bmatrix} \xi\,(t) \\ \xi\,(t-1) \\ \vdots \\ \xi\,(t+\tau_1-\tau_2+r-1) \end{bmatrix} \tag{3.24}$$

and $r \ll N$. In addition, we can also derive from (3.6) and (3.18) the following relations

$$\begin{cases} \Pi_{\mathrm{O}_r}^{\perp}\hat{\mathbf{R}}_{y\xi}\Pi_{\hat{\mathbf{R}}_{u\xi}^{\mathrm{T}}}^{\perp}\mathbf{W}_c = \Pi_{\mathrm{O}_r}^{\perp}M_{\varphi,N}\,(\tau_1)\Pi_{\hat{\mathbf{R}}_{u\xi}^{\mathrm{T}}}^{\perp}\mathbf{W}_c \\ \Pi_{\mathrm{O}_r}^{\perp}\hat{\mathbf{R}}_{y\xi}\Pi_{\hat{\mathbf{R}}_{u\xi}^{\mathrm{T}}}^{\perp}\mathbf{W}_c\hat{V}_1\hat{S}_1^{-1} = \underbrace{\Pi_{\mathrm{O}_r}^{\perp}\mathbf{W}_r^{-1}\hat{Q}_1}_{\varepsilon} \end{cases} \tag{3.25}$$

where the orthonormality of $\hat{V}_1, \hat{V}_2$ is used. Also, (3.20) implies

$$\hat{V}_1 \hat{S}_1^{-1} = \left(\hat{S}_1 \hat{V}_1^{\mathrm{T}}\right)^{\dagger} \overset{\mathrm{A}}{=} \left(\mathbf{T}_{r,c} \Delta \mathbf{W}_c\right)^{\dagger} \tag{3.26}$$

where $(\cdot)^{\dagger}$ represents the right pseudo-inverse. Thus, in an asymptotic sense,

$$\begin{aligned} \varepsilon &= \Pi_{\mathrm{O}_r}^{\perp} M_{\varphi,N}(\tau_1) \Pi_{\hat{\mathbf{R}}_{u\xi}^{\mathrm{T}}}^{\perp} \mathbf{W}_c \left(\hat{S}_1 \hat{V}_1^{\mathrm{T}}\right)^{\dagger} \\ &\overset{\mathrm{A}}{=} \Pi_{\mathrm{O}_r}^{\perp} M_{\varphi,N}(\tau_1) \Pi_{\hat{\mathbf{R}}_{u\xi}^{\mathrm{T}}}^{\perp} \mathbf{W}_c \left(\mathbf{T}_{r,c} \Delta \mathbf{W}_c\right)^{\dagger} \\ &= \Pi_{\mathrm{O}_r}^{\perp} M_{\varphi,N}(\tau_1) \mathbf{H} \\ &= \underbrace{\frac{1}{N} \sum_{t=0}^{N-1} \mathbf{n}'(t) \varphi'(t-\tau_1)^{\mathrm{T}}}_{M_{\varphi',N}(\tau_1)} \end{aligned} \tag{3.27}$$

where $\mathbf{n}'(t) = \Pi_{\mathrm{O}_r}^{\perp} \mathbf{n}(t)$, $\varphi'(t) = \mathbf{H}^{\mathrm{T}} \varphi(t)$ represent the weighted signals and

$$\mathbf{H} = \Pi_{\hat{\mathbf{R}}_{u\xi}^{\mathrm{T}}}^{\perp} \mathbf{W}_c^2 \Delta^{\mathrm{T}} \left(\Delta \mathbf{W}_c^2 \Delta^{\mathrm{T}}\right)^{-1} \mathbf{T}_{r,c}^{-1}. \tag{3.28}$$

The direct effect of the left weighting matrix $\mathbf{W}_r$ on the estimation error ε in (3.27) is asymptotically eliminated. In addition, it will be shown in Section 3.4.3 that the unknown matrix $\mathbf{T}_{r,c}$ doesn't affect the optimal accuracy. Thus, we may choose $\mathbf{W}_r = \mathbf{I}$, without loss of generality, from an asymptotic perspective. Although only finite data is available, the asymptotic properties are approximately preserved for a relatively large data length N.

The vectorized version of above is $\mathrm{vec}(\varepsilon) \overset{\mathrm{A}}{=} S_{\varphi',N}(\tau_1)$ with the form denoted in Proposition 2. According to Proposition 2, all the components of the error term ε will converge to zero w.p.1 as N goes to infinity with the trace of the variance of $\mathrm{vec}(\varepsilon)$ decreasing at the rate $\mathrm{O}\left(\frac{1}{N}\right)$. Thus, in order to minimize the variance of the estimation error regardless of the data length N, we should multiply ε by $\sqrt{N}$, which leads to the following scalar accuracy function

$$\begin{aligned}\mathrm{V}(\varepsilon) &= \mathrm{Tr}\left\{\mathrm{Var}\left\{\sqrt{N}\mathrm{vec}\left(\varepsilon\right)\right\}\right\} \\ &= \mathrm{Tr}\left\{E\left\{N\mathrm{vec}\left(\varepsilon\right)\mathrm{vec}(\varepsilon)^{\mathrm{T}}\right\}\right\} \\ &\overset{\mathrm{A}}{=} \underbrace{\mathrm{Tr}\left\{E\left\{NS_{\varphi',N}\left(\tau_1\right)S_{\varphi',N}(\tau_1)^{\mathrm{T}}\right\}\right\}}_{\hat{\mathrm{V}}(\varepsilon)}.\end{aligned} \tag{3.29}$$

Note from (3.16) that $\varphi'\left(t\right)$ can tune the variance via a weighting $\mathrm{w}_l^{i,j,\tau}$. Thus, an ideal optimization problem of finding the optimal IV $\xi(t)$ without introducing the freedom $\mathbf{W}_c$ can be summarized as

$$\begin{aligned}\min_{\xi(t)} \quad & \hat{\mathrm{V}}(\varepsilon) = \mathrm{Tr}\left\{\mathrm{Var}\left\{\sqrt{N}S_{\varphi',N}\left(\tau_1\right)\right\}\right\} \\ \text{s.t.} \quad & \mathbf{n}'\left(t\right) = \Pi^{\perp}_{\mathrm{O}_r}\mathbf{n}\left(t\right), \varphi'\left(t\right) = \mathbf{H}^{\mathrm{T}}\varphi\left(t\right), \\ & \xi\left(t\right) \in \mathscr{C}_\xi, \mathbf{W}_c = \mathbf{I},\end{aligned} \tag{3.30}$$

where $\mathscr{C}_\xi$ is a certain constraint satisfying the conditions in Section 3.2.2. However, the analytic solution is intractable since $\mathbf{H}$ is typically a nonlinear function of $\xi(t)$. Instead, a tangible alternative of tuning $\varphi'\left(t\right)$ is to solve a new optimization problem based on the positive definite freedom variable $\mathbf{W}_c$

$$\begin{aligned}\min_{\mathbf{W}_c} \quad & \hat{\mathrm{V}}(\varepsilon) = \mathrm{Tr}\left\{\mathrm{Var}\left\{\sqrt{N}S_{\varphi',N}\left(\tau_1\right)\right\}\right\} \\ \text{s.t.} \quad & \mathbf{n}'\left(t\right) = \Pi^{\perp}_{\mathrm{O}_r}\mathbf{n}\left(t\right), \varphi'\left(t\right) = \mathbf{H}^{\mathrm{T}}\varphi\left(t\right), \\ & \xi\left(t\right) \in \mathscr{C}_\xi, \mathbf{W}_c \succ 0.\end{aligned} \tag{3.31}$$

3.4.2 Results On the Scalar Accuracy Function $\hat{\mathrm{V}}(\varepsilon)$

In order to solve the optimization problem in (3.31), one should derive an explicit representation for the scalar accuracy function $\hat{\mathrm{V}}(\varepsilon)$. The result is summarized in the following theorem.

Theorem 1. *The following relation holds for the scalar accuracy function* $\hat{\mathrm{V}}(\varepsilon)$

$$\begin{aligned} & \hat{\mathrm{V}}\left(\varepsilon\right) \\ \overset{\mathrm{A}}{=} \; & \sum_{|\tau|\leq N-1} \mathrm{Tr}\left\{\mathbf{M}_{\mathrm{H}}\left(\tau\right)\right\} \mathrm{Tr}\left\{\Pi_{\mathrm{O}_r}^{\perp} R_{nn}\left(-\tau\right)\right\} \\ \overset{\mathrm{A}}{=} \; & \lim_{\bar{\tau}_n\to+\infty} \hat{\mathrm{V}}_{\bar{\tau}_n}\left(\varepsilon\right) \end{aligned} \tag{3.32}$$

where

$$\mathbf{M}_{\mathrm{H}}\left(\tau\right) = \mathbf{H}^{\mathrm{T}} \hat{R}_{\xi\xi}^{t,s}(\tau)^{\mathrm{T}} \mathbf{H} \tag{3.33}$$

$$\hat{\mathrm{V}}_{\bar{\tau}_n}\left(\varepsilon\right) = \sum_{|\tau|\leq\bar{\tau}_n} \mathrm{Tr}\left\{\mathbf{M}_{\mathrm{H}}\left(\tau\right)\right\} \mathrm{Tr}\left\{\Pi_{\mathrm{O}_r}^{\perp} \hat{R}_{nn}\left(-\tau\right)\right\} \tag{3.34}$$

with $\mathbf{H}$ *defined in (3.28), and* $R_{nn}\left(-\tau\right)$, $\hat{R}_{nn}\left(-\tau\right)$ *are defined in (3.3) and (3.4) with* $\mathbf{n}\left(t\right)$ *given by (3.23). Note that* $\hat{R}_{\xi\xi}^{t,s}\left(\tau\right)$ *is a block* $l_c \times l_c$ *matrix with the* t, s*-th element being* $\hat{R}_{\xi\xi}\left(\tau+s-t\right) \in \mathbb{R}^{p\times p}$ *and* $l_c = \tau_2 - \tau_1 - r + 2$ *is the number of the block columns of the Hankel matrices in (3.5).*

Proof. According to [46], we have

$$\begin{aligned} & \mathrm{Var}\left\{\sqrt{N}\mathrm{vec}\left(\varepsilon\right)\right\} \\ \overset{\mathrm{A}}{=} \; & \sum_{|\tau|\leq N-1} \mathbf{M}_{\mathrm{H}}\left(\tau\right) \otimes \left(\Pi_{\mathrm{O}_r}^{\perp} R_{nn}\left(-\tau\right) \Pi_{\mathrm{O}_r}^{\perp}\right) \\ \overset{\mathrm{A}}{=} \; & \lim_{\bar{\tau}_n\to+\infty} \sum_{|\tau|\leq\bar{\tau}_n} \mathbf{M}_{\mathrm{H}}\left(\tau\right) \otimes \left(\Pi_{\mathrm{O}_r}^{\perp} \hat{R}_{nn}\left(-\tau\right) \Pi_{\mathrm{O}_r}^{\perp}\right) \end{aligned}$$

Then, (3.32) can be proved straightforwardly. □

Remark 6. *In (3.32),* $\hat{\mathrm{V}}_{\bar{\tau}_n}\left(\varepsilon\right)$ *could be used as an approximation of* $\hat{\mathrm{V}}\left(\varepsilon\right)$ *if two assumptions are made. First, the summand should be negligible for* $|\tau| > \bar{\tau}_n$*. Thus, either* $R_{nn}\left(-\tau\right)$ *is of finite support with* $|\tau| \leq \bar{\tau}_n$*, or* $\hat{R}_{\xi\xi}\left(\tau\right)$ *tends to be around zero very quickly for* $|\tau| > \bar{\tau}_n$*. The former case is reasonable since* $\mathbf{n}(t)$ *can be seen as the output of a stable filter.* $\bar{\tau}_n$ *is selected such that sufficient noise information is included. Note that the selection of* $\bar{\tau}_n$ *is* *independent of $\bar{\tau}$ in (3.5). The latter case can be achieved by appropriately choosing the IV. Second, $R_{nn}(-\tau)$ is estimated by $\hat{R}_{nn}(-\tau)$ under the assumption of ergodicity.*

3.4.3 Least-Length Perturbation and Solution

From Theorem 1 and Remark 6, we could take the following approximation of $\hat{\mathrm{V}}(\varepsilon)$ for a sufficiently large N

$$\begin{aligned} & \hat{\mathrm{V}}(\varepsilon) \\ \approx\; & \hat{\mathrm{V}}_{\bar{\tau}_n}(\varepsilon) \\ \overset{\mathrm{A}}{=}\; & \mathrm{Tr}\Big\{\big(\mathrm{O}_r^{\mathrm{T}}\mathrm{O}_r\big)^{-\frac{1}{2}}\big(\Delta\mathbf{W}_c^2\Delta^{\mathrm{T}}\big)^{-1}\Delta\mathbf{W}_c^2 \\ & \underbrace{\Big(\Pi^{\perp}_{\hat{\mathbf{R}}^{\mathrm{T}}_{u\xi}}\Sigma\Pi^{\perp}_{\hat{\mathbf{R}}^{\mathrm{T}}_{u\xi}}\Big)}_{\mathbf{M}_\Sigma}\mathbf{W}_c^2\Delta^{\mathrm{T}}\big(\Delta\mathbf{W}_c^2\Delta^{\mathrm{T}}\big)^{-1}\big(\mathrm{O}_r^{\mathrm{T}}\mathrm{O}_r\big)^{-\frac{1}{2}}\Big\} \end{aligned} \tag{3.35}$$

where

$$\Sigma = \sum_{|\tau|\leq\bar{\tau}_n}\hat{R}^{t,s}_{\xi\xi}(\tau)^{\mathrm{T}}\mathrm{Tr}\left\{\Pi^{\perp}_{\mathrm{O}_r}\hat{R}_{nn}(-\tau)\right\} \tag{3.36}$$

and the relation $\mathbf{T}_{r,c}^{-1}\mathbf{T}_{r,c}^{-\mathrm{T}}\overset{\mathrm{A}}{=}\big(\mathrm{O}_r^{\mathrm{T}}\mathrm{O}_r\big)^{-1}$ [46], derived from (3.18) and (3.20), is used during the second step in (3.35). Thus, the scalar accuracy function $\hat{\mathrm{V}}(\varepsilon)$ and $\hat{\mathrm{V}}_{\bar{\tau}_n}(\varepsilon)$ are asymptotically invariant to $\mathbf{T}_{r,c}$ introduced by the similarity transformation during estimating the extended observability matrix O_r.

In practice, the accuracy function $\hat{\mathrm{V}}(\varepsilon)$ in (3.31) is replaced by $\hat{\mathrm{V}}_{\bar{\tau}_n}(\varepsilon)$ since only finite experimental data is available. Some existing results can be applied to solve the optimization problem in (3.31) analytically as long as $\mathbf{M}_\Sigma$ and $\mathbf{W}_c$ in (3.35) are positive definite. However, $\mathbf{M}_\Sigma$ is rank deficient and appears to be positive semi-definite in the ideal case. Moreover, it may sometimes have several small negative eigenvalues due to the approximation and numerical errors in practice. Note that only information within $|\tau|\leq\bar{\tau}_n$ is included in the approximation $\hat{\mathrm{V}}_{\bar{\tau}_n}(\varepsilon)$ when finite data is available. Thus, $\mathbf{M}_\Sigma$ may not be an ideal positive semi-definite matrix as suggested by the construction of $\hat{\mathrm{V}}(\varepsilon)$. We can split $\hat{\mathrm{V}}_{\bar{\tau}_n}(\varepsilon)$ into two parts: one is with a positive definite matrix $\mathbf{M}'_\Sigma$ in the middle and has an analytic minimum solution; the other one with the residue $\mathbf{M}_\Sigma - \mathbf{M}'_\Sigma$. On the other hand, it is appealing to obtain an analytic solution by ignoring the residue at the cost of losing minor accuracy instead of using nonlinear or non-smooth optimizations. Thus, the same idea of perturbing the Hessian matrix in modified Newton methods is used here to perturb $\mathbf{M}_\Sigma$ into a positive definite matrix $\mathbf{M}'_\Sigma$. This procedure can be summarized into the following optimization problem

$$\begin{aligned} \min_{\mathbf{M}'_\Sigma} \quad & \|\mathbf{M}_\Sigma - \mathbf{M}'_\Sigma\|_F \\ \text{s.t.} \quad & \mathrm{cond}\left(\mathbf{M}'_\Sigma\right) \le \beta, \quad \mathbf{M}'_\Sigma \succ 0 \end{aligned} \tag{3.37}$$

where cond$(\cdot)$ denotes the spectral condition number and β is a specified positive constant. The selection of β depends on the highest condition number that a digital device could endure. Note that $\mathbf{M}_\Sigma$ is symmetric and can be rewritten as $\mathbf{M}_\Sigma = U_\Sigma \Lambda_\Sigma U_\Sigma^{\mathrm{T}}$ through eigen-decomposition. We assume that $\Lambda_\Sigma = \mathrm{diag}\left(\lambda_1, \lambda_2, \cdots, \lambda_{n_\Sigma}\right)$ with the eigenvalues ordered such that $\lambda_1 \ge \lambda_2 \ge \cdots \ge \lambda_{n_\Sigma}$. The lower bound of the expected smallest eigenvalue is then given by $\sigma = \lambda_1/\beta$ with $\lambda_1 > 0$.

Proposition 3. *The solution for (3.37) is given by* $\mathbf{M}'_\Sigma = U_\Sigma \Lambda'_\Sigma U_\Sigma^{\mathrm{T}}$*, where* $\Lambda'_\Sigma = \mathrm{diag}(\lambda'_1, \lambda'_2, \cdots, \lambda'_{n_\Sigma})$ *and*

$$\lambda'_i = \begin{cases} \lambda_i, & \text{if } \lambda_i \ge \sigma \\ \sigma, & \text{otherwise} \end{cases}$$

Proof. This proposition can be easily proved by referring to [47]. Let $\mathbf{Y} = U_\Sigma^{\mathrm{T}} \mathbf{M}'_\Sigma U_\Sigma$. We have $\mathbf{Y} \succ 0$ since $M'_\Sigma \succ 0$. Then,

$$\begin{aligned}
\left\|\mathbf{M}_\Sigma - \mathbf{M}'_\Sigma\right\|_F^2 &= \left\|\Lambda_\Sigma - \mathbf{Y}\right\|_F^2 \\
&= \sum_{i \neq j} y_{ij}^2 + \sum_i \left(\lambda_i - y_{ii}\right)^2 \\
&\geq \sum_{\lambda_i < \lambda_1} \left(\lambda_i - y_{ii}\right)^2
\end{aligned}$$

Note that the diagonal entries of a positive definite matrix must be positive, i.e., $y_{ii} > 0$. Attaining the lower bound above implies $y_{ij} = 0$ for $i \neq j$ and $y_{11} = \lambda_1 > 0$. In this case, $\{y_{ii}\}$ are the eigenvalues of $\mathbf{M}'_\Sigma$. Also, the lower bound of the smallest eigenvalue of $\mathbf{M}'_\Sigma$ is given by $\sigma = \lambda_1/\beta$ since $\operatorname{cond}\left(\mathbf{M}'_\Sigma\right) \leq \beta$. This implies $\min\{y_{ii}\} \geq \sigma$. Further, we have $\sum\limits_{\lambda_i < \lambda_1} \left(\lambda_i - y_{ii}\right)^2 \geq \sum\limits_{\lambda_i < \sigma} \left(\lambda_i - \sigma\right)^2$, whose lower bound can be uniquely achieved with $\mathbf{Y} = \Lambda'_\Sigma = \operatorname{diag}\left(\lambda'_1, \lambda'_2, \cdots, \lambda'_{n_\Sigma}\right)$. □

Note that $\mathbf{M}'_\Sigma$ is a positive definite matrix satisfying the given constraint and with the least-length perturbation in terms of the Frobenius norm. Replace $\mathbf{M}_\Sigma$ in (3.35) by $\mathbf{M}'_\Sigma$ calculated in (3.37) and denote the perturbed expression as $\hat{\mathrm{V}}'_{\bar{\tau}_n}(\varepsilon)$. A practical version of the optimization problem in (3.31) can be written as

$$\begin{aligned}
\min_{\mathbf{W}_c} \quad & \hat{\mathrm{V}}'_{\bar{\tau}_n}(\varepsilon) \\
\text{s.t.} \quad & \mathbf{n}'(t) = \Pi^\perp_{\mathrm{O}_r}\mathbf{n}(t), \varphi'(t) = \mathbf{H}^{\mathrm{T}}\varphi(t), \\
& \xi(t) \in \mathscr{C}_\xi, \mathbf{W}_c \succ 0.
\end{aligned} \tag{3.38}$$

Theorem 2. *The analytic solution for (3.38) is given by*

$$\mathbf{W}_c^* = \left(\mathbf{M}'_\Sigma\right)^{-\frac{1}{2}} \tag{3.39}$$

Proof. The optimal solution $\mathbf{W}_c^*$ for (3.38) can be obtained according to Appendix II.2 in [2]. □

Remark 7. *The derivation and the final optimal weighting matrix for CoBRA are different from the work in* [42]*. Weightings are applied based on the covariance data instead of the raw data. Weighted CoBRA produces the optimal low-order deterministic model instead of the possibly high-order innovation model. In addition, the matrix perturbation in Section 3.4.3 is needed to get an approximate but analytic optimal weighting for (3.31).*

3.5 Two Stage Implementation of CoBRA for Closed-Loop Identification

3.5.1 Approximation of System and Noise Information

The variance reduced open-loop identification via CoBRA has been investigated in Section 3.4. In order to compute the matrix $\mathbf{M}_\Sigma$ in (3.35), one needs the system information O_r and sample covariance values $\hat{R}_{nn}\left(-\tau\right)$ of the noise. This can be done by computing an initial system model $G^{\text{int}}(q^{-1})$ through CoBRA without introducing the freedom $\{\mathbf{W}_r, \mathbf{W}_c\}$ as in (3.18). One can get an approximation of O_r directly from a state-space representation of $G^{\text{int}}(q^{-1})$. The noise signal $\mathbf{v}(t)$ in one experiment can be estimated by

$$\mathbf{v}\left(t\right) = \mathbf{y}\left(t\right) - \bar{\mathbf{y}}\left(t\right) \approx \mathbf{y}\left(t\right) - G^{\text{int}}\left(q^{-1}\right)\mathbf{u}\left(t\right). \tag{3.40}$$

All of the above discussions boil down to an iterative implementation of weighted CoBRA for open-loop MIMO system identification:

(i) Given $\{\mathbf{u}\left(t\right), \mathbf{y}\left(t\right)\}$, we compute an initial system model $G^{\text{int}}(q^{-1})$ by applying the unweighted (i.e. $\mathbf{W}_r = \mathbf{I}, \mathbf{W}_c = \mathbf{I}$) CoBRA described in Section 3.2.2;

(ii) Approximate $\mathbf{v}(t)$ using (3.40), construct $\mathbf{n}(t)$ in (3.23), and compute $\hat{R}_{nn}(-\tau)$ using (3.4). In addition, O_r is estimated from a state-space model of $G^{\text{int}}(q^{-1})$;

(iii) Compute the weighting matrices in (3.18) such that $\mathbf{W}_r = \mathbf{I}$ and $\mathbf{W}_c = \mathbf{W}_c^*$ as in (3.39);

(iv) Given the same $\{\mathbf{u}(t), \mathbf{y}(t)\}$, apply the weighted CoBRA to obtain the estimated $\{\hat{A}, \hat{B}, \hat{C}, \hat{D}\}$ in (3.1).

The last three procedures can also be iterated to seek a possibly better performance. Except for the weighting matrices, the other parameter settings (i.e., the model order, Hankel matrices, etc) for weighted and unweighted CoBRA should be the same.

3.5.2 Summary of Two Stage CoBRA

The two stage technique described in Section 3.2.1 is implemented to cope with closed-loop identification of MIMO systems. Relevant signals are denoted as follows:

$$\mathbf{r}(t) = \begin{bmatrix} \mathbf{r}_1(t) \\ \mathbf{r}_2(t) \end{bmatrix}, \mathbf{y}_J(t) = \begin{bmatrix} \mathbf{u}(t) \\ \mathbf{y}(t) \end{bmatrix}, \bar{\mathbf{y}}_J(t) = \begin{bmatrix} \bar{\mathbf{u}}(t) \\ \bar{\mathbf{y}}(t) \end{bmatrix}$$

where $\mathbf{r}(t)$ is the reference signal and $\bar{\mathbf{y}}_J(t)$ is the noise-free version of the joint input-output signal $\mathbf{y}_J(t)$ contaminated by noise in a closed-loop setting. The iterative implementation of the weighted CoBRA in Section 3.5.1 is applied at the first stage to identify a deterministic model $G_J(q^{-1})$ from $\mathbf{r}(t)$ to $\mathbf{y}_J(t)$. Then, $\bar{\mathbf{y}}_J(t)$ is estimated through a simulation output $\bar{\mathbf{y}}_J(t) = G_J(q^{-1})\mathbf{r}(t)$. At the second stage, the unweighted CoBRA is performed to identify a deterministic model from $\bar{\mathbf{u}}(t)$ to $\bar{\mathbf{y}}(t)$, leading to estimates of the parameters $\{\hat{A}, \hat{B}, \hat{C}, \hat{D}\}$ for the plant $G(q^{-1})$ in (3.1).

3.6 Simulation Results

Suppose that the plant to be identified and the controller are given by

$$\begin{cases} G\left(q^{-1}\right) & = \frac{0.03228q^{-1}+0.03164q^{-2}}{1-1.781q^{-1}+0.926q^{-2}} \\ C\left(q^{-1}\right) & = \frac{1-0.8q^{-1}}{1-0.6q^{-1}} \end{cases}$$

The noise signal $\mathbf{v}(t)$ is assumed to be generated by a stable filter $H(q^{-1})$ such that $\mathbf{v}\left(t\right) = H\left(q^{-1}\right)\mathbf{e}\left(t\right)$ and

$$H\left(q^{-1}\right) = \frac{1-1.55q^{-1}+1.045q^{-2}-0.3388q^{-3}}{1-2.4q^{-1}+2.2725q^{-2}-0.8303q^{-3}} \cdot \sigma_e$$

where $\sigma_e = 0.2$. The reference signal $\mathbf{r}_2(t) = 0$ and $\mathbf{r}_1(t)$ is a realization of Gaussian white noise with zero mean and variance $\sigma_r^2 = 3^2$. Note that the signals $\{\mathbf{u}\left(t\right), \mathbf{y}\left(t\right)\}$ are heavily contaminated by the noise in the closed-loop setting. The true open-loop poles of $G(q^{-1})$ and the closed-loop poles of the system $G_J(q^{-1})$ from $\mathbf{r}_1(t)$ to $\mathbf{y}_J(t)$ are respectively

$$p_{1,2}^{ol} = \{0.8905 \pm 0.3647i\}$$

$$p_{1,2}^{cl} = \{0.8531 \pm 0.4200i\}, p_3^{cl} = 0.6424$$

A two stage CoBRA described in Section 3.4 is implemented. In (3.5), we choose $r = 10$, $\tau_1 = -20$, $\tau_2 = 50$, and $\bar{\tau} = 50$ for both stages. At the first stage, the closed-loop model order is known as $n_{cl} = 3$, the instrument is $\xi(t) = \mathbf{r}_1(t)$ and we select $\bar{\tau}_n = 100$ in (3.36). The optimal weighting W_c^* is computed as (3.39) with $\beta = 10^6$ in (3.37). At the second stage, the open-loop model order is known as $n_{ol} = 2$ and the instrument is $\xi(t) = \bar{\mathbf{u}}(t)$. Different data lengths N are used to illustrate the efficiency of the proposed two stage CoBRA method. For a given data length N, 100 Monte Carlo simulations are performed each time. The Root

Mean Square Error (RMSE) of the estimated poles is defined as

$$\text{RMSE} = \sqrt{\frac{1}{m_p} \sum_{i=1}^{m_p} |\hat{p}_i - p_i|^2}$$

where m_p indicates the number of the estimated poles $\hat{p}_i$ and true poles p_i.

For the purpose of comparison, we perform six other closed-loop identifications simultaneously. Five of them are two stage methods but with unweighted CoBRA, ORT, CCA [11], PO-MOESP [44], and N4SID/PEM method at the first stage, respectively. Note that PEM is using an initial model estimated by N4SID. The last one is VARX-based PBSIDopt method [48], which is a direct closed-loop subspace method. ORT expects to obtain a good deterministic model since it decomposes the signal into deterministic and stochastic parts. It should also be noted that CCA, PO-MOESP, and PBSIDopt implicitly assume that the system to be identified is with an innovation model, which requires a higher order estimation due to the incorporation of noise dynamics. The drawback is obvious: one needs to estimate a higher order model at the first stage than actually needed for producing a deterministic simulation output.

For a fair comparison at the first stage, CCA and PO-MOESP method are first used to identify a 6th order closed-loop model, followed by a model order reduction to the true order of the deterministic part. As for PBSIDopt, we directly identify a 5th order open-loop model followed by a model order reduction. Note that the past window size p for PBSIDopt has to be large (e.g., $p = 25$) to get asymptotic consistent estimates.

For the first stage, the RMSEs of estimated closed-loop real and complex poles are illustrated in Fig. 3.2 and Fig. 3.3. It can be also observed that the total RMSEs can be illustrated just by Fig. 3.2 since the errors in estimated complex poles are relatively small.

Obviously, the proposed iterative implementation of weighted CoBRA produces much smaller errors for both real and complex poles even when the data length N is relatively short. However, to achieve a same level of estimation performance for an unweighted counterpart, one needs to select a much larger length N. ORT method also shows a good estimate in terms of total RMSEs but it is not good at estimating complex poles. CCA method is struggling to capture the real poles. In view of a joint performance, the unweighted CoBRA is intermediate. PEM is not always the best in practice since the estimated parameters may run into local minima via nonlinear optimization.

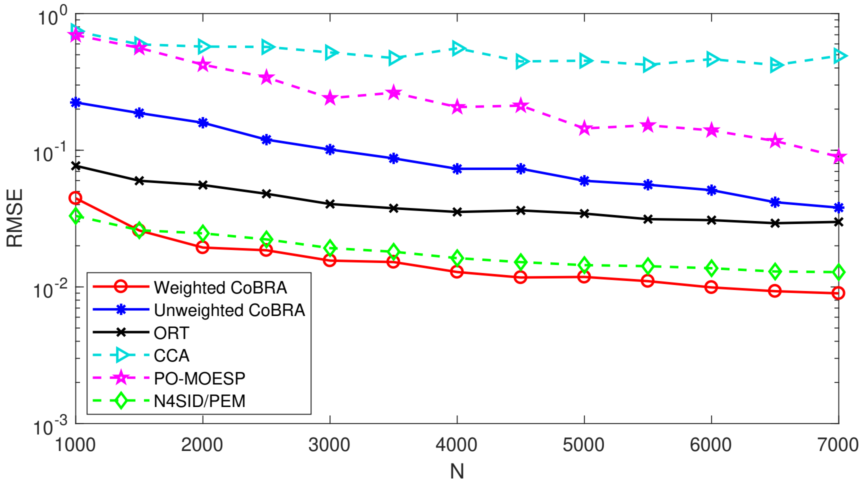


Figure 3.2: RMSE of real poles at the first stage.

For the second stage, the RMSEs of estimated open-loop poles, which are the poles of the plant to be identified, are illustrated in Fig. 3.4. The inaccuracy of the estimates from the first stage propagates to the second stage. Undoubtedly, the proposed two stage CoBRA produces a rather more accurate estimate for a relatively short data. Although the total RMSEs at the first stage are dominated by the real poles, it is interesting to observe that the accuracy of both real and complex poles at the first stage will affect that of the

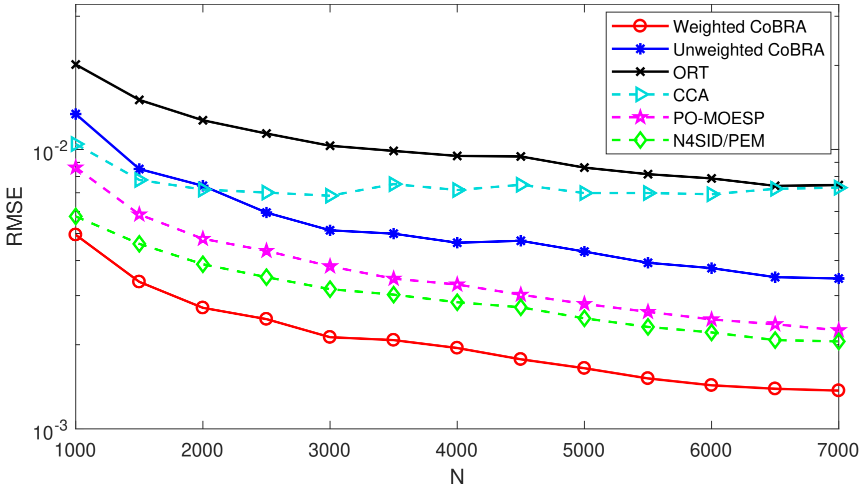


Figure 3.3: RMSE of complex poles at the first stage.

second stage. Unweighted CoBRA is still intermediate at the second stage. The direct method PBSIDopt does not show a remarkable performance in estimating the low order deterministic model. Although an exact conclusion can not be given, it is undoubted that smaller errors in both real and complex poles estimation at the first stage will lead to, with a bigger chance, a better estimate at the second stage.

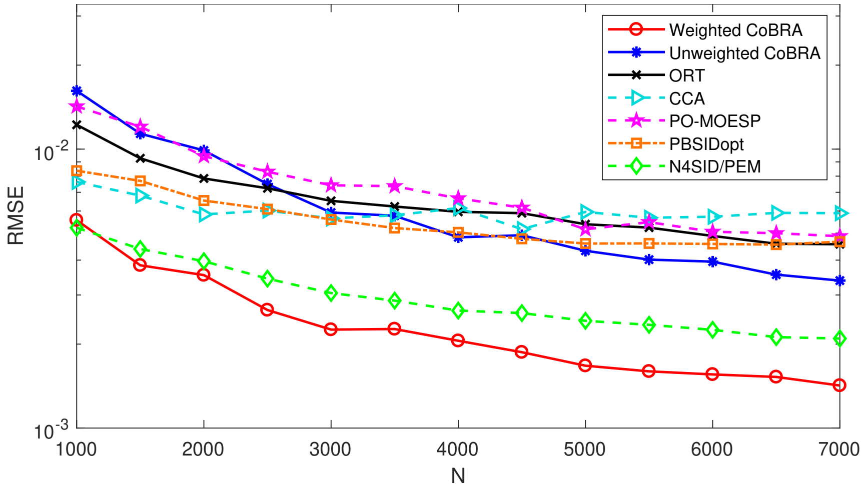


Figure 3.4: RMSE of (complex) poles at the second stage.

3.7 Conclusion

In this chapter, a variance reduced two stage CoBRA has been proposed to cope with the closed-loop identification of MIMO systems. At the first stage, a weighted CoBRA is applied to obtain an optimal closed-loop joint input-output model in terms of a minimum variance. This model produces instruments for the identification at the second stage, where a better estimate can be expected. An in-depth statistical analysis of a stochastic vector signal with bounded covariance has been given in order to show how the noise term in the CoBRA behaves as the data length goes to infinity and how the instrumental variable affects the covariance. In order to obtain an analytic solution for the optimal tuning, a least-length perturbation is performed at the cost of losing only minor accuracy. Simulation examples demonstrate the efficiency of the proposed method.

This chapter is based on the following papers that were published:

Y. Hu and R.A. de Callafon, "Optimal weighting for covariance based realization algorithm," in *Proceedings of IEEE 56th Annual Conference on Decision and Control (CDC)*, Melbourne, VIC, Australia, Dec. 12–15, 2017, pp. 5274–5279.

Y. Hu, Y. Jiang and R.A. de Callafon, " Variance reduction in Covariance Based Realization Algorithm with application to closed-loop data," *Automatica*, vol. 113, pp.108683, 2020.

Part II

MIMO Volterra System Identification

Chapter 4

Tensor Network-based Parameter Estimation of Volterra Systems

This chapter investigates the multi-input multi-output (MIMO) Volterra system identification by using a Tensor Network (TN) technique. Volterra model serves as one of the powerful alternatives to approximate nonlinear dynamics on the basis of input/output observations. The kernel representation of a Volterra model is attractive since it is linear in the kernel coefficients and always stable. Although the kernel representation suffers from the curse of dimensionality, the demand for storage requirements can be relieved by using a TN representation. This allows one to approximate complicated coupled nonlinear dynamics with high degree and even MIMO Volterra models.

4.1 Introduction

The (truncated/finite support) Volterra model is very attractive among different nonlinear models since it is linear in the kernel coefficients and always stable. A simple least squares solution can be obtained. The Volterra model is formulated based on the Volterra series functional expansion and has been widely studied since the seminal work of Wiener [49, 50, 51]. The Volterra model has been demonstrated to be effective in many applications such as equalization [52], acoustic echo cancellation [53], loudspeaker modeling [54] and linearization [55], speech modeling [56], distortion compensation in narrowband communication system [57], distortion analysis [58], identification of nonlinear systems for control [59, 60], active noise control [61], identification of an imaging system [62], and modeling of biological and physiological systems [63]. These applications usually assume that the linear dynamics still dominate although the nonlinear effects are nonnegligible. Thus, they did not exploit the Volterra models of high degree or even the MIMO case to capture complicated coupled nonlinear behaviors between signals. It has been pointed out in [64] that this is restricted by the exponentially growing number of Volterra kernel coefficients as the degree increases, i.e., the curse of dimensionality. This restriction is caused by the use of Kronecker product based structure, which implies redundancy in the kernel coefficients but will simplify notation and matrix computations. The intense storage requirement may also explain why only few contributions on the MIMO Volterra system identification can be found in the recent years [64, 65, 66, 67].

TN can relieve the curse of dimensionality by referring to techniques for multidimensional arrays, i.e., tensors. For example, the storage cost of a square $n^d \times n^d$ matrix is typically n^{2d}, which can be prohibitively large as d increases. A linear TN called Tensor Train (TT) or Matrix Product State (MPS) could reduce the storage cost to approximately dn^2r^2, where r is the maximal TN-rank and usually low in practice [68, 65]. A TN representation for the MIMO Volterra system was proposed in [64]. The Alternating Linear Scheme (ALS) and the Modified ALS (MALS) algorithms were derived therein to compute the TN-cores with low TN-ranks. The MALS algorithm can search the appropriate TN-ranks automatically whereas the ALS algorithm is required to fix the TN-ranks. Both algorithms are implemented to update the TN-cores in an iterative fashion. However, a particular initial guess for the TN-cores to be estimated should be provided and there is no guarantee for both algorithms to converge to an appropriate solution in the end via the alternating iterations. In this chapter, a noniterative TN-based algorithm for MIMO Volterra system identification is proposed to overcome these drawbacks. Two tuning factors are adjustable to obtain low-rank TN-cores for the Volterra kernel coefficients. The resulting Volterra model shows better prediction accuracy than those from the ALS and MALS algorithms as illustrated in the simulation examples.

The remaining part of this chapter is organized as follows. Section 4.2 gives the tensor description of a MIMO Volterra system. The problem formulation is presented in Section 4.3. The noniterative TN-based algorithms are developed in Section 4.4. In Section 4.5, simulation examples are given to demonstrate the efficiency of the proposed algorithms. Section 4.6 summarizes this chapter.

4.2 The Tensor Description of Volterra System

The normal description of Volterra system can be found in [69] and the persistent excitation condition was derived therein and also presented in Chapter 5. Besides the normal description, we can also use a tensor description of a Volterra system to reduce memory storage and improve numerical computations. Tensors are referring to the multi-dimensional arrays in this chapter. A d-way tensor is denoted as $\boldsymbol{\mathcal{T}} \in \mathbb{R}^{n_1 \times n_2 \times \cdots \times n_d}$ where a boldface calligraphic letter is used to differentiate it from vectors and matrices. The integers $n_1, n_2, \cdots, n_d$ are the dimensions and the entries of the tensor are referred to by $\boldsymbol{\mathcal{T}}^{(i_1 i_2 \cdots i_d)}$ via d integer indices $(i_1 i_2 \cdots i_d)$. Note that a vector is a 1-way tensor and a matrix is a 2-way tensor.

Definition 2. *(The k-Mode Product [68]) For a tensor $\boldsymbol{\mathcal{T}} \in \mathbb{R}^{n_1 \times \cdots \times n_k \times \cdots \times n_d}$ and a matrix $\mathbf{U} \in \mathbb{R}^{p_k \times n_k}$, the k-mode product $\boldsymbol{\mathcal{X}} = \boldsymbol{\mathcal{T}} \times_k \mathbf{U}$ is defined as*

$$\boldsymbol{\mathcal{X}}^{(i_1 \cdots i_{k-1} j i_{k+1} \cdots i_d)} = \sum_{i_k=1}^{n_k} \mathbf{U}^{(j i_k)} \boldsymbol{\mathcal{T}}^{(i_1 \cdots i_{k-1} i_k i_{k+1} \cdots i_d)}$$

and $\boldsymbol{\mathcal{X}} \in \mathbb{R}^{n_1 \times \cdots \times n_{k-1} \times p_k \times n_{k+1} \times \cdots \times n_d}$.

Definition 3. *(Tensor Train (TT) Decomposition [68]) The d-way tensor $\boldsymbol{\mathcal{T}}$ can be represented by a linear Tensor Network (TN) such that*

$$\boldsymbol{\mathcal{T}}^{(i_1 i_2 \cdots i_d)} = \sum_{\alpha_0, \cdots, \alpha_d} \boldsymbol{\mathcal{T}}_1^{(\alpha_0 i_1 \alpha_1)} \boldsymbol{\mathcal{T}}_2^{(\alpha_1 i_2 \alpha_2)} \cdots \boldsymbol{\mathcal{T}}_d^{(\alpha_{d-1} i_d \alpha_d)}$$

where $\boldsymbol{\mathcal{T}}_1, \cdots, \boldsymbol{\mathcal{T}}_d$ are called TT-cores. Each $\boldsymbol{\mathcal{T}}_k$ is a 3-way tensor of dimensions $r_{k-1} \times n_k \times r_k$, where r_{k-1}, r_k are called the TT-ranks and $r_0 = r_d = 1$ here.

The redundancy in kernel coefficients does not affect the characteristics of the specified Volterra system whereas it allows one to represent it as a compact form. To represent the

MIMO Volterra system via the tensor description, an alternative form is [64]

$$\mathbf{y}\left(t\right)=\mathbf{V}\bar{\mathbf{u}}_t+\mathbf{w}(t) \tag{4.1}$$

where $\mathbf{V}\in\mathbb{R}^{m\times(pM+1)^d}$ contains the kernel coefficients and

$$\begin{aligned}
&\bar{\mathbf{u}}_t=\underbrace{(\mathbf{u}_t\otimes\cdots\otimes\mathbf{u}_t)}_{d\text{ terms}}\in\mathbb{R}^{(pM+1)^d}\\
&\mathbf{u}_t=\left[1,\mathbf{u}_M^{\mathrm{T}}\left(t\right)\right]^{\mathrm{T}}\in\mathbb{R}^{pM+1}\\
&\mathbf{u}_M\left(t\right)=\left[\mathbf{u}^{\mathrm{T}}\left(t\right),\cdots,\mathbf{u}^{\mathrm{T}}\left(t-M+1\right)\right]^{\mathrm{T}}\in\mathbb{R}^{pM}\\
&\mathbf{u}\left(t\right)=\left[u^{(1)}\left(t\right),\cdots,u^{(p)}\left(t\right)\right]^{\mathrm{T}}\in\mathbb{R}^{p}
\end{aligned}$$

The matrix $\mathbf{V}$ can be written into the $(d+1)$-way Volterra tensor $\boldsymbol{\mathcal{V}}$ of dimensions $(pM+1)\times\cdots\times(pM+1)\times m$ satisfying

$$\begin{aligned}
\mathbf{y}^{\mathrm{T}}\left(t\right)&=\boldsymbol{\mathcal{V}}\times_1\mathbf{u}_t^{\mathrm{T}}\times_2\mathbf{u}_t^{\mathrm{T}}\cdots\times_d\mathbf{u}_t^{\mathrm{T}}+\mathbf{w}^{\mathrm{T}}(t)\\
&=(\boldsymbol{\mathcal{V}}_1\times_2\mathbf{u}_t^{\mathrm{T}})(\boldsymbol{\mathcal{V}}_2\times_2\mathbf{u}_t^{\mathrm{T}})\cdots(\boldsymbol{\mathcal{V}}_d\times_2\mathbf{u}_t^{\mathrm{T}})+\mathbf{w}^{\mathrm{T}}(t)
\end{aligned} \tag{4.2}$$

where $\boldsymbol{\mathcal{V}}_1,\cdots,\boldsymbol{\mathcal{V}}_d$ are the TN-cores of the Volterra tensor $\boldsymbol{\mathcal{V}}$. Note that the last core $\boldsymbol{\mathcal{V}}_d$ is of dimensions $r_{d-1}\times(pM+1)\times m$ and the other cores $\boldsymbol{\mathcal{V}}_i$ are of dimensions $r_{i-1}\times(pM+1)\times r_i$ with $r_0=1$. Thus, the TN representation $\boldsymbol{\mathcal{V}}_1,\cdots,\boldsymbol{\mathcal{V}}_d$ here is actually a generalization of the TT-decomposition with the last TT-rank $r_d=m$. The $(d+1)$-way Volterra tensor, originally containing $m(pM+1)^d$ elements, can be stored with $O((d-1)r^2(pM+1)+mr(pM+1))$ elements via the TN representation, where r is the maximal TN-rank. The storage requirement can be greatly reduced if the TN-ranks are small. For a MIMO Volterra system of high degree, the TN representation in (4.2) is preferred considering a limited memory.

4.3 Problem Formulation

The MIMO Volterra system identification in this chapter is to estimate the TN representation $\boldsymbol{\mathcal{V}}_1, \cdots, \boldsymbol{\mathcal{V}}_d$ of the Volterra tensor $\boldsymbol{\mathcal{V}}$ in (4.2), given the input and output data $\{\mathbf{u}(t), \mathbf{y}(t)\}$ for $t = 1, \cdots, N$. In other words, the parameter estimation described in (4.1) will be performed in the TN representation. Before we discuss the solution in the TN representation, the original optimization problems in different scenarios are presented. Define

$$\mathbf{Y}^{\mathrm{T}} = \left(\mathbf{y}\left(1\right), \mathbf{y}\left(2\right), \cdots, \mathbf{y}\left(N\right)\right) \in \mathbb{R}^{m \times N},$$

$$\mathbf{U}^{\mathrm{T}} = \left(\bar{\mathbf{u}}_1, \bar{\mathbf{u}}_2, \cdots, \bar{\mathbf{u}}_N\right) \in \mathbb{R}^{(pM+1)^d \times N}$$

and then we have to solve the following linear equations

$$\mathbf{Y}^{\mathrm{T}} = \mathbf{V}\mathbf{U}^{\mathrm{T}} \tag{4.3}$$

from (4.1) if the noise signal $\mathbf{w}(t)$ is neglected. However, there will be modeling errors and measurement noise in practice and thus least squares estimators are used.

First, solving (4.1) can be described by the linear regression (ordinary least squares optimization) problem:

$$\min_{\mathbf{V}} \|\mathbf{Y}^{\mathrm{T}} - \mathbf{V}\mathbf{U}^{\mathrm{T}}\|_F^2 \tag{4.4}$$

Since the optimization problem (4.4) is inherently underdetermined due to the redundant kernel formulation, one may resort to the ridge regression (l_2-norm regularization) as below:

$$\min_{\mathbf{V}} \frac{1}{N} \|\mathbf{Y}^{\mathrm{T}} - \mathbf{V}\mathbf{U}^{\mathrm{T}}\|_F^2 + \lambda \|\mathbf{V}\|_F^2 \tag{4.5}$$

whose closed-form solution is usually computed as

$$\mathbf{V} = \mathbf{Y}^{\mathrm{T}}\mathbf{U}\left(\mathbf{U}^{\mathrm{T}}\mathbf{U} + N\lambda\mathbf{I}_{(pM+1)^d}\right)^{-1} \tag{4.6}$$

or

$$\mathbf{V} = \mathbf{Y}^{\mathrm{T}} \left(\mathbf{U}\mathbf{U}^{\mathrm{T}} + N\lambda \mathbf{I}_N \right)^{-1} \mathbf{U} \tag{4.7}$$

with certain $\lambda > 0$. Choosing the first expression (4.6) or the second expression (4.7) depends on how $(pM+1)^d$ and N compare. The one corresponding to the smaller between $(pM+1)^d$ and N is usually selected to reduce the storage requirement.

Finally, one may also want to emphasize the possible sparse structure of the Volterra system. This corresponds to the LASSO regression (l_1-norm regularization) as below

$$\min_{\mathbf{V}} \frac{1}{2N} \|\mathbf{Y}^{\mathrm{T}} - \mathbf{V}\mathbf{U}^{\mathrm{T}}\|_F^2 + \beta \|\mathrm{vec}\,(\mathbf{V})\|_1 \tag{4.8}$$

where $\beta > 0$ and vec$(\cdot)$ is the vectorization operator. Without much effort, (4.8) can be written into a normal LASSO regression over a vector variable as

$$\min_{\mathbf{v}} \frac{1}{2N} \|\mathrm{vec}(\mathbf{Y}^{\mathrm{T}}) - (\mathbf{U} \otimes \mathbf{I}_m)\,\mathbf{v}\|_2^2 + \beta \|\mathbf{v}\|_1 \tag{4.9}$$

where $\mathbf{v} = \mathrm{vec}(\mathbf{V})$. In general, there is no closed-form solution for (4.8) except for the univariate case [70].

In this chapter, (4.4) and (4.5) will be solved in the TN representation in a noniterative way and their numerical results will be used for comparison during the sparse Volterra system identification.

4.4 Noniterative Tensor Network based Algorithm

4.4.1 TN-based Ordinary Least Squares

The Alternating Linear Scheme (ALS) and Modified ALS (MALS) were used to solve (4.4) to determine the TN-cores in (4.2) in an iterative fashion in [64]. Both ALS and MALS estimate the TN-cores directly without referring back to the original $\mathbf{V}$. This avoids the complexity of computing the matrix inversion of $\mathbf{U}^{\mathrm{T}}$ and the infinite amount of solutions caused by the redundancy in kernel coefficients. On the other hand, the ALS method requires that the TN-ranks are fixed, which is less flexible. In addition, both methods require a particular initialization and do not guarantee the convergence to an appropriate solution.

In this subsection, a noniterative approach involving four steps, described in Algorithm 1–4 respectively, is proposed to calculate the solution for (4.4) with minimum Frobenius norm in terms of the TN representation.

Lemma 1. *If the Singular Value Decomposition (SVD) of* $\mathbf{U}^{\mathrm{T}}$ *is*

$$\mathbf{U}^{\mathrm{T}} = \mathbf{L}\mathbf{S}\mathbf{Z}^{\mathrm{T}} = \begin{bmatrix} \mathbf{L}_1 & \mathbf{L}_2 \end{bmatrix} \begin{bmatrix} \mathbf{S}_1 & \\ & \mathbf{0} \end{bmatrix} \begin{bmatrix} \mathbf{Z}_1^{\mathrm{T}} \\ \mathbf{Z}_2^{\mathrm{T}} \end{bmatrix} \tag{4.10}$$

where the diagonal entries of S_1 *consist of the nonzero singular values. Then, the solution of (4.4) with minimum Frobenius norm is*

$$\mathbf{V} = \mathbf{Y}^{\mathrm{T}}\mathbf{Z}_1\mathbf{S}_1^{-1}\mathbf{L}_1^{\mathrm{T}} \tag{4.11}$$

Proof. Follows from SVD properties. □

In order to obtain the TN representation of the matrix $\mathbf{V}$, the SVD in Lemma 1 should be carried out in the TN form. We first calculate the TN representation of $\mathbf{U}^{\mathrm{T}}$. Note that

$$\mathbf{U}^{\mathrm{T}} = \underbrace{\tilde{\mathbf{U}}^{\mathrm{T}} \odot \cdots \odot \tilde{\mathbf{U}}^{\mathrm{T}}}_{d \text{ terms}}$$

where $\odot$ denotes the column-wise Kronecker product, i.e., Khatri-Rao product, and $\tilde{\mathbf{U}}^{\mathrm{T}} = [\mathbf{u}_1, \mathbf{u}_2, \cdots, \mathbf{u}_N]$. The TN representation of $\mathbf{U}^{\mathrm{T}}$ can then be efficiently constructed by $(d-1)$

SVDs and Khatri-Rao products [65]. The upper bounds of the corresponding TN-ranks are given in Lemma 2. In practice, these upper bounds are attained if the input signals are persistently exciting. Different from [65], a δ-truncated SVD is introduced in this chapter during computing the TN representation of $\mathbf{U}^{\mathrm{T}}$ to actively remove less significant modes and reduce the computational cost. For each SVD cycle, the sum of squares of all truncated singular values is not greater than δ^2. The pseudocode following a MATLAB fashion is summarized in Algorithm 1. Note that the idea of using SVD for low-rank system identification is also used in subspace methods [71, 46, 72].

Algorithm 1: Compute the TN representation of $\mathbf{U}^{\mathrm{T}}$

```
Input: $\tilde{\mathbf{U}}^{\mathrm{T}} \in \mathbb{R}^{(pM+1)\times N}$, degree $d$, expected accuracy $\varepsilon_u$ in percentage
Output: TN-cores $\boldsymbol{\mathcal{U}}_1, \cdots, \boldsymbol{\mathcal{U}}_d$ of $\mathbf{U}^{\mathrm{T}}$
Initialization: $r_0 \leftarrow 1, r_d \leftarrow N$
$\boldsymbol{\mathcal{U}}_1 \leftarrow$ reshape($\tilde{\mathbf{U}}^{\mathrm{T}}, [r_0, pM+1, r_d]$)
for $j = 1, \cdots, d-1$ do
    $\mathbf{T} \leftarrow$ reshape($\boldsymbol{\mathcal{U}}_j, [r_{j-1}(pM+1), r_d]$)
    $\mathbf{T} \leftarrow \mathbf{T} \odot \tilde{\mathbf{U}}^{\mathrm{T}}$
    $\mathbf{T} \leftarrow$ reshape($\mathbf{T}, [r_{j-1}(pM+1), (pM+1)r_d]$)
    $[\mathbf{L}, \mathbf{S}, \mathbf{Z}] \leftarrow$ SVD($\mathbf{T}$,'econ')
    $\delta \leftarrow \varepsilon_u \|\mathbf{S}\|_F$
    $r_j \leftarrow$ numerical TN-rank determined by $\delta$-truncated SVD
    $\mathbf{L}_1 \leftarrow \mathbf{L}(:, 1 : r_j)$
    $\mathbf{S}_1 \leftarrow \mathbf{S}(1 : r_j, 1 : r_j)$
    $\mathbf{Z}_1 \leftarrow \mathbf{Z}(:, 1 : r_j)$
    $\boldsymbol{\mathcal{U}}_j \leftarrow$ reshape($\mathbf{L}_1, [r_{j-1}, pM+1, r_j]$)
    $\boldsymbol{\mathcal{U}}_{j+1} \leftarrow$ reshape($\mathbf{S}_1\mathbf{Z}_1^{\mathrm{T}}, [r_j, pM+1, r_d]$)
```

Lemma 2. *The TN-ranks of* $\mathbf{U}^{\mathrm{T}}$ *have upper bounds*

$$r_j \leq \begin{pmatrix} j + pM \\ \\ pM \end{pmatrix}, \; j = 1, \cdots, d-1$$

Proof. Similar to the proof of Theorem 4.1 in [65]. □

Since the first $(d-1)$ tensors $\boldsymbol{\mathcal{U}}_1, \cdots, \boldsymbol{\mathcal{U}}_{d-1}$ are orthogonal via the SVD in Algorithm 1, only the last tensor $\boldsymbol{\mathcal{U}}_d$ needs to be modified to get the TN-based SVD of $\mathbf{U}^{\mathrm{T}}$. This modification is summarized in Algorithm 2. Note that $\tilde{r} = \min(r_{d-1}(pM+1), N)$.

Algorithm 2: Compute the TN-based SVD of $\mathbf{U}^{\mathrm{T}}$

```
Input: TN-cores $\mathcal{U}_1, \cdots, \mathcal{U}_d$ of $\mathbf{U}^{\mathrm{T}}$
Output: TN-cores $\mathcal{L}_1, \cdots, \mathcal{L}_d$ of orthogonal matrix $\mathbf{L}$, diagonal matrix $\mathbf{S}$, and
        orthogonal matrix $\mathbf{Z}$ in (4.10)
for $j = 1, \cdots, d-1$ do
    $\mathcal{L}_j \leftarrow \mathcal{U}_j$
$\mathbf{T} \leftarrow$ reshape($\mathcal{U}_d$, $[r_{d-1}(pM+1), r_d]$)
$[\mathbf{L}, \mathbf{S}, \mathbf{Z}] \leftarrow$ SVD($\mathbf{T}$,'econ')
$\tilde{r} \leftarrow$ size($\mathbf{L}$, 2)
$\mathcal{L}_d \leftarrow$ reshape($\mathbf{L}$, $[r_{d-1}, pM+1, \tilde{r}]$)
```

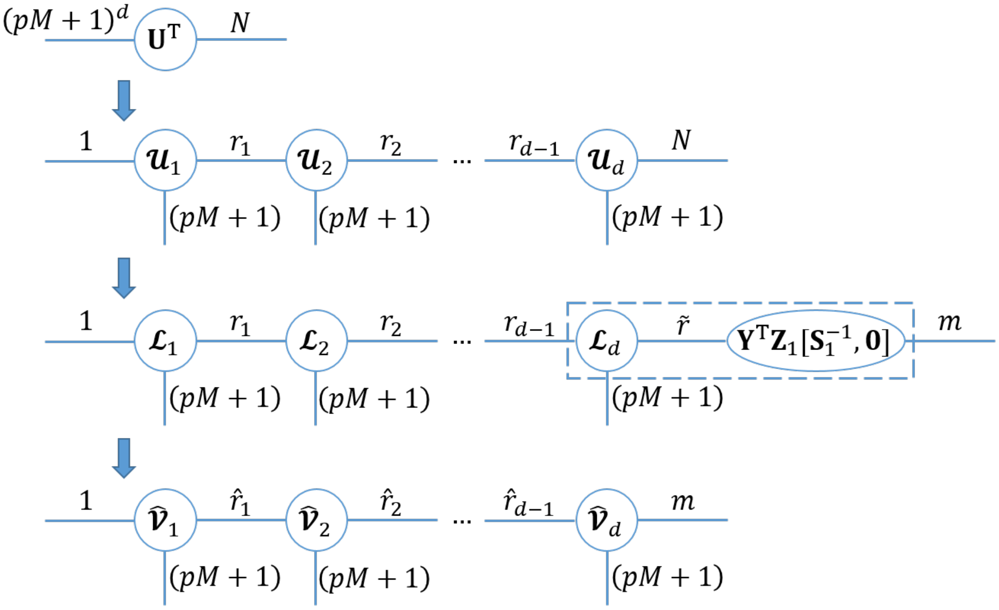


Figure 4.1: Illustration of the noniterative TN-based algorithm for estimating Volterra kernel coefficients.

Then, the TN representation of (4.11) can be written out straightforwardly as in Algorithm 3. It is worth noting that the TN-ranks of the obtained $\boldsymbol{\mathcal{V}}_1, \cdots, \boldsymbol{\mathcal{V}}_d$ are not changed and can be still very large. In practice, the TN representation of the Volterra tensor can be expected to be of low TN-ranks. Thus, a TT-rounding proposed in [68] is followed to do the TN-recompression and summarized in Algorithm 4. It reduces the ranks with a guaranteed accuracy. A δ-truncated SVD is also performed during recompression. The whole noniterative TN-based algorithm described as above for estimating the Volterra kernel coefficients is illustrated in Fig. 4.1.

Algorithm 3: Compute the TN representation of the kernel coefficients in (4.11)

```
**Input:** TN-cores $\boldsymbol{\mathcal{L}}_1, \cdots, \boldsymbol{\mathcal{L}}_d$, matrix $\mathbf{S}$ and $\mathbf{Z}$ obtained in Algorithm 2
**Output:** TN-cores $\boldsymbol{\mathcal{V}}_1, \cdots, \boldsymbol{\mathcal{V}}_d$ of the solution (4.11)
for $j = 1, \cdots, d-1$ do
    $\boldsymbol{\mathcal{V}}_j \leftarrow \boldsymbol{\mathcal{L}}_j$
$r \leftarrow$ numerical rank of $\mathbf{U}^{\mathrm{T}}$ determined from matrix $\mathbf{S}$
$\mathbf{S}_1 \leftarrow \mathbf{S}(1:r, 1:r)$
$\mathbf{Z}_1 \leftarrow \mathbf{Z}(:, 1:r)$
$\boldsymbol{\mathcal{V}}_d \leftarrow \boldsymbol{\mathcal{L}}_d \times_3 (\mathbf{Y}^{\mathrm{T}}\mathbf{Z}_1[\mathbf{S}_1^{-1}, \mathbf{0}])$
```

Algorithm 4: Perform TN-recompression [68]

```
**Input:** TN-cores $\boldsymbol{\mathcal{V}}_1, \cdots, \boldsymbol{\mathcal{V}}_d$ obtained in Algorithm 3, expected accuracy $\varepsilon$ in
           percentage
**Output:** TN-cores $\widehat{\boldsymbol{\mathcal{V}}}_1, \cdots, \widehat{\boldsymbol{\mathcal{V}}}_d$ with reduced TN-ranks
Initialization: $\delta \leftarrow \frac{\varepsilon}{\sqrt{d-1}}\|\mathbf{V}\|_F$
Right to left orthogonalization via LQ decomposition
for $j = d, d-1, \cdots, 2$ do
    $[d_1, d_2, d_3] \leftarrow \mathrm{size}(\widehat{\boldsymbol{\mathcal{V}}}_j)$
    $[\mathbf{L}, \mathbf{Q}] \leftarrow \mathrm{lq}(\mathrm{reshape}(\widehat{\boldsymbol{\mathcal{V}}}_j, [d_1, d_2 d_3]))$
    $\widehat{\boldsymbol{\mathcal{V}}}_j \leftarrow \mathrm{reshape}(\mathbf{Q}, [\mathrm{size}(\mathbf{Q}, 1), d_2, d_3])$
    $\widehat{\boldsymbol{\mathcal{V}}}_{j-1} \leftarrow \boldsymbol{\mathcal{V}}_{j-1} \times_3 \mathbf{L}$
Left to right compression via $\delta$-truncated SVD
for $j = 1, 2, \cdots, d-1$ do
    $[d_1, d_2, d_3] \leftarrow \mathrm{size}(\widehat{\boldsymbol{\mathcal{V}}}_j)$
    $[\mathbf{L}, \mathbf{S}, \mathbf{Z}] \leftarrow \mathrm{SVD}(\mathrm{reshape}(\widehat{\boldsymbol{\mathcal{V}}}_j, [d_1 d_2, d_3], \text{'econ'}))$
    $\widehat{r}_j \leftarrow$ numerical TN-rank determined by $\delta$-truncated SVD
    $\widehat{\boldsymbol{\mathcal{V}}}_j \leftarrow \mathrm{reshape}(\mathbf{L}(:, 1:\widehat{r}_j), [d_1, d_2, \widehat{r}_j])$
    $\widehat{\boldsymbol{\mathcal{V}}}_{j+1} \leftarrow \widehat{\boldsymbol{\mathcal{V}}}_{j+1} \times_1 (\mathbf{S}(1:\widehat{r}_j, 1:\widehat{r}_j)\mathbf{Z}(:, 1:\widehat{r}_j)^{\mathrm{T}})$
```

Remark 8. *The Volterra system identification can be seen as the corner case of identifying the polynomial state-space model in [65] when the linear state sequence is ignored. However, the algorithm in [65] can not directly handle the corner case. The first novelty of the proposed algorithm in this subsection is using δ-truncated SVD to actively remove less significant modes rather than the rank-gap as in [65]. The second novelty is introducing the TN-recompression.*

Remark 9. *Factors $\varepsilon_u \in [0,1)$ in Algorithm 1 and $\varepsilon \in [0,1)$ in Algorithm 4 are adjustable to reduce both computational cost and TN-ranks. Note that ε_u emphasizes the low-rank expression by actively neglecting the less significant modes in the input matrix $\mathbf{U}^{\mathrm{T}}$ when the degree d is high, whereas ε focuses on the low-rank features or sparsity of the obtained kernels during recompression.*

4.4.2 TN-based Ridge Regression

Before solving the ridge regression in the TN representation, two types of block matrix operations are introduced as below.

Definition 4. *(The Strong Kronecker Product [73]) The strong Kronecker product between a block $r_i \times r_j$ matrix $\mathbf{X} = [\mathbf{X}_{ij}]$ with $\mathbf{X}_{ij} \in \mathbb{R}^{n_x \times m_x}$ and a block $r_j \times r_k$ matrix $\mathbf{Y} = [\mathbf{Y}_{jk}]$ with $\mathbf{Y}_{jk} \in \mathbb{R}^{n_y \times m_y}$ is defined as*

$$\mathbf{X} \,|\!\otimes\!|\, \mathbf{Y} = [\mathbf{K}_{ik}] := \mathbf{K}$$

where $\mathbf{K}$ is a block $r_i \times r_k$ matrix with

$$\mathbf{K}_{ik} = \sum_{j=1}^{n_j} \mathbf{X}_{ij} \otimes \mathbf{Y}_{jk} \in \mathbb{R}^{n_x n_y \times m_x m_y}$$

Definition 5. *(The C-Product [73]) The core contracted product (C-product) between a block $r_1 \times r_2$ matrix $\mathbf{X} = [\mathbf{X}_{i_1 i_2}]$ with $\mathbf{X}_{i_1 i_2} \in \mathbb{R}^{n\times p}$ and a block $r_3 \times r_4$ matrix $\mathbf{Y} = [\mathbf{Y}_{i_3 i_4}]$ with $\mathbf{Y}_{i_3 i_4} \in \mathbb{R}^{p\times m}$ is defined as*

$$\mathbf{X} \,|\bullet|\, \mathbf{Y} = [\mathbf{Z}_{j_1 j_2}] := \mathbf{Z}$$

where $\mathbf{Z}$ is a block $r_1 r_3 \times r_2 r_4$ matrix with $\mathbf{Z}_{j_1 j_2} = \mathbf{X}_{i_1 i_2}\mathbf{Y}_{i_3 i_4} \in \mathbb{R}^{n\times m}$, $j_1 = i_1 + (i_3 - 1)r_1$, and $j_2 = i_2 + (i_4 - 1)r_2$.

As shown in (4.6) and (4.7), there are two close-form expressions for the ridge regression. If $(pM + 1)^d \gg N$ due to the curse of dimensionality, the second expression should be selected and implemented in the TN representation to alleviate the storage requirement. Otherwise, one can just select the first expression and compute it in the original matrix form. For the rest of this subsection, a noniterative approach to computing the second closed-form expression in (4.7) is derived.

The large-scale matrix $\mathbf{U}^{\mathrm{T}}$ can be approximated by the TT decomposition $\boldsymbol{\mathcal{U}}_1, \cdots, \boldsymbol{\mathcal{U}}_d$, which can be obtained in Algorithm 1. As pointed out in [73], the TT decomposition can be compactly represented as strong Kronecker products of block matrices, which leads to

$$\mathbf{U}^{\mathrm{T}} = \boldsymbol{\mathcal{U}}_1^{\mathrm{blk}} \,|\otimes| \cdots |\otimes|\, \boldsymbol{\mathcal{U}}_d^{\mathrm{blk}} \tag{4.12}$$

where the block $r_{j-1} \times r_j$ matrix $\boldsymbol{\mathcal{U}}_j^{\mathrm{blk}} = \left[\boldsymbol{\mathcal{U}}_{j,i_1 i_2}^{\mathrm{blk}}\right]$, $j \in \{1, \cdots, d-1\}$, $i_1 \in \{1, \cdots, r_{j-1}\}$, $i_2 \in \{1, \cdots, r_j\}$ with $\boldsymbol{\mathcal{U}}_{j,i_1 i_2}^{\mathrm{blk}} = \mathrm{reshape}(\boldsymbol{\mathcal{U}}_j(i_1, :, i_2), [pM + 1, 1])$. Since $r_d = N$, one can write $\boldsymbol{\mathcal{U}}_d^{\mathrm{blk}}$ into a block $r_{d-1} \times 1$ matrix instead such that $\boldsymbol{\mathcal{U}}_d^{\mathrm{blk}} = \left[\boldsymbol{\mathcal{U}}_{d,i_1 1}^{\mathrm{blk}}\right]$ with $\boldsymbol{\mathcal{U}}_{d,i_1 1}^{\mathrm{blk}} = \mathrm{reshape}(\boldsymbol{\mathcal{U}}_d(i_1, :, :), [pM + 1, N])$ for $i_1 \in \{1, \cdots, r_{d-1}\}$. The reason for using the strong Kronecker products is to simplify the computations between tensors. Similarly, the large-scale matrix $\mathbf{U}$ can be represented as strong Kronecker products of block matrices by transposing the block components of $\boldsymbol{\mathcal{U}}_j^{\text{blk}}$ in (4.12), which leads to

$$\mathbf{U} = \boldsymbol{\mathcal{U}}_1^{\text{blk,T}} \,|\otimes|\cdots|\otimes|\, \boldsymbol{\mathcal{U}}_d^{\text{blk,T}} \tag{4.13}$$

where $\boldsymbol{\mathcal{U}}_j^{\text{blk,T}} = \left[(\boldsymbol{\mathcal{U}}_{j,i_1i_2}^{\text{blk}})^{\text{T}}\right]$, $j \in \{1, \cdots, d\}$.

Proposition 4. *Given* $\mathbf{U}^{\text{T}}$ *and* $\mathbf{U}$ *represented as strong Kronecker products in (4.12) and (4.13), the matrix* $\mathbf{U}\mathbf{U}^{\text{T}} + N\lambda\mathbf{I}_N$*, whose inverse is required in (4.7), can be written in strong Kronecker products of block matrices as below*

$$\mathbf{U}\mathbf{U}^{T} + N\lambda\mathbf{I}_N = \boldsymbol{\mathcal{C}}_1^{blk} \,|\otimes|\cdots|\otimes|\, \boldsymbol{\mathcal{C}}_d^{blk} \tag{4.14}$$

where the block $r_{j-1}^{\lambda} \times r_j^{\lambda}$ *matrix* $\boldsymbol{\mathcal{C}}_j^{\text{blk}} = \left[\boldsymbol{\mathcal{C}}_{j,k_1k_2}^{\text{blk}}\right]$*,* $j \in \{1, \cdots, d\}$*,* $k_1 \in \{1, \cdots, r_{j-1}^{\lambda}\}$*,* $k_2 \in \{1, \cdots, r_j^{\lambda}\}$ *for certain* $\{r_0^{\lambda}, r_1^{\lambda}, \cdots, r_d^{\lambda}\}$*.*

Proof. The basic matrix operations can be efficiently computed by matrix TT decomposition for large-scale matrices [73]. Thus, the matrix $\mathbf{U}\mathbf{U}^{\text{T}} \in \mathbb{R}^{N\times N}$ can be computed based on the C-product from (4.12) and (4.13) as below

$$\mathbf{U}\mathbf{U}^{\text{T}} = \boldsymbol{\mathcal{W}}_1^{\text{blk}} \,|\otimes|\cdots|\otimes|\, \boldsymbol{\mathcal{W}}_d^{\text{blk}} \tag{4.15}$$

where $\boldsymbol{\mathcal{W}}_j^{\text{blk}} = \boldsymbol{\mathcal{U}}_j^{\text{blk,T}} \,|\bullet|\, \boldsymbol{\mathcal{U}}_j^{\text{blk}}$ is a block $r_{j-1}^2 \times r_j^2$ matrix for $j \in \{1, \cdots, d-1\}$ and a block $r_{d-1}^2 \times 1$ matrix for $j = d$.

On the other hand, the diagonal matrix $N\lambda\mathbf{I}_N$ can be represented as

$$N\lambda\mathbf{I}_N = \boldsymbol{\mathcal{I}}_1^{\text{blk}} \,|\otimes|\cdots|\otimes|\, \boldsymbol{\mathcal{I}}_d^{\text{blk}} \tag{4.16}$$

where the block 1×1 matrices $\boldsymbol{\mathcal{I}}_j^{\text{blk}} = [1]$ for $j \in \{1, \cdots, d-1\}$ and $\boldsymbol{\mathcal{I}}_d^{\text{blk}} = [N\lambda\mathbf{I}_N]$.

Finally, the addition of two matrices can be efficiently computed in terms of matrix TT decomposition such that

$$\begin{aligned}\mathbf{U}\mathbf{U}^{\mathrm{T}} + N\lambda\mathbf{I}_N = \begin{bmatrix}\boldsymbol{\mathcal{W}}_1^{\mathrm{blk}} \, \boldsymbol{\mathcal{I}}_1^{\mathrm{blk}}\end{bmatrix} |\otimes| \begin{bmatrix}\boldsymbol{\mathcal{W}}_2^{\mathrm{blk}} & \mathbf{0} \\ \mathbf{0} & \boldsymbol{\mathcal{I}}_2^{\mathrm{blk}}\end{bmatrix} |\otimes| \cdots \\ |\otimes| \begin{bmatrix}\boldsymbol{\mathcal{W}}_{d-1}^{\mathrm{blk}} & \mathbf{0} \\ \mathbf{0} & \boldsymbol{\mathcal{I}}_{d-1}^{\mathrm{blk}}\end{bmatrix} |\otimes| \begin{bmatrix}\boldsymbol{\mathcal{W}}_d^{\mathrm{blk}} \\ \boldsymbol{\mathcal{I}}_d^{\mathrm{blk}}\end{bmatrix}\end{aligned} \tag{4.17}$$

where one can denote

$$\boldsymbol{\mathcal{C}}_1^{\mathrm{blk}} = \begin{bmatrix}\boldsymbol{\mathcal{W}}_1^{\mathrm{blk}} \, \boldsymbol{\mathcal{I}}_1^{\mathrm{blk}}\end{bmatrix}, \, \boldsymbol{\mathcal{C}}_d^{\mathrm{blk}} = \begin{bmatrix}\boldsymbol{\mathcal{W}}_d^{\mathrm{blk}} \\ \boldsymbol{\mathcal{I}}_d^{\mathrm{blk}}\end{bmatrix}$$

and

$$\boldsymbol{\mathcal{C}}_j^{\mathrm{blk}} = \begin{bmatrix}\boldsymbol{\mathcal{W}}_j^{\mathrm{blk}} & \mathbf{0} \\ \mathbf{0} & \boldsymbol{\mathcal{I}}_j^{\mathrm{blk}}\end{bmatrix}$$

for $j \in \{2, \cdots, d-1\}$. Note that $r_0^\lambda = r_d^\lambda = 1$ and $r_j^\lambda = r_j^2 + 1$ for $j \in \{1, \cdots, d-1\}$. □

The TN-cores $\boldsymbol{\mathcal{C}}_1, \cdots, \boldsymbol{\mathcal{C}}_d$ of the matrix $\mathbf{U}\mathbf{U}^{\mathrm{T}} + N\lambda\mathbf{I}_N$ can be derived directly from (4.14). Its inverse matrix can be calculated by referring to the TN-based SVD described in Algorithm 2. Since the final representation of $\mathbf{U}\mathbf{U}^{\mathrm{T}} + N\lambda\mathbf{I}_N$ is an $N \times N$ matrix, which does not suffer the curse of dimensionality, one can actually contract all the TN-cores to recover the original matrix and then compute the matrix $\mathbf{Y}^{\mathrm{T}}(\mathbf{U}\mathbf{U}^{\mathrm{T}} + N\lambda\mathbf{I}_N)^{-1}$ without referring to its dual TN representation. To get the TN-cores $\boldsymbol{\mathcal{V}}_1, \cdots, \boldsymbol{\mathcal{V}}_d$ of the final ridge solution as described in (4.6), one needs to connect the matrix $\mathbf{Y}^{\mathrm{T}}(\mathbf{U}\mathbf{U}^{\mathrm{T}} + N\lambda\mathbf{I}_N)^{-1}$ to the TN representation of $\mathbf{U}$, which can be done in a similar way as for (4.11). The TN-ranks will increase during matrix operations, so the TN-recompression is also required in the next step to obtain $\widehat{\boldsymbol{\mathcal{V}}}_1, \cdots, \widehat{\boldsymbol{\mathcal{V}}}_d$ with reduced TN-ranks. Computing the TN-based ridge solution is summarized in Algorithm 5.

Algorithm 5: TN-based ridge regression

```
Input: TN-cores $\boldsymbol{\mathcal{U}}_1, \cdots, \boldsymbol{\mathcal{U}}_d$ of $\mathbf{U}^{\mathrm{T}}$ via Algorithm 1, $l_2$-norm regularization
       coefficient $\lambda$, expected accuracy $\varepsilon$ in percentage during TN-recompression
Output: TN-cores $\widehat{\boldsymbol{\mathcal{V}}}_1, \cdots, \widehat{\boldsymbol{\mathcal{V}}}_d$ of the ridge solution in (4.7) with reduced TN-ranks
$\boldsymbol{\mathcal{U}}_1^{\mathrm{blk}}, \cdots, \boldsymbol{\mathcal{U}}_d^{\mathrm{blk}} \leftarrow$ determined by (4.12)
$\boldsymbol{\mathcal{U}}_1^{\mathrm{blk,T}}, \cdots, \boldsymbol{\mathcal{U}}_d^{\mathrm{blk,T}} \leftarrow$ determined by (4.13)
for $j = 1, 2, \cdots, d$ do
    $\boldsymbol{\mathcal{W}}_j^{\mathrm{blk}} \leftarrow \boldsymbol{\mathcal{U}}_j^{\mathrm{blk,T}} |\bullet| \boldsymbol{\mathcal{U}}_j^{\mathrm{blk}}$
    $\boldsymbol{\mathcal{I}}_j^{\mathrm{blk}} \leftarrow [1]$
    $\boldsymbol{\mathcal{V}}_j \leftarrow \boldsymbol{\mathcal{U}}_j$
$\boldsymbol{\mathcal{I}}_d^{\mathrm{blk}} \leftarrow [N\lambda \mathbf{I}_N]$
$\boldsymbol{\mathcal{C}}_1^{\mathrm{blk}}, \cdots, \boldsymbol{\mathcal{C}}_d^{\mathrm{blk}} \leftarrow$ determined by (4.17)
$\mathbf{C} \leftarrow \boldsymbol{\mathcal{C}}_1^{\mathrm{blk}} |\otimes| \cdots |\otimes| \boldsymbol{\mathcal{C}}_d^{\mathrm{blk}}$
$\boldsymbol{\mathcal{V}}_d \leftarrow \boldsymbol{\mathcal{V}}_d \times_3 (\mathbf{Y}^{\mathrm{T}} \mathbf{C}^{-1})$
$\widehat{\boldsymbol{\mathcal{V}}}_1, \cdots, \widehat{\boldsymbol{\mathcal{V}}}_d \leftarrow$ TN-recompression by Algorithm 4
```

4.5 Simulation Results

All simulations in this section were performed on an Intel Core i5-10210U CPU with 16 GB RAM.

4.5.1 SISO Volterra System

The SISO Volterra system with kernel coefficients being decaying exponentials in [64] is used here for comparison. The degree $d = 10$ and the memory length $M = 7$. Thus, $pM + 1 = 8$. Denote the noniterative TN-based algorithm proposed in Section 4.4.1 to solve the linear regression (4.4) as NIteTN and that proposed in Section 4.4.2 to solve the ridge regression (4.5) as ridgeTN. As in [64], the first 700 samples of the input/output signals are used for system identification and the 4300 remaining data points are used for model validation. The output signal for identification is contaminated by Gaussian white noise with a SNR of 25dB. Denote the relative residue of the output signals as

$$res = \frac{\|\mathbf{Y} - \widehat{\mathbf{Y}}\|_F}{\|\mathbf{Y}\|_F} \tag{4.18}$$

where $\mathbf{Y}$ and $\widehat{\mathbf{Y}}$ are the practical and simulated output signals defined as in (4.3). As suggested in Remark 9, NIteTN and ridgeTN algorithms are implemented with $\varepsilon_u = 0.19$ and $\varepsilon = 0.01$ to obtain a Volterra model with both good prediction performance and low TN-ranks. In addition, the regularization coefficient for ridgeTN is set as $\lambda = 1$. Two iterative algorithms proposed in [64], i.e., MALS and ALS algorithms, are implemented for comparison. The TN-ranks determined by the MALS algorithm are used for the ALS algorithm where fixed TN-ranks are required. The tolerance on the relative residual res should be specified for both MALS and ALS algorithms to terminate the iterations. There is a trade-off: if res is set too large, the fitting accuracy will be low; if res is set too small, it is more likely to be overfitting since the resulting models also tend to fit the noise during system identification. Setting the residue threshold $res = 0.06$ gives the best performance while maintaining low TN-ranks for this example in our simulation. 100 Monte Carlo simulations are performed and the results are showing in Fig. 4.2. The y-axis is the relative residue between the true and simulated outputs. For the x-axis, the left four are for identification data and the right four are for validation data. Table. 4.1 shows the estimated TN-ranks for $\widehat{\boldsymbol{\mathcal{V}}}_1, \cdots, \widehat{\boldsymbol{\mathcal{V}}}_{10}$ and the average run time for four algorithms.

The TN-ranks for all four algorithms are small, whereas they are determined in different ways. The training performances are all acceptable. However, the MALS and ALS

Table 4.1: TN-ranks and run time for the SISO example.

	$\widehat{r}_1$	$\widehat{r}_2$	$\widehat{r}_3$	$\widehat{r}_4$	$\widehat{r}_5$	$\widehat{r}_6$	$\widehat{r}_7$	$\widehat{r}_8$	$\widehat{r}_9$	Run time [s]
NIteTN	6	7	7	8	9	9	9	9	8	0.095
MALS	8	8	8	8	8	8	8	8	8	0.849
ALS	8	8	8	8	8	8	8	8	8	0.075
ridgeTN	6	7	7	8	9	9	9	9	8	2.386

algorithms are not performing well during validation. One explanation is that both MALS and ALS algorithms are fitting the data in an iterative fashion and convergence to an appropriate solution is not guaranteed. In contrast, the NIteTN and ridgeTN algorithms intrinsically give solutions in Least Squares sense where the statistical features of the noise have been taken into account. The ridgeTN algorithm show slightly better prediction accuracy by using the regularization. The time costs shown in Table. 4.1 illustrate that the ridgeTN algorithm is the most time-consuming since more TN operations are involved.

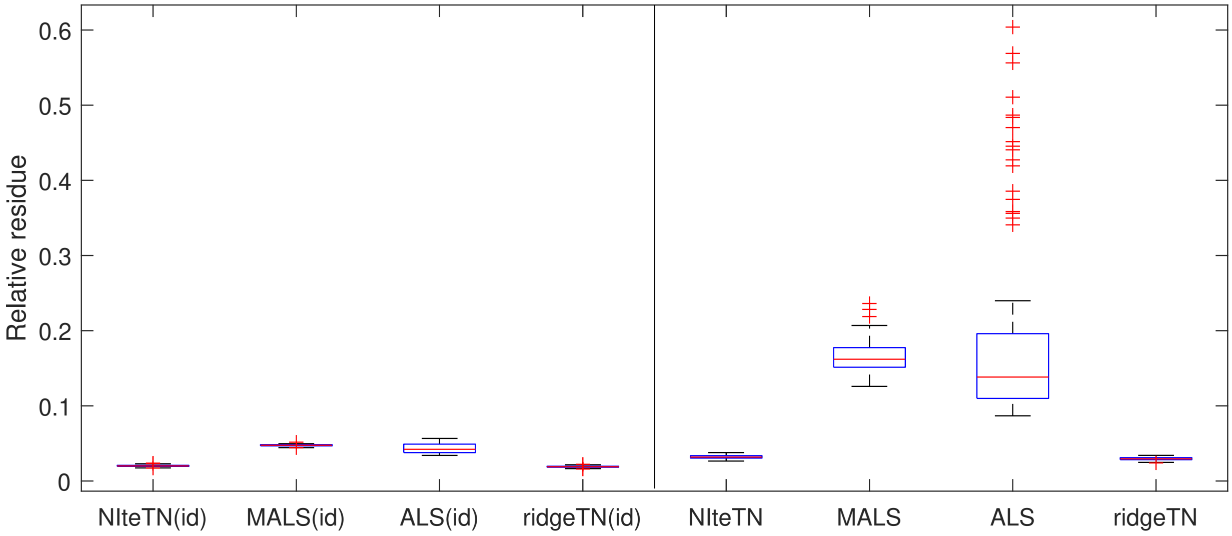


Figure 4.2: The box plot of the results of 100 Monte Carlo experiments for the SISO Volterra system identification. The symbol '+' denotes the outlier.

4.5.2 MIMO Volterra System with Sparse Kernels

Consider a 2-input 2-output Volterra system of degree $d = 4$

$$\mathbf{y}\left(t\right) = \begin{bmatrix} y^{(1)}\left(t\right) \\ y^{(2)}\left(t\right) \end{bmatrix}$$

where

$$\begin{aligned} y^{(1)}\left(t\right) =& \beta_{11}u^{(1)}\left(t\right) + \beta_{12}u^{(1)}(t)^3 + \beta_{13}u^{(1)}(t)^4 + \beta_{14}u^{(2)}(t-3)^4 \\ y^{(2)}\left(t\right) =& \beta_{21}u^{(2)}\left(t\right) + \beta_{22}u^{(2)}(t)^2 + \beta_{23}u^{(2)}\left(t-1\right)u^{(1)}\left(t-2\right) + \beta_{24}u^{(1)}(t-3)^4 \end{aligned}$$

and $\beta_{11} = \beta_{21} = 1, \beta_{12} = \beta_{22} = 0.1, \beta_{13} = \beta_{23} = 0.01$, and $\beta_{14} = \beta_{24} = 0.05$. The memory length $M = 4$. Note that the corresponding matrix $\mathbf{V}$ in (4.1) containing the kernel coefficients is sparse. Also, $pM + 1 = 9$ in this example. The Gaussian white signals with covariance matrix $\sigma^2 = \mathbf{I}$ are used as input signals. The input signals are scaled to be within $[-1, 1]$. The first 700 samples of the total 5000 generated input/output data are used for system identification and the remaining 4300 samples are for validation as in Section 4.5.1.

The NIteTN, MALS, ALS, ridgeTN, and LASSO algorithms are implemented for comparison. As suggested in Remark 9, NIteTN is implemented with $\varepsilon_u = 0.001$ and $\varepsilon = 0.05$. Note that smaller ε_u helps maintain more input excitation information whereas larger ε emphasizes the sparsity within the kernel coefficients. Similarly, the TN-ranks determined by the MALS algorithm are used for the ALS algorithm. The residue threshold $res = 0.1$ is selected, which takes into account the trade-off between the prediction accuracy and low TN-ranks. Since the TN-ranks will increase in a square sense temporarily during running TN-based ridge regression, one should select a larger ε_u to simplify the TN representation at the beginning which incurs more approximation errors for Volterra coefficients. Thus,

$\varepsilon_u = 0.4$ and $\varepsilon = 0.05$ are selected to compute the TN-based ridge solution. The l_2-norm regularization coefficient $\lambda = 10^{-4}$ is used to penalize the Volterra coefficients. The estimation may be biased, but the variance may also be reduced. Larger λ will smooth out more Volterra coefficients. The LASSO regularization coefficient $\beta = 10^{-6}$.

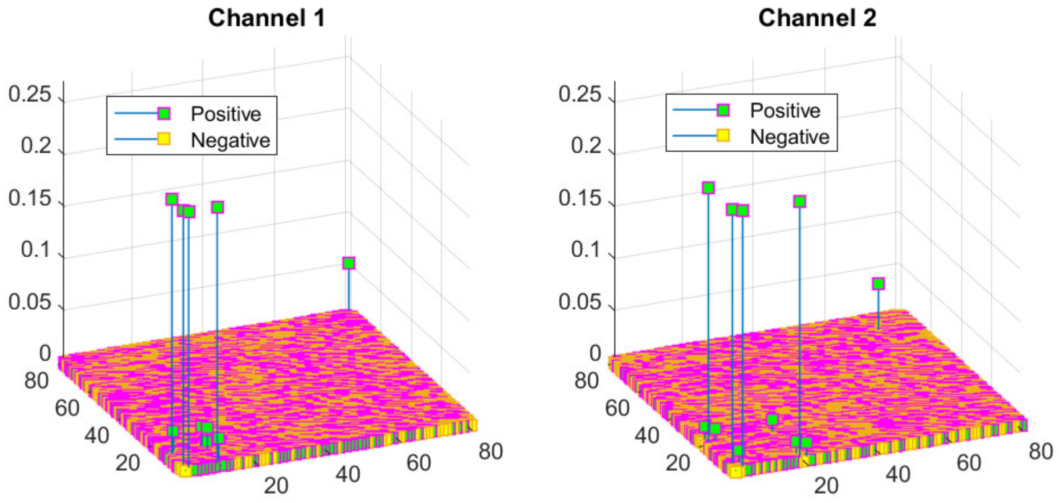


Figure 4.3: The estimated kernel coefficients via the NIteTN algorithm.

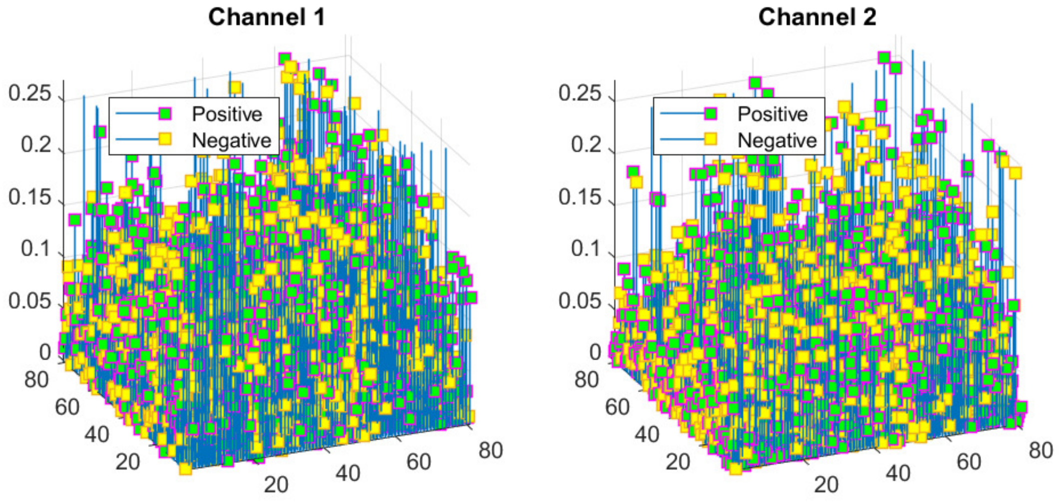


Figure 4.4: The estimated kernel coefficients via the MALS algorithm.

The estimated TN-ranks for $\widehat{\boldsymbol{\mathcal{V}}}_1, \cdots, \widehat{\boldsymbol{\mathcal{V}}}_4$ and the run time from one simulation are

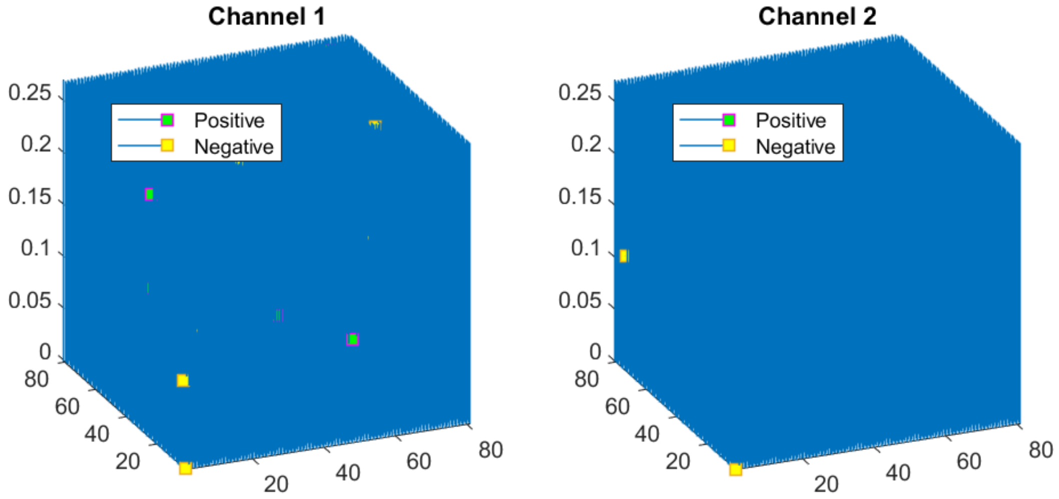


Figure 4.5: The estimated kernel coefficients via the ALS algorithm.

Table 4.2: TN-ranks and prediction errors for the MIMO example.

	$\widehat{r}_1$	$\widehat{r}_2$	$\widehat{r}_3$	Run time [s]	res (Train)	res (Validation)
NIteTN	5	6	6	0.453	0.003	0.003
MALS	9	9	9	0.684	0.043	0.122
ALS	9	9	9	0.242	0.040	0.201
ridgeTN	6	8	10	5.211	0.017	0.023
LASSO	-	-	-	2.450	4e-4	6e-4

shown in Table. 4.2. The LASSO algorithm shows the most accurate prediction, because it is based on an optimization over the original data while the TN-based algorithms imply approximations of the data matrices. Among the TN-based algorithms, the proposed NIteTN algorithm performs best during both training and validation periods in terms of low TN-ranks and accurate output prediction. Both MALS and ALS algorithms stabilize to relatively low TN-ranks and produce small prediction errors for the training data. However, their validation errors are relatively large since the iterative algorithms do not guarantee to converge to an appropriate solution. The proposed NIteTN algorithm is noniterative and will not suffer from such problems due to iterations. In addition, the parameter ε emphasizes the inherent

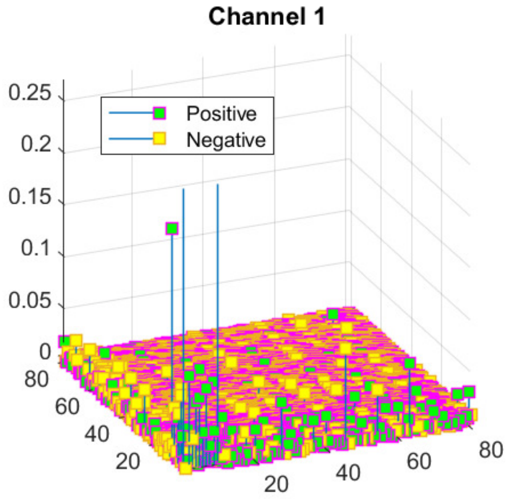


Figure 4.6: The estimated kernel coefficients via the ridgeTN algorithm.

sparsity of the given Volterra system and results in even lower TN-ranks than both MALS and ALS algorithms. The training and validation errors for the model from the ridgeTN algorithm are almost at the same level since the l_2-norm regularization will help reduce the overfitting.

Fig. 4.3-4.7 illustrate the estimated kernel coefficients for different algorithms. There are $(pM + 1)^d = 6561$ kernel coefficients in total for each output channel and they are reshaped into two 81×81 matrices. For better illustration, each kernel coefficient is shown using its absolute value with a positive or negative indicator. For the TN-based algorithms, the performance of the sparse estimation of the inherent kernel coefficients is different. Although both MALS and ALS algorithms result in a model with relatively low TN-ranks for $\boldsymbol{\mathcal{V}}_1, \cdots, \boldsymbol{\mathcal{V}}_4$, its dual representation $\mathbf{V}$ which contains the estimated kernel coefficients is not sparse. One can observe from Fig. 4.4 and Fig. 4.5 that the iterative algorithms suffer from overfitting problems. The ALS algorithm encounters more overfitting since it is refining the parameters to further reduce the prediction errors based on the initial estimation from the

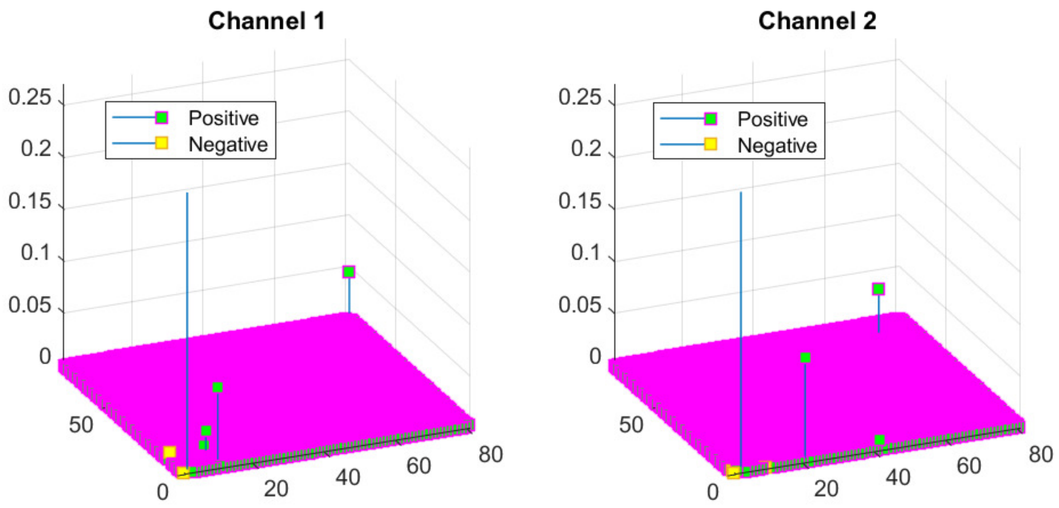


Figure 4.7: The estimated kernel coefficients via the LASSO regression.

MALS algorithm. The NIteTN algorithm not only shows low TN-rank approximation, but also maintains the sparsity of its dual representation as shown in Fig. 4.3. The distribution of trivial kernel coefficients is very flat around zero. The distribution of the estimated kernel coefficients by the ridgeTN is fluctuated as shown in Fig. 4.6 since more approximation errors are introduced by setting a large ε_u. However, the l_2-norm regularization has helped smooth out insignificant kernel coefficients. Keeping increasing λ will finally force the estimates of all the kernel coefficients to zero. As expected, the LASSO algorithm produces the best sparse estimation as shown in Fig. 4.7 and the distribution of trivial kernel coefficients is even smoother than that in Fig. 4.3.

In terms of the storage requirement, all the kernel coefficients should be explicitly listed in the LASSO regression, which will be likely infeasible as the degree is getting higher. The TN-based algorithms perform all the calculations based on the TN-representation without referring back to its dual matrix representation which suffers from the curse of dimensionality. In addition, it is interesting to observe that the NIteTN algorithm implies a symmetric kernel constraint as shown in Fig. 4.3, while the LASSO algorithm does not obey this constraint during optimization.

It is worth noting that the sparsity of the original kernel model is not guaranteed even though a low TN-rank model can be obtained. In other words, all four TN-based algorithms may not regenerate the original sparse kernel coefficients in general. However, the TN-based algorithms allow one to seek the sparsity in the sense of simple TN representation which can still give good prediction. When the data is noise contaminated and different amount of data points N are used, the comparison of validation errors among the four different TN-based algorithms is shown in Fig. 4.8. All the settings are the same except that $\varepsilon_u = 0.4$ for the NIteTN to make it more robust to noise. It is interesting to observe that the ridgeTN algorithm can improve the prediction performance of the NIteTN algorithm when the SNR is large. The MALS and ALS algorithms are more likely to fit the noise and result in a less accurate model with even higher TN-ranks. For example, for a SNR of 15dB and $N = 700$, the estimated TN-ranks $(\widehat{r}_1, \widehat{r}_2, \widehat{r}_3)$ are: NIteTN (6,8,13), MALS (9,77,18), ALS (9,77,18), ridgeTN (6,8,13).

To illustrate the effect of different pairs of the hyperparameters $(\varepsilon_u, \varepsilon)$, the estimated TN-ranks and the average relative residues from the NIteTN algorithm are shown in Table. 4.3 for a SNR of 25dB and $N = 700$. The tuning factor ε_u plays a more important role in actively reducing the TN-ranks whereas ε will reduce the TN-ranks once the TN-representation could be further simplified. In addition, low TN-ranks also mean low computational complexities.

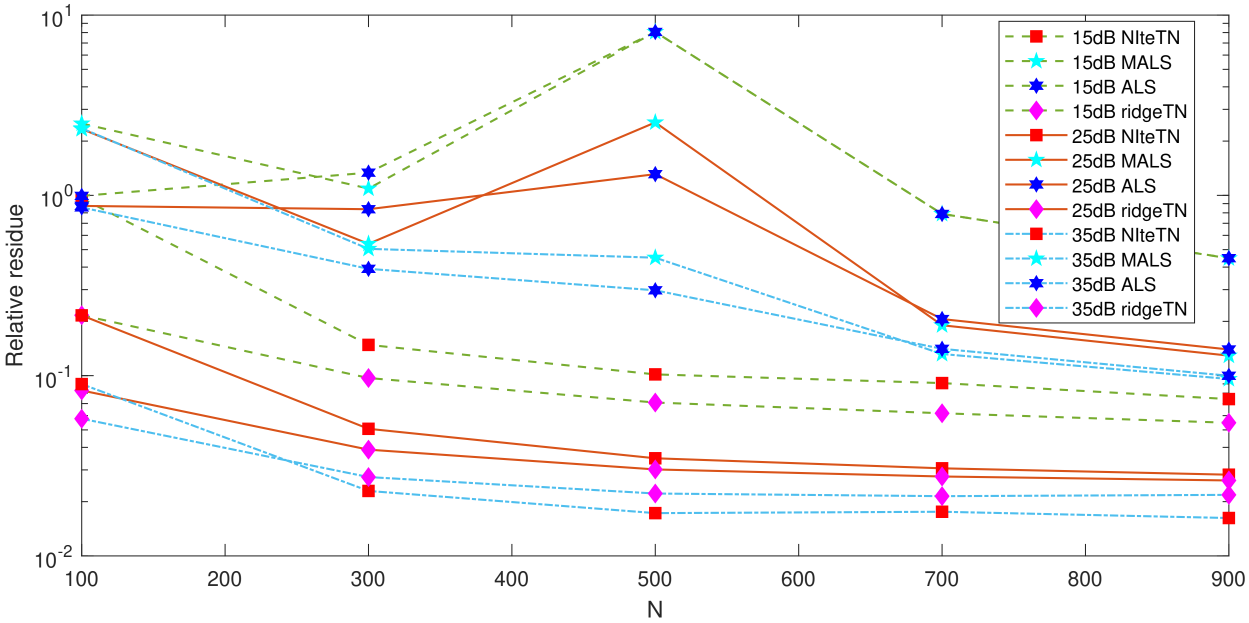


Figure 4.8: Prediction errors for different TN-based algorithms.

Table 4.3: TN-ranks and prediction errors for different $(\varepsilon_u, \varepsilon)$.

$(\varepsilon_u, \varepsilon)$	$\widehat{r}_1$	$\widehat{r}_2$	$\widehat{r}_3$	Run time [s]	res (Train)	res (Validation)
$(0.001, 0.01)$	9	45	18	0.369	0.047	0.246
$(0.01, 0.01)$	9	45	18	0.346	0.047	0.244
$(0.1, 0.01)$	9	41	18	0.264	0.047	0.245
$(0.2, 0.01)$	9	32	18	0.150	0.042	0.145
$(0.4, 0.01)$	6	8	13	0.031	0.023	0.029
$(0.6, 0.01)$	2	2	2	0.010	0.050	0.052
$(0.001, 0.05)$	9	45	18	0.369	0.047	0.246
$(0.01, 0.05)$	9	43	18	0.332	0.053	0.249
$(0.1, 0.05)$	9	40	18	0.265	0.051	0.249
$(0.2, 0.05)$	8	27	18	0.150	0.201	0.243
$(0.4, 0.05)$	5	7	10	0.031	0.027	0.031
$(0.6, 0.05)$	2	2	2	0.011	0.049	0.051
$(0.001, 0.1)$	9	39	18	0.344	0.075	0.252
$(0.01, 0.1)$	9	40	18	0.334	0.074	0.251
$(0.1, 0.1)$	9	41	18	0.269	0.142	0.289
$(0.2, 0.1)$	9	27	18	0.147	0.940	0.955
$(0.4, 0.1)$	5	7	10	0.031	0.037	0.040
$(0.6, 0.1)$	2	2	2	0.010	0.047	0.050

4.6 Conclusion

Noniterative TN-based algorithms for either a linear or ridge regression are proposed in this chapter to estimate kernel coefficients of a MIMO Volterra model. It is demonstrated to have an advantage over the ALS and MALS methods by avoiding the need for iterations. Two adjustable factors in the proposed algorithms help estimate a TN representation of the Volterra kernel coefficients with both good prediction accuracy and low TN-ranks. The TN technique relieves the curse of dimensionality and makes it possible to approximate complicated coupled nonlinear dynamics via a high degree and even MIMO Volterra model. Note that the TN representation is an approximation of the original kernel coefficients. The simulation examples show the numerical efficiency and storage saving of the proposed noniterative algorithms, and illustrate their potential for sparse Volterra system identification.

This chapter is based on the following paper that were submitted:

Y. Hu, L. Tan and R.A. de Callafon, "Noniterative tensor network-based estimation for MIMO Volterra system identification," *IEEE Transactions on Automatic Control*, under review.

Chapter 5

Persistent Excitation Condition with Gaussian Distributed Input Signals

This chapter gives a persistent excitation condition for the parameter estimation in MIMO Volterra system identification in the case of zero mean, Gaussian distributed (not necessarily white) input signals. The persistent excitation condition shows that under those input conditions a MIMO Volterra system can be identified consistently via an appropriately sized input signal.

5.1 Introduction

The nonlinear systems to be modelled by a MIMO Volterra structure are usually assumed to be persistently excited without further discussion in most of relevant contributions. However, this assumption does not hold if the input signals are not appropriately selected. For example, a Pseudo-Random Binary Sequence (PRBS), which is widely used for linear system identification due to its similar spectrum to a white noise and easy realization, is inappropriate for nonlinear system identification in general [74]. More precisely, PRBS is not persistently exciting for Volterra systems of degree higher than one [75, 76]. The system is assumed to have symmetric Volterra kernel coefficients without loss of generality.

In this chapter, a symmetric constraint on the Volterra kernel coefficients is enforced to handle the redundancy. In fact, the proposed persistent condition still holds as long as the redundant part is constrained. The idea of moments to cumulants conversion in [77] is extended to the MIMO case. It is shown in this chapter that each moments equation can be equivalently split into several cumulants equations, from which we can appropriately select one to simplify the original least squares problem. The given persistent excitation condition is very easy to satisfy in practice and further demonstrates the persistence of excitation of the Gaussian input signal for the MIMO Volterra model system identification. The derived persistent excitation condition implies that a persistently exciting Gaussian signal will not lose this property by feeding it through a standard linear filter, whose transfer function can only have finite number of zeros on the unit circle.

The remaining part of this chapter is organized as follows. Section 5.2 provides the notation and necessary preliminaries on the properties of the joint cumulant. The normal description of Volterra system is described in Section 5.3. In Section 5.4, the moments to cumulants conversion is applied to reformulate the problem and the persistent excitation condition with the zero mean Gaussian input is given. Section 5.5 investigates a simulation example to demonstrate the proposed persistent excitation condition. Section 5.6 summarizes this chapter.

5.2 Notations and Preliminaries of Cumulant

For the reference and convenience of the readers, only the properties of cumulants relevant to this chapter will be listed in this section. Most notations are following the convention in [78].

5.2.1 Notations

We use $\{\cdot\}$ to represent a set. Denote $\mathscr{P}(b)$ as the set of partitions of a finite nonempty set b. An element π of $\mathscr{P}(b)$ is collection of nonempty and disjoint subsets (i.e., blocks) of b such that their union equals b. $|b|$ denotes the number of all possible partitions. For example, $\{\{1\},\{2\},\{3\}\}$ is an element of $\mathscr{P}(\{1,2,3\})$. For every $\sigma,\pi \in \mathscr{P}(b)$, $\sigma \vee \pi$ denotes the join of them. Each block of $\sigma \vee \pi$ is the union of all the blocks of σ and π that have at least one component in common. Thus, each component of $\sigma \vee \pi$ can be seen as the least upper bound with respect to certain overlapped components of both σ and π. Note that $\sigma \vee \pi$ is also an element of $\mathscr{P}(b)$. The trivial partition $\hat{1} = \{b\}$ is the maximal element of $\mathscr{P}(b)$. The minimal element of $\mathscr{P}(b)$ is the partition $\hat{0}$, each block of which consists of exactly one component of b.

For $n \geq 1$, we denote $\mathbf{X}_{[n]} = (X_1, \cdots, X_n)$ as a vector of real-valued random variables such that the moments $\mathbb{E}|X_j|^n < \infty, \forall j = 1, \cdots, n$. Note that it is not necessary to be a series generated from a single stochastic process. Without causing much ambiguity in the following sections, two notations respectively with subscript and superscript are defined as below:

$$\mathbf{X}_b = (X_{j_1}, \cdots, X_{j_k}), \quad \mathbf{X}^b = X_{j_1} \cdots X_{j_k}$$

where the set $b = \{j_1, \cdots, j_k\} \subseteq [n] = \{1, \cdots, n\}$, $\mathbf{X}_b$ is a vector, and $\mathbf{X}^b$ is the product of all components.

5.2.2 Properties of Cumulant

The joint cumulant of $\mathbf{X}_b$ is denoted by $\operatorname{cum}(\mathbf{X}_b)$ and defined as

$$\operatorname{cum}(\mathbf{X}_b) = \operatorname{cum}(X_{j_1}, \cdots, X_{j_k}) = (-i)^k \frac{\partial^k}{\partial\theta_1 \cdots \partial\theta_k} \ln \mathbb{E}\left[\exp\left(i \sum_{l=1}^{k} \theta_l X_{j_l}\right)\right]\Bigg|_{\theta_1=\cdots=\theta_k=0}$$

where i is the imaginary unit for a complex number. If all k random variables are the same, then it is actually the kth ordinary cumulant.

Lemma 3. *(i) The joint cumulant with one random variable equals its mean and that of two random variables equals their covariance;*

(ii) The application of $\mathbf{X}_b \mapsto \operatorname{cum}(\mathbf{X}_b)$ is homogeneous:

$$\operatorname{cum}(h_1 X_{j_1}, \cdots, h_k X_{j_k}) = \left(\prod_{l=1}^{k} h_l\right) \operatorname{cum}(\mathbf{X}_b) \tag{5.1}$$

(iii) The joint cumulant is multilinear:

$$\operatorname{cum}(X_{j_1}, \cdots, X_{j_k}, Z_1 + Z_2) = \operatorname{cum}(X_{j_1}, \cdots, X_{j_k}, Z_1) + \operatorname{cum}(X_{j_1}, \cdots, X_{j_k}, Z_2) \tag{5.2}$$

(iv) If $Y = \{Y_j, j \in \mathbb{N}\}$ is a Gaussian family and $\mathbf{X}_{[n]}$ is a vector obtained by juxtaposing $n \geq 3$ elements of Y (with possible repetitions), then $\operatorname{cum}\left(\mathbf{X}_{[n]}\right) = 0$.

Proof. See [78] and [79] for reference. □

The following lemma presents two important relations between higher moments and cumulants. It shows a potential way to perform certain transformations to simplify a relevant problem.

Lemma 4. *(i) The higher moments can be expressed as the sum of products of cumulants over all possible partitions:*

$$\mathbb{E}\left[\mathbf{X}^{b}\right] = \sum_{\pi=\{b_1,\cdots,b_k\}\in\mathscr{P}(b)} \mathrm{cum}\left(\mathbf{X}_{b_1}\right)\cdots\mathrm{cum}\left(\mathbf{X}_{b_k}\right) \tag{5.3}$$

(ii) For any partition $\pi = \{b_1, \cdots, b_k\} \in \mathscr{P}(b)$, a relation between cumulants is

$$\mathrm{cum}\left(\mathbf{X}^{b_1}, \cdots, \mathbf{X}^{b_k}\right) = \sum_{\substack{\sigma=\{t_1,\cdots,t_s\}\in\mathscr{P}(b),\\ \sigma\vee\pi=\hat{1}}} \mathrm{cum}\left(\mathbf{X}_{t_1}\right)\cdots\mathrm{cum}\left(\mathbf{X}_{t_s}\right) \tag{5.4}$$

Proof. See Proposition 3.1 in [78]. □

We illustrate the results of the above Lemmas with an example.

Example 1. *Consider a simple set $b = \{1,2,3,4,5\}$. If the partitions $\pi = \{\{1\},\{2\},\{3,4,5\}\}$ and $\sigma = \{\{1,3\},\{2,4\},\{5\}\}$, then $\sigma\vee\pi = \hat{1} = \{b\}$. If the partitions $\pi = \{\{1\},\{2\},\{3\},\{4,5\}\}$ and $\sigma = \{\{1,4\},\{2,5\},\{3\}\}$, then $\sigma\vee\pi = \{\{1,2,4,5\},\{3\}\} \neq \hat{1}$.*

These properties in Lemma 3 and Lemma 4 will be repeatedly used in the following analysis.

5.3 The Normal Description of Volterra System

A discrete-time single-input single-output (SISO) Volterra model of degree d can be described as

$$y\left(t\right) = h_0 + \sum_{i=1}^{d} \sum_{k_1,\cdots,k_i=0}^{M_i-1} h_i\left(k_1,\cdots,k_i\right)\prod_{j=1}^{i} u\left(t-k_j\right) + v\left(t\right) \tag{5.5}$$

where $\{u(t), y(t)\}$ are the scalar input and output sampling signals, $v(t)$ is an independent noise signal, M_i is the memory length, and $h_i(\cdot)$ is called the ith Volterra kernel. Although we are investigating the persistent excitation condition for a MIMO Volterra system, it is enough to consider a MISO one as below

$$y\left(t\right)=h_0+\sum_{i=1}^{d}\sum_{k_1,\cdots,k_i=0}^{M_i-1}\sum_{\alpha_1,\cdots,\alpha_i=0}^{p-1}h_i\left(k_1,\alpha_1;\cdots;k_i,\alpha_i\right)\prod_{j=1}^{i}u^{\left(\alpha_i+1\right)}\left(t-k_j\right)+v\left(t\right) \tag{5.6}$$

where $u^{(\alpha)}(t)$ is the αth component of the vector input signal $\mathbf{u}(t)\in\mathbb{R}^p$. We assume that the Volterra kernels are symmetric sequences, i.e.,

$$h_i\left(k_1,\alpha_1;\cdots;k_i,\alpha_i\right)=h_i\left(k_{\rho(1)},\alpha_{\rho(1)};\cdots;k_{\rho(i)},\alpha_{\rho(i)}\right) \tag{5.7}$$

where $\rho\left(\cdot\right)$ is an arbitrary permutation of $\{1,\cdots,i\}$. Obviously there is a redundancy in the unknown kernel parameters due to the symmetric structure. However, this redundancy enables us to rewrite the model into a neat linear regression form via Kronecker product as follows:

$$y\left(t\right)=\mathbf{\Phi}^{\mathrm{T}}\left(t\right)\mathbf{H}+v\left(t\right) \tag{5.8}$$

The regressor vector $\mathbf{\Phi}(t)$ is defined as

$$\mathbf{\Phi}\left(t\right)=\left[1,\varphi_1^{\mathrm{T}}\left(t\right),\cdots,\varphi_d^{\mathrm{T}}\left(t\right)\right]^{\mathrm{T}}\in\mathbb{R}^{1+\sum\limits_{i=1}^{d}\left(pM_i\right)^i}$$

where

$$\begin{aligned}\varphi_i\left(t\right)&=\underbrace{\mathbf{u}_{M_i}\left(t\right)\otimes\cdots\otimes\mathbf{u}_{M_i}\left(t\right)}_{i\,\text{terms}}\in\mathbb{R}^{\left(pM_i\right)^i}\\ \mathbf{u}_{M_i}\left(t\right)&=\left[\mathbf{u}^{\mathrm{T}}\left(t\right),\cdots,\mathbf{u}^{\mathrm{T}}\left(t-M_i+1\right)\right]^{\mathrm{T}}\\ \mathbf{u}\left(t\right)&=\left[u^{(1)}\left(t\right),\cdots,u^{(p)}\left(t\right)\right]^{\mathrm{T}}\end{aligned}$$

The coefficients vector $\mathbf{H}$ is defined as

$$\mathbf{H}=\left[h_0,\mathbf{H}_1^{\mathrm{T}},\cdots,\mathbf{H}_d^{\mathrm{T}}\right]^{\mathrm{T}}\in\mathbb{R}^{1+\sum\limits_{i=1}^{d}\left(pM_i\right)^i}$$

where

$$\begin{aligned}\mathbf{H}_i = & \left[h_i\left(0,0;\cdots;0,0\right), h_i\left(0,0;\cdots;0,1\right),\right.\\ & \left.\cdots, h_i\left(M_i-1,p-1;\cdots;M_i-1,p-1\right)\right]^{\mathrm{T}}\end{aligned}$$

If the stationary stochastic signal is chosen to be the excitation (input) signal, the model in (5.8) leads to the following equivalent expression

$$\mathbf{R}_{\Phi\Phi}\mathbf{H} = \mathbf{R}_{\Phi y} \tag{5.9}$$

where $\mathbf{R}_{\Phi\Phi} = \mathbb{E}\left[\mathbf{\Phi}\left(t\right)\mathbf{\Phi}^{\mathrm{T}}\left(t\right)\right]$ and $\mathbf{R}_{\Phi y} = \mathbb{E}\left[\mathbf{\Phi}\left(t\right)y\left(t\right)\right]$ denote the auto- and cross-correlation functions between the regressor and the output observation. The expression in (5.9) effectively illustrates the possibility to use least squares minimization to estimate the coefficients vector $\mathbf{H}$.

Note that the noise effect is eliminated since it is independent of the input signals. The matrix $\mathbf{R}_{\Phi\Phi}$ is singular due to the redundancy. But (5.9) can have a unique solution if the symmetric constraint in (5.7) is taken into account. In this chapter, we are trying to establish a condition for a unique solution $\mathbf{H}$ to the model (5.8) under the assumption (5.7), i.e., the persistent excitation condition.

5.4 The Development of Persistent Excitation Condition

Redundancy in the model usually exists for a compact model description and thus infinite solutions exist. For this reason, the persistent excitation condition is said to be satisfied if the kernel coefficients can be uniquely determined with the redundancy being constrained in some way. Also, the normal description in (5.6), rather than the tensor description, is used in this chapter since it is easier to develop the persistent excitation condition. Without loss of generality, the symmetric constraint in (5.7) is assumed. Each channel of the inputs is chosen to be a zero mean Gaussian signal, not necessarily independent of each other.

5.4.1 Moments to Cumulants Conversion

Two problems make it hard to determine whether (5.9) has a unique solution:

1. $\mathbf{R}_{\Phi\Phi}$ is typically a full square matrix and will not reflect the invertibility obviously;
2. $\mathbf{R}_{\Phi\Phi}$ is singular due to the redundancy in the kernel coefficients as formulated.

For the first problem, we will transform the full matrix into a block triangular square matrix whose invertibility can be determined by the diagonal blocks. For the second problem, the symmetric condition (5.7) will be used to rearrange the components in the diagonal blocks corresponding to the symmetric kernels to remove the redundancy effect and make it possible for the linear equations to have a unique solution.

Denote the scalar term $x(t) = \mathbf{\Phi}^{\mathrm{T}}(t)\,\mathbf{H}$. Then, (5.9) becomes

$$\begin{bmatrix} \mathbb{E}\left[x\left(t\right)\right] \\ \mathbb{E}\left[\varphi_1\left(t\right)x\left(t\right)\right] \\ \vdots \\ \mathbb{E}\left[\varphi_d\left(t\right)x\left(t\right)\right] \end{bmatrix} = \begin{bmatrix} \mathbb{E}\left[y\left(t\right)\right] \\ \mathbb{E}\left[\varphi_1\left(t\right)y\left(t\right)\right] \\ \vdots \\ \mathbb{E}\left[\varphi_d\left(t\right)y\left(t\right)\right] \end{bmatrix} \tag{5.10}$$

where the jth block row is corresponding to all the possible jth order moments of input signals with $x(t)$ or $y(t)$. Each row of the block entry $\mathbb{E}\left[\varphi_i\left(t\right)x\left(t\right)\right] \in \mathbb{R}^{(pM)^i}$ can be written in the form

$$\mathbb{E}[\mathbf{X}^{_ib}] = \mathbb{E}[\underbrace{u^{(\alpha_1+1)}(t-k_1)}_{X_1}\cdots\underbrace{u^{(\alpha_i+1)}(t-k_i)}_{X_i}\underbrace{x(t)}_{X_{i+1}}] \tag{5.11}$$

where the indices set ${}_ib = \{1,\cdots,i,i+1\}$. Similarly, the corresponding row in $\mathbb{E}[\varphi_i(t)\,y(t)] \in \mathbb{R}^{(pM)^i}$ is written as

$$\mathbb{E}[\mathbf{X}^{_ib'}] = \mathbb{E}[\underbrace{u^{(\alpha_1+1)}(t-k_1)}_{X_1}\cdots\underbrace{u^{(\alpha_i+1)}(t-k_i)}_{X_i}\underbrace{y(t)}_{X_{i+2}}] \tag{5.12}$$

where the indices set ${}_ib' = \{1,\cdots,i,i+2\}$, $\mathbb{E}[\mathbf{X}^{_ib}] = \mathbb{E}[\mathbf{X}^{_ib'}]$, and $|{}_ib| = |{}_ib'|$. Lemma 4 tells us that the higher moments in both $\mathbb{E}[\mathbf{X}^{_ib}]$ and $\mathbb{E}[\mathbf{X}^{_ib'}]$ can be split into the sum of $|{}_ib|$ different products of cumulants terms. Then, each moments equation $\mathbb{E}[\mathbf{X}^{_ib}] = \mathbb{E}[\mathbf{X}^{_ib'}]$ produces $|{}_ib|$ different cumulants equations since the $|{}_ib|$ terms of both can be paired via one-to-one correspondence. For each cumulants equation, the unknown coefficients $\mathbf{H}$ within $x(t)$ could be extracted by using the multilinear and homogeneous properties in Lemma 3, which leads to a raw cumulants based linear system equivalent to (5.9):

$$\underbrace{\begin{bmatrix} 1 & \mathbf{R}_{01}^{\mathrm{cum}} & \cdots & \mathbf{R}_{0d}^{\mathrm{cum}} \\ \mathbf{R}_{10}^{\mathrm{cum}} & \mathbf{R}_{11}^{\mathrm{cum}} & \cdots & \mathbf{R}_{1d}^{\mathrm{cum}} \\ \vdots & \vdots & \ddots & \vdots \\ \mathbf{R}_{d0}^{\mathrm{cum}} & \mathbf{R}_{d1}^{\mathrm{cum}} & \cdots & \mathbf{R}_{dd}^{\mathrm{cum}} \end{bmatrix}}_{\mathbf{R}^{\mathrm{cum}}} \underbrace{\begin{bmatrix} h_0 \\ \mathbf{H}_1 \\ \vdots \\ \mathbf{H}_d \end{bmatrix}}_{\mathbf{H}} = \underbrace{\begin{bmatrix} \mathbf{Q}_0^{\mathrm{cum}} \\ \mathbf{Q}_1^{\mathrm{cum}} \\ \vdots \\ \mathbf{Q}_d^{\mathrm{cum}} \end{bmatrix}}_{\mathbf{Q}^{\mathrm{cum}}}. \tag{5.13}$$

Denote

$$N_h = 1 + \sum_{i=1}^{d} (pM)^i, N_{\mathrm{cum}} = 1 + \sum_{i=1}^{d} |{}_ib|\,(pM)^i$$

The number of the unknown kernel coefficients is N_h. Note that there are N_h equations in (5.9) whereas N_{cum} equations in (5.13). Arbitrary N_h of the N_{cum} linear equations in (5.13) that guarantees a unique solution under the symmetric constraint is sufficient to provide the persistent excitation condition.

Example 2. *Among the $|{}_ib|$ different cumulants equations from each $\mathbb{E}[\mathbf{X}^{{}_ib}] = \mathbb{E}[\mathbf{X}^{{}_ib'}]$, the one corresponding to a minimal partition $\hat{0}$ is*

$$\begin{aligned} &\operatorname{cum}\left(u^{(\alpha_1+1)}\left(t-k_1\right), \cdots, u^{(\alpha_i+1)}\left(t-k_i\right), \mathbf{\Phi}^{\mathrm{T}}\left(t\right)\right)\mathbf{H} \\ =&\operatorname{cum}\left(u^{(\alpha_1+1)}\left(t-k_1\right), \cdots, u^{(\alpha_i+1)}\left(t-k_i\right), y\left(t\right)\right) \end{aligned} \tag{5.14}$$

which forms one equation in (5.13).

We could possibly sift out from (5.13) those special equations that can reduce the complexity of the problem. If we can reduce $\mathbf{R}^{\text{cum}}$ into a block triangular square matrix, then it would be much easier to tell the invertibility by only looking at the diagonal blocks. Another lemma is given before we show that all the N_h cumulants equations corresponding to a minimal partition $\hat{0}$ as in (5.14) are such special equations.

Lemma 5. *Let the set $b = \{1, \cdots, i, i+1, \cdots, i+j\}$ and $\mathbf{X}^b$ be a vector obtained by juxtaposing elements of a zero mean Gaussian family (with possible repetitions). Consider a partition*

$$\pi = \{\underbrace{\{1\}}_{b_1}, \cdots, \underbrace{\{i\}}_{b_i}, \underbrace{\{i+1, \cdots, i+j\}}_{b_{i+1}}\} \in \mathscr{P}\left(b\right).$$

If $i > j$ or $(i+j)$ is odd, then

$$\operatorname{cum}\left(\mathbf{X}^{b_1}, \cdots, \mathbf{X}^{b_i}, \mathbf{X}^{b_{i+1}}\right) = 0.$$

If $i \leq j$ and $(i+j)$ is even, then

$$\begin{aligned}
&\operatorname{cum}\left(\mathbf{X}^{b_1}, \cdots, \mathbf{X}^{b_i}, \mathbf{X}^{b_{i+1}}\right) \\
=&\sum_{\delta}\sum_{\rho}\sum_{\gamma} \operatorname{cum}\left(X_1, X_{\rho(\delta(1))}\right) \cdots \operatorname{cum}\left(X_i, X_{\rho(\delta(i))}\right) \qquad (5.15)\\
&\times \operatorname{cum}\left(X_{\gamma(\delta(i+1))}, X_{\gamma(\delta(i+2))}\right) \cdots \operatorname{cum}\left(X_{\gamma(\delta(j-1))}, X_{\gamma(\delta(j))}\right)
\end{aligned}$$

where the $\frac{j!}{i!(j-i)!}$ possible ways to select i elements from j elements in b_{i+1} without putting back are denoted as

$$\{\underbrace{\{\delta(1), \cdots, \delta(i)\}}_{i\,\text{elements}}, \{\delta(i+1), \cdots, \delta(j)\}\} \in \mathscr{P}(b_{i+1})$$

the $i!$ possible permutations of $\{\delta(1), \cdots, \delta(i)\}$ are denoted as $\rho(\cdot)$, and the $\frac{(j-i)!}{((j-i)/2)!2^{(j-i)/2}}$ possible unordered ways to pair elements of $\{\delta(i+1), \cdots, \delta(j)\}$ are denoted as $\gamma(\cdot)$.

Proof. For elements from a Gaussian family, only the first and second cumulants are nonzero via the item 4 in Lemma 3. Since it is also of zero mean, only the second cumulants remain nonzero via the item 1 in Lemma 3. Thus, further using the item 2 in Lemma 4, we could express $\operatorname{cum}\left(\mathbf{X}^{b_1}, \cdots, \mathbf{X}^{b_i}, \mathbf{X}^{b_{i+1}}\right)$ as the sum of products of all possible second cumulants satisfying the given condition $\sigma \vee \pi = \hat{1}$. If $(i+j)$ is odd, σ can never be arranged into a partition with only pairs and thus the sum will be zero.

Then, let's consider the case when $(i+j)$ is even. If $i > j$, one can never find a σ both including only pairs and satisfying $\sigma \vee \pi = \hat{1}$. See Example 1 for a simple demonstration. If $i \leq j$, the partitions being only pairs and satisfying $\sigma \vee \pi = \hat{1}$ can only be formulated in the following way: pair each element in $\{b_1, \cdots, b_i\}$ with a different element in $\{b_{i+1}\}$ and the remaining $(j-i)$ elements are paired with each other without specifying an order. This turns out to be (5.15). □

Remark 10. *Lemma 5 only assumes that each random variable is of zero mean and Gaussian distributed. It does not require that they are from the same channel of Gaussian processes.*

5.4.2 Reduction of Cumulants Equations

It has been pointed out in Section 5.4.1 that the moments to cumulants conversion produces at most N_{cum} different linear equations. In fact, we only need N_h of them to determine the unknown kernels. Theoretically speaking, we have $C_{N_{\text{cum}}}^{N_h}$ different choices. But we prefer a structure that would simplify the problem of determining a unique solution.

For each original equation described as (5.12), now we only choose the cumulants equation (5.14) from $|_ib|$ different options, which is corresponding to a minimal partition $\hat{0}$. Then, the row number of $\mathbf{R}^{\text{cum}}$ and $\mathbf{Q}^{\text{cum}}$ in (5.13) will be reduced to N_h. Denote the reduced N_h cumulants equations as

$$\bar{\mathbf{R}}^{\text{cum}}\mathbf{H} = \bar{\mathbf{Q}}^{\text{cum}} \tag{5.16}$$

where $\bar{\mathbf{R}}^{\text{cum}}$ is a square matrix of the same dimension as $\mathbf{R}_{\Phi\Phi}$.

Lemma 5 implies that matrix $\bar{\mathbf{R}}^{\text{cum}}$ can be further reduced to a block triangular form

$$\bar{\mathbf{R}}^{\text{cum}} = \begin{bmatrix} 1 & \mathbf{0} & \mathbf{R}_{02}^{\text{cum}} & \mathbf{0} & \cdots & \cdots \\ \mathbf{0} & \mathbf{R}_{11}^{\text{cum}} & \mathbf{0} & \mathbf{R}_{13}^{\text{cum}} & \cdots & \cdots \\ \mathbf{0} & \mathbf{0} & \mathbf{R}_{22}^{\text{cum}} & \mathbf{0} & \cdots & \cdots \\ \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{R}_{33}^{\text{cum}} & \cdots & \cdots \\ \vdots & \vdots & \vdots & \vdots & \ddots & \vdots \\ \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \cdots & \mathbf{R}_{dd}^{\text{cum}} \end{bmatrix} \tag{5.17}$$

Note that $\bar{\mathbf{R}}^{\mathrm{cum}}$ is still singular due to the redundancy of kernel coefficients but it is possible to have a unique solution if the symmetric condition in (5.7) is enforced. The columns in $\bar{\mathbf{R}}^{\mathrm{cum}}$ corresponding to the symmetric kernel coefficients are regarded as identical and can be rearranged to make the modified $\bar{\mathbf{R}}^{\mathrm{cum}}$ able to be nonsingular without changing the solution in the sense of symmetric kernel coefficients. One only needs to take care of the block diagonal entries $\mathbf{R}_{ii}^{\mathrm{cum}}$ and develop the persistent excitation condition to make the modified block diagonal entries nonsingular.

Proposition 5. *Consider the reduced cumulants equation (5.16) with $\mathbf{R}^{\mathrm{cum}}$ being a matrix of dimension $N_h \times N_h$ given by (5.17). If the symmetric constraint in (5.7) is assumed, then (5.17) can be replaced, without affecting the solution $\mathbf{H}$, by*

$$\widehat{\mathbf{R}}^{\mathrm{cum}} = \begin{bmatrix} 1 & \mathbf{0} & \mathbf{R}_{02}^{\mathrm{cum}} & \mathbf{0} & \cdots & \cdots \\ \mathbf{0} & \widehat{\mathbf{R}}_{11}^{\mathrm{cum}} & \mathbf{0} & \mathbf{R}_{13}^{\mathrm{cum}} & \cdots & \cdots \\ \mathbf{0} & \mathbf{0} & \widehat{\mathbf{R}}_{22}^{\mathrm{cum}} & \mathbf{0} & \cdots & \cdots \\ \mathbf{0} & \mathbf{0} & \mathbf{0} & \widehat{\mathbf{R}}_{33}^{\mathrm{cum}} & \cdots & \cdots \\ \vdots & \vdots & \vdots & \vdots & \ddots & \vdots \\ \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \cdots & \widehat{\mathbf{R}}_{dd}^{\mathrm{cum}} \end{bmatrix} \tag{5.18}$$

where the modified block diagonal entries are

$$\begin{aligned} &\widehat{\mathbf{R}}_{ii}^{\mathrm{cum}} \\ =&i!\,\underbrace{\mathrm{cum}\left(\mathbf{u}_{M_i}(t)\,\mathbf{u}_{M_i}^{\mathrm{T}}(t)\right) \otimes \cdots \otimes \mathrm{cum}\left(\mathbf{u}_{M_i}(t)\,\mathbf{u}_{M_i}^{\mathrm{T}}(t)\right)}_{i \text{ terms}} \\ =&i!\,\underbrace{\mathbb{E}\left[\mathbf{u}_{M_i}(t)\,\mathbf{u}_{M_i}^{\mathrm{T}}(t)\right] \otimes \cdots \otimes \mathbb{E}\left[\mathbf{u}_{M_i}(t)\,\mathbf{u}_{M_i}^{\mathrm{T}}(t)\right]}_{i \text{ terms}} \end{aligned} \tag{5.19}$$

and $\mathbf{u}_{M_i}(t)$ is defined as in (5.8).

Proof. Let's look at the (r, c)-th component of the matrix $\mathbf{R}_{ii}^{\text{cum}} \in \mathbb{R}^{(pM)^i \times (pM)^i}$. First, the row index r can be written as a grouped index $(k_1, \alpha_1; \cdots ; k_i, \alpha_i)$ such that

$$\begin{aligned} r - 1 = &(k_1 p + \alpha_1)\,(pM)^{i-1} \\ &+ \cdots + (k_{i-1} p + \alpha_{i-1})\,(pM)^1 + k_i p + \alpha_i \end{aligned}$$

where $0 \le k_l \le M - 1, 0 \le \alpha_l \le p - 1$. Similarly, the column index c can be written as a grouped index $(\bar{k}_1, \bar{\alpha}_1; \cdots ; \bar{k}_i, \bar{\alpha}_i)$. Then, the (r, c)-th component could be represented in the following form:

$$\begin{aligned} &(\mathbf{R}_{ii}^{\text{cum}})_{r,c} \\ =&\text{cum}\Big(\underbrace{u^{(\alpha_1+1)}\,(t - k_1)}_{X_1}, \cdots , \underbrace{u^{(\alpha_i+1)}\,(t - k_i)}_{X_i}, \\ &\underbrace{u^{(\bar{\alpha}_1+1)}\left(t - \bar{k}_1\right)}_{X_{i+1}} \cdots \underbrace{u^{(\bar{\alpha}_i+1)}\left(t - \bar{k}_i\right)}_{X_{i+i}}\Big) \end{aligned} \tag{5.20}$$

Using Lemma 5 and Remark 10, one further has

$$(\mathbf{R}_{ii}^{\text{cum}})_{r,c} = \sum_{\rho} \text{cum}\left(X_1, X_{\rho(\delta(1))}\right) \cdots \text{cum}\left(X_i, X_{\rho(\delta(i))}\right) \tag{5.21}$$

where there is only one way for the selection δ and there are no partitions introduced by γ since $j = i$ in this case. There are $i!$ items, some of which are possibly identical, to add up in (5.21). For each row r, it can be observed from (5.20) that there is a redundancy in the columns. All the components in the column c' satisfying

$$\begin{aligned} c' - 1 = &\left(\bar{k}_{\rho(1)} p + \bar{\alpha}_{\rho(1)}\right)(pM)^{i-1} \\ &+ \cdots + \left(\bar{k}_{\rho(i-1)} p + \bar{\alpha}_{\rho(i-1)}\right)(pM)^1 \\ &+ \bar{k}_{\rho(i)} p + \bar{\alpha}_{\rho(i)} \end{aligned}$$

are exactly with the same value as in the column c, where $\rho(\cdot)$ is a permutation of $\{1, \cdots , i\}$. There are at most $i!$ possible columns with the same value if each pair $(\bar{k}_l, \bar{\alpha}_l)$ is different.

Note that there will be less than $i!$ identical columns if some of $(\bar{k}_l, \bar{\alpha}_l)$ are with the same values. But this also implies repeated variables in $\{X_{i+1}, \cdots, X_{i+i}\}$ and thus will not affect the results of the following manipulation. We can interchange the terms under summation among these identical columns such that all the equal terms are gathered together and the (r, c)-th component of the modified $\widehat{\mathbf{R}}_{ii}^{\text{cum}}$ is

$$
\begin{aligned}
&\left(\widehat{\mathbf{R}}_{ii}^{\text{cum}}\right)_{r,c} \\
=&i!\text{cum}\left(X_1, X_{i+1}\right)\cdots\text{cum}\left(X_i, X_{i+i}\right) \\
=&i!\text{cum}\Big(u^{(\alpha_1+1)}\left(t-k_1\right), u^{(\bar{\alpha}_1+1)}\left(t-\bar{k}_1\right)\Big) \\
&\cdots\text{cum}\Big(u^{(\alpha_i+1)}\left(t-k_i\right), u^{(\bar{\alpha}_i+1)}\left(t-\bar{k}_i\right)\Big)
\end{aligned}
$$

which is actually the (r, c)-th component of $\widehat{\mathbf{R}}_{ii}^{\text{cum}}$ in (5.19), where the last step is obtained by using the item 1 in Lemma 3. Note that $\text{cum}(X_l, X_s) = \text{cum}(X_l X_s)$ for zero mean signals. □

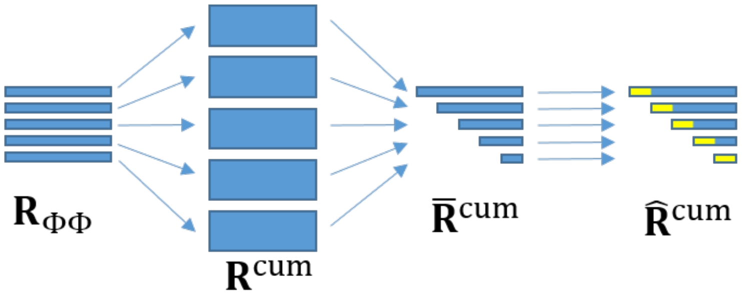

$\mathbf{R}_{\Phi\Phi}$
$\mathbf{R}^{\text{cum}}$
$\overline{\mathbf{R}}^{\text{cum}}$
$\widehat{\mathbf{R}}^{\text{cum}}$

Figure 5.1: Illustration of the moments to cumulants conversion to simplify the original problem.

The whole procedure for moments to cumulants conversion and relevant modifications is illustrated in Fig. 5.1.

5.4.3 Persistent Excitation Condition

All above discussions show that the original system (5.6) is persistently excited if $\widehat{\mathbf{R}}^{\mathrm{cum}}\mathbf{H} = \mathbf{Q}^{\mathrm{cum}}$ has a unique solution, where $\widehat{\mathbf{R}}^{\mathrm{cum}}$ is defined in (5.18).

Theorem 3. *Assume that each channel of the input signal for a Volterra system (5.6) is a zero mean Gaussian signal and not necessarily independent of the other channels. Let* $M = \max\{M_1, \cdots, M_d\}$*. The equation* $\widehat{\mathbf{R}}^{\mathrm{cum}}\mathbf{H} = \mathbf{Q}^{\mathrm{cum}}$ *under the symmetric condition (5.7) has a unique solution if and only if* $\widehat{\mathbf{R}}^{\mathrm{cum}}_{ii}, i = 1, \cdots, d$ *are nonsingular. A sufficient condition is that the power spectrum matrix of the input signal* $\mathbf{u}(t)$ *is positive definite at least at* M *distinct frequencies.*

Proof. The unique solution implies that the matrix $\widehat{\mathbf{R}}^{\mathrm{cum}}$ should be nonsingular. Thus, the block diagonal matrices $\widehat{\mathbf{R}}^{\mathrm{cum}}_{ii}, i = 1, \cdots, d$ should be nonsingular. Note that the multiple Kronecker product has the property $(\mathbf{A}_1 \otimes \cdots \otimes \mathbf{A}_i)^{-1} = \mathbf{A}_1^{-1} \otimes \cdots \otimes \mathbf{A}_i^{-1}$. So matrices $\mathbb{E}[\mathbf{u}_{M_i}(t)\,\mathbf{u}_{M_i}^{\mathrm{T}}(t)], i = 1, \cdots, d$ should be positive definite. This is equivalently requiring that matrix $\mathbb{E}[\mathbf{u}_M(t)\,\mathbf{u}_M^{\mathrm{T}}(t)]$ is positive definite since those matrices are its leading principal minors. Select an arbitrary vector

$$\mathbf{g} = \left(g_{01}, \cdots, g_{0p}, \cdots, g_{(M-1)1}, \cdots, g_{(M-1)p}\right)^{\mathrm{T}}$$

and define a matrix transfer function

$$\mathbf{G}\left(q^{-1}\right) = \left[G_1\left(q^{-1}\right), \cdots, G_p\left(q^{-1}\right)\right]$$

where $G_l\left(q^{-1}\right) = \sum\limits_{j=0}^{M-1} g_{jl} q^{-1}$ and q^{-1} is a time delay operator. Then,

$$\begin{aligned} 0 &\leq \mathbf{g}^{\mathrm{T}} \mathbb{E}\left[\mathbf{u}_M\left(t\right)\mathbf{u}_M^{\mathrm{T}}\left(t\right)\right]\mathbf{g} \\ &= \mathbb{E}\Big[\big(\mathbf{G}\left(q^{-1}\right)\mathbf{u}\left(t\right)\big)^2\Big] \\ &= \frac{1}{2\pi}\int_{-\pi}^{\pi}\mathbf{G}\left(e^{-j\omega}\right)\phi_{uu}\left(\omega\right)\mathbf{G}^{\mathrm{T}}\left(e^{j\omega}\right)\mathrm{d}\omega \end{aligned}$$

where $\phi_{uu}\left(\omega\right)$ is the power spectrum matrix of the input signal $\mathbf{u}(t)$. Since there are at most $(M-1)$ zeros for $G_l\left(e^{-j\omega}\right)$, the integral must be strictly positive if $\phi_{uu}\left(\omega\right)$ is positive definite at least at M distinct frequencies. □

5.5 Simulation Results

The 2-input 2-output sparse Volterra system in Section 4.5.2 is still used here for demonstration. In this example, only the NIteTN algorithm is implemented to test if the input signal is persistently exciting. The nonzero kernels will be estimated and compared with the true ones. Note that the coefficients corresponding to the symmetric kernels should be combined together for comparison. We set $\varepsilon_u = 0.001$ and $\varepsilon = 0.001$ so that all information is maintained during system identification except machine errors. Gaussian white signals, filtered Gaussian signals, and PRBS are used as different excitation signals. The first 1500 samples of the total 5000 generated input/output data are used for system identification and the remaining 3500 samples are for validation. The output signals for identification are noise-free.

The covariance matrix of the Gaussian white signals is chosen as $\sigma^2 = \mathbf{I}$. The filtered Gaussian signals are generated by feeding the above Gaussian white signals through the

Table 5.1: The estimated nonzero kernel coefficients.

	β_{11}	β_{12}	β_{13}	β_{14}	β_{21}	β_{22}	β_{23}	β_{24}
True	1	0.1	0.01	0.05	1	0.1	0.01	0.05
White	1.000	0.100	0.010	0.050	1.000	0.100	0.010	0.050
Filtered	1.000	0.100	0.010	0.050	1.000	0.100	0.010	0.050
PRBS	0.092	0.092	-0.013	0.002	0.082	-0.044	-0.000	0.001

following low-pass filter $G_0(z) = \frac{0.75}{z-0.75}$. The PRBS signals for the two input channels are generated independently such that they are randomly switching between ± 1 for the first 2499 samples, ± 0.9 for the range $[2500, 3749]$, and ± 1.1 for the range $[3750, 5000]$.

The estimates of those nonzero kernel coefficients for different excitation signals are given in Table. 5.1. The Gaussian white signals and filtered Gaussian signals are persistently exciting as suggested. Note that the estimates of the kernel coefficients that are supposed to be zero, are around machine precision level. The PRBS signals can not produce correct estimates since they are never persistently exciting for Volterra systems of high degree. It is also interesting to observe that the estimates of both β_{11} and β_{12} are the same. The reason is that $u^{(1)}(t)$ and $u^{(1)}(t)^3$ are indistinguishable under PRBS with magnitude 1 and thus considered as symmetric kernels during system identification. Fig. 5.2 compares the true and simulated $y^{(1)}(t)$. The identified model fails to predict the output when the input magnitude changes.

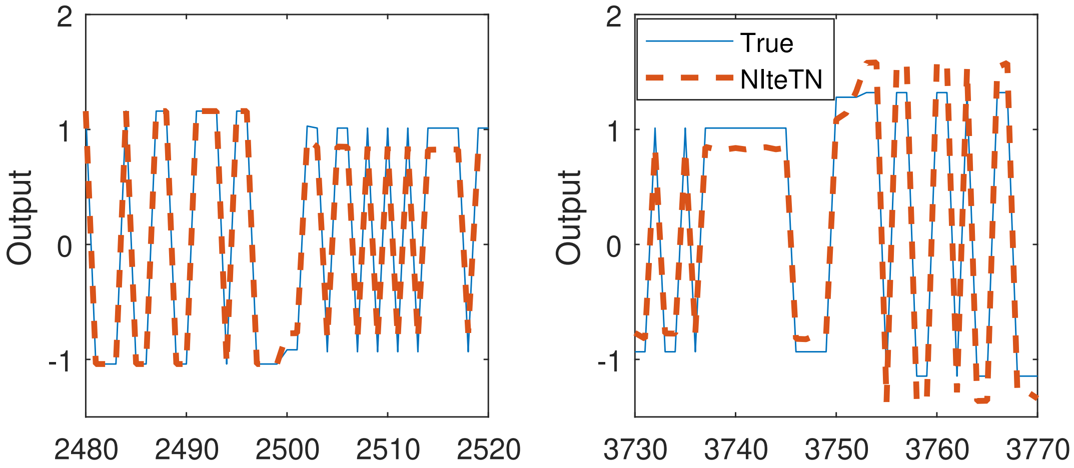


Figure 5.2: The comparison for the first output channel. Starting from sample 2500, the input magnitude changes to 0.9. Starting from sample 3750, the input magnitude then changes to 1.1.

5.6 Conclusion

It is shown that the least squares solution typically solved for Volterra models can be simplified via moments to cumulants conversion. A symmetric constraint is enforced during the least squares estimation to handle the redundancy in the kernel coefficients and allows for the formulation of a persistent excitation condition. It is worthwhile to note that the imposed constraint does not have to be symmetric, as the persistent excitation condition applies naturally to cases where the redundant kernels are constrained in other ways. Techniques like tensor network exploit the potential to capture complicated coupled nonlinear behaviors via higher degree and even MIMO Volterra models. The resulting persistent excitation condition in this chapter demonstrates that Gaussian distributed input signals provide sufficient excitation to a MIMO Volterra system. Such a persistent excitation condition may not hold for non-Gaussian distributed input signals, such as a pseudo-random binary sequence.

This chapter is based on the following papers that were published or submitted:

Y. Hu, L. Tan and R.A. de Callafon, "Persistent excitation condition for MIMO Volterra system identification with Gaussian distributed input signals," in *Proceedings of IEEE 58th Annual Conference on Decision and Control (CDC)*, Nice, France, Dec. 11–13, 2019, pp. 1752–1757.

Y. Hu, L. Tan and R.A. de Callafon, "Noniterative tensor network-based estimation for MIMO Volterra system identification," *IEEE Transactions on Automatic Control*, under review.

Part III

Applications to Energy Systems

Chapter 6

Estimation of Microgrid Power Flow Dynamics via CoBRA

This chapter considers the application of the Covariance Based Realization Algorithm (CoBRA) to microgrid dynamic modeling. More specifically, we propose a data-driven approach to modeling Distributed Energy Resources (DERs) power flow dynamics within a microgrid by using the discrete-time dependent covariance between active and reactive power flow. The microgrid network with its DERs is regarded as a multi-input multi-output (MIMO) linear dynamic system concatenated by a static nonlinear part to model power loss within the microgrid. The linear part is identified by the CoBRA that can handle noisy power flow data and focus on identifying a low-order dynamic model, where a static gain of the model can be enforced during the estimation. The nonlinear part is formulated by the estimation of the admittance matrix of the microgrid network. The proposed model structure takes the physical characteristics of the microgrid into account and will be suitable for a relatively wide range of operating conditions even if the parameter estimation is performed from surrounding one local equilibrium. It is shown via a simulation and an experimental study that simulated and measured power flow transient effects can be matched with high accuracy.

6.1 Introduction

Microgrids typically consist of a set of power producing DERs interconnected with multiple power consuming loads such as lighting, HVAC system, Electric Vehicles (EV), etc. The DERs may include fossil fuel generation, sustainable energy generation (e.g., wind, solar, etc.), and energy storage system (e.g., battery powered inverters). The micrigrid can be working in either grid-connected mode or islanded mode when the microgrid is disconnected from the main grid [80]. This flexibility allows autonomous operation of the microgrid during main grid faults possibly due to events such as extreme weather, blackouts, and attacks on the power infrastructure and thus increases the resiliency of the whole power systems [81, 82, 83]. In addition, DERs created by modern inverters with fast transient dynamics can balance and reduce fluctuations in power flow within the microgrid in real-time by absorbing and providing energy [84, 85, 86]. This further mitigates the need for volatile modulations of any fossil fuel powered generators to improve efficiency.

The ability to control power flow within a microgrid becomes a very important task in order to stably and optimally operate the microgrid [87]. Real-time feedback control of both active and reactive bi-directional power flow between the main grid and local microgrid allows seamless mode transition between grid-connected and islanded modes [88]. For such feedback power control, a dynamic model describing the power flow from DERs to the point of common coupling (PCC) is desired to design control algorithms with guaranteed stability or robustness margins. Although component- and physics-based modeling may provide a dynamic model of the microgrid, uncertainty in component specifications and model complexity may render the resulting model unsuitable for actual feedback control design. In contrast, a data-driven approach tailored to learn or estimate DER and microgrid dynamics provides an alternative to both identify a low-order model and capture the relevant dynamics for control [2, 3].

Synchronized phasor data (voltage, current, and frequency) obtained via Phasor Measurement Units (PMUs) [89] can be exploited in a data-driven approach for microgrid dynamic modeling. For example, in [90], it has been demonstrated that dynamic system identification can be used to identify the model of the power dynamics around a working equilibrium. In [88], a synchrophasor based model estimation is implemented to design the controller for seamless mode switching, whereas the model can be updated for auto-tuning controller parameters. The work in this chapter builds on these observations, but the objective is to develop a data-driven approach that support modeling of microgrid dynamics with a multitude of DERs operating simultaneously and incorporate power loss though the microgrid. The end result is a MIMO microgrid dynamic model that consists of a linear dynamic part and a static nonlinear mapping that is suitable for (distributed) feedback control design. It will be shown that the proposed microgrid dynamic model will capture power flow dynamics in the microgrid for a large range of operating conditions.

MIMO system identification methods are instrumental for the development of microgrid dynamic model when DERs are operating simultaneously. Within the class of MIMO methods, subspace methods are popular due to the computational robustness and noniterative feature compared to the prediction error methods (PEM) [11, 10]. Among different subspace methods, the CoBRA has been demonstrated to be quite efficient in estimating the inherent low-order deterministic model of the given system and handling a large amount of input/output data heavily contaminated by noise with unknown and possibly high-order spectral characteristics [28, 35, 29]. An optimal implementation for the CoBRA is proposed in [46, 72] and shows the best performance among different subspace methods under closed-loop settings as illustrated in Chapter 3.

The remaining part of the chapter is organized as follows. In Section 6.2, the problem of microgrid dynamic modeling is formulated and a novel model structure is proposed. In Section 6.3, we apply the CoBRA to estimate the linear part of the proposed model. Some practical issues during application of the CoBRA are discussed in Section 6.4. Section 6.5 presents the estimation of the nonlinear part of the proposed model. A simulation example is illustrated in Section 6.6 and a practical experiment is investigated in Section 6.7. Finally, a conclusion is drawn in Section 6.8.

6.2 Problem Formulation

6.2.1 Framework of Microgrid Network

To formalize the data-driven approach for microgrid dynamic modeling, consider a microgrid with a network structure as shown in Fig. 6.1. The figure illustrates that the microgrid can be either connected or disconnected (islanded) from the main grid through the PCC and consists of a hybrid collection of DERs interconnected with loads. The multitude of DERs in the microgrid may include: (a) A battery energy storage system (BESS) with a source switchable battery/inverter structure which can operate either in grid following (current source) or grid forming (voltage source) mode [91]; (b) An emergency synchronous diesel generator to support the microgrid when working in the islanded mode [92]; (c) Photo Voltaic (PV) or solar inverter that outputs AC power from DC solar power [93]. The loads include day-to-day residential or industrial appliances that could be modeled as resistive and inductive (R-L) elements.

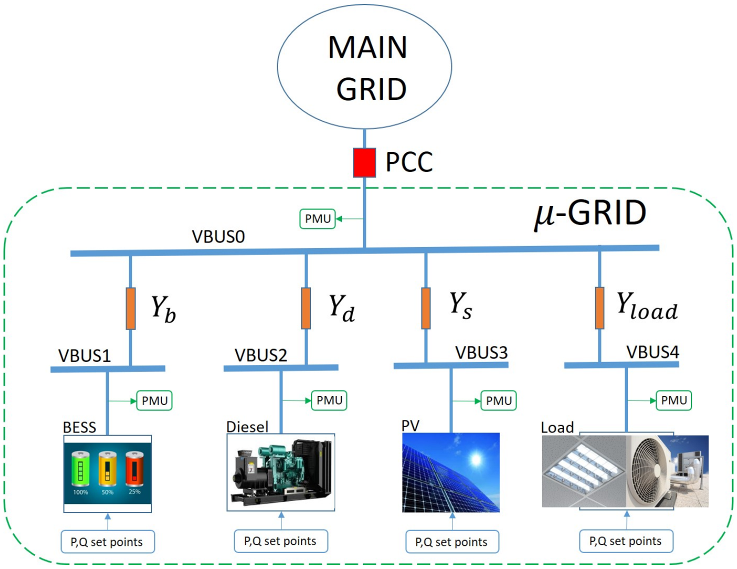


Figure 6.1: The microgrid framework with a multitude of DERs.

DERs and loads within the microgrid are assumed to be tied to individual buses, which are further connected to the main bus at the PCC through distribution lines with certain values of admittance to model power loss. As mentioned in [88], fast switching between islanding and grid connection is a strict requirement for the microgrid to survive under main grid or microgrid faults [94]. In addition, a model based controller design can be used for quick and smooth transition from grid-connected mode to islanded mode without the need for load shedding. The islanded to grid-connected transition can be smoothly realized by referring to phase angle control [95]. Clearly, modeling the dynamics of power flow is instrumental in understanding grid-connected to islanded power flow transitions.

6.2.2 Identification Problem

Given a simultaneous operation of DERs and PMU-based measurements of the flow of active power P and reactive power Q within the microgrid, the objective is to model the dynamics of the power flow from the DERs to the PCC, while taking into account nonlinearities associated with power loss within the microgrid. This problem formulation can be stated as MIMO system identification or estimation problem, which will then be solved by the CoBRA. Note that PMUs typically measure the pair of complex-valued (positive sequence) current $\mathbf{I}$ and voltage $\mathbf{V}$ phasors and the corresponding pair of active power P and reactive power Q represent the same information via $P = \text{Re}\{\mathbf{S}\}$, $Q = \text{Im}\{\mathbf{S}\}$, where $\mathbf{S} = \mathbf{V}\mathbf{I}^*$. The identification problem can then be stated as: given the requested DER active and reactive power set points $\mathbf{u}_0(t)$ and PMU-based measurements of the DER active and reactive power output $\mathbf{y}(t)$ in the microgrid, identify a model of the active power $\mathrm{P}_{PCC}(t)$ and reactive power $\mathrm{Q}_{PCC}(t)$ flow at the PCC.

Referring back to Fig. 6.1, the input signal $\mathbf{u}_0(t)$ is a vector of DERs set points, while

$\mathbf{y}(t)$ is a vector of the PMU measurements of the DERs output power flow according to

$$\mathbf{u}_0(t) = \begin{bmatrix} \mathrm{P}_b^{\mathrm{in}}(t) \\ \mathrm{Q}_b^{\mathrm{in}}(t) \\ \mathrm{P}_d^{\mathrm{in}}(t) \\ \mathrm{Q}_d^{\mathrm{in}}(t) \\ \mathrm{P}_s^{\mathrm{in}}(t) \\ \mathrm{Q}_s^{\mathrm{in}}(t) \end{bmatrix}, \mathbf{y}(t) = \begin{bmatrix} \mathrm{P}_b^{\mathrm{out}}(t) \\ \mathrm{Q}_b^{\mathrm{out}}(t) \\ \mathrm{P}_d^{\mathrm{out}}(t) \\ \mathrm{Q}_d^{\mathrm{out}}(t) \\ \mathrm{P}_s^{\mathrm{out}}(t) \\ \mathrm{Q}_s^{\mathrm{out}}(t) \end{bmatrix} \tag{6.1}$$

The discrete-time nature of the signals is represented by $t = k\Delta T, k \in \mathbb{N}$ with ΔT being the sampling time, and $\mathrm{P}_b/\mathrm{Q}_b$, $\mathrm{P}_d/\mathrm{Q}_d$, and $\mathrm{P}_s/\mathrm{Q}_s$ represent active/reactive power flow for inverter based battery system, diesel generator, and solar inverter, respectively. The corresponding PMU measurements of active and reactive power flow at PCC are $\mathrm{P}_{PCC}(t)$ and $\mathrm{Q}_{PCC}(t)$.

6.2.3 Proposed Model Structure

It is worth recognizing that DERs may have imposed power ramp constraints to limit requested power flow fluctuations. In addition, different DERs may have different transient dynamics and the outputs may not reach the set points immediately. Last, but not least, transmission lines with complex admittance within a microgrid may impose coupling between active and reactive power, along with power loss at the PCC. These effects can be modeled in a data-driven approach by a careful selection of dynamic and static components in the microgrid model as summarized in Fig. 6.2.

Consistent with the definition of signal in (6.1), the input $\mathbf{u}_0(t)$ is a vector of DERs set points, but $\mathbf{u}(t)$ is an intermediate signal after the multi-input rate limiter. The different

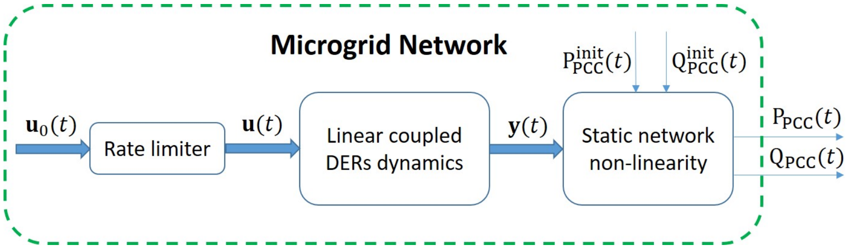


Figure 6.2: Model structure for dynamic microgrid network model.

transient dynamics observed in the the PMU measurements of the DERs power outputs $\mathbf{y}(t)$ are modeled by the MIMO linear coupled DER dynamics. Finally, the possible static nonlinear coupling and power loss effects due to transmission lines within a microgrid are modelled by the static network nonlinearity included in the model structure of Fig. 6.2. The effect of the power loss is inherently nonlinear, since it is proportional to the current squared.

6.3 Linear Dynamic Model Estimation

6.3.1 Formulation of Covariance Data Equation

In order to obtain the input/output data depicted in Fig. 6.2, the DERs should be perturbed around their working equilibrium by small power excitation signals without affecting the stability of the microgrid. Simultaneous excitation of all DERs to reduce experimentation time is allowed, but to guarantee indentifiability of the linear MIMO dynamic model part, the DER power perturbations should ideally be uncorrelated. Since typically only small power perturbations are allowed, the signal-to-noise ratio (SNR) of the measurement data may be relatively low due to existence of measurement noise and/or possible random load changes. The CoBRA serves as a powerful tool to extract a low-order deterministic model from noise contaminated data by first averaging the time-dependent covariance between the data.

Let the dynamic relation between the input/output signals $\mathbf{u}(t)$ and $\mathbf{y}(t)$ in Fig. 6.2 be described by a MIMO linear time-invariant minimal system $G(q^{-1})$ with the following state-space form:

$$\begin{cases} \mathbf{x}\left(t+1\right) = A\mathbf{x}\left(t\right) + B\mathbf{u}\left(t\right) \\ \\ \mathbf{y}\left(t\right) = C\mathbf{x}\left(t\right) + D\mathbf{u}\left(t\right) + \mathbf{v}\left(t\right) \end{cases} \tag{6.2}$$

where $A \in \mathbb{R}^{n\times n}$, $B \in \mathbb{R}^{n\times p}$, $C \in \mathbb{R}^{m\times n}$, $D \in \mathbb{R}^{m\times p}$ and $\mathbf{v}(t)$ is a stochastic noise signal of zero mean, with unknown spectral properties, and uncorrelated with the input $\mathbf{u}(t)$. The objective is to estimate the model parameters $\{A, B, C, D\}$ (up to a similarity transformation) as a low-order deterministic dynamic model for the sampled input/output data $\{\mathbf{u}(t), \mathbf{y}(t)\}$.

As described in Section 3.2.2 from Chapter 3, one can introduce an instrumental variable (IV) $\xi(t) \in \mathbb{R}^{p\times 1}$ satisfying certain conditions to perform the covariance pre-processing for the CoBRA. Note that $\xi(t)$ can also be of the dimension higher than p. However, one can simply select $\xi(t) = \mathbf{u}(t)$ in practice. Then, we can come up with the following equations using sample expectation defined in (3.4):

$$\begin{cases} \hat{R}_{x\xi}\left(\tau+1\right) = A\hat{R}_{x\xi}\left(\tau\right) + B\hat{R}_{u\xi}\left(\tau\right) \\ \\ \hat{R}_{y\xi}\left(\tau\right) = C\hat{R}_{x\xi}\left(\tau\right) + D\hat{R}_{u\xi}\left(\tau\right) + \hat{R}_{v\xi}\left(\tau\right) \end{cases} \tag{6.3}$$

One can finally formulate the following covariance data equation:

$$\hat{\mathbf{R}}_{y\xi} = \mathrm{O}_r\hat{\mathbf{R}}_{x\xi} + \Psi\hat{\mathbf{R}}_{u\xi} + \hat{\mathbf{R}}_{v\xi} \tag{6.4}$$

where

$$\mathrm{O}_r = \begin{bmatrix} C^{\mathrm{T}} & (CA)^{\mathrm{T}} & \cdots & (CA^{r-1})^{\mathrm{T}} \end{bmatrix}^{\mathrm{T}}$$

$$\hat{\mathbf{R}}_{x\xi} = \mathrm{Hankel}\left(\hat{R}_{x\xi}(\tau), 1, \tau_1, \tau_2 - r + 1\right)$$

$$\hat{\mathbf{R}}_{f\xi} = \mathrm{Hankel}\left(\hat{R}_{f\xi}(\tau), r, \tau_1, \tau_2\right), f \in \{y, u, v\}$$

$$\mathrm{Hankel}\left(h(\tau), \tau_0, \tau_1, \tau_2\right) = \begin{bmatrix} h(\tau_1) & h(\tau_1+1) & \cdots & h(\tau_2-\tau_0+1) \\ h(\tau_1+1) & h(\tau_1+2) & \cdots & h(\tau_2-\tau_0+2) \\ \vdots & \vdots & & \vdots \\ h(\tau_1+\tau_0-1) & h(\tau_1+\tau_0) & \cdots & h(\tau_2) \end{bmatrix}$$

and the Toeplitz matrix Ψ is including the unknown Markov parameters to be identified. Note that the extended observability matrix O_r consists of the information of the system matrices $\{A, C\}$.

6.3.2 Estimation of Matrices A and C via Covariance Data

As mentioned in Chapter 3, the noise effect $\hat{\mathbf{R}}_{v\xi}$ will be asymptotically eliminated. In addition, one can easily obtain a shifted version of the covariance data equation in (6.4). The two modified equations are rewritten as follows:

$$\hat{\mathbf{R}}_{y\xi} = \mathrm{O}_r \hat{\mathbf{R}}_{x\xi} + \Psi \hat{\mathbf{R}}_{u\xi} \tag{6.5}$$

$$\hat{\mathbf{R}}_{\vec{y\xi}} = \mathrm{O}_r A \hat{\mathbf{R}}_{x\xi} + \vec{\Psi} \hat{\mathbf{R}}_{u\xi} \tag{6.6}$$

where $\hat{\mathbf{R}}_{\vec{y\xi}} = \mathrm{Hankel}\big(\hat{R}_{y\xi}(\tau), r, \tau_1 + 1, \tau_2 + 1\big)$ is the one-step time-shifted version of $\hat{\mathbf{R}}_{y\xi}$ and $\vec{\Psi}$ is the corresponding shifted Toeplitz matrix. As will be illustrated in the following derivation, there is no need to pay much attention to the detailed structures of both Ψ and $\vec{\Psi}$.

An orthogonal projection

$$\Pi^{\perp}_{\hat{\mathbf{R}}^{\mathrm{T}}_{u\xi}} = \mathbf{I} - \hat{\mathbf{R}}^{\mathrm{T}}_{u\xi}\left[\hat{\mathbf{R}}_{u\xi}\hat{\mathbf{R}}^{\mathrm{T}}_{u\xi}\right]^{-1}\hat{\mathbf{R}}_{u\xi} \tag{6.7}$$

is introduced to remove the part containing the information Ψ and $\vec{\Psi}$. Thus, (6.5) and (6.6) can be further reduced to

$$\hat{\mathbf{R}}_{y\xi}\Pi^{\perp}_{\hat{\mathbf{R}}^{\mathrm{T}}_{u\xi}} = \mathrm{O}_r\hat{\mathbf{R}}_{x\xi}\Pi^{\perp}_{\hat{\mathbf{R}}^{\mathrm{T}}_{u\xi}} \tag{6.8}$$

$$\hat{\mathbf{R}}_{\vec{y}\xi}\Pi^{\perp}_{\hat{\mathbf{R}}^{\mathrm{T}}_{u\xi}} = \mathrm{O}_r A\hat{\mathbf{R}}_{x\xi}\Pi^{\perp}_{\hat{\mathbf{R}}^{\mathrm{T}}_{u\xi}} \tag{6.9}$$

The matrix A can be approximated by solving the optimization problem

$$\min_{A}\left\|\mathrm{O}_r A\hat{\mathbf{R}}_{x\xi}\Pi^{\perp}_{\hat{\mathbf{R}}^{\mathrm{T}}_{u\xi}} - \hat{\mathbf{R}}_{\vec{y}\xi}\Pi^{\perp}_{\hat{\mathbf{R}}^{\mathrm{T}}_{u\xi}}\right\|_F \tag{6.10}$$

where an estimation of O_r and $\hat{\mathbf{R}}_{x\xi}\Pi^{\perp}_{\hat{\mathbf{R}}^{\mathrm{T}}_{u\xi}}$ are required. Directly solving (6.10) is typically impossible. However, as mentioned in Section 6.3.1, the estimated model parameters $\{\hat{A}, \hat{B}, \hat{C}, \hat{D}\}$ are just required to be equivalent to the true ones in the sense of being within a similarity transformation. The reason is that both of them are equivalent from the perspective of the input-output description. Their mathematical relation is

$$A = M\hat{A}M^{-1}, B = M\hat{B}, C = \hat{C}M^{-1}, D = D$$

where M is a certain nonsingular matrix. Thus, it is sufficient to solve the following relaxed version of (6.10) instead

$$\min_{\hat{A}}\left\|\hat{\mathrm{O}}_r\hat{A}\hat{\mathbf{R}}_{\hat{x}\xi}\Pi^{\perp}_{\hat{\mathbf{R}}^{\mathrm{T}}_{u\xi}} - \hat{\mathbf{R}}_{\vec{y}\xi}\Pi^{\perp}_{\hat{\mathbf{R}}^{\mathrm{T}}_{u\xi}}\right\|_F \tag{6.11}$$

where $\hat{\mathrm{O}}_r = \mathrm{O}_r M$ and $\hat{\mathbf{x}}(t) = M^{-1}\mathbf{x}(t)$. One appropriate selection of the pair $\left\{\hat{\mathrm{O}}_r, \hat{\mathbf{R}}_{\hat{x}\xi}\Pi^{\perp}_{\hat{\mathbf{R}}^{\mathrm{T}}_{u\xi}}\right\}$

is obtained through the singular value decomposition of (6.8)

$$
\begin{aligned}
\hat{\mathbf{R}}_{y\xi}\Pi^{\perp}_{\hat{\mathbf{R}}^{\mathrm{T}}_{u\xi}} &= [U_1, U_2] \begin{bmatrix} \Sigma_1 & 0 \\ 0 & \Sigma_2 \end{bmatrix} \begin{bmatrix} V_1^{\mathrm{T}} \\ V_2^{\mathrm{T}} \end{bmatrix} \\
&\approx \underbrace{U_1\Sigma_1^{1/2}}_{\hat{\mathrm{O}}_r} \underbrace{\Sigma_1^{1/2}V_1^{\mathrm{T}}}_{\hat{\mathbf{R}}_{\hat{x}\xi}\Pi^{\perp}_{\hat{\mathbf{R}}^{\mathrm{T}}_{u\xi}}}
\end{aligned}
\tag{6.12}
$$

where $\Sigma_1 \in \mathbb{R}^{n\times n}$ contains the n dominant singular values and thus can be used to determine the system order n. The matrix $\hat{A}$ can be computed by solving (6.11) and the matrix $\hat{C}$ is directly given as the first m rows of $\hat{\mathrm{O}}_r$.

Even if the actual dynamic system is stable, the estimated matrix A may be unstable due to poor SNR conditions of the experimental data. A stable approximation of matrix A can be guaranteed by solving a modified version of (6.11) with the pole location constraints

$$
\begin{aligned}
\min_{Q,P} \quad & \left\| \hat{\mathrm{O}}_r Q - \hat{\mathbf{R}}^{\rightarrow}_{y\xi}\Pi^{\perp}_{\hat{\mathbf{R}}^{\mathrm{T}}_{u\xi}} \left(\hat{\mathbf{R}}_{\hat{x}\xi}\Pi^{\perp}_{\hat{\mathbf{R}}^{\mathrm{T}}_{u\xi}} \right)^{\dagger} P \right\|_F \\
\text{s.t.} \quad & a \otimes P + b \otimes Q + b^T \otimes Q^{\mathrm{T}} \geq 0 \\
& P = P^{\mathrm{T}} > 0 \\
& \mathrm{Tr}\,(P) = n
\end{aligned}
\tag{6.13}
$$

where $a = 1-\delta, \delta \in [0,1]$, $b = \begin{bmatrix} 0 & 1 \\ 0 & 0 \end{bmatrix}$, and $(\cdot)^{\dagger}$ represents the right pseudo-inverse. Denote the optimum of (6.13) as $\{Q^*, P^*\}$. Then, matrix $\hat{A}$ is estimated by $A = Q^* P^{*-1}$ and the eigenvalues of $\hat{A}$ are enforced to be within the disk with radius $(1-\delta)$ in the complex plain. One can find other different pole location constraints in [32].

6.3.3 Estimation of Matrices B and D

Once the matrices A and C have been estimated through the extended observability matrix, the estimation of B and D is about solving a linear regression optimization based on either the covariance data $\{\hat{R}_{u\xi}(\tau), \hat{R}_{y\xi}(\tau)\}$ or the time-domain data $\{u(t), y(t)\}$. In addition, the identified model is expected to be applicable for a wide range of working power levels although the data for identification is obtained by exciting the system with small perturbations around a given equilibrium. Thus, a priori information of the system should be taken into account to guarantee a reliable model structure.

From the physical perspective, the DERs have unit static gains since the output active and reactive power will always be adjusted to be equal to the set points. For this reason, the unit static gain will be enforced during the estimation of B and D. On the other hand, the mean values will usually be removed beforehand from the covariance data $\{\hat{R}_{u\xi}(\tau), \hat{R}_{y\xi}(\tau)\}$ to focus on the dynamics. Thus, the raw data $\{u(t), y(t)\}$ will be a better choice for the optimization with a gain constraint. The constrained optimization to be solved is then given as follows

$$\begin{aligned} &\min_{x_0, B, D} \quad \left\| \mathbf{Y} - \mathbf{X} \begin{bmatrix} \mathrm{vec}\,(x_0) \\ \mathrm{vec}\,(B) \\ \mathrm{vec}\,(D) \end{bmatrix} \right\|_2 \\ &\text{s.t.} \quad [\hat{C}(\mathbf{I}_n - \hat{A})^{-1} B + D]_{(k_r, k_c)} = \mathrm{Gain}_{(k_r, k_c)} \end{aligned} \tag{6.14}$$

where (k_r, k_c) represents the index of a matrix element in row k_r and column k_c, $\mathrm{Gain}_{(k_r,k_c)}$ is the corresponding predetermined static gain from the k_cth input to the k_rth output, $\hat{A}$

and $\hat{C}$ are estimated from Section 6.3.2, and

$$\mathbf{Y} = \begin{bmatrix} y(0) \\ y(1) \\ \vdots \\ y(N-1) \end{bmatrix}, \mathbf{X} = \begin{bmatrix} \varphi(0) \\ \varphi(1) \\ \vdots \\ \varphi(N-1) \end{bmatrix}$$

$$\varphi(k) = \begin{bmatrix} \hat{C}\hat{A}^k & \sum_{i=0}^{k-1} u(i)^{\mathrm{T}} \otimes \hat{C}\hat{A}^{k-i-1} & u(k)^{\mathrm{T}} \otimes \mathbf{I}_m \end{bmatrix}$$

Note that there is no need to specify the static gains for all the subsystems. In our case, only the six diagonal elements of the transfer function matrix is required to be constrained, i.e.,

$$\mathrm{Gain}_{(j,j)} = 1, j \in \{1, 2, \cdots, 6\} \tag{6.15}$$

6.4 Issues Related to Application of CoBRA

6.4.1 Persistent Excitation

As one important part in system identification, input signal design should be carefully considered to meet persistent excitation requirement. For the traditional subspace methods using (6.2), the persistent excitation is defined as follows [10]:

Definition 6. *[PE for (6.2)]. Denote the input signal as $\mathbf{u}(t)$ for $t \in [0, N-1]$ and build a Hankel matrix as below:*

$$\mathbf{U}_{r,l} = \mathrm{Hankel}\left(\mathbf{u}(t), r, 0, N-1\right), \tag{6.16}$$

where r and $l = N - r + 1 = O(N)$ are the numbers of block rows and columns respectively. Then, input signal $\mathbf{u}(t)$ is persistently exciting of order r if the following $r \times r$ block matrix *has full rank pr:*

$$PE_1 = \frac{1}{l}\mathbf{U}_{r,l}\mathbf{U}_{r,l}^{\mathrm{T}} \tag{6.17}$$

If $\mathbf{u}(t)$ is a stochastic signal, column number l in the above definition should go to infinity or be sufficiently large. In addition, the dimension of $\mathbf{U}_{r,l}$ in (6.16) will become extremely large as the data length N goes to infinity. This is quite challenging for limited memory storage.

However, the CoBRA will have much less storage trouble. Although only the values of sample covariance functions $\{\hat{R}_{u\xi}(\tau), \hat{R}_{y\xi}(\tau)\}$ for bounded τ are used to construct the Hankel matrices, all the information up to $t = N - 1$ is compressed into the covariance functions. The dimensions of Hankel data matrices are pre-determined and will not increase as the data length N goes to infinity. It would be interesting to get an insight into the persistent excitation condition for the CoBRA.

Definition 7. *[PE for (6.3)]. Denote the input signal as $\mathbf{u}(t)$ for $t \in [0, N-1]$ and the instrumental variable as $\xi(t)$ with the same dimension, and build a Hankel matrix as below:*

$$\hat{\mathbf{R}}_{u\xi} = \mathrm{Hankel}\left(\hat{R}_{u\xi}(\tau), r, \tau_1, \tau_2\right) \tag{6.18}$$

where $\hat{R}_{u\xi}(\tau)$ is defined as in (3.4), and $-\bar{\tau} < \tau_1 < \tau_2 < \bar{\tau}$. Note that matrix $\hat{\mathbf{R}}_{u\xi}$ has r block rows and $l_c = \tau_2 - (\tau_1 + r - 2) = O(1)$ block columns. Then, $\{\mathbf{u}(t), \xi(t)\}$ is persistently exciting of order r if the following $r \times r$ block matrix has full rank pr:

$$PE_2 = \hat{\mathbf{R}}_{u\xi}\hat{\mathbf{R}}_{u\xi}^{\mathrm{T}} \tag{6.19}$$

Note that the dimension of $\hat{\mathbf{R}}_{u\xi}$ is fixed as long as r, τ_1, τ_2 are chosen, regardless of the data length N.

An intuitive idea is to apply the excitation signals for traditional subspace methods to the CoBRA. One of the most popular options is the white noise input $\mathbf{u}(t) \sim \mathscr{N}\left(0, \sigma^2 \mathbf{I}\right)$. To make $\xi(t)$ be correlated with $\mathbf{u}(t)$, we simply choose $\xi(t) = \mathbf{u}(t)$. Then, we have $R_{u\xi}(\tau) = \sigma^2 \mathbf{I} \delta(\tau)$. Since the stochastic signals are considered, PE_1 and PE_2 should take their asymptotic version as N goes to infinity. It is well known that PE_1 must be of full rank since it becomes an identity matrix multiplied by σ^2. As to PE_2 for the CoBRA, τ_1, τ_2 should then be carefully chosen to make $\mathbf{R}_{u\xi}$ take the form:

$$\mathbf{R}_{u\xi} = \begin{bmatrix} \cdots & 0 & 0 & 0 & 0 & R_{u\xi}(0) & 0 & \cdots \\ \cdots & 0 & 0 & 0 & R_{u\xi}(0) & 0 & 0 & \cdots \\ & \vdots & \vdots & \iddots & & \vdots & \vdots & \\ \cdots & 0 & R_{u\xi}(0) & 0 & 0 & 0 & 0 & \cdots \end{bmatrix} \tag{6.20}$$

where the positive definite component $R_{u\xi}(0) = \sigma^2 \mathbf{I}$ should appear in every block row.

6.4.2 Arbitrary Data Segments

As mentioned in Section 6.4.1, white noise input can serve as a good excitation signal if the Hankel data matrices are appropriately constructed. In some cases, a long period of excitation signal is not available. For example, in some working conditions, it is possible that the system shows dynamics for some time segments where the input suddenly changes, but stays around a steady state for most of the time. It will be suitable to use those unsteady segments for system identification. Similar as the idea in [96], we come up with a modified CoBRA which can handle arbitrary data segments. In the following discussions, we assume that data series in each segments are sequential and with relatively large length.

Denote the chosen time segments where the system shows dynamics as $[t_i, t_i + L_i) \in [0, N-1]$, where L_i is the length of the ith segments. Using the definition of sample expectation in (3.4) and the ith data segments, one can obtain the following equations similar as (6.3):

$$\begin{cases} \hat{R}^i_{x\xi}(\tau+1) = A\hat{R}^i_{x\xi}(\tau) + B\hat{R}^i_{u\xi}(\tau) \\ \hat{R}^i_{y\xi}(\tau) = C\hat{R}^i_{x\xi}(\tau) + D\hat{R}^i_{u\xi}(\tau) + \hat{R}^i_{v\xi}(\tau) \end{cases} \tag{6.21}$$

Then, the above equations for different segments can be combined together into one form:

$$\begin{cases} \underbrace{\sum_i \hat{R}^i_{x\xi}(\tau+1)}_{\hat{R}_{x\xi}(\tau+1)} = A\underbrace{\sum_i \hat{R}^i_{x\xi}(\tau)}_{\hat{R}_{x\xi}(\tau)} + B\underbrace{\sum_i \hat{R}^i_{u\xi}(\tau)}_{\hat{R}_{u\xi}(\tau)} \\ \underbrace{\sum_i \hat{R}^i_{y\xi}(\tau)}_{\hat{R}_{y\xi}(\tau)} = C\underbrace{\sum_i \hat{R}^i_{x\xi}(\tau)}_{\hat{R}_{x\xi}(\tau)} + D\underbrace{\sum_i \hat{R}^i_{u\xi}(\tau)}_{\hat{R}_{u\xi}(\tau)} + \underbrace{\sum_i \hat{R}^i_{v\xi}(\tau)}_{\hat{R}_{v\xi}(\tau)} \end{cases} \tag{6.22}$$

where $\sum_i \hat{R}^i_{v\xi}(\tau)$ can still be regarded as small in practice and the same notations of (6.3) are used here to unify the equations. Note that there is no need for different segments to be overlapped as in [96]. The reason is that the overlapped data are duplicate information when using the sample covariance functions calculated as (3.4).

6.4.3 Sample Approximation and Imperfect Data

In practice, we may not produce ideal white noise with infinite length. Instead, some other signals with approximately the same spectra of white noise can be used. For example, the finite pseudo-random signals or pseudo-random binary sequences (PRBS) can approximate the white noise quite well. In a worse case, only signals within a given frequency band may be available. Also, the covariance functions in the CoBRA are computed for the finite data using the sample approximation given in (3.4). All these cases will cause the

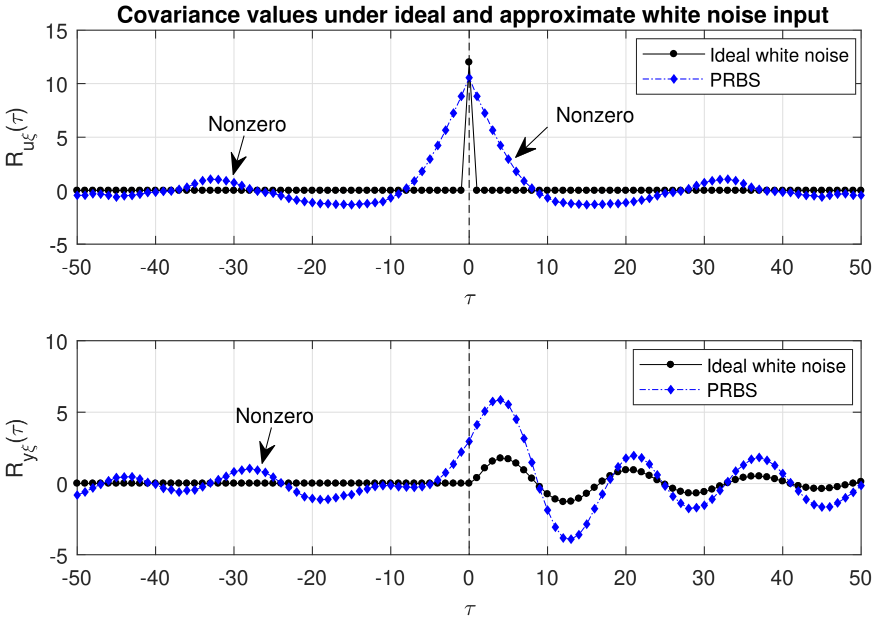


Figure 6.3: The values of covariance functions when using ideal white noise and PRBS as input respectively. Due to sample approximation and imperfect data, those nonzero values indicated in the figure show up in practice.

components, which are zero in ideal case, to be nonzero. For example, those zero components of $\mathbf{R}_{u\xi}$ in (6.20) may also become nonzero. See Fig. 6.3 for an illustration of a single-input-single-out system. However, the equality relations in (6.3) hold rigorously since the sample calculation in (6.3) can be interpreted as a computation of weighting and sum on (6.2). This also means that we don't care about the real covariance values for a possibly stochastic input. Thus, it should be noted that those nonzero data are not approximation errors. Rather, they are containing parts of system dynamics and can be used for parameter estimation in the CoBRA. In addition, power spectrum of a stochastic signal is not used here, so the overlapping technique to reduce the estimation variance in Welch method [97] is not needed. This also implies that data segments overlapping is not needed as mentioned in Section 6.4.2.

6.5 Modeling of the static nonlinearity

6.5.1 Admittance Matrix of the Microgrid Network

Following the conventional definition of the (positive sequence) voltage phasor $\mathbf{V} = V^{rms}e^{j\phi^v}$ and current phasor $\mathbf{I} = I^{rms}e^{j\phi^i}$, along with the (positive sequence) apparent power

$$\mathbf{S} = \mathbf{V}\mathbf{I}^* = V^{rms}I^{rms}e^{j\left(\phi^v - \phi^i\right)} = P + jQ \tag{6.23}$$

where rms denotes the root mean square of a signal and is omitted in the following discussions for notational simplicity. The static coupling between phasors and power loss at the PCC can be characterized by the admittance matrix of the microgrid network. In particular for the microgrid framework of Fig. 6.2, the relation between the phasors is given by

$$\mathbf{M} \begin{bmatrix} \mathbf{V}_b \\ \mathbf{V}_d \\ \mathbf{V}_s \\ \mathbf{V}_{PCC} \end{bmatrix} = \begin{bmatrix} \mathbf{I}_b \\ \mathbf{I}_d \\ \mathbf{I}_s \\ \mathbf{I}_{PCC} \end{bmatrix} \tag{6.24}$$

where $\mathbf{I}_b, \mathbf{I}_d$, and $\mathbf{I}_s$ are the currents going through the BESS, diesel, and solar/PV respectively, $\mathbf{I}_{PCC} = -\mathbf{I}_b - \mathbf{I}_d - \mathbf{I}_s$. The admittance matrix $\mathbf{M}$ in (6.24) is defined by

$$\mathbf{M} = \begin{bmatrix} Y_b & 0 & 0 & -Y_b \\ 0 & Y_d & 0 & -Y_d \\ 0 & 0 & Y_s & -Y_s \\ -Y_b & -Y_d & -Y_s & Y_b + Y_d + Y_s \end{bmatrix} \tag{6.25}$$

and it should be noted that only the power flows between the main grid and DERs are considered since we are only interested in the dynamic modeling from the DERs to the

PCC. The load is regarded as either known or constant during a relatively short period for system identification. Any slight change of the power flow through the load can be regarded as the disturbance.

The admittance matrix $\mathbf{M}$ of the network is dependent on the structure of the local transmission lines and can be approximated by using (6.24). Only the admittance between PCC and each DER is taken into account here. A more general case is also considering the admittance between the DERs, which results in a similar admittance matrix. However, for the structure (6.25) used in this chapter, each line admittance can be simply estimated separately instead of computing the whole admittance matrix as

$$Y_k = \frac{\mathrm{I}_k}{\mathrm{V}_k - \mathrm{V}_{PCC}}, k = b, d, s \tag{6.26}$$

6.5.2 Estimation of the Power Loss

The transmission line with admittance Y_k ($k = b, d, s$) is modelled as a serial connection of a resistor R_k and an inductor L_k, where $Y_k = \frac{1}{R_k + jL_k}$. The total power loss is computed as $\mathbf{I}_k^2(R_k + jL_k)$, which is proportional to the through current squared. Since the main grid is working as the voltage source while in grid-connected mode, one can denote the nominal voltage at the PCC as

$$\mathbf{V}_{PCC} = V_{PCC} e^{j\phi^v_{PCC}} = V_{PCC}^{re} + jV_{PCC}^{im} \tag{6.27}$$

Similar notations can be defined for $Y_k, V_k, k = b, d, s$. Then, the following nonlinear equations can be obtained by combining (6.23), (6.24), (6.26), and (6.27)

$$\begin{cases} Y_k^{re}\left(V_k^{re}\right)^2 - \left(Y_k^{re}V_{PCC}^{re} - Y_k^{im}V_{PCC}^{im}\right)V_k^{re} \\ \quad +Y_k^{re}\left(V_k^{im}\right)^2 - \left(Y_k^{re}V_{PCC}^{im} + Y_k^{im}V_{PCC}^{re}\right)V_k^{im} = P_k \\ -Y_k^{im}\left(V_k^{re}\right)^2 + \left(Y_k^{re}V_{PCC}^{im} + Y_k^{im}V_{PCC}^{re}\right)V_k^{re} \\ \quad -Y_k^{im}\left(V_k^{im}\right)^2 - \left(Y_k^{re}V_{PCC}^{re} - Y_k^{im}V_{PCC}^{im}\right)V_k^{im} = Q_k \end{cases} \tag{6.28}$$

where $k = b, d, s$ to represent the battery, diesel, and solar channel respectively. Thus, there are actually 3 pairs of nonlinear equations in total to solve $\mathbf{V}_k = V_k^{re} + jV_k^{im}, k = b, d, s$. The practical solution can be obtained directly by solving (6.28) via nonlinear optimization with the initial value as $(V_{PCC}^{re}, V_{PCC}^{im})$. The values of P_k, Q_k can be obtained from the outputs of the linear DERs system identified using the techniques in Section 6.3. In terms of Fig. 6.2,

$$\begin{cases} P_b = \mathrm{P}_b^{\mathrm{out}}, Q_b = \mathrm{Q}_b^{\mathrm{out}} \\ P_d = \mathrm{P}_d^{\mathrm{out}}, Q_d = \mathrm{Q}_d^{\mathrm{out}} \\ P_s = \mathrm{P}_s^{\mathrm{out}}, Q_s = \mathrm{Q}_s^{\mathrm{out}} \end{cases} \tag{6.29}$$

It is worthwhile to mention that the problem can be further simplified by ignoring the wire inductance if it is much smaller compared to the resistance. Thus, $Y_k \approx \frac{1}{R_k}$ by neglecting the imaginary part and (6.28) can be further reduced to

$$\begin{cases} \left(V_k^{re}\right)^2 - V_k^{re}V_{PCC}^{re} + \left(V_k^{im}\right)^2 - V_k^{im}V_{PCC}^{im} = R_kP_k \\ V_k^{re}V_{PCC}^{im} - V_k^{im}V_{PCC}^{re} = R_kQ_k \end{cases} \tag{6.30}$$

The corresponding currents $\mathbf{I}_k = I_k^{re} + jI_k^{im}$ can be computed from (6.24) once (V_k^{re}, V_k^{im}) is obtained. The active power loss going through each power channel is then $P_k^{loss} = [(I_k^{re})^2 +$

$(I_k^{im})^2]R_k$. The reactive power loss is $Q_k^{loss} = [(I_k^{re})^2 + (I_k^{im})^2]L_k$. Note that Q_k^{loss} can be neglected for the case of small inductance in the transmission lines and then $Q_k^{loss} = 0$. The total active power loss reflected at the PCC is $\mathrm{P}_{\mathrm{PCC}}^{\mathrm{loss}} = \sum_k P_k^{loss}$ while the total reactive power loss $\mathrm{Q}_{\mathrm{PCC}}^{\mathrm{loss}} = \sum_k Q_k^{loss}(= 0)$.

Denote the initial active and reactive power levels at the PCC as $\mathrm{P}_{\mathrm{PCC}}^{\mathrm{init}}$ and $\mathrm{Q}_{\mathrm{PCC}}^{\mathrm{init}}$ when the three channels are neither producing or consuming power. Then, the active and reactive power at the PCC shown in Fig. 6.2 considering the power loss are estimated by

$$\mathrm{P}_{\mathrm{PCC}} = \mathrm{P}_{\mathrm{PCC}}^{\mathrm{init}} - \sum_k P_k + \mathrm{P}_{\mathrm{PCC}}^{\mathrm{loss}} \tag{6.31}$$

$$\mathrm{Q}_{\mathrm{PCC}} = \mathrm{Q}_{\mathrm{PCC}}^{\mathrm{init}} - \sum_k Q_k + \mathrm{Q}_{\mathrm{PCC}}^{\mathrm{loss}} \tag{6.32}$$

6.6 Simulation Examples

In this chapter, a microgrid that is connected to the main grid and consists of a hybrid collection of DERs interconnected with transmission lines and loads as shown in Fig. 6.1 is considered. The detailed component microgrid model is constructed using the toolbox Simscape Power Systems$^{\mathrm{TM}}$ in MATLAB/Simulink. Simscape simulator allows one to produce simulation data under realistic conditions and validate identification and control techniques under practical scenarios.

The transmission lines are all modelled as a serial concatenation of a 1 Ω resistor and $10e^{-5}$ H inductor. The corresponding impedance can be approximated as 1 Ω by neglecting the inductance. Thus, the admittance $Y_k \approx 1$ S for $k = b, d, s, load$ and the reduced version (6.30) can be used. Note that $\mathrm{Q}_{\mathrm{PCC}}^{\mathrm{loss}} = 0$ in the simplified case. The nominal voltage at the PCC with the phasor form in (6.27) is set as $V_{PCC} = 480$ V and $\phi_{PCC}^{v} = 10$ degree with nominal frequency 60 Hz. Note that, in practice, the frequency within the microgrid will not be constant but oscillating around the nominal value regulated by the voltage source. The magnitude and angle of the PCC voltage also show a similar behavior. We will still use the nominal PCC values to solve (6.30) and the estimation result will show that the small oscillation is not a problem. The load is modelled as the parallel RLC load with 20 kW nominal active power, 4 kVAR nominal inductive reactive power, and -100 VAR nominal capacitive reactive power. In addition, the rate limiter is not considered in the simulation example for easy discussion. However, the rate limiter will be taken into account in the next section where the practical data is used.

Five working scenarios, with 100 seconds total simulation time and corresponding to different power levels, are considered and described in Table. 6.1. The initial power levels at the PCC are $\mathrm{P}_{\mathrm{PCC}}^{\mathrm{init}} = 18.3$ kW and $\mathrm{Q}_{\mathrm{PCC}}^{\mathrm{init}} = 3.45$ kVAR. The nominal power set points of the solar inverter are assumed to be constant during the whole simulation. There will be only one DER power level step change between two scenarios. For example, the nominal active power level of the battery inverter system will drop down to -20 kW from 0 kW at $t = 60$s while the other set points are not changed.

Table 6.1: Nominal values for five scenarios.

Input	Initial	1 (0-60s)	2 (60-70s)	3 (70-80s)	4 (80-90s)	5 (90-100s)
$\mathrm{P}_{\mathrm{b}}^{\mathrm{in}}$	0 kW	0 kW	**–20 kW**	–20 kW	–20 kW	–20 kW
$\mathrm{Q}_{\mathrm{b}}^{\mathrm{in}}$	0 kVAR	0 kVAR	0 kVAR	**–5 kVAR**	–5 kVAR	–5 kVAR
$\mathrm{P}_{\mathrm{d}}^{\mathrm{in}}$	0 kW	0 kW	0 kW	0 kW	**40 kW**	40 kW
$\mathrm{Q}_{\mathrm{d}}^{\mathrm{in}}$	0 kVAR	0 kVAR	0 kVAR	0 kVAR	0 kVAR	**10 kVAR**
$\mathrm{P}_{\mathrm{s}}^{\mathrm{in}}$	0 kW	**30 kW**	30 kW	30 kW	30 kW	30 kW
$\mathrm{Q}_{\mathrm{s}}^{\mathrm{in}}$	0 kVAR	0 kVAR	0 kVAR	0 kVAR	0 kVAR	0 kVAR
Output	Initial					
$\mathrm{P}_{\mathrm{PCC}}$	18.3 kW					
$\mathrm{Q}_{\mathrm{PCC}}$	3.45 kVAR					

Six independent PRBS signals with magnitude 3 kW/kVAR are added around the nominal set points to excite the system. The data of the first 50 seconds under the scenario 1 is used for identification by using the techniques proposed in Section 6.3 and 6.5. The remaining data including all five scenarios is used for validation. The measurements are sampled at 60 Hz and contaminated by Gaussian white noise with SNR=25 dB.

For the sake of comparison, we also use the CoBRA method to directly identify a linear model from the DER set points to the output power $\mathrm{P_{PCC}}$ and $\mathrm{Q_{PCC}}$ without compensating the transmission line loss. The corresponding results are shown in Fig. 6.4. The model output is not precisely matching but acceptable for scenario 1 as expected since the model is identified by using scenario 1 data. However, the model is getting off when the microgrid system is moving to other working power levels. Thus, one can use a single linear model without considering the power loss if the microgrid system is working within a relatively small range of power levels. Otherwise, the nonlinearity caused by the power loss due to the transmission line impedance should be taken into account.

For the proposed method in this chapter, the parameters in the covariance data equation (6.4) are chosen as $r = 30, \tau_1 = 0, \tau_2 = 150$. The linear DERs system is identified as a 14th order model. The final active and reactive power at PCC are estimated as in (6.31) and (6.32), where the power loss due to transmission lines is compensated. The estimation results are illustrated in Fig. 6.5. The identified model by using the proposed method shows high accuracy under not only scenario 1 but also the other scenarios. This is desirable in practice since one would like the model identified around one working power level to be applicable for other levels.

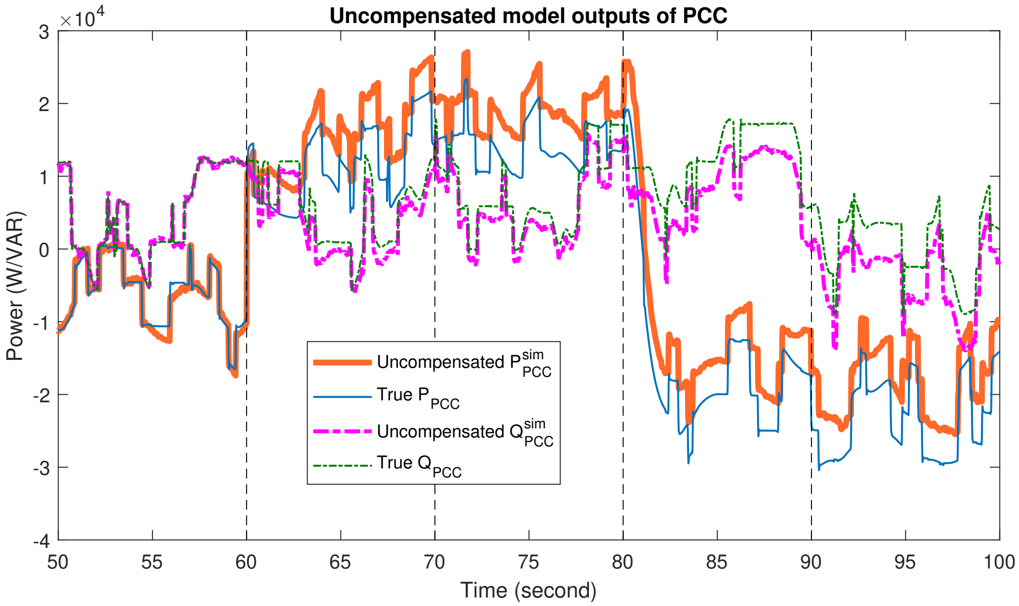


Figure 6.4: The estimation results of direct identification from the input set points to the output power at the PCC without compensating the line loss. The vertical black dashed line separates the time window into the five scenarios given in Table. 6.1.

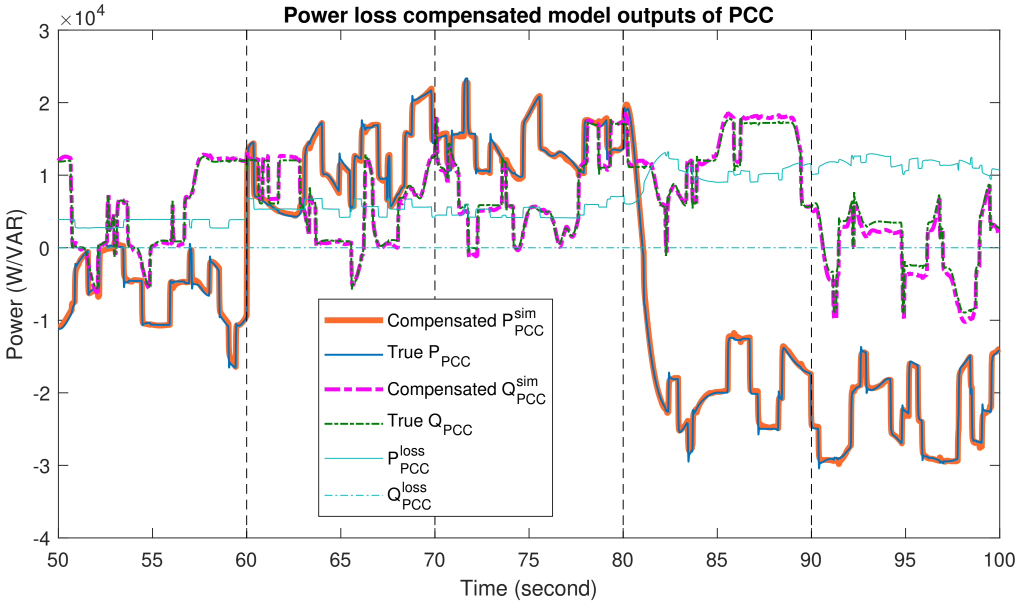


Figure 6.5: The estimation results of the proposed method using (6.31) and (6.32). The vertical black dashed line separates the time window into the five scenarios given in Table. 6.1.

6.7 Experimental Results

The proposed algorithm is also applied to data obtained from a local area distribution network (LADN), operating as a grid-connected microgrid where DERs are placed behind local distribution transformer. A schematic overview of the LADN with its Point of Interconnect (POI) or PCC is shown in Fig 6.6, illustrating the presence of a BESS DER and a PV DER each behind their own distribution transformers. In addition, a non-controllable but measurable load is present in the LADN behind a distribution transformer. The data of this LADN is chosen to illustrate that presence of local distribution transformer can be taken into account in the microgrid dynamic model by estimating the dynamics and power loss as part of the dynamic model.

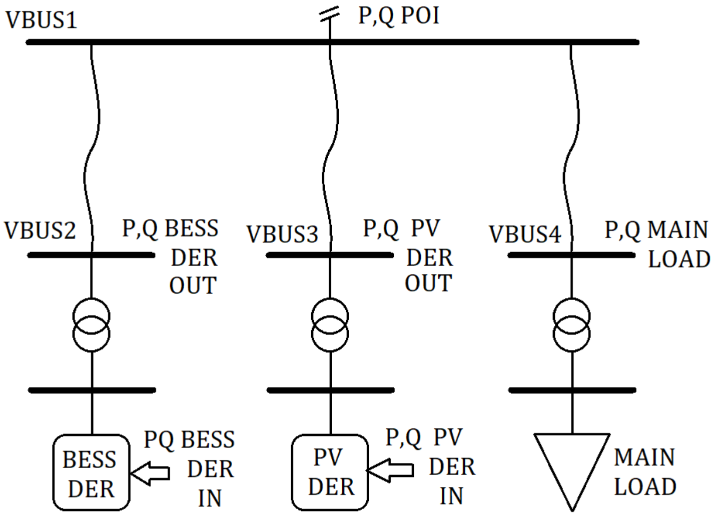


Figure 6.6: Schematic overview of Local Area Distribution Network.

The BESS DER is rated at 250 kW and the PV DER at 500 kW. The BESS DER is excited by random step perturbations on the active and reactive power, while the PV DER is excited with slowly changing random power variations, curtailing solar power production. In terms of notation, these excitation signals correspond to $\mathrm{P}_{\mathrm{b}}^{\mathrm{in}}$, $\mathrm{Q}_{\mathrm{b}}^{\mathrm{in}}$, $\mathrm{P}_{\mathrm{s}}^{\mathrm{in}}$, $\mathrm{Q}_{\mathrm{s}}^{\mathrm{in}}$ as discussed earlier, while there is no diesel generator in the system under test. The network is connected to a main load that can be up to 1.5 MW. The BESS, PV, and load power outputs are measured on bus 2,3,4 respectively and can be denoted as $\mathrm{P}_{\mathrm{b}}^{\mathrm{out}}$, $\mathrm{Q}_{\mathrm{b}}^{\mathrm{out}}$, $\mathrm{P}_{\mathrm{s}}^{\mathrm{out}}$, $\mathrm{Q}_{\mathrm{s}}^{\mathrm{out}}$, $\mathrm{P}_{\mathrm{load}}^{\mathrm{out}}$, and

$\mathrm{Q}_{\mathrm{load}}^{\mathrm{out}}$ accordingly. The admittance of the transmission line between bus 1 and bus 2,3,4 is then denoted as Y_b, Y_s, and Y_{load}.

The nominal $V_{POI} = 12$ kV in the voltage phasor $\mathbf{V}_{POI} = V_{POI}e^{j\phi_{POI}^v}$ at the POI similar to (6.27). It is worthwhile to note that the positive sequence voltage angle ϕ_{POI}^v at the POI is not constant due to the time variations in the AC grid frequency to which the LADN is connected via the POI. However, since the active and reactive power components of the apparent power $\mathbf{S}$ in (6.23) are used, only $\cos(\cdot)$ and $\sin(\cdot)$ of the angle difference $\phi_{POI}^v - \phi_{POI}^i$ are relevant. All PMU data is sampled at 30 Hz and data is collected over a time window of 1 hour long. For estimation and modeling purposes, only the first 11 minutes are used as training data and the rest for validation.

It is obvious that the network under test is similar to the one in Fig. 6.1 and the proposed method can be implemented. Similar to (6.1), the input/output signals for system identification is modified as

$$\mathbf{u}_0(t) = \begin{bmatrix} \mathrm{P}_b^{\mathrm{in}}(t) \\ \mathrm{Q}_b^{\mathrm{in}}(t) \\ \mathrm{P}_s^{\mathrm{in}}(t) \\ \mathrm{Q}_s^{\mathrm{in}}(t) \end{bmatrix}, \mathbf{y}(t) = \begin{bmatrix} \mathrm{P}_b^{\mathrm{out}}(t) \\ \mathrm{Q}_b^{\mathrm{out}}(t) \\ \mathrm{P}_s^{\mathrm{out}}(t) \\ \mathrm{Q}_s^{\mathrm{out}}(t) \end{bmatrix} \tag{6.33}$$

The initial active and reactive power at the POI are computed as $\mathrm{P}_{\mathrm{POI}}^{\mathrm{init}} = -\mathrm{P}_{\mathrm{load}}^{\mathrm{out}}$ and $\mathrm{Q}_{\mathrm{POI}}^{\mathrm{init}} = -\mathrm{Q}_{\mathrm{load}}^{\mathrm{out}}$. The admittance Y_b, Y_s, and Y_{load} can be determined by (6.24). As shown in Fig. 6.7, the rate limiter for BESS and PV DERs can be determined from the slope of two selected points in the power output when there is a high rate input change. The intermediate signal $\mathbf{u}(t)$ illustrated in Fig. 6.2 can be then computed. An 11th order linear state-space model from $\mathbf{u}(t)$ to $\mathbf{y}(t)$ is identified by the CoBRA. This model can be used to predict $\mathbf{y}(t)$ given

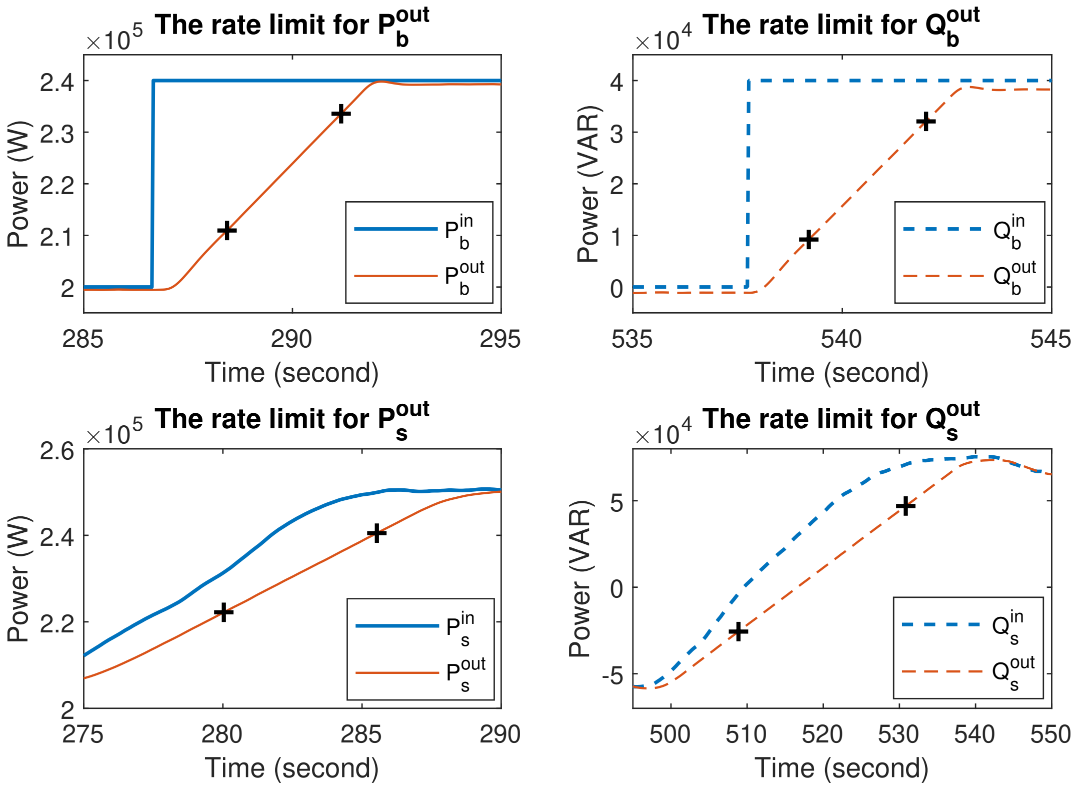


Figure 6.7: Determine the rate limiter from the input/output data by calculating the slope between the two selected points denoted by '+'.

any $\mathbf{u}(t)$ without using PMUs. The active and reactive power losses, denoted as $\mathrm{P}_{\mathrm{POI}}^{\mathrm{loss}}$ and $\mathrm{Q}_{\mathrm{POI}}^{\mathrm{loss}}$, are calculated as $\mathrm{P}_{\mathrm{POI}}^{\mathrm{loss}} = \sum_k P_k^{loss}, \mathrm{Q}_{\mathrm{POI}}^{\mathrm{loss}} = \sum_k Q_k^{loss}$, where $k = b, s, load$ for this test and (6.28) will be solved in the process. Similar to (6.31) and (6.32), the active and reactive power at the POI can be estimated as

$$\mathrm{P}_{\mathrm{POI}} = \mathrm{P}_{\mathrm{POI}}^{\mathrm{init}} - (\mathrm{P}_{\mathrm{b}}^{\mathrm{out}} + \mathrm{P}_{\mathrm{s}}^{\mathrm{out}}) + \mathrm{P}_{\mathrm{POI}}^{\mathrm{loss}} \tag{6.34}$$

$$\mathrm{Q}_{\mathrm{POI}} = \mathrm{Q}_{\mathrm{POI}}^{\mathrm{init}} - (\mathrm{Q}_{\mathrm{b}}^{\mathrm{out}} + \mathrm{Q}_{\mathrm{s}}^{\mathrm{out}}) + \mathrm{Q}_{\mathrm{POI}}^{\mathrm{loss}} \tag{6.35}$$

The comparison between the model prediction from simulation and the true power output at the POI is illustrated in Fig. 6.8. The model prediction considering the power loss in the transmission line is more accurate, as also demonstrated in the earlier simulation

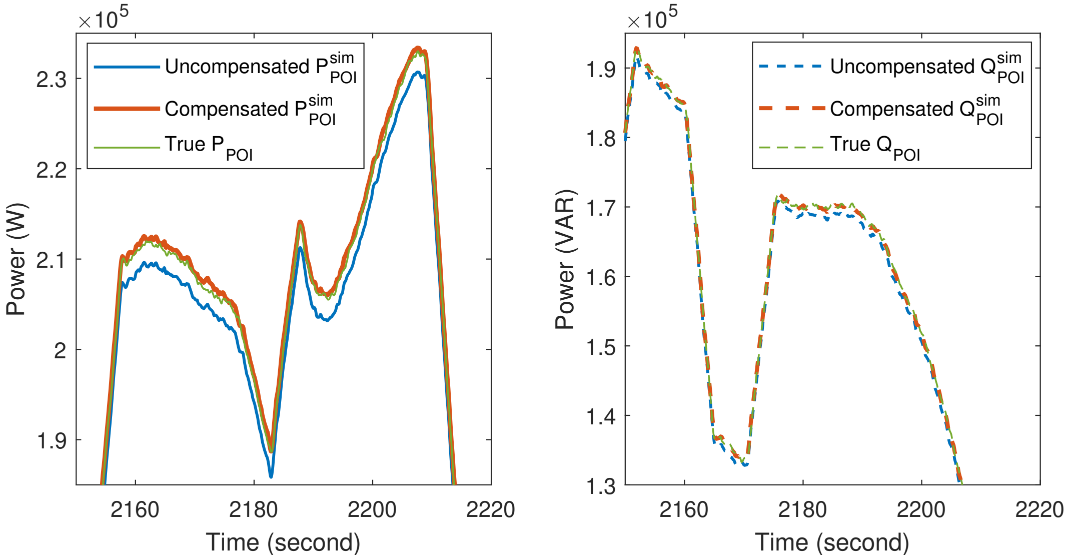


Figure 6.8: The comparison between the model prediction and the true power output at the POI. Compensating the power loss in the model leads to a more accurate prediction.

example. It is worth noting that the CoBRA can accurately capture the coupling between different channels as shown in Fig. 6.9.

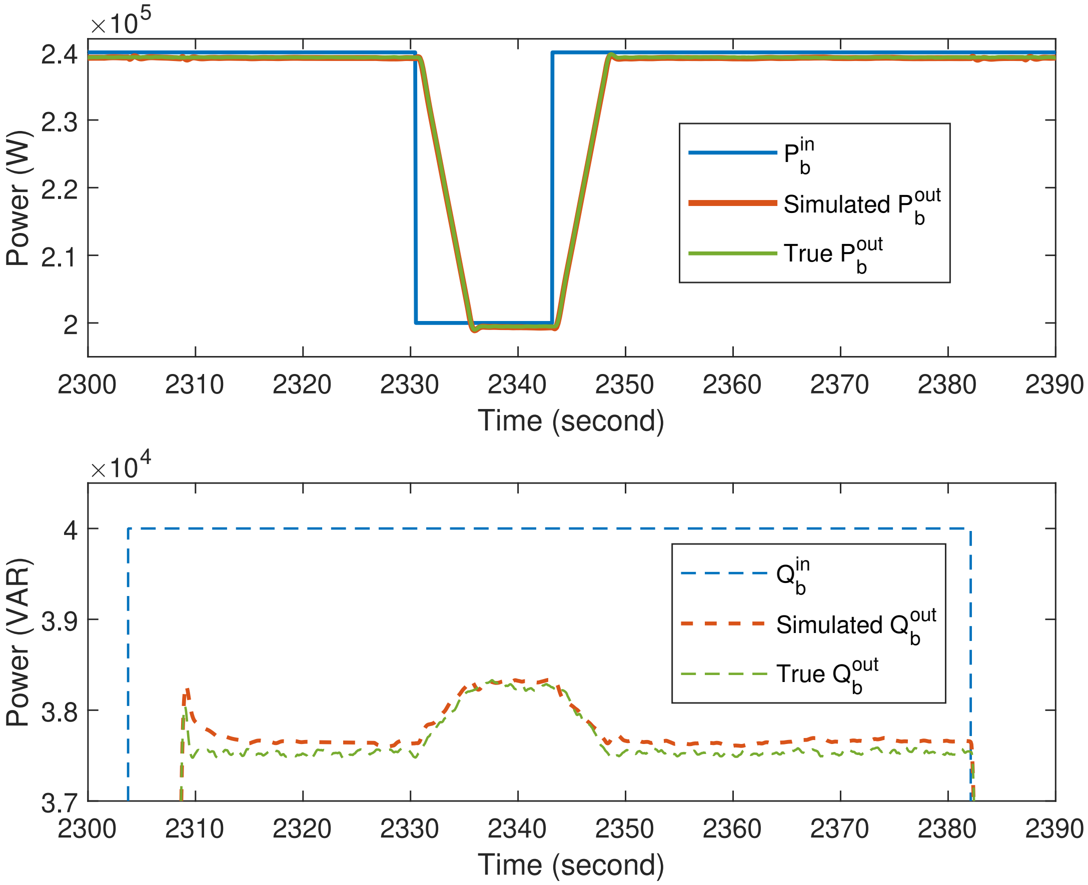


Figure 6.9: The identified model by the CoBRA is able to capture the coupling between $\mathrm{P}_\mathrm{b}^\mathrm{out}$ and $\mathrm{Q}_\mathrm{b}^\mathrm{out}$.

6.8 Conclusion

A model structure combining a linear dynamic system and static nonlinearity due to the power loss is proposed to model the power flow dynamics of a microgrid with multiple DERs. The linear part is identified via the CoBRA while the nonlinear part is formulated by the estimation of the admittance matrix of the network of the microgrid. The model parameters are estimated by using power flow input/output data obtained via PMUs using small perturbations of power flow around an operating condition of the microgrid. The simulation and experimental results demonstrate that the resulting model is suitable for not only the current working power levels but also the other operating conditions. This is desirable in practice since we would like a model identified around an equilibrium to be applicable when the system is moving to another equilibrium. Ignoring the nonlinear part results in a model that can not capture the nonlinear feature due to the line loss and is only acceptable within a small range around the current operating level. Once the proposed model is formulated, one can carry out the model based controller design to manipulate the set point values of DERs to achieve the desired objective and dynamic performance at the PCC while taking the power loss at transmission lines into account.

This chapter is based on the following papers that were published or submitted:

Y. Hu, S.A.R. Konakalla and R.A. de Callafon, "Covariance based estimation for reduced order models of microgrid power flow dynamics," in *Proceedings of IFAC 18th Symposium on System Identification (SYSID 2018)*, Stockholm, Sweden, Jul. 9–11, 2018, pp. 903–908.

Y. Hu and R.A. de Callafon, "Microgrid dynamic modeling with power flow covariance data," *IEEE Transactions on Power Systems*, under review.

Chapter 7

Tensor Network Based MIMO Volterra Model for Lithium-ion Batteries

In this chapter, we develop a Tensor Network (TN) based Volterra double-capacitor (VDC) model for lithium-ion batteries to improve the prediction performance of the nonlinear double-capacitor (NDC) model. It is shown that the VDC model maintains the advantages of the NDC model to account for the rate capacity effect and the voltage recovery effect. In addition, the VDC model is capable of predicting both static and dynamic nonlinearities simultaneously in a more accurate way.

7.1 Introduction

Rechargeable batteries have become a major driver for applications ranging from portable consumer electronics to electric vehicles and microgrid applications. Among them, lithium-ion batteries (LiBs), having a high power/energy density, a long life span, high energy storage efficiency, and environment friendliness, are attracting more and more attention in both research and application fields [98, 99, 100, 101, 102]. There are roughly two major developed efforts in LiBs: one is focusing on the battery cell design, materials selection for electrodes and electrolytes, pack structure optimization etc., to develop high-performance batteries; the other one is implementing advanced battery management systems (BMS) to maximize the utilization efficiency and safety for given types of batteries generally based on estimation of the State-of-Charge (SoC), State-of-Power (SoP), and State-of-Health (SoH).

Algorithms for BMS generally require mathematical models that describe battery physics and dynamics. Two main types of battery models have been intensively investigated: physical models based on electrochemical principles and equivalent circuit models (ECMs) by referring to electric circuit theory and system theory [103, 104, 105, 106, 99, 107, 108, 109, 110, 111, 112, 113, 114, 115, 116, 117]. A brief comparison between the two models can be found in [115], where the NDC model was proposed to bridge the gap between physical models and ECM. The NDC model is a modification of a linear double-capacitor model [118] while maintaining feature to account for rate capacity effects and energy recovery effects. The nonlinear mapping introduced in the NDC model is based on the observation of the nonlinear SoC-OCV (Open Circuit Voltage) curve. However, the SoC-OCV curve is a static feature for the LiBs fully at rest and does not reflect the full nonlinearity when the batteries are used dynamically. As shown in the validation experiment for identification 2.0 in [115], the resulting SoC-OCV curve identified by the NDC model does not fully match the measured curve when the training process was trying to capture both the static and dynamic nonlinearities. Despite the less accurate static prediction, the dynamic prediction was still improved by using the NDC model. This observation provided the motivation for this chapter to propose a new model to account for the two types of nonlinearities simultaneously.

Both electrochemical models and ECMs have physical explanations for the underlying processes happening inside of the battery during charging/discharging. However, due to the limited knowledge about the complicated dynamic processes involved in charging/discharging, they are impossible to model all the processes. Pure data-driven modeling approaches such as system identification or machine learning provide an alternative to first-principles modeling [2, 3, 119, 46, 72]. Instead of figuring out the detailed electrochemical and physical processes, data-driven modeling uncovers the underlying relationship between the historical input/output data from the perspectives of statistics and optimization. These techniques have been demonstrated effective in different battery modeling applications [120, 90, 121, 122, 123, 124, 125]. However, a large amount of data is generally required to guarantee such an effective model, and the resulting model can easily suffer overfitting problems.

In this chapter, a hybrid combination of an ECM and TN-based supervised learning is used for battery modeling to enable physics-informed data-driven learning of the unknown nonlinear dynamics inherent to batteries. For the ECM part, the linear double-capacitor model is used to extract the voltage and SoC features within the battery. These features are then fed into a supervised learning framework as shown in [126] to learn the nonlinear dynamics between the raw input features and output terminal voltage. However, compared to [126], this chapter introduces a new modification that the input features are further encoded into a feature map described by a Volterra model which can also be equivalently written into a TN representation for better computational efficiency. The reason for using TN representation is that it can mitigate the curse of dimensionality encounterd by a normal Volterra model with a large memory length and a high model degree [64]. Thus, a TN representation allows the Volterra model to capture complex coupled nonlinear dynamics from the numerical perspective. In addition, we propose the bond core sweeping algorithm, which uses the ε-truncated singular value decomposition (SVD) [68], to ignore less significant modes and seek the low-rank property intrinsic to data to avoid overfitting.

The contributions of this chapter can be summarized as follows.

First, a novel TN-based VDC model is developed. A linear double-capacitor model proposed in [118] with the currents as the input signals is used as an initial feature extractor. The raw features produced by the linear double-capacitor model are then fed into a supervised learning structure described by a TN-based Volterra model to predict the output terminal voltages. The proposed VDC model maintains some advantages of the linear double-capacitor model such as capturing the rate capacity effect and the voltage recovery effect. On the other hand, there are different sources of the dynamic nonlinearities including the hysteresis for charging/discharging, unmodelled dynamics, etc. The linear double-capacitor model or the NDC model, as an approximation of the true dynamics inherent to a LiB, involves model mismatch inevitably. The proposed VDC model then has an advantage over the NDC model proposed in [115] in terms of accounting for both static and dynamic nonlinearities simultaneously in a more accurate way by incorporating a data-driven technique.

To our best knowledge, this study is the first to apply TN-based Volterra techniques in the application of battery modeling.

Second, a bond core sweeping algorithm is developed to estimate the TN-cores of the proposed VDC model. The proposed algorithm is developed by modifying the sweeping optimization algorithm in [126]. The first major difference is that a least squares technique is used to estimate each bond core instead of using gradient descent algorithm. The reason is that the convergence of the gradient descent algorithm can be rather slow for a problem in which the gradient is rank deficient. This problem is encountered in our application. The second major difference is that an ε-truncated singular value decomposition is used when splitting the bond core into two individual TN-cores. The advantage is that the model complexity can be adjusted and simplified by emphasizing the low-rank estimation. Seeking a low-rank approximation is important to avoid overfitting.

Third, experimental validation is performed to assess the performance of the proposed VDC model. The approach to generating the training data for a nonlinear dynamic battery modeling is presented. The effect of the range of current magnitudes and SoC values swept by the training data on the model prediction is also investigated. A comparison of the VDC model with the NDC model is also given to show the efficacy of the proposed VDC model.

The remaining part of this chapter is organized as follows. Section 7.2 reviews the TN representation of a MIMO Volterra model. Section 7.3 presents the modeling and parameter estimation of a LiB. We propose the bond core sweeping algorithm to estimate the parameters in the VDC model. The formulation of training data is also discussed to facilitate the parameter estimation. An experimental validation is investigated in Section 7.4. We summarize the chapter in Section 7.5.

7.2 MIMO Volterra Model in the TN Representation

In this section, the TN representation of a MIMO Volterra model is recalled. Denote the output signal as $\mathbf{y}(t) \in \mathbb{R}^m$ and the input signal $\mathbf{u}(t) \in \mathbb{R}^p$. The αth element of a vector $\mathbf{x}(t)$ is written as $x^{(\alpha)}(t)$. Assume that there is a nonlinear relationship between $\mathbf{u}(t)$ and $\mathbf{y}(t)$ that can be described by a MIMO Volterra model.

A normal description of a discrete-time p-input m-output Volterra system of degree d can be described as

$$\mathbf{y}\left(t\right) = \mathbf{h}_0 + \sum_{i=1}^{d} \sum_{k_1,\cdots,k_i=0}^{M-1} \sum_{\alpha_1,\cdots,\alpha_i=0}^{p-1} \mathbf{h}_{1:i}^{k,\alpha} \phi_{1:i}^{k,\alpha}\left(\mathbf{u}\left(t\right)\right) \tag{7.1}$$

where M is the memory length, $\phi_{1:i}^{k,\alpha}(\mathbf{u}(t)) \in \mathbb{R}$ and $\mathbf{h}_{1:i}^{k,\alpha} \in \mathbb{R}^m$ are the ith Volterra kernel function and corresponding kernel coefficient. Specifically, $\phi_{1:i}^{k,\alpha}$ and $\mathbf{h}_{1:i}^{k,\alpha}$ are the abbreviations of functions of $\{k_1, \alpha_1; k_2, \alpha_2; \cdots; k_i, \alpha_i\}$ such that

$$\phi_{1:i}^{k,\alpha}\left(\mathbf{u}\left(t\right)\right) = \prod_{j=1}^{i} u^{(\alpha_j+1)}\left(t - k_j\right) \tag{7.2}$$

$$\mathbf{h}_{1:i}^{k,\alpha} = \mathbf{h}_i\left(k_1, \alpha_1; k_2, \alpha_2; \cdots; k_i, \alpha_i\right) \tag{7.3}$$

The kernel function $\phi_{1:i}^{k,\alpha}(\cdot)$ is a multiplication of different degrees among the possibly different input components at possibly different time instants. This allows one to capture high order nonlinearity and even coupled dynamics among different input channels and past states. Thus, this model structure has great potential for nonlinear system identification.

However, it should be noted that the number of all the kernel coefficients is $1 + \sum_{i=1}^{d}\left(pM\right)^i$ and suffers from the curse of dimensionality as the degree d goes up. Avoiding the choice of a large d prevents an overburdened storage requirement for a computer, but limits the model's capability of capturing complex and even coupled dynamics. On the other hand, even if the storage demand is not a problem, recording every detailed kernel coefficient may be unnecessary since not every kernel will be dominantly active in practice, which motivates us to seek a low-rank representation of (7.1).

TN representation can be used to address this issue. Following the discussion and formulation in [64], one can incorporate all the kernel coefficients into a $(d+1)$-way Volterra tensor $\boldsymbol{\mathcal{V}} \in \mathbb{R}^{(pM+1)\times\cdots\times(pM+1)\times m}$ such that

$$\begin{aligned} \mathbf{y}^{\mathrm{T}}(t) &= \boldsymbol{\mathcal{V}} \times_1 \mathbf{u}_t^{\mathrm{T}} \times_2 \mathbf{u}_t^{\mathrm{T}} \cdots \times_d \mathbf{u}_t^{\mathrm{T}} \\ &= (\boldsymbol{\mathcal{V}}_1 \times_2 \mathbf{u}_t^{\mathrm{T}})(\boldsymbol{\mathcal{V}}_2 \times_2 \mathbf{u}_t^{\mathrm{T}}) \cdots (\boldsymbol{\mathcal{V}}_d \times_2 \mathbf{u}_t^{\mathrm{T}}) \end{aligned} \tag{7.4}$$

where

$$\mathbf{u}_t = \left[1, \mathbf{u}^{\mathrm{T}}(t), \mathbf{u}^{\mathrm{T}}(t-1), \cdots, \mathbf{u}^{\mathrm{T}}(t-M+1)\right]^{\mathrm{T}} \in \mathbb{R}^{pM+1}$$

and $\{\boldsymbol{\mathcal{V}}_1, \cdots, \boldsymbol{\mathcal{V}}_d\}$ are the TN-cores of the Volterra tensor $\boldsymbol{\mathcal{V}}$. Note that the last core $\boldsymbol{\mathcal{V}}_d \in \mathbb{R}^{r_{d-1}\times(pM+1)\times m}$ and the remaining $\boldsymbol{\mathcal{V}}_i \in \mathbb{R}^{r_{i-1}\times(pM+1)\times r_i}$ with $r_0 = 1$. In fact, the TN-representation used here is a generalization of the TT-decomposition with $r_d = m$. Once the TN-cores are obtained, one can simulate the system using the second row of (7.4) without referring back to its dual representation $\boldsymbol{\mathcal{V}}$, which suffers from the curse of dimensionality.

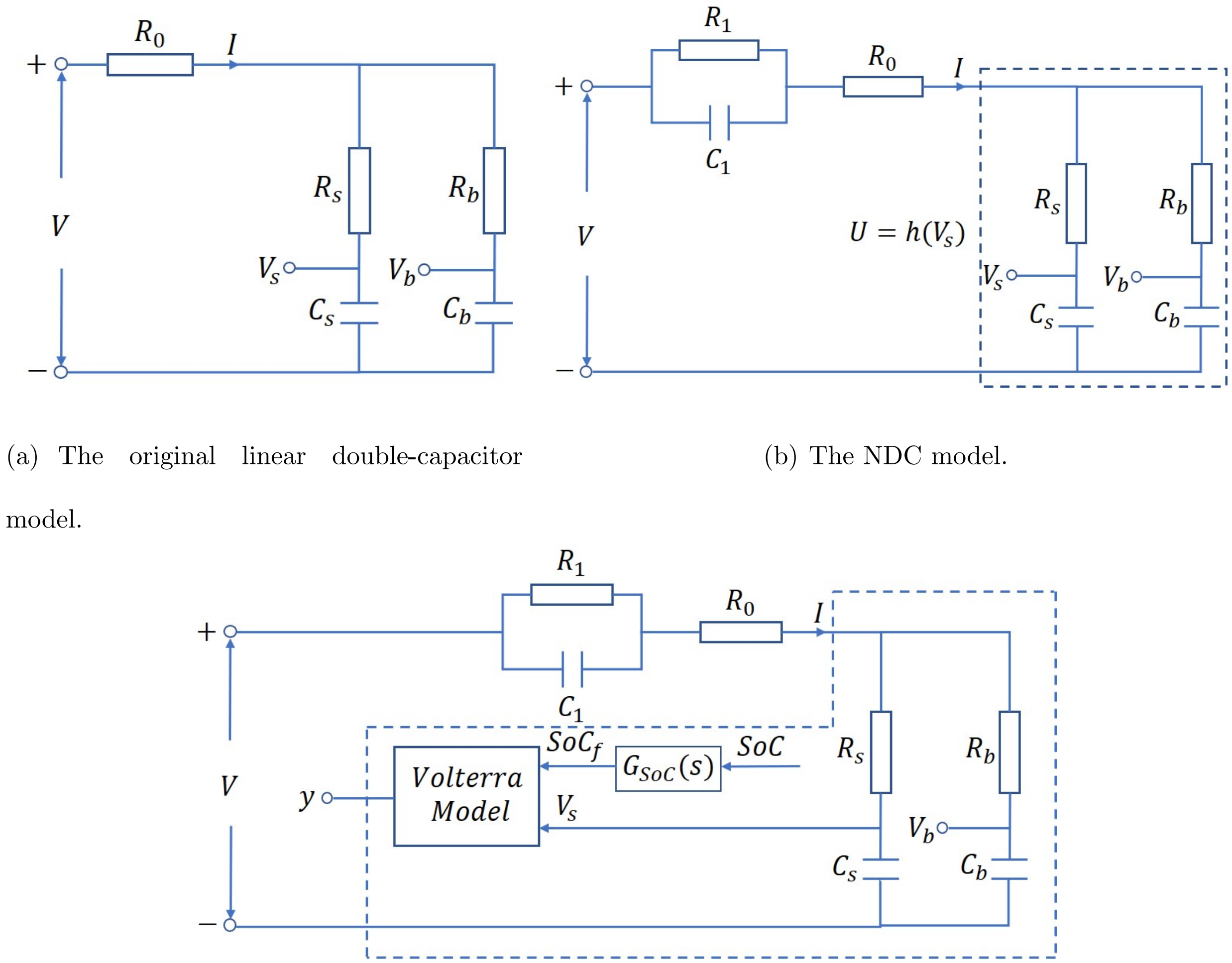
The number of stored elements in the TN representation $\{\boldsymbol{\mathcal{V}}_1, \cdots, \boldsymbol{\mathcal{V}}_d\}$ is at a magnitude of $O((d-1)r^2(pM+1)+mr(pM+1))$, which will greatly reduce the storage requirement if the maximal TN-rank $r = \max\{r_1, \cdots, r_d\}$ is sufficiently small. Fortunately, a low value for the maximal TN-rank r is very common in practical applications, as is the case with battery modeling. Thus, (7.4) computed with the TN representation $\{\boldsymbol{\mathcal{V}}_1, \cdots, \boldsymbol{\mathcal{V}}_d\}$ serves as a low-rank representation of (7.1). The corresponding computational complexity of using (7.4) is approximately $O(d(pM+1)r^2)$.

7.3 Battery Modeling and Parameter Estimation

7.3.1 Linear and Nonlinear Double-Capacitor Model

The original linear double-capacitor model for a battery was proposed in [118] and consists of two serial Resistor-Capacitor circuits in parallel (i.e., R_b-C_b and R_s-C_s) and a resistor R_0, as shown in Fig. 7.1(a). The resistor R_0 represents the collected instantaneous ohmic resistance. The R_s-C_s circuit corresponds to the electrode surface region exposed to the electrolyte, while the R_b-C_b to the bulk inner part of the electrode. The double-capacitor structure can capture both the rate capacity effect and the voltage recovery effect since it is modeling the charge migration between the near-surface and bulk inner domains of an electrode from the perspective of a single-particle model (SPM) [127, 128, 110].

However, the linear double-capacitor model is unable to describe a defining characteristic of batteries, for instance, the nonlinear SoC-OCV curve. Thus, the linear double-capacitor model is working well only around a certain SoC range so that the linear approximation is reasonable. In order to resolve this issue, the NDC model was proposed in [115], where a nonlinear mapping of V_s was introduced to approximate the SoC-OCV curve. A parallel RC circuit R_1-C_1 was also introduced there to account for the voltage transients related to the charge transfer on the electrode/electrolyte interface and the ion mass diffusion in the battery [129]. The NDC model is shown in Fig. 7.1(b). Two scenarios can be considered for the parameter estimation of the NDC model: constant-current charging/discharging; variable-current charging/discharging. Experiment validation illustrated that the NDC model can have higher accuracy than other different ECMs under comparison. Also, the basic NDC model without the R_1-C_1 part is almost comparable to the full



(a) The original linear double-capacitor model.

(b) The NDC model.

(c) The VDC model.

Figure 7.1: Illustration of different double-capacitor models. The components within the rectangle are treated as a whole. In (b), $h(\cdot)$ is a nonlinear function. In (c), $G_{\mathrm{SoC}}(s)$ is a linear filter and y is the voltage prediction of the part within the rectangle.

model, especially if the product R_1C_1 is small.

7.3.2 Hybrid Double-Capacitor Model Development

Despite the advantages of the NDC model, the validation results in [115] indicate a space for further improvements: the validation data show current-dependence of the parameters, which is not considered by the model; the NDC model is estimated independently for constant and variable current scenarios, and the model estimated for one scenario offers less predictive accuracy for the other; the SoC-OCV curve identification for the scenario of variable current matches the true one with relatively limited accuracy since the identifiability of this defining characteristic becomes less with many parameters to be estimated therein. As a final remark, the SoC-OCV curve is a static feature for batteries when they are fully at rest. Thus, the simple nonlinear mapping introduced in the NDC model may not fully account for the nonlinear transient dynamics. Also, when the battery stops discharging, the true terminal voltage transient takes a long time to reach the full rest state whereas the predicted one by the NDC model arrives at the steady state almost instantaneously. This phenomenon implies that additional dynamics can be modelled within the battery after the discharging is stopped and before it achieves the fully steady state.

To improve the above mentioned points of the NDC model, a hybrid double-capacitor model is proposed in this chapter as shown in Fig. 7.1(c). A linear double-capacitor is used to extract meaningful features of the batteries and a multi-input single-output (MISO) Volterra system is followed to synthesize all important features to predict the voltage. The proposed model is referred to as the VDC model in the following discussion.

The state-space equation of the linear part in Fig. 7.1(b) or Fig. 7.1(c), which corresponds to a modified version of the linear double-capacitor model appended with one R-C component, is given by

$$\begin{bmatrix} \dot{V}_b\left(t\right) \\ \dot{V}_s\left(t\right) \\ \dot{V}_1\left(t\right) \end{bmatrix} = A \begin{bmatrix} V_b\left(t\right) \\ V_s\left(t\right) \\ V_1\left(t\right) \end{bmatrix} + BI\left(t\right) \tag{7.5}$$

similar to [115], where

$$A = \begin{bmatrix} \frac{-1}{C_b(R_b+R_s)} & \frac{1}{C_b(R_b+R_s)} & 0 \\ \frac{1}{C_s(R_b+R_s)} & \frac{-1}{C_s(R_b+R_s)} & 0 \\ 0 & 0 & \frac{-1}{R_1C_1} \end{bmatrix}, B = \begin{bmatrix} \frac{R_s}{C_b(R_b+R_s)} \\ \frac{R_b}{C_s(R_b+R_s)} \\ \frac{1}{C_1} \end{bmatrix}$$

Note that $I > 0$ for charging, $I < 0$ for discharging, $V_b = V_s = 1\,\mathrm{V}$ for $\mathrm{SoC} = 1$, $V_b = V_s = 0\,\mathrm{V}$ for $\mathrm{SoC} = 0$, and SoC is computed as

$$\mathrm{SoC} = \frac{C_bV_b + C_sV_s}{C_b + C_s} \tag{7.6}$$

Additional dynamics, which is not captured by the NDC model, can be attributed to the slow charge diffusion process inside the battery. The battery becomes fully at rest when the the process reaches its equilibrium. The additional and slow dynamics is not obvious by using the cycling data with a high switching frequency. This phenomenon can be modeled as SoC dynamics formulated as a first-order model

$$G_{\mathrm{SoC}}(s) = \frac{1}{\alpha s + 1} \tag{7.7}$$

shown in Fig. 7.1(c). The static gain of the first-order model is normalized to 1, since the SoC as a whole within the battery remains the same whereas it takes some time for all the charges to diffuse toward a steady state. From the electrochemical perspective, it describes the dynamics of the lithium concentration at the surface of the particle around the average lithium concentration in the solid. The time constant α would change in practice and thus requires to be adjusted during the parameter estimation. Denote the output of $G_{\mathrm{SoC}}(s)$ in (7.7) as $\mathrm{SoC}_f(t)$ at the time instant t.

The battery impedance is typically a function of SoC [130]. However, one can break it down into a constant part R_0 due to the materials of current collectors and the inside impedance which is related to the SoC. The inside impedance can be attributed to the SoC status.

The V_s is the reflection of the surface charge and related to the major part of the terminal voltage. Different current profile will affect the dynamic changing of V_s and SoC_f. Thus, the current dependence of the model parameters mentioned in the experimental validation in [115] can be described by the nonlinear interaction between V_s and SoC_f.

Finally, the voltage V_s and SoC dynamics SoC_f are selected as the input features for the nonlinear mapping described by the Volterra system. The transient voltage V_1 due to charge transfer and the constant resistance R_0 are separated from the nonlinear mapping. The input vector $\mathbf{u}(t)$ of the Volterra system is given as

$$\mathbf{u}\left(t\right) = \left[\mathrm{SoC}_f\left(t\right), V_s\left(t\right)\right]^{\mathrm{T}} \tag{7.8}$$

One should be careful about the initial state when simulating $\mathrm{SoC}_f(t)$ using the filter $G_{\mathrm{SoC}}(s)$. For example, for a fully charged battery, the initial $\mathrm{SoC}(0) = 1$ and it should be set as the initial state of $G_{\mathrm{SoC}}(s)$. The other approach is to filter the signal $\mathrm{SoC}(t) - 1$ instead for a fully charged battery if the initial state of $G_{\mathrm{SoC}}(s)$ is set as zero.

Note that the VDC model consists of a linear double-capacitor model and a Volterra system. The linear part can be determined separately by parameter estimation, where R_0 along with other parameters within the linear model will be determined. Then, the Volterra system serves as a nonlinear correction to improve the voltage prediction.

The terminal voltage $V(t)$ consists of $V_1(t)$, $I(t)R_0$, and the output $\mathbf{y}(t)$ of the nonlinear mapping described by the Volterra system, i.e.,

$$\mathbf{y}(t) = V(t) - I(t)R_0 - V_1(t) \tag{7.9}$$

The Volterra model is capable of grasping the nonlinearity and complex coupling among the input features. TN-based representation makes it possible to capture nonlinearity of high degree while seeking the low rank to simplify the parameterization and prevent overfitting. Moreover, a Volterra system is always bounded-input bounded-output stable, allowing a reliable learning of parameters in battery applications.

7.3.3 The Bond Core Sweeping Algorithm

Given the input features $\mathbf{u}(t)$ and the output prediction $\mathbf{y}(t)$, the next is to estimate the TN-cores $\{\boldsymbol{\mathcal{V}}_1, \cdots, \boldsymbol{\mathcal{V}}_d\}$ in (4.2). Although a MISO Volterra system identification is required for our application, the estimation algorithm will be presented in terms of a MIMO one for general purpose. The idea of sweeping optimization proposed in [126] is used in this chapter. However, the inherent ill-condition feature of the Volterra structure in this chapter will result in a very slow convergence rate for the gradient method, which is used during sweeping in [126]. Thus, the local optimization during sweeping will be modified to handle the ill-condition problem.

Note that the dual representation $\boldsymbol{\mathcal{V}}$, which suffers from the curse of dimensionality, will not be referred to during the whole estimation process. Instead, the TN-cores are optimized directly, and the TN-ranks grow and shrink adaptively during training to concentrate resources on the most useful correlations within the data for learning.

The cost function to be minimized is

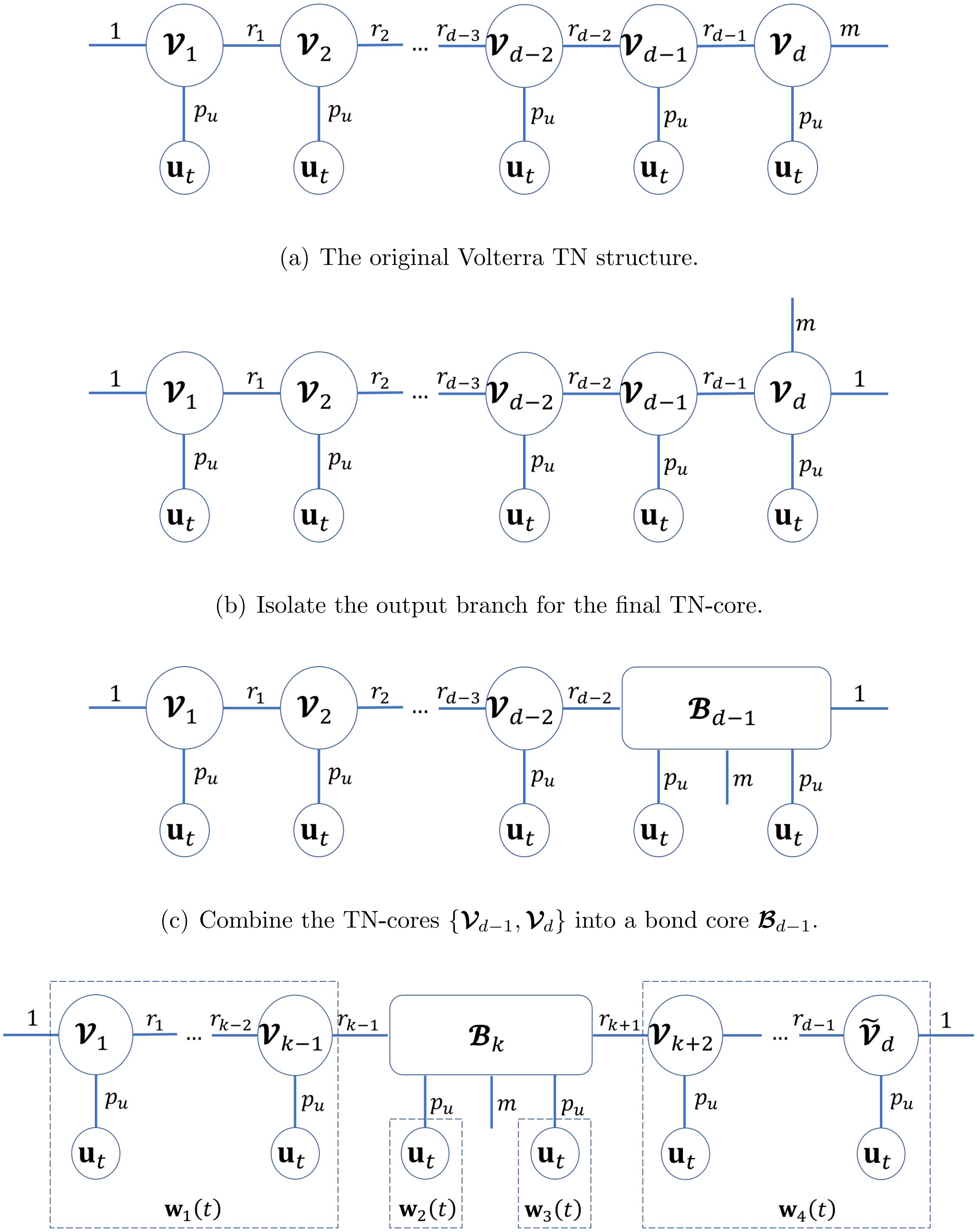
$$J = \frac{1}{2} \sum_{t=1}^{N} \sum_{l=1}^{m} \left(f^{(l)} \left(\mathbf{u} \left(t \right) \right) - y^{(l)} \left(t \right) \right)^2 \tag{7.10}$$

where $f^{(l)}(\mathbf{u}(t)) := \hat{y}^{(l)}(t)$ is the lth prediction output $\hat{\mathbf{y}}(t)$ of the proposed Volterra model in (4.2). The outline of the bond core sweeping algorithm is illustrated in Fig. 7.2. In order to adaptively estimate the internal TN-ranks, two neighboring TN-cores are merged into a bond core during each local optimization whereas the other TN-cores are fixed. The local optimization will sweep left and right to iteratively minimize the cost function. The output branch will be moving along with the bond core so that the parameters can be adjusted regarding different outputs during each local optimization.

The original representation of TN-cores are shown in Fig. 7.2(a). The sweeping algorithm starts from the rightmost. Thus, the rightmost TN-core is firstly modified by adding one more dimension and isolating the output branch as shown in Fig. 7.2(b). Then, the rightmost two TN-cores $\{\boldsymbol{\mathcal{V}}_{d-1}, \boldsymbol{\mathcal{V}}_{d}\}$ are merged to formulate a bond core $\boldsymbol{\mathcal{B}}_{d-1}$ as shown in Fig. 7.2(c). The bond core parameters are estimated by minimizing the cost function (7.10) while fixing the other TN-cores. The estimated bond core should be split up back into two updated TN-cores. Similar steps are then implemented for the next two TN-cores $\{\boldsymbol{\mathcal{V}}_{d-2}, \boldsymbol{\mathcal{V}}_{d-1}\}$. When the bond core sweeping reaches the leftmost, the sweeping continues by changing to a right direction. The sweeping procedures go left and right for several iterations of local optimizations until the estimation error is below a given threshold. In order to maintain the original TN structure shown in Fig. 7.2(a), the sweeping algorithm should also stop at the rightmost finally.

For simplicity of notations, denote $p_u = pM + 1$. In terms of an intermediate iteration shown in Fig. 7.2(d) where the TN-cores $\{\boldsymbol{\mathcal{V}}_{k}, \boldsymbol{\mathcal{V}}_{k+1}\}$ form the bond tensor $\boldsymbol{\mathcal{B}}_{k} \in \mathbb{R}^{r_{k-1} \times p_u \times m \times p_u \times r_{k+1}}$, the output prediction $\hat{\mathbf{y}}(t)$ can be computed as

$$\hat{\mathbf{y}}^{\mathrm{T}}(t) = \boldsymbol{\mathcal{B}}_{k} \times_1 \mathbf{w}_1(t) \times_2 \mathbf{w}_2^{\mathrm{T}}(t) \times_4 \mathbf{w}_3^{\mathrm{T}}(t) \times_5 \mathbf{w}_4^{\mathrm{T}}(t) \tag{7.11}$$



(a) The original Volterra TN structure.

(b) Isolate the output branch for the final TN-core.

(c) Combine the TN-cores $\{\mathcal{V}_{d-1}, \mathcal{V}_d\}$ into a bond core $\mathcal{B}_{d-1}$.

(d) Four fixed parts $\{\mathbf{w}_1(t), \mathbf{w}_2(t), \mathbf{w}_3(t), \mathbf{w}_4(t)\}$ during the local optimization to estimate an intermediate bond core $\mathcal{B}_k$. Note that the output branch is shifting along with $\mathcal{B}_k$.

Figure 7.2: Illustration of estimating the bond cores.

where

$$\begin{aligned}
\mathbf{w}_1(t) &= \left(\boldsymbol{\mathcal{V}}_1 \times_2 \mathbf{u}_t^{\mathrm{T}}\right) \cdots \left(\boldsymbol{\mathcal{V}}_{k-1} \times_2 \mathbf{u}_t^{\mathrm{T}}\right) \\
\mathbf{w}_2(t) &= \mathbf{w}_3(t) = \mathbf{u}_t \\
\mathbf{w}_4(t) &= \left(\boldsymbol{\mathcal{V}}_{k+2} \times_2 \mathbf{u}_t^{\mathrm{T}}\right) \cdots \left(\widetilde{\boldsymbol{\mathcal{V}}}_d \times_2 \mathbf{u}_t^{\mathrm{T}}\right)
\end{aligned} \tag{7.12}$$

and $\widetilde{\boldsymbol{\mathcal{V}}}_d \in \mathbb{R}^{r_{d-1} \times p_u \times 1}$ is a modified final TN-core with the output branch on $\boldsymbol{\mathcal{V}}_d \in \mathbb{R}^{r_{d-1} \times p_u \times m}$ shifted to the bond core $\boldsymbol{\mathcal{B}}_k$.

The local optimization is to fix the other TN-cores $\{\boldsymbol{\mathcal{V}}_1, \cdots, \boldsymbol{\mathcal{V}}_{k-1}, \boldsymbol{\mathcal{V}}_{k+2}, \cdots, \boldsymbol{\mathcal{V}}_d\}$ and estimate the bond core $\boldsymbol{\mathcal{B}}_k$ from (7.11). Due to the ill-condition feature of the Volterra model, the gradient method as in [126] is not used here. Instead, one can solve the local optimization via least squares. Similar ideas were also used in [64]. Denote $\boldsymbol{\mathcal{B}}_k^{(l)} \in \mathbb{R}^{r_{k-1} \times p_u \times 1 \times p_u \times r_{k+1}}$ as the lth sub-tensor of $\boldsymbol{\mathcal{B}}_k$ corresponding to the lth output, where $l = 1, 2, \cdots, m$. Also, denote the pseudo-inverse operator as $\mathrm{pinv}(\cdot)$. Then, $\boldsymbol{\mathcal{B}}_k$ can be estimated as

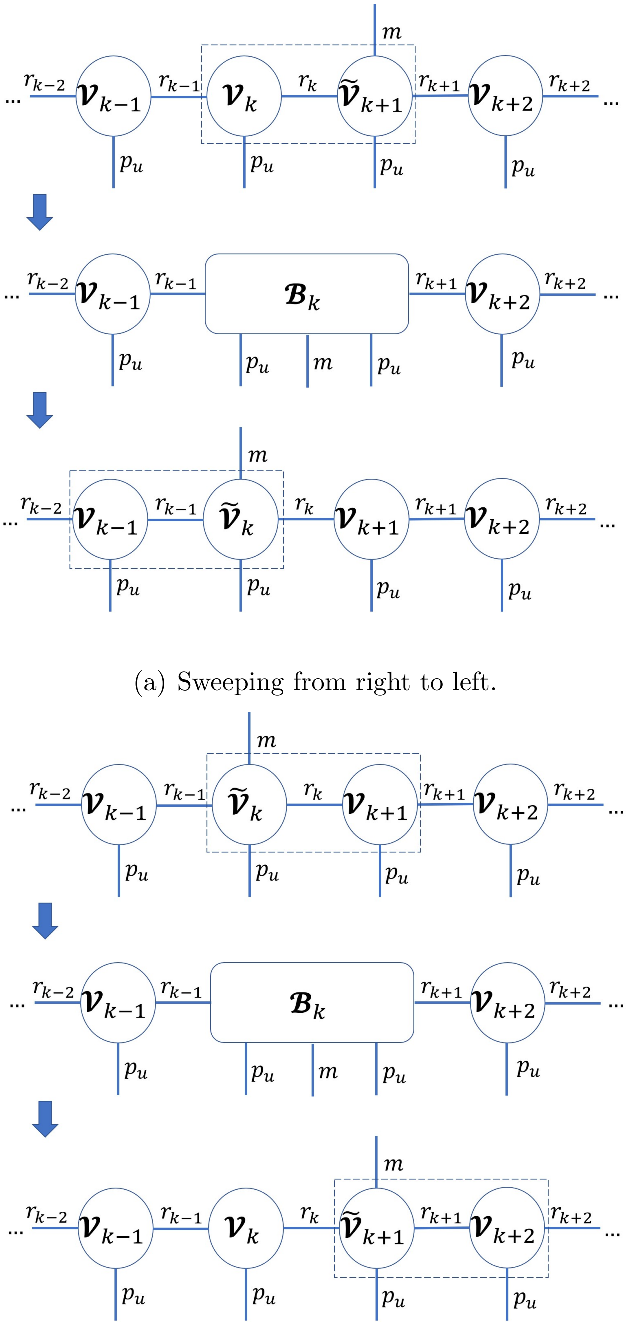
$$\boldsymbol{\mathcal{B}}_k(:, :, l, :, :) = \boldsymbol{\mathcal{B}}_k^{(l)} \tag{7.13}$$

where

$$
\begin{aligned}
&\boldsymbol{\mathcal{B}}_k^{(l)} = \operatorname{reshape}(\operatorname{vec}(\boldsymbol{\mathcal{B}}_k^{(l)}), [r_{k-1}, p_u, 1, p_u, r_{k+1}]) \\
&\operatorname{vec}(\boldsymbol{\mathcal{B}}_k^{(l)}) = \operatorname{pinv}\left(\mathbf{W}\right)\mathbf{Y}_l \\
&\mathbf{W} = \begin{bmatrix} \mathbf{w}_4^{\mathrm{T}}\left(1\right) \otimes \mathbf{w}_3^{\mathrm{T}}\left(1\right) \otimes \mathbf{w}_2^{\mathrm{T}}\left(1\right) \otimes \mathbf{w}_1\left(1\right) \\ \mathbf{w}_4^{\mathrm{T}}\left(2\right) \otimes \mathbf{w}_3^{\mathrm{T}}\left(2\right) \otimes \mathbf{w}_2^{\mathrm{T}}\left(2\right) \otimes \mathbf{w}_1\left(2\right) \\ \vdots \\ \mathbf{w}_4^{\mathrm{T}}\left(N\right) \otimes \mathbf{w}_3^{\mathrm{T}}\left(N\right) \otimes \mathbf{w}_2^{\mathrm{T}}\left(N\right) \otimes \mathbf{w}_1\left(N\right) \end{bmatrix} \\
&\mathbf{Y}_l = \begin{bmatrix} y^{(l)}\left(1\right) \\ y^{(l)}\left(2\right) \\ \vdots \\ y^{(l)}\left(N\right) \end{bmatrix}
\end{aligned}
\tag{7.14}
$$

and N is the number of available training data points. Note that $\mathbf{W}$ only needs to be computed for once during each iteration.

Once the bond core $\boldsymbol{\mathcal{B}}_k$ has been updated, one can split it up into two new TN-cores and move the output branch to the next TN-core. The singular value decomposition (SVD) is used to collapse the bond core while the TN-rank r_k is updated by observing the number of dominant singular values. In order to determine the dominant singular values, an ε-truncated SVD [68], presented in Algorithm 6 following a MATLAB fashion, is performed with the sum of squares of the truncated singular values not being greater than a percentage of ε^2 of the total one. The next step is then to update the bond core $\boldsymbol{\mathcal{B}}_{k-1}$ if sweeping left or $\boldsymbol{\mathcal{B}}_{k+1}$ if sweeping right. The implementation of splitting up the bond core and shifting the output branch for both left and right sweeping is illustrated in Fig. 7.3, and the complete



(a) Sweeping from right to left.

(b) Sweeping from left to right.

Figure 7.3: Illustration of shifting the output branch during bond core sweeping. The rectangular box drawn with dashed lines is showing the pair of TN-cores to be merged into a bond core.

algorithm in a MATLAB fashion is summarized in Algorithm 7. In the figure, a tilde sign will be added over the notation of the TN-core whose dimension is different from its original one. Since the algorithm is set to stop sweeping at the rightmost TN-core, $\boldsymbol{\mathcal{V}}_d$ should be permuted back to the original three-way tensor with TN-rank $r_d = m$ in the final step.

```
Algorithm 6: ε-Truncated SVD
  Input: Matrix $\mathbf{B}$, expected amount $\varepsilon$ in percentage to be truncated
  Output: Estimated rank $r$; truncated $\{\mathbf{L}_1, \mathbf{S}_1, \mathbf{Z}_1\}$ via SVD
  $[\mathbf{L}, \mathbf{S}, \mathbf{Z}] \leftarrow$ SVD($\mathbf{B}$,'econ')
  $r \leftarrow$ numerical rank determined by removing at most $\varepsilon$ portion of insignificant
    singular values in $\mathbf{S}$ such that their sum of squares is not greater than $\varepsilon^2\|\mathbf{S}\|_F^2$
  $\mathbf{L}_1 \leftarrow \mathbf{L}(:, 1 : r)$
  $\mathbf{S}_1 \leftarrow \mathbf{S}(1 : r, 1 : r)$
  $\mathbf{Z}_1 \leftarrow \mathbf{Z}(:, 1 : r)$
```

7.3.4 Acquisition of Training Data

We have proposed a VDC battery model to capture the complex and coupled dynamics and the bond core sweeping algorithm to estimate its parameters. In addition, the experiments to acquire the training data should be carefully designed to excite the major dynamics within the battery. An accurate prediction of the static SoC-OCV characteristic is also desired for the proposed model since the batteries are usually in a storage state for most of its life time in practice. Thus, we would like to train a model with both good dynamic and static predictions.

In order to achieve an accurate dynamic prediction, the model should be fed with data containing rich modes. This requirement is also known as persistent excitation in the field of system identification [2, 69]. A white excitation input signal sweeping the whole working range is desired but not practical in many applications. Instead, a relatively small data set with several different variable-current profiles are sufficient for the proposed VDC model. Denote this part of training data as Dynamic Training Data (DTD).

The proposed VDC model is a hybrid model with both physically meaningful modeling and complexity-adjustable nonlinear mapping. One advantage of the Volterra structure is that the nonlinear mapping is always stable, and thus there is no need to worry about a divergent prediction during validation or in real-time battery monitoring. The adaptive adjustment of the TN-ranks of the proposed model during training also reduces the overfitting problem since the low-rank feature within the data is explored. Thus, it is unnecessary to use a very large data set as in training pure black-box models such as deep neural networks.

In order to capture the static feature, the SoC-OCV information should also be fed into the VDC model. Thus, a characterization experiment should be performed to obtain the SoC-OCV curve, denoted as OCV $= h(\text{SoC})$. Let $\{SoC_i, OCV_i\}, i = 1, \cdots, N_d$ be the N_d sampled points of the SoC-OCV curve. Then, the corresponding training data for the static feature can be constructed such that, for each pair $\{SoC_i, OCV_i\}$, the input/output data $\{\mathbf{u}(t), \mathbf{y}(t)\}$ in (7.8) and (7.9) is formulated as

$$\begin{aligned} &V_s\left(t\right) = \text{SoC}_f\left(t\right) = SoC_i \\ &\mathbf{y}\left(t\right) = h\left(SoC_i\right) \end{aligned} \tag{7.15}$$

for a time interval with certain duration, i.e., $t \in [L_i, R_i]$. The duration depends on how much emphasis we would like to put on the static feature during training. Since there are N_d pairs of $\{SoC_i, OCV_i\}$, there will be N_d such individual subintervals for (7.15). Denote this part of training data as Static Training Data (STD).

The reason why (7.15) reflects the static relation between the SoC and OCV is that the proposed VDC model is capable of describing the state of the battery when it is fully at rest. The battery will be at equilibrium when it is fully at rest after stopping charging/discharging. Thus, the current $I(t)$ and the transient voltage $V_1(t)$ will become zero and the other inner states $V_s(t)$, SoC(t), and SoC$_f(t)$ will become the same. This inherent property makes the VDC model more powerful in simultaneously capturing both dynamic and static features of the battery.

In sum, two types of training data are required: DTD and STD. The first part focuses on the dynamics and the second part focuses on the static feature. The STD is constructed based on the SOC-OCV curve and its duration can be selected based on how much weight to put on the static feature.

Algorithm 7: Bond Core Sweeping Algorithm

```
Input: $N$ input/output sampled data $\{\mathbf{u}(t), \mathbf{y}(t)\}$ in (7.8) and (7.9), memory
       length $M$, degree $d$, expected accuracy $\varepsilon$ in percentage
Output: TN-cores $\{\boldsymbol{\mathcal{V}}_1, \cdots, \boldsymbol{\mathcal{V}}_d\}$ in (4.2) minimizing (7.10)
Initialization: Construct $\mathbf{u}_t$ in (4.2); $r_0 \leftarrow 1, r_d \leftarrow m$; initialize left orthogonal
  TN-cores $\{\boldsymbol{\mathcal{V}}_1, \cdots, \boldsymbol{\mathcal{V}}_d\}$ of ranks 1, i.e., $r_i \leftarrow 1, i = 1, 2, \cdots, d-1$; the starting
  index $k \leftarrow d-1$; sweeping direction R2L ← 'left'
$\boldsymbol{\mathcal{V}}_d \leftarrow$ reshape($\boldsymbol{\mathcal{V}}_d, [r_{d-1}, p_u, m, 1]$)
$r_d \leftarrow 1$
while the termination criterion is not satisfied do
    Compute $\{\mathbf{w}_1(t), \mathbf{w}_2(t), \mathbf{w}_3(t), \mathbf{w}_4(t)\}$ in (7.12)
    for $l = 1, \cdots, m$ do
        $\boldsymbol{\mathcal{B}}_k(:, :, l, :, :) \leftarrow \boldsymbol{\mathcal{B}}_k^{(l)}$ as in (7.13)
    if R2L is 'left' then
        $\mathbf{B}_k \leftarrow$ reshape($\boldsymbol{\mathcal{B}}_k, [r_{k-1}p_u m, p_u r_{k+1}]$)
        $\{r, \mathbf{L}_1, \mathbf{S}_1, \mathbf{Z}_1\} \leftarrow \varepsilon$-truncated SVD on $\mathbf{B}_k$
        $r_k \leftarrow r$
        Split $\boldsymbol{\mathcal{B}}_k$ while keeping left orthogonal:
        $\boldsymbol{\mathcal{V}}_k \leftarrow$ reshape($\mathbf{L}_1\mathbf{S}_1, [r_{k-1}, p_u, m, r_k]$)
        $\boldsymbol{\mathcal{V}}_{k+1} \leftarrow$ reshape($\mathbf{Z}_1^{\mathrm{T}}, [r_k, p_u, r_{k+1}]$)
        if $k > 1$ then
            $k \leftarrow k - 1$
        else
            R2L←'right'
    else
        $\mathbf{B}_k \leftarrow$ reshape($\boldsymbol{\mathcal{B}}_k, [r_{k-1}p_u, m p_u r_{k+1}]$)
        $\{r, \mathbf{L}_1, \mathbf{S}_1, \mathbf{Z}_1\} \leftarrow \varepsilon$-truncated SVD on $\mathbf{B}_k$
        $r_k \leftarrow r$
        Split $\boldsymbol{\mathcal{B}}_k$ while keeping right orthogonal:
        $\boldsymbol{\mathcal{V}}_k \leftarrow$ reshape($\mathbf{L}_1, [r_{k-1}, p_u, r_k]$)
        $\boldsymbol{\mathcal{V}}_{k+1} \leftarrow$ reshape($\mathbf{S}_1\mathbf{Z}_1^{\mathrm{T}}, [r_k, m, p_u, r_{k+1}]$)
        if $k < d - 1$ then
            $k \leftarrow k + 1$
        else
            R2L←'left'
$\boldsymbol{\mathcal{V}}_d \leftarrow$ permute($\boldsymbol{\mathcal{V}}_d, [1, 3, 2, 4]$)
$r_d \leftarrow m$
```

7.4 Experimental Validation

This section presents the experimental validation of the proposed TN-based VDC model and the comparison with the other relevant methods. All the experiments in this section were conducted on a PEC SBT4050 battery tester. It can support charging/discharging with arbitrary current-, voltage-, or power- based loads up to 40 V and 50 A. A dedicated server prepares and configures a test offline and collects sampled experimental data online via the associated software, LifeTest. Charging/discharging tests were performed on a Panasonic NCR18650B lithium-ion battery cell, which was also used in [115], to generate experimental data. Note that the battery cell was set to operate between 3.2 V (fully discharged) and 4.2 V (fully charged). In this section, only the discharging case is considered.

Two perspectives of validation are performed in the rest of this section. The first one shows the arrangement of training and validation data, the parameter estimation of TN-cores via the bond core sweeping algorithm, and the prediction accuracy of the estimated VDC model. The second one shows the requirements about training data to ensure the success of the identification algorithm, serving as a note for practitioners.

7.4.1 Training and Validation Results

The proposed hybrid VDC model consists of a linear double-capacitor model and a Volterra model. The parameters of the linear double-capacitor can be directly identified or using the linear part extracted from the identified NDC model in [115]. Note that the linear double-capacitor model here serves as a feature extractor to produce useful features that are fed into the Volterra system. Thus, slight distortion of using a different identified linear model is not a problem and will be accounted for during training of the Volterra model. As presented in Section 7.3.4, two types of training data, i.e., DTD and STD, are required.

In order to persistently excite the VDC model, the dynamic modes in the training data should be sufficiently rich. Thus, variable-current discharging data should be included. Since the VDC model describes the nonlinear dynamics among the current $I(t)$, the voltage $V_s(t)$, and SoC dynamics $\mathrm{SoC}_f(t)$, the training data should sweep across the preset working range of current and SoC. Otherwise, the prediction of the nonlinear model outside the uncovered working range will show unexpected behaviors due to unmodeled dynamics. This phenomenon will be illustrated in Section 7.4.2.

In order to maintain the static SoC-OCV feature of the lithium-ion battery, additional training data described in (7.15) should also be included. This portion of training data will guarantee that the voltage prediction of the VDC model arrives at the corresponding value according to the SoC-OCV curve when the battery is fully at rest.

In order to better describe the low-frequency dynamics, at least one constant-current discharging data set should be included in the training data. The reason is that the variable-current discharging data focuses on the high frequency dynamics due to fast switching and thus contains little information about the modes lying in the low-frequency band.

Finally, the training data should include three parts: DTD1 (variable-current profile for high frequency dynamics), STD (SoC-OCV characteristics for static features), and DTD2 (constant-current profile for low frequency dynamics).

In our experiment, DTD1 consists of two variable-current profiles (0∼3 A and 0∼6 A) shown in Fig. 8 and Fig. 9 in [115]. Note that the second one sweeps mainly across about 2∼6A. Thus, the two profiles allow the nonlinear dynamic modeling for working range 0∼6 A.

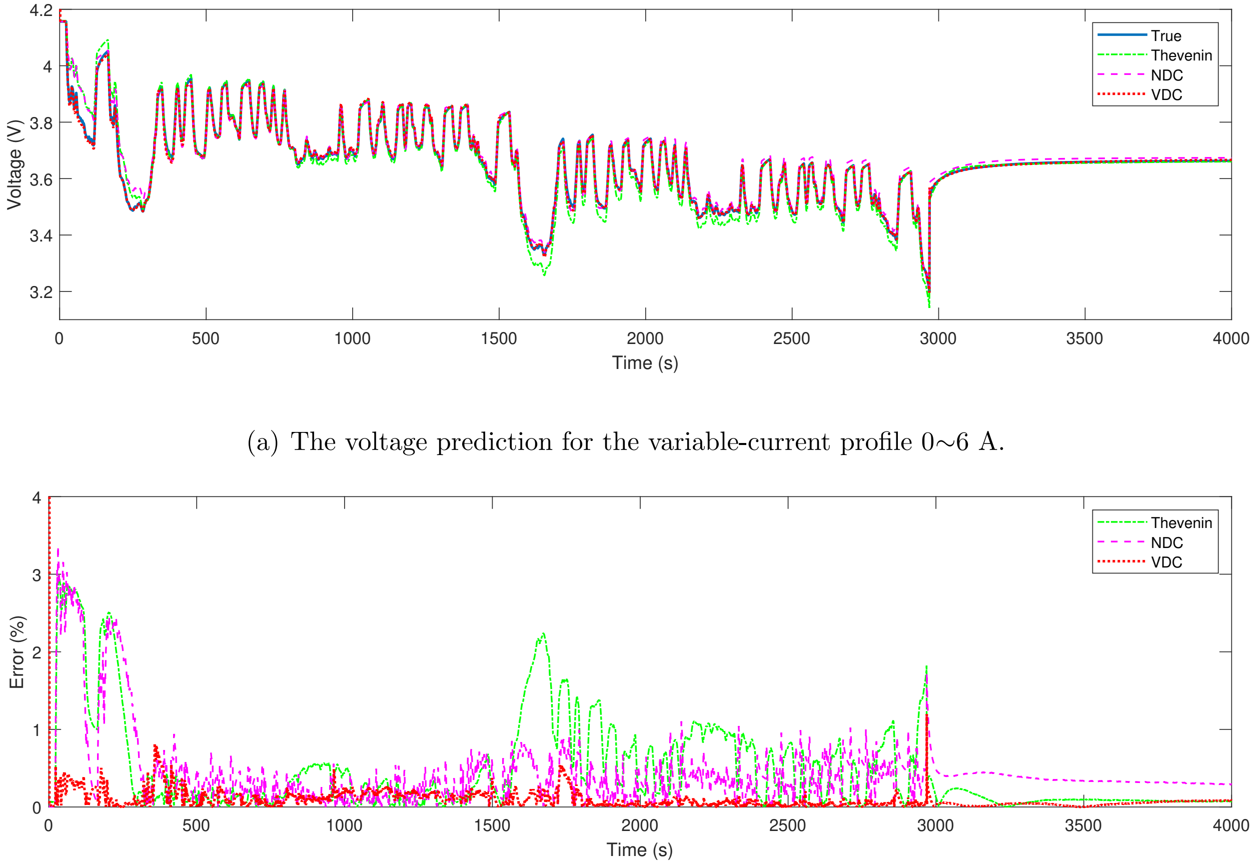
STD describing the static features is formulated as in (7.15) with $\{SoC_i, OCV_i\}, i = 1, \cdots, 21$ being the 21 equidistantly data points sampled every 5% SoC from the SoC-OCV curve and the duration $R_i - L_i = 500$ seconds. DTD2 consists of one constant-current profile (3.5 A).

The resulting physical parameters were estimated by the 2.0 identification approach in [115] and are given as follows: $C_b = 10,031\,\mathrm{F}$, $C_s = 979\,\mathrm{F}$, $R_b = 0.063\,\Omega$, $R_s = 0\,\Omega$, $R_1 = 0.003\,\Omega$, $C_1 = 2,449\,\mathrm{F}$, and $R_0 = 0.069\,\Omega$. These parameters determine the feature extractor described by the linear double-capacitor model in (7.5). Then, the TN-based Volterra model with input/output $\{\mathbf{u}(t), \mathbf{y}(t)\}$ defined in (7.8) and (7.9) was estimated using the bond core sweeping algorithm proposed in Section 7.3.3 with memory length $M = 3$, degree $d = 5$, filter time constant $\alpha = 1/0.003$, and $\varepsilon = 0.4$. Note that the truncation factor ε is adjusted to seek the low-rank estimation as much as possible while maintaining the prediction accuracy. This procedure will prevent the resulting model from being overfitted. In practice, one should increase ε if the available dynamic modes are getting richer with increasing model complexity. The estimated TN-ranks are illustrated in Table 7.1. The training algorithm takes about 23 seconds on an Intel Core i5-10210U CPU with 16 GB RAM. The training results are shown in Fig. 7.4, Fig. 7.5, Fig. 7.6, and Fig. 7.7.

Table 7.1: Estimated TN-ranks for $\{\boldsymbol{\mathcal{V}}_1, \cdots, \boldsymbol{\mathcal{V}}_5\}$.

TN-rank	$\widehat{r}_1$	$\widehat{r}_2$	$\widehat{r}_3$	$\widehat{r}_4$
value	4	7	6	7

For variable-current profiles, the VDC model produces the best voltage prediction both during cycling and resting. The NDC model uses a static nonlinear mapping estimated to balance the trade-off between the nonlinear dynamics and static SoC-OCV relationship. Despite the improvement in the dynamic prediction, the trade-off is obvious: the NDC model

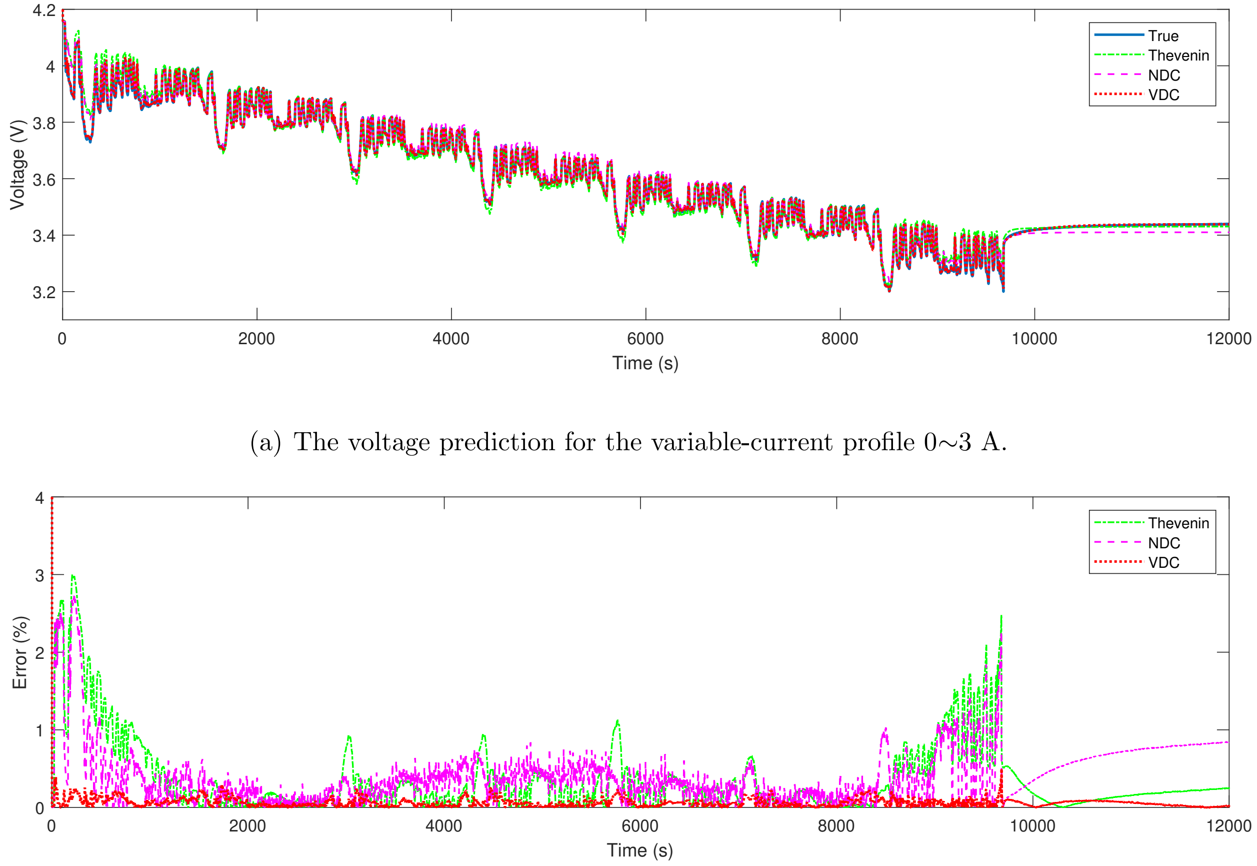


(a) The voltage prediction for the variable-current profile 0∼6 A.

(b) Fitting error in percentage.

Figure 7.4: Illustration of the voltage prediction by the proposed model for the variable-current profile 0∼6 A in the training data set DTD1.

cannot fit the dynamics and static feature simultaneously. Thus, the NDC model shows better prediction accuracy than the Thevenin model during discharging, whereas the voltage prediction of the NDC model is less accurate when at rest. Fig. 7.6 shows the prediction of SoC-OCV relationship using VDC model. It illustrates that the VDC model can be used to produce a more accurate dynamic voltage prediction without deteriorating the static prediction of the SoC-OCV feature. For the constant-current profile (3.5 A), one can observe that the Thevenin model is doing its best to minimize the error during discharging but is limited due to the lack of model fidelity. The VDC model is the best and can accurately



(a) The voltage prediction for the variable-current profile 0∼3 A.

(b) Fitting error in percentage.

Figure 7.5: Illustration of the voltage prediction of the proposed model for the variable-current profile 0∼3 A in the training data set DTD1.

capture the diffusion dynamics during the idling period.

The validation data consists of four constant-current profiles (1 A, 1.5 A, 2 A, 2.5 A) and one variable-current profile (1.5∼2.5 A). The prediction accuracy for the validation data is shown in Fig. 7.8 and Fig. 7.9.

For the constant-current profile in the validation data, the VDC model is showing the best voltage prediction for all four cases in the following aspects as shown in Fig. 7.8: the dynamic nonlinearities are being accurately predicted during discharging; the transient diffusion dynamics are being accurately tracked during the idling period; the errors of static

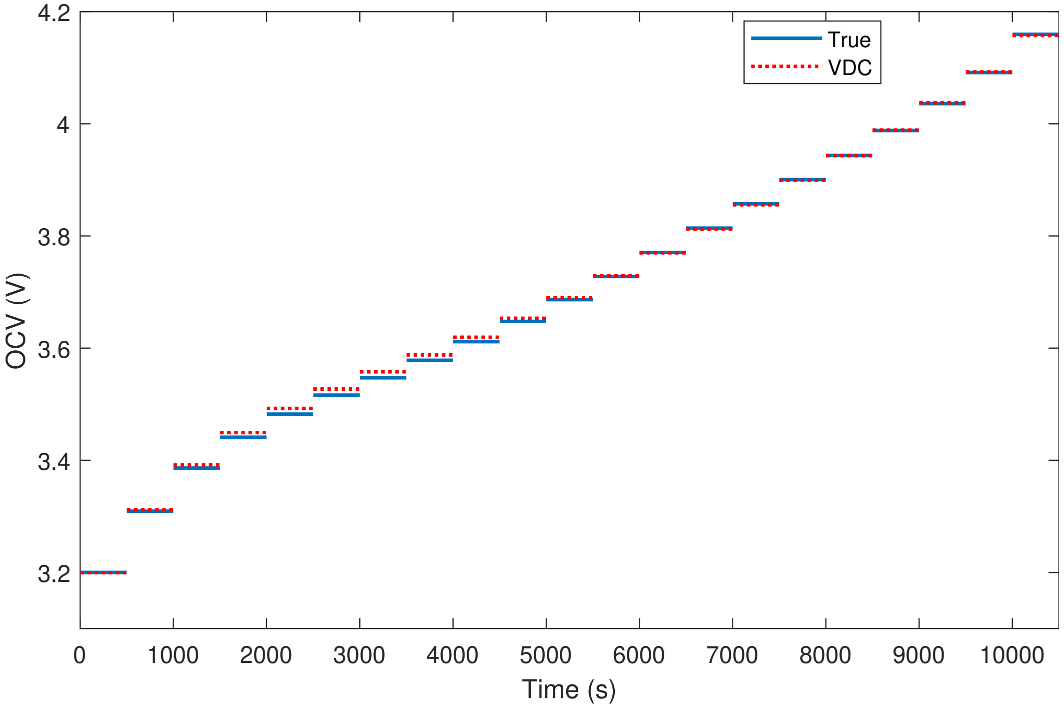


Figure 7.6: The model prediction of SoC-OCV relationship for the training data set STD.

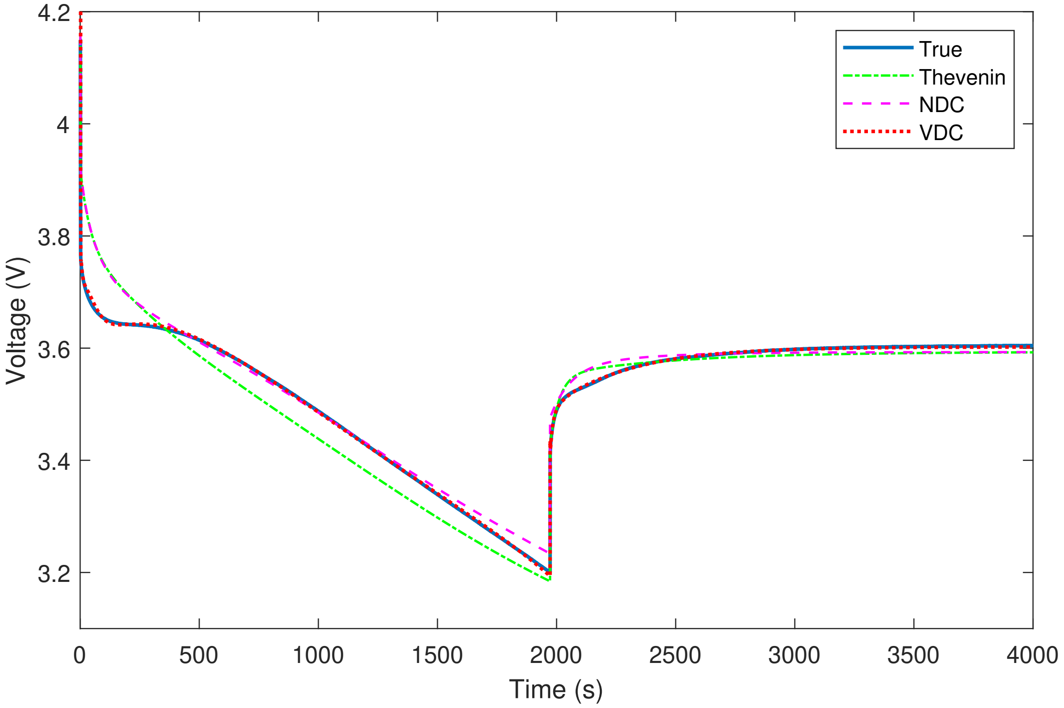


Figure 7.7: The voltage prediction for the constant-current profile (3.5 A) in the training data set DTD2.

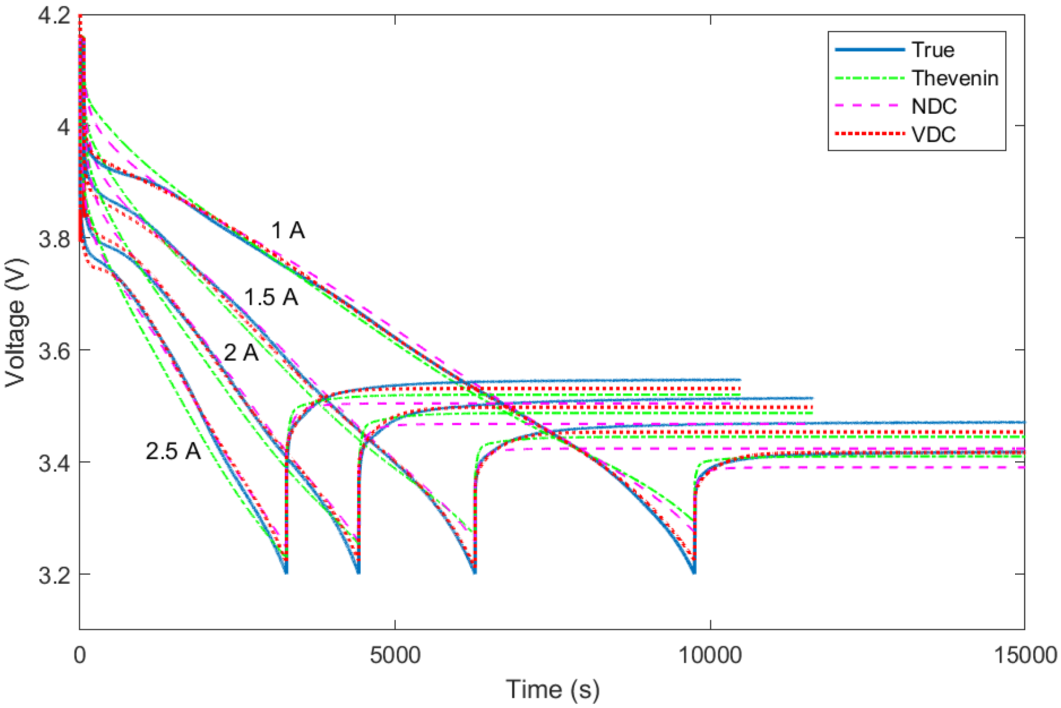
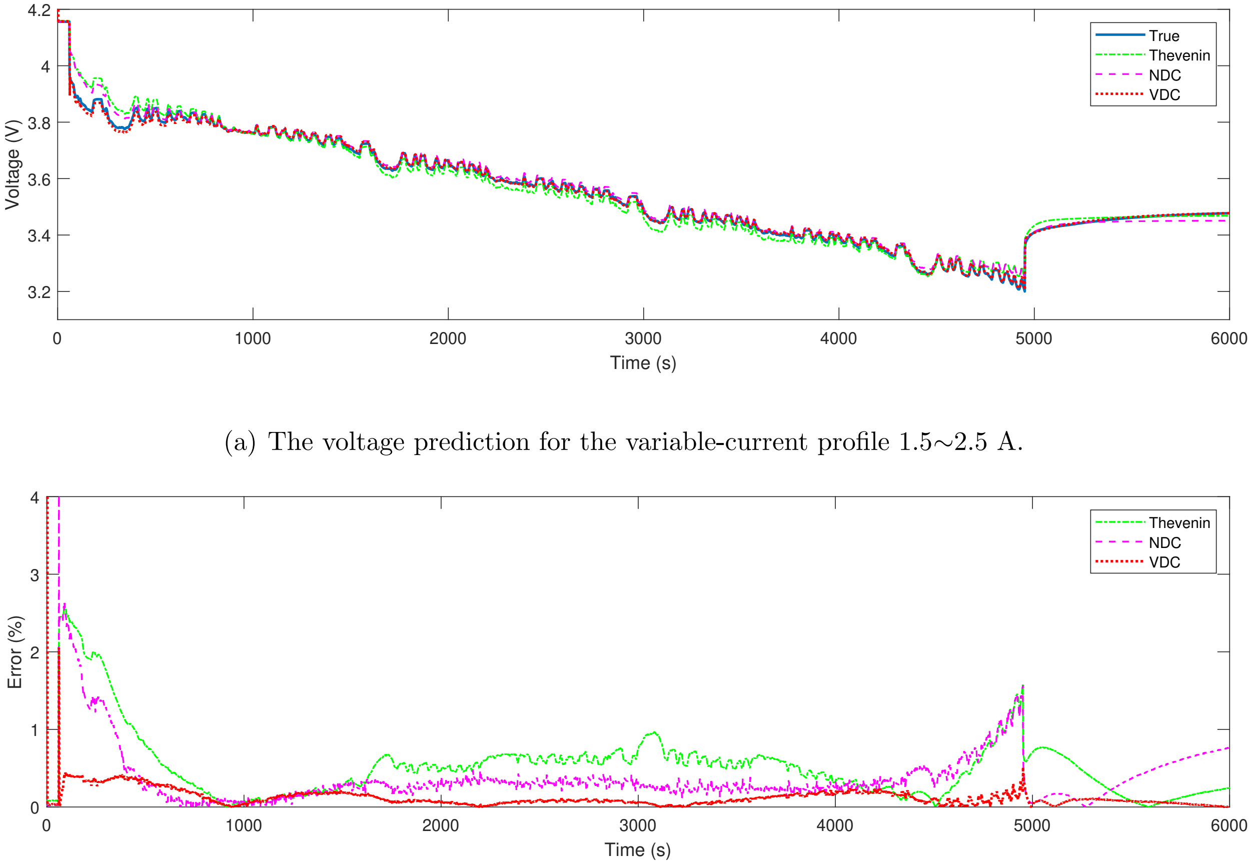


Figure 7.8: The voltage prediction for the constant-current profiles (1 A, 1.5A, 2 A, 2.5A) in the validation data set.

voltage predictions when at rest are small. As expected, the NDC model is showing the largest bias in the voltage prediction when at rest. The Thevenin model is struggling to predict the true dynamics. For the variable-current profile in the validation data, the VDC model performs the best during both the discharging period and idling period. Similar to the training result, the NDC model is better in dynamic prediction than the Thevenin model whereas it produces still relatively large bias during resting.

7.4.2 The Effect of Training Data Range

Section 7.4.1 has demonstrated the efficacy of the proposed method. Since nonlinear dynamics is modelled, it should be mentioned that the VDC model, due to its data-driven nature, will only capture the nonlinearity within the observed range of input current magni-



(a) The voltage prediction for the variable-current profile 1.5∼2.5 A.

(b) Fitting error in percentage.

Figure 7.9: Illustration of the voltage prediction of the proposed model for the variable-current profile 1.5∼2.5 A in the validation data set.

tudes and SoC values. Thus, it is necessary to select the training data sweeping across the normal working range. The nonlinearity is directly related to the input current magnitude and the battery dynamics keeps changing along with the SoC. For this reason, the current profiles should cover those common magnitudes in practice, and the training data should spread from full SoC to almost empty.

In order to demonstrate the effect of training data range, the investigation of parameter estimation was performed on the two variable-current profiles in DTD1. The 0∼3 A variable-current profile spreads from SoC = 100% to SoC = 14% (almost empty). The

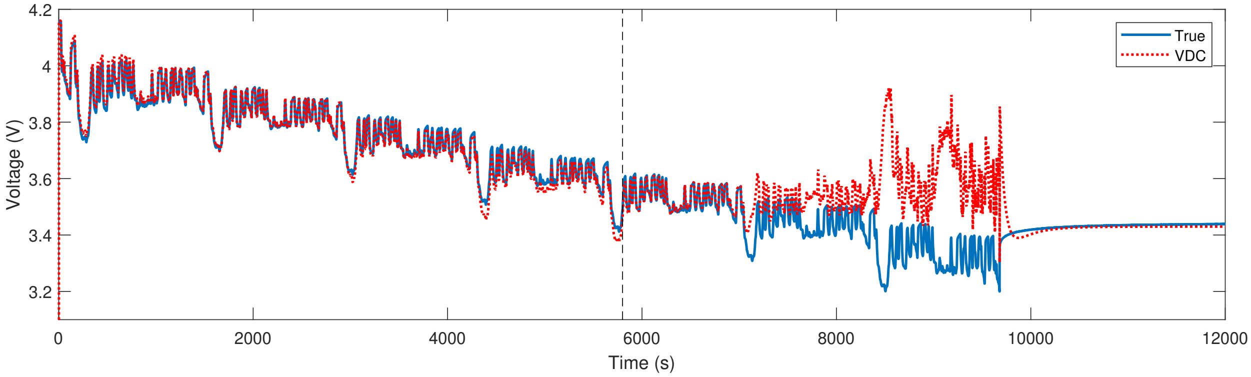


Figure 7.10: Illustration of the effect of training data range on the model fitting. The training data includes STD and a variable-current profile 0∼6 A with the SoC spreading from 100% to 47%.

0∼6 A variable-current profile spreads from SoC = 100% to SoC = 47%. The STD and the experimental data corresponding to 0∼6 A were used to train the TN Volterra model. The other one (0∼3 A) was used for validation. The results are shown in Fig. 7.10. The figure shows the voltage prediction on the validation data and the black vertical dashed line indicates the time instant where SoC = 47%.

For the stage when the SoC is above 47%, the voltage prediction is good. However, when the SoC is dropping below 47%, the voltage prediction is getting worse. The reason is that no training data is sweeping below SoC = 47%. Thus, the VDC model is not persistently excited in the SoC range below 47%. This observation indicates a common and important need when performing nonlinear modeling: the training data should cover the normal working range of the batteries.

7.5 Conclusion

This chapter proposes a novel TN-based VDC model. The VDC model consists of two parts: a linear double-capacitor model and a TN-based Volterra model to capture nonlinear dynamics. The proposed VDC model has the advantage to model both static and dynamic nonlinearities simultaneously in a more accurate way. It is by nature an SoC-dependent model, which will capture the changes in the battery dynamics as the SoC is varying. Parameters of the VDC model are estimated via the bond core sweeping algorithm. It allows the Volterra model to seek a low-rank representation during training which also reduces the overfitting problem. The experimental results show that the VDC model produces a much more accurate voltage prediction than the Thevenin model and NDC model. The prediction error of the proposed model can be less than 0.5% as shown in the experimental validation. It is illustrated that the TN-based VDC model can serve as a powerful tool in modeling unknown and complex nonlinearities within the batteries using a data-driven model.

This chapter is based on the following paper that was submitted:

Y. Hu, R.A. de Callafon, N. Tian and H. Fang, "Tensor network based MIMO Volterra model for lithium-ion batteries," *IEEE Transactions on Control Systems Technology*, under review.

Chapter 8

Conclusion and Future Work

8.1 Conclusion

This dissertation presents both theoretical and applied research on low-rank approximations for estimation of multivariable dynamics by using MIMO system identification techniques. Regarding MIMO linear systems, an in-depth study on the CoBRA, a branch of subspace methods, is conducted and the CoBRA is implemented in microgrid dynamic modeling. Regarding MIMO nonlinear systems, TN-based algorithms on MIMO Volterra systems are investigated and applied in battery modeling. Throughout the work in this dissertation, the SVD tool plays an important role in low-rank approximations, which result in low-order models or simplified parameterization, during estimation of multivariable dynamics.

The CoBRA has been demonstrated to be effective in obtaining low-order models in practice using input/output data heavily contaminated by noise with unknown spectrum characteristics in previous work. Noting the advantages of the CoBRA in handling a large amount of noise-contaminated data and focusing on low-order model estimation, we did several further research in the CoBRA. First, we incorporated the CoBRA into LTP system identification and accurately captured the periodic dynamics from data. This research expands the scope of application of the CoBRA. Second, we carried out a comprehensive statistical analysis on the CoBRA and illustrated how the noise effects were reduced by introducing covariance functions and the instrumental variable. An optimal implementation of the CoBRA by using weighting matrices based on these results was then developed therein. It was shown in a closed-loop setting that the optimal CoBRA shows higher accuracy in parameter estimation than other competitive subspace methods. Finally, we used the CoBRA to estimate a linear dynamic model in the application of microgrid power flow estimation and further demonstrated the efficacy of the CoBRA in practice.

A MIMO Volterra model is attractive in nonlinear system identification due to its capability to describe complex coupled nonlinear dynamics. In addition, it is always bounded-input bounded-output stable, which makes it a good model candidate for many real-world applications. Noting that the TN representation proposed in previous work enables us to exploit a more powerful Volterra model of high degree and long memory length, we did several studies in the TN-based algorithms and input excitation signal selection for MIMO Volterra system identification. First, we modified the existing iterative TN-based algorithms and proposed noniterative implementations. The noniterative TN-based algorithms are advantageous over the iterative ones such as the ALS and MALS which require an initialization and do not guarantee the convergence to an appropriate solution. In addition, the proposed noniterative algorithms actively remove less significant modes and emphasize low-rank approximations. Second, we derived a persistent excitation condition in the case of zero mean, Gaussian distributed (not necessarily white) input signals and illustrated it in a simulation example. Finally, we used the TN representation of a Volterra model to learn a nonlinear mapping in the application of modeling a lithium-ion battery and showed improved prediction accuracy compared with other competitive methods. This provides insights into applying TN-based MIMO Volterra model to other applications where nonlinear mappings may need to be estimated.

8.2 Recommendations for Future Work

This dissertation derives several useful results for system identification and demonstrates the efficacy by two interesting applications. However, there are still potential directions for future research.

First, it would be interesting to investigate the implementation of the CoBRA for a linear parameter-varying (LPV) system and a more general nonlinear model. This motivation is obvious since most systems in practice are nonlinear by nature although they can be approximated by a linear model around an equilibrium. In addition, it is worth studying the CoBRA in the frequency domain since the covariance data can be equivalently transformed into power spectrum.

Second, the TN representation can be incorporated in other different system identification problems, where either a large-scale data matrix or a large-scale parameter vector needs to be formulated. This attempt will further exploit the potential of certain model structures to describe complex behaviors in the data. This is promising especially nowadays since more and more data is available and all kinds of complicated phenomena are being modeled.

Finally, it is interesting to look at more control relevant applications such as autonomous driving, aircraft flight control, robotic control, etc, where a good model is typically required for an effective controller design. System identification will play an important role in the modeling process and facilitate the feedback controller design thereafter. Exploring specific applications will bring up important inherent modeling problems in practice and push forward the development of system identification.

Bibliography

[1] T. Söderström and P. Stoica, *System identification.* Prentice-Hall International, 1989.

[2] L. Ljung, *System Identification: Theory for the User*, 2nd ed. Upper Saddle River, NJ: Prentice-Hall, 1999.

[3] M. Verhaegen and V. Verdult, *Filtering and System Identification: A Least Squares Approach.* Cambridge university press, 2007.

[4] L. Ljung, "Perspectives on system identification," *Annual Reviews in Control*, vol. 34, no. 1, pp. 1–12, 2010.

[5] K. J. Åström and B. Torsten, "Numerical identification of linear dynamic systems from normal operating records," in *Proceedings of IFAC 2nd Symposium on the Theory of Self-Adaptive Control Systems*, vol. 2, no. 2. Elsevier, 1965, pp. 96–111.

[6] B. L. Ho and R. E. Kálmán, "Effective construction of linear state-variable models from input/output functions," *at-Automatisierungstechnik*, vol. 14, no. 1-12, pp. 545–548, 1966.

[7] H. Hotelling, "Analysis of a complex of statistical variables into principal components." *Journal of educational psychology*, vol. 24, no. 6, p. 417, 1933.

[8] B. Moore, "Principal component analysis in linear systems: Controllability, observability, and model reduction," *IEEE transactions on automatic control*, vol. 26, no. 1, pp. 17–32, 1981.

[9] S. Wold, K. Esbensen, and P. Geladi, "Principal component analysis," *Chemometrics and intelligent laboratory systems*, vol. 2, no. 1-3, pp. 37–52, 1987.

[10] P. Van Overschee and B. L. De Moor, *Subspace Identification for Linear Systems: Theory-—Implementation-—Applications.* Kluwer Academic Publishers, 1996.

[11] T. Katayama, *Subspace Methods for System Identification.* Springer Science & Business Media, 2005.

[12] G. Van der Veen, J.-W. van Wingerden, M. Bergamasco, M. Lovera, and M. Verhaegen, "Closed-loop subspace identification methods: an overview," *IET Control Theory & Applications*, vol. 7, no. 10, pp. 1339–1358, 2013.

[13] F. Giri and E.-W. Bai, *Block-oriented Nonlinear System Identification.* Springer-Verlag Berlin Heidelberg, 2010.

[14] J. Schoukens and L. Ljung, "Nonlinear system identification: A user-oriented road map," *IEEE Control Systems Magazine*, vol. 39, no. 6, pp. 28–99, 2019.

[15] G. Pillonetto, F. Dinuzzo, T. Chen, G. De Nicolao, and L. Ljung, "Kernel methods in system identification, machine learning and function estimation: A survey," *Automatica*, vol. 50, no. 3, pp. 657–682, 2014.

[16] A. Chiuso and G. Pillonetto, "System identification: A machine learning perspective," *Annual Review of Control, Robotics, and Autonomous Systems*, vol. 2, pp. 281–304, 2019.

[17] A. Hansson and L. Ljung, "Some topics in system identification," 2020.

[18] M. Niedzwiecki, *Identification of Time-Varying Processes.* Wiley New York, 2000.

[19] S. Bittanti and P. Colaneri, *Periodic Systems: Filtering and Control.* Springer Science & Business Media, 2009, vol. 5108985.

[20] M. S. Allen, "Frequency-domain identification of linear time-periodic systems using lti techniques," *Journal of Computational and Nonlinear Dynamics*, vol. 4, no. 4, 2009.

[21] M. S. Allen, M. W. Sracic, S. Chauhan, and M. H. Hansen, "Output-only modal analysis of linear time-periodic systems with application to wind turbine simulation data," *Mechanical Systems and Signal Processing*, vol. 25, no. 4, pp. 1174–1191, 2011.

[22] N. R. Chaudhuri, R. Oliveira, and A. Yazdani, "Stability analysis of vector-controlled modular multilevel converters in linear time-periodic framework," *IEEE Transactions on Power Electronics*, vol. 31, no. 7, pp. 5255–5269, 2015.

[23] D. Logan, T. Kiemel, and J. J. Jeka, "Using a system identification approach to investigate subtask control during human locomotion," *Frontiers in computational neuroscience*, vol. 10, p. 146, 2017.

[24] J. Kwon, X. Wang, F. Blaabjerg, and C. L. Bak, "Comparison of lti and ltp models for stability analysis of grid converters," in *2016 IEEE 17th Workshop on Control and Modeling for Power Electronics (COMPEL).* IEEE, 2016, pp. 1–8.

[25] S. Bittanti and P. Colaneri, "Invariant representations of discrete-time periodic systems," *Automatica*, vol. 36, no. 12, pp. 1777–1793, 2000.

[26] E. Louarroudi, J. Lataire, R. Pintelon, P. Janssens, and J. Swevers, "Frequency domain, parametric estimation of the evolution of the time-varying dynamics of periodically time-varying systems from noisy input–output observations," *Mechanical Systems and Signal Processing*, vol. 47, no. 1-2, pp. 151–174, 2014.

[27] 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.

[28] D. N. Miller and R. A. de Callafon, "Efficient identification of input dynamics for correlation function-based subspace identification," *IFAC Proceedings Volumes*, vol. 44, no. 1, pp. 6511–6516, 2011.

[29] R. A. de Callafon and D. N. Miller, "Cobra-software for a covariance based realization algorithm with convex constraints on pole locations," in *2015 IFAC 17th Symposium on System Identification (SYSID)*, vol. 48, no. 28. IFAC, 2015, pp. 763–768.

[30] D. N. Miller, "Realization-based system identification with applications," Ph.D. dissertation, UC San Diego, 2012.

[31] S. Bittanti, "Deterministic and stochastic linear periodic systems," in *Time series and linear systems*. Springer, 1986, pp. 141–182.

[32] D. N. Miller and R. A. De Callafon, "Subspace identification with eigenvalue constraints," *Automatica*, vol. 49, no. 8, pp. 2468–2473, 2013.

[33] 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.

[34] P. Van den Hof, "Closed-loop issues in system identification," *Annual reviews in control*, vol. 22, pp. 173–186, 1998.

[35] D. N. Miller, R. A. De Callafon, and M. J. Brenner, "Covariance-based realization algorithm for the identification of aeroelastic dynamics," *Journal of Guidance, Control, and Dynamics*, vol. 35, no. 4, pp. 1169–1177, 2012.

[36] W. E. Larimore, "System identification, reduced-order filtering and modeling via canonical variate analysis," in *1983 American Control Conference*. IEEE, 1983, pp. 445–451.

[37] K. Liu and R. E. Skelton, "Q-markov covariance equivalent realization and its application to flexible structure identification," *Journal of Guidance, Control, and Dynamics*, vol. 16, no. 2, pp. 308–319, 1993.

[38] K. Liu and D. W. Miller, "Time domain state space identification of structural systems," *Journal of Dynamic Systems, Measurement, and Control*, vol. 117, no. 4, pp. 608–618, 1995.

[39] P. Stoica and T. Soderstrom, "Optimal instrumental variable estimation and approximate implementations," *IEEE Transactions on Automatic Control*, vol. 28, no. 7, pp. 757–772, 1983.

[40] P. C. Young, "Refined instrumental variable estimation: maximum likelihood optimization of a unified box–jenkins model," *Automatica*, vol. 52, pp. 35–46, 2015.

[41] P. Stoica, M. Cedervall, and A. Eriksson, “Combined instrumental variable and subspace fitting approach to parameter estimation of noisy input-output systems,” *IEEE Transactions on Signal Processing*, vol. 43, no. 10, pp. 2386–2397, 1995.

[42] T. Gustafsson, “Subspace-based system identification: weighting and pre-filtering of instruments,” *Automatica*, vol. 38, no. 3, pp. 433–443, 2002.

[43] M. Viberg, B. Wahlberg, and B. Ottersten, “Analysis of state space system identification methods based on instrumental variables and subspace fitting,” *Automatica*, vol. 33, no. 9, pp. 1603–1616, 1997.

[44] M. Verhaegen, “Identification of the deterministic part of mimo state space models given in innovations form from input-output data,” *Automatica*, vol. 30, no. 1, pp. 61–74, 1994.

[45] T. Söderström and P. G. Stoica, “Instrumental variable methods for system identification,” *Lecture notes in control and information sciences*, vol. 57, 1983.

[46] Y. Hu and R. A. de Callafon, “Optimal weighting for covariance based realization algorithm,” in *2017 IEEE 56th Annual Conference on Decision and Control (CDC).* IEEE, 2017, pp. 5274–5279.

[47] N. J. Higham, “Computing a nearest symmetric positive semidefinite matrix,” *Linear algebra and its applications*, vol. 103, pp. 103–118, 1988.

[48] I. Houtzager, J.-W. van Wingerden, and M. Verhaegen, “Varmax-based closed-loop subspace model identification,” in *Proceedings of the 48h IEEE Conference on Decision and Control (CDC) held jointly with 2009 28th Chinese Control Conference.* IEEE, 2009, pp. 3370–3375.

[49] N. Wiener, “Nonlinear problems in random theory,” *Nonlinear Problems in Random Theory, by Norbert Wiener, pp. 142. ISBN 0-262-73012-X. Cambridge, Massachusetts, USA: The MIT Press, August 1966.(Paper)*, p. 142, 1966.

[50] M. Schetzen, *The Volterra and Wiener theories of nonlinear systems.* Wiley, 1980.

[51] ——, “Nonlinear system modeling based on the wiener theory,” *Proceedings of the IEEE*, vol. 69, no. 12, pp. 1557–1573, 1981.

[52] B. F. Beidas, “Intermodulation distortion in multicarrier satellite systems: analysis and turbo volterra equalization,” *IEEE Transactions on Communications*, vol. 59, no. 6, pp. 1580–1590, 2011.

[53] L. A. Azpicueta-Ruiz, M. Zeller, A. R. Figueiras-Vidal, J. Arenas-García, and W. Kellermann, “Adaptive combination of volterra kernels and its application to nonlinear acoustic echo cancellation,” *IEEE Transactions on Audio, Speech, and Language Processing*, vol. 19, no. 1, pp. 97–110, 2011.

[54] P. M. S. Burt and J. H. de Morais Goulart, "Efficient computation of bilinear approximations and volterra models of nonlinear systems," *IEEE Transactions on Signal Processing*, vol. 66, no. 3, pp. 804–816, 2018.

[55] Y. Kajikawa, "Subband parallel cascade volterra filter for linearization of loudspeaker systems," in *2008 16th European Signal Processing Conference.* Lausanne, Switzerland: IEEE, Aug. 2008, pp. 1–5.

[56] E. Mumolo and D. Francescato, "Adaptive predictive coding of speech by means of volterra predictors," in *IEEE Winter Workshop on Nonlinear Digital Signal Processing.* IEEE, 1993, p. 2.1.

[57] C.-H. Cheng and E. J. Powers, "Optimal volterra kernel estimation algorithms for a nonlinear communication system for psk and qam inputs," *IEEE Transactions on Signal processing*, vol. 49, no. 1, pp. 147–163, 2001.

[58] P. Wambacq and W. M. Sansen, *Distortion Analysis of Analog Integrated Circuits.* New York: Springer Science & Business Media, 1998.

[59] M. Kortmann, K. Janiszowski, and H. Unbehauen, "Application and comparison of different identification schemes under industrial conditions," *International Journal of Control*, vol. 48, no. 6, pp. 2275–2296, 1988.

[60] F. J. Doyle III, R. K. Pearson, and B. Ogunnaike, *Identification and Control Using Volterra Models.* Godalming, United Kingdom: Springer-Verlag London, 2002.

[61] L. Tan and J. Jiang, "Adaptive volterra filters for active control of nonlinear noise processes," *IEEE Transactions on signal processing*, vol. 49, no. 8, pp. 1667–1676, 2001.

[62] J.-N. Lin and R. Unbehauen, "2-d adaptive volterra filter for 2-d nonlinear channel equalisation and image restoration," *Electronics Letters*, vol. 28, no. 2, pp. 180–182, 1992.

[63] M. J. Korenberg and I. W. Hunter, "The identification of nonlinear biological systems: Volterra kernel approaches," *Annals of biomedical engineering*, vol. 24, no. 2, pp. 250–268, 1996.

[64] K. Batselier, Z. Chen, and N. Wong, "Tensor network alternating linear scheme for mimo volterra system identification," *Automatica*, vol. 84, pp. 26–35, 2017.

[65] K. Batselier, C.-Y. Ko, and N. Wong, "Tensor network subspace identification of polynomial state space models," *Automatica*, vol. 95, pp. 187–196, 2018.

[66] C. A. R. Fernandes, J. C. M. Mota, G. Favier *et al.*, "Mimo volterra modeling for nonlinear communication channels," *Learning and Nonlinear Models*, vol. 2, no. 8, pp. 71–92, 2010.

[67] K. Batselier, Z. Chen, and N. Wong, "A tensor network kalman filter with an application in recursive mimo volterra system identification," *Automatica*, vol. 84, pp. 17–25, 2017.

[68] I. V. Oseledets, "Tensor-train decomposition," *SIAM Journal on Scientific Computing*, vol. 33, no. 5, pp. 2295–2317, 2011.

[69] Y. Hu, L. Tan, and R. A. de Callafon, "Persistent excitation condition for mimo volterra system identification with gaussian distributed input signals," in *Proceedings of 2019 IEEE 58th Annual Conference on Decision and Control (CDC).* IEEE, 2019, pp. 1752–1757.

[70] J. Friedman, T. Hastie, H. Höfling, R. Tibshirani *et al.*, "Pathwise coordinate optimization," *Annals of applied statistics*, vol. 1, no. 2, pp. 302–332, 2007.

[71] T. Katayama, *Subspace methods for system identification.* Springer Science & Business Media, 2006.

[72] Y. Hu, Y. Jiang, and R. A. de Callafon, "Variance reduction in covariance based realization algorithm with application to closed-loop data," *Automatica*, vol. 113, p. 108683, 2020.

[73] N. Lee and A. Cichocki, "Fundamental tensor operations for large-scale data analysis using tensor network formats," *Multidimensional Systems and Signal Processing*, vol. 29, no. 3, pp. 921–960, 2018.

[74] E.-W. Bai, "Decoupling the linear and nonlinear parts in hammerstein model identification," *Automatica*, vol. 40, no. 4, pp. 671–676, 2004.

[75] I. Leontaritis and S. Billings, "Experimental design and identifiability for non-linear systems," *International Journal of Systems Science*, vol. 18, no. 1, pp. 189–202, 1987.

[76] R. Nowak and B. Van Veen, "Random and pseudorandom excitation sequences for nonlinear system identification," in *IEEE Winter Workshop on Nonlinear Digital Signal Processing.* IEEE, 1993, p. 2.1.

[77] G.-O. A. Glentis, P. Koukoulas, and N. Kalouptsidis, "Efficient algorithms for volterra system identification," *IEEE Transactions on Signal Processing*, vol. 47, no. 11, pp. 3042–3057, 1999.

[78] G. Peccati and M. S. Taqqu, "Moments, cumulants and diagram formulae for nonlinear functionals of random measures," *arXiv preprint arXiv:0811.1726*, 2008.

[79] M. Rosenblatt, "Cumulants and cumulant spectra," *Handbook of Statistics*, vol. 3, pp. 369–382, 1983.

[80] R. H. Lasseter, "Microgrids and distributed generation," *Journal of Energy Engineering*, vol. 133, no. 3, pp. 144–149, 2007.

[81] L. G. Weber, A. Nasiri, and H. Akbari, “Dynamic modeling and control of a synchronous generator in an ac microgrid environment,” *IEEE Transactions on Industry Applications*, vol. 54, no. 5, pp. 4833–4841, 2018.

[82] A. Khodadadi, M. Mohammadian, and S. H. Hosseinian, “Robust power sharing control of an autonomous microgrid featuring secondary controller,” in *2016 Smart Grids Conference (SGC)*. IEEE, 2016, pp. 1–7.

[83] A. Sheikh, T. Youssef, and O. Mohammed, “Ac microgrid control using adaptive synchronous reference frame pll,” in *2017 Ninth Annual IEEE Green Technologies Conference (GreenTech)*. IEEE, 2017, pp. 46–51.

[84] B. T. Gwynn and R. A. de Callafon, “Robust real-time inverter-based reactive power compensation,” in *2019 IEEE 58th Conference on Decision and Control (CDC)*. IEEE, 2019, pp. 6578–6583.

[85] T. Jiang, L. M. Costa, N. Siebert, and P. Tordjman, “Automated microgrid control systems,” *CIRED-Open Access Proceedings Journal*, vol. 2017, no. 1, pp. 961–964, 2017.

[86] A. D. Bintoudi, L. Zyglakis, T. Apostolos, D. Ioannidis, S. Al-Agtash, J. L. Martinez-Ramos, A. Onen, B. Azzopardi, L. Hadjidemetriou, N. Martensen *et al.*, “Novel hybrid design for microgrid control,” in *2017 IEEE PES Asia-Pacific Power and Energy Engineering Conference (APPEEC)*. IEEE, 2017, pp. 1–6.

[87] G. B. Giannakis, V. Kekatos, N. Gatsis, S.-J. Kim, H. Zhu, and B. F. Wollenberg, “Monitoring and optimization for power grids: A signal processing perspective,” *IEEE Signal Processing Magazine*, vol. 30, no. 5, pp. 107–128, 2013.

[88] S. A. R. Konakalla, A. Valibeygi, and R. A. de Callafon, “Microgrid dynamic modeling and islanding control with synchrophasor data,” *IEEE Transactions on Smart Grid*, vol. 11, no. 1, pp. 905–915, 2019.

[89] E. O. Schweitzer and D. E. Whitehead, “Real-time power system control using synchrophasors,” in *2008 61st Annual Conference for Protective Relay Engineers*. IEEE, 2008, pp. 78–88.

[90] Y. Hu, S. A. R. Konakalla, and R. A. de Callafon, “Covariance based estimation for reduced order models of microgrid power flow dynamics,” in *Proceedings of IFAC 18th Symposium on System Identification (SYSID 2018)*, vol. 51, no. 15. Elsevier, 2018, pp. 903–908.

[91] S.-H. Ko, S. R. Lee, H. Dehbonei, and C. V. Nayar, “Application of voltage-and current-controlled voltage source inverters for distributed generation systems,” *IEEE Transactions on Energy Conversion*, vol. 21, no. 3, pp. 782–792, 2006.

[92] K. Yeager and J. Willis, “Modeling of emergency diesel generators in an 800 megawatt nuclear power plant,” *IEEE Transactions on Energy Conversion*, vol. 8, no. 3, pp. 433–441, 1993.

[93] S. Jain and V. Agarwal, “A single-stage grid connected inverter topology for solar pv systems with maximum power point tracking,” *IEEE transactions on power electronics*, vol. 22, no. 5, pp. 1928–1940, 2007.

[94] M. A. Abusara, J. M. Guerrero, and S. M. Sharkh, “Line-interactive ups for microgrids,” *IEEE Transactions on Industrial Electronics*, vol. 61, no. 3, pp. 1292–1300, 2013.

[95] S. D’Arco and J. A. Suul, “A synchronization controller for grid reconnection of islanded virtual synchronous machines,” in *2015 IEEE 6th International Symposium on Power Electronics for Distributed Generation Systems (PEDG)*. IEEE, 2015, pp. 1–8.

[96] E. Naderi and K. Khorasani, “Subspace-based identification of linear systems using arbitrary data segments,” in *2016 American Control Conference (ACC)*. IEEE, 2016, pp. 48–53.

[97] P. Welch, “The use of fast fourier transform for the estimation of power spectra: a method based on time averaging over short, modified periodograms,” *IEEE Transactions on Audio and Electroacoustics*, vol. 15, no. 2, pp. 70–73, 1967.

[98] J. B. Goodenough and K.-S. Park, “The li-ion rechargeable battery: a perspective,” *Journal of the American Chemical Society*, vol. 135, no. 4, pp. 1167–1176, 2013.

[99] G. L. Plett, *Battery management systems, Volume II: Equivalent-circuit methods*. Artech House, 2015.

[100] Y. Liang, C.-Z. Zhao, H. Yuan, Y. Chen, W. Zhang, J.-Q. Huang, D. Yu, Y. Liu, M.-M. Titirici, Y.-L. Chueh *et al.*, “A review of rechargeable batteries for portable electronic devices,” *InfoMat*, vol. 1, no. 1, pp. 6–32, 2019.

[101] B. Scrosati, J. Garche, and W. Tillmetz, *Advances in battery technologies for electric vehicles*. Woodhead Publishing, 2015.

[102] D. Tenfen, E. C. Finardi, B. Delinchant, and F. Wurtz, “Lithium-ion battery modelling for the energy management problem of microgrids,” *IET Generation, Transmission & Distribution*, vol. 10, no. 3, pp. 576–584, 2016.

[103] G. L. Plett, *Battery management systems, Volume I: Battery modeling*. Artech House, 2015.

[104] S. Santhanagopalan, Q. Guo, P. Ramadass, and R. E. White, “Review of models for predicting the cycling performance of lithium ion batteries,” *Journal of power sources*, vol. 156, no. 2, pp. 620–628, 2006.

[105] D. Di Domenico, A. Stefanopoulou, and G. Fiengo, “Lithium-ion battery state of charge and critical surface charge estimation using an electrochemical model-based extended kalman filter,” *Journal of dynamic systems, measurement, and control*, vol. 132, no. 6, 2010.

[106] J. C. Forman, S. J. Moura, J. L. Stein, and H. K. Fathy, “Genetic identification and fisher identifiability analysis of the doyle–fuller–newman model from experimental cycling of a lifepo4 cell,” *Journal of Power Sources*, vol. 210, pp. 263–275, 2012.

[107] J. Li, K. Adewuyi, N. Lotfi, R. G. Landers, and J. Park, “A single particle model with chemical/mechanical degradation physics for lithium ion battery state of health (soh) estimation,” *Applied energy*, vol. 212, pp. 1178–1190, 2018.

[108] G. K. Prasad and C. D. Rahn, “Model based identification of aging parameters in lithium ion batteries,” *Journal of power sources*, vol. 232, pp. 79–85, 2013.

[109] F. M. Kindermann, J. Keil, A. Frank, and A. Jossen, “A sei modeling approach distinguishing between capacity and power fade,” *Journal of The Electrochemical Society*, vol. 164, no. 12, p. E287, 2017.

[110] M. B. Pinson and M. Z. Bazant, “Theory of sei formation in rechargeable batteries: capacity fade, accelerated aging and lifetime prediction,” *Journal of the Electrochemical Society*, vol. 160, no. 2, p. A243, 2012.

[111] C. Zou, C. Manzie, and D. Nešić, “Model predictive control for lithium-ion battery optimal charging,” *IEEE/ASME Transactions on Mechatronics*, vol. 23, no. 2, pp. 947–957, 2018.

[112] A. Farmann and D. U. Sauer, “A comprehensive review of on-board state-of-available-power prediction techniques for lithium-ion batteries in electric vehicles,” *Journal of Power Sources*, vol. 329, pp. 123–137, 2016.

[113] X. Lin, H. E. Perez, S. Mohan, J. B. Siegel, A. G. Stefanopoulou, Y. Ding, and M. P. Castanier, “A lumped-parameter electro-thermal model for cylindrical batteries,” *Journal of Power Sources*, vol. 257, pp. 1–11, 2014.

[114] B. Xia, X. Zhao, R. De Callafon, H. Garnier, T. Nguyen, and C. Mi, “Accurate lithium-ion battery parameter estimation with continuous-time system identification methods,” *Applied energy*, vol. 179, pp. 426–436, 2016.

[115] N. Tian, H. Fang, J. Chen, and Y. Wang, “Nonlinear double-capacitor model for rechargeable batteries: Modeling, identification, and validation,” *IEEE Transactions on Control Systems Technology*, vol. 29, no. 1, pp. 370–384, 2021.

[116] X. Hu, S. Li, and H. Peng, “A comparative study of equivalent circuit models for li-ion batteries,” *Journal of Power Sources*, vol. 198, pp. 359–367, 2012.

[117] C. Zou, X. Hu, S. Dey, L. Zhang, and X. Tang, “Nonlinear fractional-order estimator with guaranteed robustness and stability for lithium-ion batteries,” *IEEE Transactions on Industrial Electronics*, vol. 65, no. 7, pp. 5951–5961, 2017.

[118] V. H. Johnson, A. A. Pesaran, and T. Sack, “Temperature-dependent battery models for high-power lithium-ion batteries,” National Renewable Energy Lab., Golden, CO (US), Tech. Rep., 2001.

[119] K. P. Murphy, *Machine learning: a probabilistic perspective.* MIT press, 2012.

[120] D. Yang, X. Zhang, R. Pan, Y. Wang, and Z. Chen, "A novel gaussian process regression model for state-of-health estimation of lithium-ion battery using charging curve," *Journal of Power Sources*, vol. 384, pp. 387–395, 2018.

[121] R. R. Ardeshiri, B. Balagopal, A. Alsabbagh, C. Ma, and M.-Y. Chow, "Machine learning approaches in battery management systems: State of the art: Remaining useful life and fault detection," in *2020 2nd IEEE International Conference on Industrial Electronics for Sustainable Energy Systems (IESES)*, vol. 1. IEEE, 2020, pp. 61–66.

[122] E. Chemali, P. J. Kollmeyer, M. Preindl, and A. Emadi, "State-of-charge estimation of li-ion batteries using deep neural networks: A machine learning approach," *Journal of Power Sources*, vol. 400, pp. 242–255, 2018.

[123] Y. L. Murphey, J. Park, Z. Chen, M. L. Kuang, M. A. Masrur, and A. M. Phillips, "Intelligent hybrid vehicle power control—part i: Machine learning of optimal vehicle power," *IEEE Transactions on Vehicular Technology*, vol. 61, no. 8, pp. 3519–3530, 2012.

[124] X. Hu, S. E. Li, and Y. Yang, "Advanced machine learning approach for lithium-ion battery state estimation in electric vehicles," *IEEE Transactions on Transportation electrification*, vol. 2, no. 2, pp. 140–149, 2015.

[125] M. A. Hannan, M. H. Lipu, A. Hussain, P. J. Ker, T. Mahlia, M. Mansor, A. Ayob, M. H. Saad, and Z. Dong, "toward enhanced state of charge estimation of lithium-ion batteries using optimized machine learning techniques," *Scientific reports*, vol. 10, no. 1, pp. 1–15, 2020.

[126] E. Stoudenmire and D. J. Schwab, "Supervised learning with tensor networks," in *Advances in Neural Information Processing Systems*, 2016, pp. 4799–4807.

[127] M. Guo, G. Sikha, and R. E. White, "Single-particle model for a lithium-ion cell: Thermal behavior," *Journal of The Electrochemical Society*, vol. 158, no. 2, p. A122, 2010.

[128] H. Fang, Y. Wang, and J. Chen, "Health-aware and user-involved battery charging management for electric vehicles: Linear quadratic strategies," *IEEE Transactions on Control Systems Technology*, vol. 25, no. 3, pp. 911–923, 2016.

[129] D. Andre, M. Meiler, K. Steiner, C. Wimmer, T. Soczka-Guth, and D. Sauer, "Characterization of high-power lithium-ion batteries by electrochemical impedance spectroscopy. i. experimental investigation," *Journal of Power Sources*, vol. 196, no. 12, pp. 5334–5341, 2011.

[130] M. Dubarry, N. Vuillaume, and B. Y. Liaw, "From single cell model to battery pack simulation for li-ion batteries," *Journal of Power Sources*, vol. 186, no. 2, pp. 500–507, 2009.