The Pennsylvania State University
The Graduate School

DATA-DRIVEN MODELING FOR ANALYSIS AND CONTROL OF DYNAMICAL SYSTEMS

A Dissertation in
Aerospace Engineering
by
Damien Guého

Submitted in Partial Fulfillment
of the Requirements
for the Degree of

Doctor of Philosophy

December 2022

The dissertation of Damien Guého was reviewed and approved by the following:

Puneet Singla
Professor of Aerospace Engineering
Dissertation Co-Advisor, Chair of Committee

Robert G. Melton
Professor of Aerospace Engineering
Dissertation Co-Advisor

Vishal Monga
Professor of Electrical Engineering and Computer Science

Daning Huang
Assistant Professor of Aerospace Engineering

Manoranjan Majji
Professor of Aerospace Engineering
Special Member

Amy R. Pritchett
Professor of Aerospace Engineering
Head of the Department of Aerospace Engineering

Abstract

This dissertation advances the understanding of data-driven modeling and delivers tools to pursue the ambition of complete unsupervised identification of dynamical systems.

From measured data only, the proposed framework consists of a series of modules to derive accurate mathematical models for the state prediction of a wide range of linear and nonlinear dynamical systems. Identified models are constructed to be of low complexity and amenable for analysis and control. This developed framework provides a unified mathematical structure for the identification of nonlinear systems based on the Koopman operator.

A main contribution of this dissertation is to introduce the concept of time-varying Koopman operator for accurate modeling of dynamical systems in a given domain around a reference trajectory. Subspace identification methods coupled with sparse approximation techniques deliver accurate models both in the continuous and discrete time domains. This allows for perfect reconstruction of several classes of nonlinear dynamical systems, from the chaotic behavior of the Lorenz oscillator to identifying the Newton's law of gravitation.

The connection between the Koopman operator and higher-order state transition matrices (STMs) is explicitly discussed. It is shown that subspace methods based on the Koopman operator are able to accurately identify the linear time varying model for the propagation of higher order STMs when polynomial basis are used as lifting functions.

Such algorithms are validated on a wide range of nonlinear dynamical systems of varying complexity and are proven to be very effective on nonlinear systems of higher dimension where traditional methods either fail or perform poorly. Applications include model-order reduction in hypersonic aerothermoelasticity and reduced-order dynamics in a high-dimensional finite-element model of the Von Kàrmàn Beam. Numerical simulation results confirm better prediction accuracy by several orders of magnitude using this framework.

Additionally, a major objective of this research is to enhance the field of data-driven uncertainty quantification for nonlinear dynamical systems. Uncertainty propagation through nonlinear dynamics is computationally expensive. Conventional approaches focus on finding a reduced order model to alleviate the computational complexity associated with the uncertainty propagation algorithms. This dissertation exploits the fact that the moment propagation equations form a linear time-varying (LTV) system and use system theory to identify this LTV system from data only. By estimating and propagating higher-order moments of an initial probability density function, two new approaches are presented and compared to analytical and quadrature-based methods for estimating the uncertainty associated with a system's states. In all test cases considered in this dissertation, a newly-introduced indirect method using a time-varying subspace identification technique jointly with a quadrature method achieved the best results.

This dissertation also extends the Koopman operator theoretic framework for controlled dynamical systems and offers a global overview of bilinear system identification techniques as well as perspectives and advances for bilinear system identification. Nonlinear dynamics with a control action are approximated as a bilinear system in a higher-dimensional space, leading to increased accuracy in the prediction of the system's response. In the same context, a data-driven parameter sensitivity method is developed using bilinear system identification algorithms.

Finally, this dissertation investigates new ways to alleviate the effect of noise in the data, leading to new algorithms with data-correlations and rank optimization for optimal subspace identification.

Table of Contents

List of Figures

List of Tables

List of Symbols

n	State dimension, generally unknown
m	Number of measured outputs
r	Number of inputs
t, t_k, k	Time variable, continuous or discrete
$\boldsymbol{x}, \boldsymbol{x}(t), \boldsymbol{x}(t_k), \boldsymbol{x}_k$	State vector
$\boldsymbol{y}, \boldsymbol{y}(t), \boldsymbol{y}(t_k), \boldsymbol{y}_k$	Output vector
$\boldsymbol{u}, \boldsymbol{u}(t), \boldsymbol{u}(t_k), \boldsymbol{u}_k$	Input vector
$\boldsymbol{f} : \mathbb{R}^n \to \mathbb{R}^n$	Dynamical flow, usually unknown
$\boldsymbol{F}_c, \boldsymbol{F} : \mathbb{R}^n \to \mathbb{R}^n$	Flow-map associated to $\boldsymbol{f}$, continuous or discrete
$\phi_i : \mathbb{R}^n \to \mathbb{R}$	$i-$th basis function
$\boldsymbol{\chi}, \boldsymbol{\chi}_k$	Augmented vector of measurements
$\mathcal{K}_c, \mathcal{K}$	Koopman operator, continuous and discrete
$A_c, A_c(t), A_c(t_k)$	$n \times n$ continuous-time state matrix, constant or time-varying
A, A_k	$n \times n$ discrete-time state matrix, constant or time-varying
$B_c, B_c(t), B_c(t_k)$	$n \times r$ continuous-time input influence matrix, constant or time-varying
B, B_k	$n \times r$ discrete-time input influence matrix, constant or time-varying
$C, C(t), C(t_k), C_k$	$m \times n$ output influence matrix, constant or time-varying
$D, D(t), D(t_k), D_k$	$m \times r$ direct transmission matrix, constant or time-varying
$\boldsymbol{R}^{(q)}, \boldsymbol{R}_k^{(q)}$	$n \times qr$ block controllability matrix, constant or time-varying
$\boldsymbol{O}^{(p)}, \boldsymbol{O}_k^{(p)}$	$pm \times n$ block observability matrix, constant or time-varying
$\boldsymbol{H}^{(p,q)}, \boldsymbol{H}_k^{(p,q)}$	$pm \times qr$ Hankel matrix, constant or time-varying

Acknowledgments

This dissertation and the research work associated to it is the culmination of several years of work; the completion of which would not have been possible without selfless and continuous support from others along the way. I would like to express my gratitude to my advisor, Dr. Puneet Singla, for his encouragement, guidance and motivation throughout my academic career and my time at Penn State. For these past few years, I have taken constant inspiration from Dr. Singla in many fields of engineering, especially in dynamical system theory, probability theory, mathematics, and astrodynamics. It is indeed a great honor to have shared active research activities with him; I learned more than I could have possibly imagined and I would not be where I am today without his support. Dr. Singla has always been going above and beyond his duties as an academic advisor, Professor and friend, always available, accommodating when I was struggling. He has always been someone I could trust and talk to. Thank you.

I shared memorable research collaboration with Dr. Robert Melton in the process of conducting the research underlying this dissertation. I would like to express my gratitude to Dr. Melton for his mentorship over these past five years for all the motivation, guidance, interaction and insight; he encouraged, directed me and brought new challenges, which contributed to the completion of this work. My interaction and collaboration with Dr. Melton enriched my graduate studies at Penn State. Both Dr. Singla and Dr. Melton believed in me like nobody else very soon and gave me strong support. I am very much looking forward to our continued collaborations in the future.

I would also like to extend my gratitude to the members of my doctoral committee for their guidance and feedback throughout the process of writing my dissertation. Dr. Majji, Dr. Huang and Dr. Monga provided great advice, support, guidance and insight on several joint research topics. I also thank the rest of the faculty and staff in the Department of Aerospace Engineering at Penn State.

Next I would like to thank my family for their love, encouragements and support. They supported me in pursuing my dreams for as long as I can remember. Nobody has been more important to me in the pursuit of this project than my family. My parents Emmanuel and Isabelle are among the best role models I could ever ask for. They assisted and encouraged me for every step in this modern world and supported every decision I made. My first brother Quentin is a major source of encouragement and undoubtedly a role model. He is someone I can look up to and I know will be extremely successful thanks to his hard-work and dedication. Yohan, my second brother, is immensely talented and is someone I can rely on in any circumstance. He has been accompanying me in this journey and I am highly proud of the man you've become. I would also like to give a special thanks to my partner in life, Alec, for being there to support me, even far from home, and for reassuring me and loving me all along this great adventure. My uncle Christophe is my number one supporter and has always been a great source of inspiration with my grand-parents. All together with the rest of my family, they form the best support structure I could ever ask for.

I am also grateful to all the people I met in Penn State in general who helped me live a great life far from home. Special thanks to all lab partners I have had the occasion to work with in the CASS Lab. You guys have ensured a smooth transition from France to the United States and showed great assistance, humor and support and I hope I will have many more occasions to grab some drinks and play some volleyball/tennis/pool/poker with all of you. You all form a good team at Penn State, but

don't make Dr. Singla's life too easy!

A special thanks goes for all my friends in State College and beyond, for your support, encouragements and your availability all these years. I have made countless memories and I am immensely grateful to be surrounded by such talented, well-rounded and fun people. I hope this is just the first chapter of a long story that we will write together for the years to come, either in Paris, State College, Boston, or Miami... ;)

Throughout the years, this material is based upon work supported through United States Air Force Office of Scientific Research (AFOSR) grants FA9550-15-1-0313, FA9550-17-1-0088, FA9550-20-1-0176 and National Science Foundation (NSF) Award No. CMMI-1826990.

Chapter 1
Introduction

1.1 The Field of Data-Driven System Identification

The general purpose of data-driven system identification is to invent methods and algorithms to develop or improve a mathematical representation of an unknown dynamical process using data. The word *system* relates to any dynamical process that evolves with time and can represent a real physical process or a simulated experiment. In the case of a real system, there is a need for accurate data-driven system identification when the underlying dynamical process is unknown or if state-of-the-art physics-based models are not able to capture the relevant dynamics to explain observable data. History as shown that the task of characterizing the dynamics of complex structures is a difficult one, especially in challenging environments. When the system refers to a simulated experiment, the need for system identification is paramount as very few high-fidelity simulations are amenable for analysis and control. Usually of high complexity (in terms of fidelity, dimensionality), these systems require reduced-order modeling and subspace identification to design strategies for analysis and control. Figure 1.1 provides a few examples of dynamical systems.

The field of system identification has been an important discipline within the field of guidance, navigation and control (GNC), the automatic control area, structural engineering, reduced-order modeling and model testing [1, 2]. Over the past six decades, the field of system identification has provided multiple tools for design, analysis and control of engineering systems. Earlier efforts have concentrated on linear system identification with a special emphasis on identifying the minimal state-space representation to define the subspace over which the system dynamics evolves. State-space models are particularly suitable since they lend themselves to linear algebra techniques, robust numerical simulation, and modern control design methods. Among the first were the works of Gilbert and Kalman [3, 4], which introduced the important principles of realization theory in terms of the concepts of controllability and observability. They first introduced the concept of state-variable equations which realize the external description via an equivalent internal description of a dynamical system. Whether the analyst or the engineer is interested in the synthesis or the analysis, these equations are an efficient and useful model with which one can proceed to further analysis and optimization.

The problem of realization for linear time invariant systems was first stated by Gilbert [3] who gave

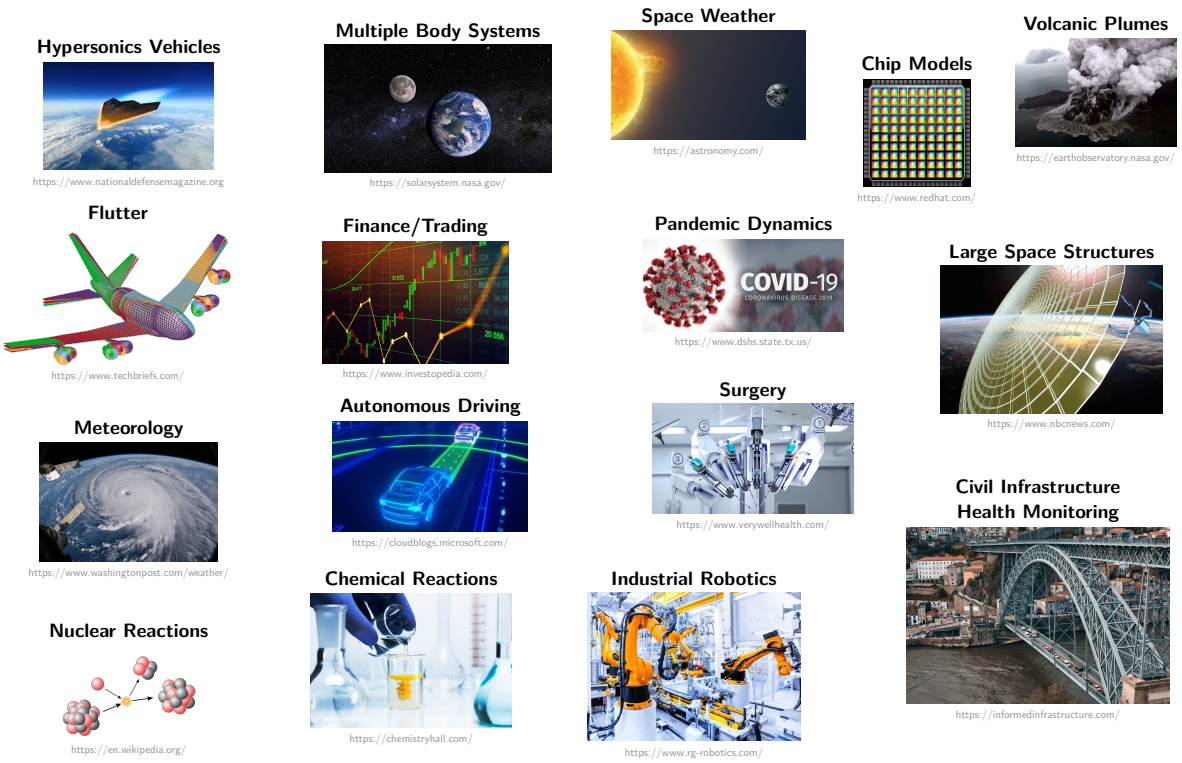


Figure 1.1: A few examples of dynamical processes

an algorithm for computing the map transfer function matrix to state-variable differential equations. A second algorithm for the same problem was given at the same time by Kalman [5] using the theory of controllability and observability and requiring linear algebra-type computation. A few years later, Ho and Kalman approached this problem from a new viewpoint. They showed that the minimum realization problem is equivalent to a representation problem involving a sequence of real matrices known as Markov parameters (pulse response samples). Minimum realization means a model with the smallest state-space dimension among all possible system realizations that have the same input-output relations. Although several techniques of minimum realization are available in the literature, formal direct application to modal parameter identification for flexible structures was not addressed until 1984.

Building upon initial work by Gilbert [3] and Kalman [4, 5], several methods have been developed to identify most observable and controllable subspaces of the system from given input-output (I/O) data [6–16]. Subspace methods for system identification such as the Eigensystem Realization Algorithm (ERA) introduced in 1985 [13] were developed for modal parameter identification and model reduction of dynamic systems using test data. The ERA was originally introduced to identify the system matrices from an impulse response or initial condition response but a few years after the introduction of the ERA, the Observer/Kalman Identification Algorithm (OKID) was introduced to generalize the ERA for generic input-output data. It is formulated entirely in the time-domain and computes the Markov parameters of a linear system, from which the state-space model and a corresponding observer are determined simultaneously [15, 17–19]. Similar subspace methods have been re-discovered later in 2010

in the field of fluid mechanics, like Dynamic Mode Decomposition (DMD) and its variants [20–23]. ERA and DMD have been useful for extracting spatial-temporal coherent structures from data, and happen to be very similar in their implementation [21].

State-space realizations methods have been shown to work very well for numerical simulations, and for experimental results from structures with modes that are well separated in frequency. The popularity of these methods lies in the simplicity with which the model order can be selected. In general, linear system identification methods are able to capture the main physics as well as the subspace in which the dynamics is evolving. However, linear methods are unable to capture nonlinearities and limitations appear when these state-space methods are applied to complex structures or when collected data include too much noise. In those cases, a state-space model does not necessarily converge and using more data in the identification process may not necessarily result in a better approximation as the computations become unfeasible. In addition, the effect of noise in the data has not been studied in the literature extensively. Some techniques have been developed involving data correlation to minimize the influence of noise during the identification process, but no practical linear identification framework has been developed to be robust enough in the presence of noise.

As a step towards system identification of nonlinear dynamical systems, several efforts have been made to develop time-varying models and to generalize subspace system identification algorithms like the ERA to the case of time-varying systems. Earliest efforts in the development of methods for linear time-varying systems involved recursive and fast implementations of the time invariant methods by exploring structural properties of the input/output realizations. Subsequently, significant results were obtained by Shokoohi and Silverman [24] and Dewilde and Van der Veen [25] that generalized several concepts in the classical linear time invariant system theory consistently. The idea of repeated experiments has been introduced [26, 27] and presented as practical methods to realize the conceptual state space model identification strategies. From a perspective of generalizing the ERA to the case of time-varying systems, Majji, Juang and Junkins [28, 29] developed a time-varying version of ERA. In parallel, using an asymptotically stable observer (to remedy the problem of unbounded growth in the number of experiments), they developed the time-varying observer/Kalman-filter system identification (TVOKID) to work with TVERA.

In contrast to linear system identification, nonlinear system identification problems are still treated mostly on a system-by-system basis with popular methods being Volterra series models [2, 30, 31], global-local learning [32,33] and neural network (NN) models [34]. The main essence of nonlinear system identification methods has been to expand the nonlinear unknown function as a linear combination of basis functions or kernels and their amplitude. Many of these methods differ in their choice of basis functions and their learning methodology. Methods like Volterra series approximation utilize Volterra kernels to provide a global approximation of the underlying dynamics while global-local approximation methods merge various local approximations valid in a local region to find a global approximation of the underlying dynamics [32]. More prevalent machine learning methods such as multi-layered NNs (also known as deep NNs) use a composition of nonlinear transformations to approximate the unknown I/O mapping. Each layer of the NN corresponds to one nonlinear transformation which is represented by a linear combination of fixed bases such as sigmoid functions known as neurons or perceptrons. According

to Cover [35, 36] and Kolmogorov's theorems, multi-layer NNs can serve as universal approximators but, in actuality, they offer no guarantee on accuracy in practice for a reasonable dimensionality (global and distributed approximation can be at the expense of high parametric dimensionality). Furthermore, the learning of parameters for multi-layer NN often involves nonlinear optimization due to composition of multiple nonlinear transformations. All of these methods focus on improving approximation accuracy by increasing the number of parameters of the models in a brute force manner through an increase in the number of basis functions, local models and/or layers of the network. A key issue arises where, if one fixes the architecture and basis functions, a given method's ability to approximate a given system's behavior can be deduced only after the learning process is over. Adaptation of the approximation architecture, not simply adjusting weights in a fixed architecture, has emerged as the key to convergence reliability and accuracy. Therefore, approximation capabilities of state-of-the-art machine learning approaches (particularly deep learning) in capturing the underlying physical characteristics of a dynamical system remain poorly understood due to the fact that these algorithms are unable to learn underlying physical features (or characteristics) of the system.

Additionally, despite the abundance of experimental data and the development of new technologies, the ability of these algorithms to extrapolate a general physical model and find governing equations from data has been very limited, especially beyond the scope of the domain where they were sampled and constructed. The major challenge in nonlinear system identification comes with the number of basis functions (or nodes) needed to extract relevant features from I/O data. A small number of units may not be enough to capture a given system's complex I/O mapping and, alternatively, a large number of units may overfit the data (and fit the noise) and may not generalize the behavior. As science attempts to understand ever more complex systems, and as data sets become larger and more difficult to process, new architectures and methods based on artificial intelligence (AI) will need to generate deep insights based on physics, mathematics, and context-aware representations that succinctly capture relevant dynamics and behaviors of complex systems under study and be able to generalize across domains.

More recently, the dynamical system discovery problem from the perspective of sparse regression and compressed sensing has been addressed by leveraging the fact that most physical systems have only a few relevant terms that define the dynamics, making the governing equations sparse in a high-dimensional nonlinear function space [33, 37–40]. Even more recently, advances in compressed sensing and sparse regression have been exploited to learn appropriate basis functions from an over-complete dictionary of basis functions without performing an exhaustive search [39–41]. Although this sparse representation through the iterative least-squares problem guarantees the balance between model complexity and accuracy, the resulting algorithm is susceptible to noise in state measurement. In [42, 43], an approach named subsampling-based threshold sparse Bayesian regression (Subtsbr) is presented to accomodate high noise in the measurments for states and state derivatives and a Galerkin formulation that involves projecting the errors on a set of basis functions known as test functions is considered in [44] to avoid estimating time derivatives of the state variables. While this formulation provides better results in the presence of noise, the choice of test functions severely affects the performance of the algorithm. Alternate formulations [45] consider a direct integral form of the dynamics for first-order systems in conjunction with a regularized ℓ_1 optimization problem to find the appropriate basis functions to approximate the unknown system dynamics.

In a similar fashion, recent advances in nonlinear system identification have used the Koopman operator-theoretic approach to obtain precise predictions of a nonlinear dynamical system as the output of a truncated linear dynamical system. The Koopman analysis complements more standard geometric [46] and probabilistic approaches by governing the evolution of scalar observables of the system state, lifting the nonlinear dynamics into a higher-dimensional space where the evolution of the flow of the system can be linear, hence trading dimensionality for linearity [47, 48]. More precisely, a nonlinear dynamical system associated with a Hamiltonian flow can be analyzed with an infinite dimensional linear operator on the Hilbert space of observable functions. This approach is no different than other nonlinear system identification techniques: it relies on a set of basis functions (or *measurement functions* as found in the literature) to represent the dynamics of the system. Many efforts have focused on adapting the architecture of the network by selecting appropriate models from a pre-defined dictionary of models [37, 38, 49–51] and advances in compressed sensing and sparse regression have been exploited to select appropriate basis functions without performing an exhaustive search [39–45, 52, 53]. The key aspect of the Koopman operator approach is to rearrange the dynamics using these measurement functions, hence obtaining predictions of a nonlinear dynamical system as the output of a linear dynamical system. While the core challenge of the Koopman operator-theoretic approach is to specify (directly or indirectly through decomposition) this Hilbert space of measurement functions of the state of the system, the theory has been applied for uncontrolled [54, 55] and controlled systems [56, 57] with promising results using popular subspace realization methods such as ERA or DMD and its extensions [58]. The resulting linear operator is a local approximator of the nonlinear dynamical system valid in the neighborhood of a nominal point and the domain of validity of this local linear approximation improves as the dimension of the lifting space is increased (i.e. more basis functions). Therefore, this time-invariant operator represents the amplitude of the modes selected to characterize the dynamics.

At present, the majority of methods available in the literature for nonlinear system identification present several shortcomings. For nonlinear dynamics evolving in regimes near equilibrium points, linear identification methods based on subspace methods are found to be effective for short-term prediction but rapidly collapse for systems with higher degrees of nonlinearity, operating farther away from the equilibrium point or for long-term propagation. While some approaches based on the Koopman operator have the potential to mitigate some of these issues, Koopman-based techniques presented in the literature are only suited for low-dimensionality uncontrolled problems. Additionally, the vast majority of these methods assume full state information which does not portray the reality for real engineering applications. Even though AI-based methods are able to handle high-dimension I/O mappings, deep neural networks are usually unable to learn and capture the underlying physical features of the system, making them inoperative for analysis and control. More importantly, AI algorithms or other identification methods whose treatment requires nonlinear optimization do not identify a subspace over which the dominant dynamics evolve. Also, presence of noise in the experimental data is problem-dependent and, since system uncertainties and measurement noise are unknown, system identification algorithms should be able to reduce their impact during the identification process. Hence, one of the primary contributions of this dissertation is to propose methods and algorithms to alleviate some of these shortcomings and offer a unified and robust data-driven framework for reduced-order modeling and system identification that combines the latest techniques in time-varying subspace realization methods, sparse representation and embeddings.

A major objective of this research is to utilize this unified data-driven framework to enhance the field of data-driven uncertainty quantification for nonlinear dynamical systems. Uncertainty propagation through nonlinear dynamics is computationally expensive and conventional approaches focus on finding a reduced order model to alleviate the computational complexity associated with the uncertainty propagation algorithms. Quantitative measures to accommodate uncertainties (in control as well as state variables) from a designed trajectory are rigorous and exact for linear systems but nonlinearities in the representation of the dynamics lead to significant challenges in estimating uncertainties associated with the state vector. Approximate measures to quantify uncertainties in the real world are poor and can lead to significant compromises in the overall performance and safety. This dissertation exploits the fact that the moment propagation equations form a linear time-varying (LTV) system and use system theory to identify this LTV system from data only. By estimating and propagating higher-order moments of an initial probability density function, two new approaches are presented and compared to analytical and quadrature-based methods for estimating the uncertainty associated with a system's states.

Shifting from autonomous systems to controlled systems, nonlinear dynamics with an external control action represented in a lifted space are *not* linear anymore, nor are control affine in a Koopman framework. Approximating the controlled nonlinear system by a linear system with affine control would yield poor results. Some previous attempts in this direction showed very mixed results [56, 57] and the theory around the Koopman operator for controlled system is not mature enough. Other research works have considered introducing a control input to model chaos dynamics in certain settings [59] but only for autonomous nonlinear systems. Secondly, one could argue that a new type of lifting functions could be introduced, function of both the state and the control vectors. This would lead to identify a Koopman operator that would not only predict future values of the state but also future values of the control input, which is not desirable. Instead, this dissertation introduces the concept of bilinear Koopman operator. Bilinear state-space model identification is of interest for two main reasons. Some physical systems are inherently bilinear and bilinear models of high dimension can approximate a broad class of nonlinear systems. Nevertheless, no well-established technique for bilinear system identification is available yet, even less in the context of Koopman. The aim of this dissertation is to offer some perspectives and advances for bilinear system identification and sensitivity analysis, working towards a bilinear Koopman operator.

1.2 Research Objectives

The overarching goal of this dissertation is to consolidate the complex field of system identification by combining popular time invariant or time-varying identification methods with newly developed nonlinear sparse identification techniques. This dissertation will focus on the following specific objectives:

1. The first objective is to reconnect major subspace time-invariant and time-varying identification methods with the Koopman operator theoretic framework and offer a unified system identification structure for autonomous systems.

2. The second objective is to develop a computationally efficient method to identify a Koopman operator leveraging filtering methods and sparse approximation techniques.

3. The third objective is to extend time-varying subspace system identification methods for the identification of a time-varying Koopman operator of arbitrary order.

4. The fourth objective is to demonstrate the utility and performance of this framework through rigorous analysis and simulations for complex and high-dimensional nonlinear systems, where traditional methods either fail or perform poorly.

5. The fifth objective is to utilize this framework to enhance uncertainty quantification for nonlinear dynamical systems.

6. The sixth objective is to extend the Koopman operator theoretic framework for controlled dynamical systems and to analyze parameter sensitivity using bilinear system identification techniques.

7. The seventh and final objective is to introduce new methods for mitigating noise in the data.

The core of this dissertation research is to lay the foundation for a computationally tractable framework for enhanced data-driven system identification of dynamical systems. Broadly speaking, the developed tools are anticipated to advance the state-of-the-art in the general area of data-driven linear time-invariant and time-varying system identification, sparse system identification, data-driven uncertainty quantification, bilinear system identification and offering tools to mitigate noise in the data for any engineering application. The chapters in this dissertation are structured in the following manner.

Chapter 2 offers a global view of the system identification problem and provides a background for linear system identification that will be used throughout the dissertation and employed to develop the basis of the identification framework in the next chapters. More importantly, this chapter relates the system identification problem with the Koopman operator theoretic framework, and defines the mathematical background for a continuous and discrete-time Koopman operator.

Chapter 3 presents a generalized approach to identify the structure of governing nonlinear equations of motion from the time history of state variables and control functions. An integral form involving a low-pass filter in conjunction with sparse approximation tools is used to find a parsimonious model for underlying true dynamics from noisy measurement data.

Chapter 4 introduces the concept of time-varying Koopman operator to predict the flow of a nonlinear dynamical system. The Koopman operator provides a linear prediction model for nonlinear systems in a lifted space of infinite dimension. An extension of time-invariant subspace realization methods known as the time-varying Eigensystem Realization Algorithm (TVERA) is used to derive a finite dimensional approximation of the infinite dimensional Koopman operator at each time step.

Chapter 5 considers two challenging high-dimensional nonlinear systems and applies the methods previously developed to come up with a reduced-order model suitable for analysis.

Chapter 6 outlines different methods to estimate the moments of the probability density function associated with the dynamical states of a system. Two new approaches (one direct and one indirect) based on the previously developed framework are presented and compared to analytical and quadrature-based methods.

Chapter 7 introduces the concept of bilinear Koopman operator for controlled systems and offers a global overview of bilinear system identification techniques as well as perspectives and advances for bilinear system identification.

Chapter 8 investigates techniques to mitigate noise in the data for more accurate and reliable data-driven identification algorithms. It introduces a data-correlation approach to the time-varying

eigensystem realization algorithm (TVERA/DC) but also a method to minimize the nuclear norm of the Hankel matrix as a way to offer an interesting alternative as a heuristic for low-rank approximation problems where a clear-cut criterion for model selection cannot be deduced from the SVD.

Chapter 9 summarizes the contributions of this dissertation, comments on open areas for future research and provides concluding remarks.

Chapter 2
The System Identification Problem

2.1 Introduction

This dissertation is aimed at providing a strong mathematical background and analytical tools to participate in the growing field of data-driven system identification. The accuracy of the mathematical models directly affects the accuracy of the system capabilities and performance. As a consequence, there is a great demand for the development of advanced modeling algorithms that can adequately represent a system behavior. However, different system processes have their own unique characteristics which they do not share with other structurally different systems. Obviously the mathematical structure of engineering models are very diverse; they can be simple algebraic models, may involve differential, integral or difference equations or may be a hybrid of these. Further, many different factors, like intended use of the model, problem dimensionality, quality of the measurement data, offline or online learning, etc., will have an impact on the choice of model architecture [32]. For the simplest input-output relationship, the mapping from the input to the output (I/O mapping) is

$$\boldsymbol{y} = \mathcal{D}(\boldsymbol{u}) \tag{2.1}$$

where $\boldsymbol{y} \in \mathbb{R}^m$ is the vector of measurable quantities and $\boldsymbol{u} \in \mathbb{R}^r$ is the control action (or bifurcation parameter). $\mathcal{D}$ simply represents the overall dynamical process. Given a time-history of $\boldsymbol{u}$ and $\boldsymbol{y}$, the system identification problem reduces to the estimation of a dynamical model for $\mathcal{D}$. In other words, the role of system identification is to apply reverse engineering to input-output (I/O) data to characterize a physical system with a mathematical model. Since the initial efforts, the system identification community has sought a reliable methodology to derive a mathematical model capturing the main characteristics of dynamical systems. The most mature part of the theory deals with linear systems using well-established techniques of linear algebra and the theory of ordinary differential or difference equations. In contrast, nonlinear system identification problems are still treated mostly on a system-by-system basis. Over the past few decades, artificial neural networks (ANN) have emerged as a powerful set of tools in pattern classification, time series analysis, signal processing, dynamical system modeling and control. The popularity of ANNs can be attributed to the fact that these network

models are frequently able to learn behavior when traditional modeling is very difficult to generalize. However, as discussed in the previous chapter, approximation capabilities of state-of-the-art machine learning approaches (particularly deep learning) in capturing the underlying physical characteristics of a dynamical system remain poorly understood because these algorithms are unable to learn the underlying physical features (or characteristics) of the system. More importantly, ANNs fail to provide a physical interpretable space over which the dynamics evolve. The I/O relationship described in Eq. (2.1) can be written in terms of that subspace of dimension n:

$$\dot{\boldsymbol{x}}(t) = \boldsymbol{f}(t, \boldsymbol{x}, \boldsymbol{u}) \tag{2.2a}$$

$$\boldsymbol{y}(t) = \boldsymbol{h}(t, \boldsymbol{x}, \boldsymbol{u}) \tag{2.2b}$$

where $t \in \mathbb{R}$, $\boldsymbol{x} \in \mathbb{X} \subseteq \mathbb{R}^n$ is the state of the system (also usually the *unknown* minimal set of variables needed to describe the evolution of the system), $\boldsymbol{f} : \mathbb{R} \times \mathbb{X} \times \mathbb{U} \to \mathbb{R}^n$ is a function of a vector field that describes how the system changes at a given state in time. Coming up with a model that is able to approximate Eq. (2.2a) and Eq. (2.2b), as well as the dimension of the state of the system n, is at the heart of system identification and is considered the most challenging part. Figure 2.1 presents a typical overview of the procedure for the identification of an unknown system from data only. Basically, the procedure involves five key steps including:

1. Simulations and/or experiments;
2. Data collection and features extraction;
3. Dynamic system identification and reduced-order modeling;
4. Observer identification and controller design;
5. Verification and testing.

The mathematical model thus obtained can then be used for analysis (performance evaluation, prediction, optimization, uncertainty quantification, sensitivity analysis) and control. At the heart of the procedure is the system identification step that involves a series of algorithms and methods to derive a mathematical model from measured data. When the system is linear, i.e. $\boldsymbol{f}$ and $\boldsymbol{g}$ are linear functions of $\boldsymbol{x}$ and $\boldsymbol{u}$ in Eq. (2.2a) and Eq. (2.2b), linear subspace identification methods can be employed. The theory around linear system identification is very well developed and the next section will provide a brief overview of some notorious algorithms in the field and equip the reader with some common notations and definitions to help support the work introduced in the next chapters.

2.2 Linear Subspace Identification

The goal in this section is not to provide a comprehensive list of methods and algorithms used for the identification of linear systems but rather to highlight crucial concepts needed to understand the fundamentals of time-invariant linear subspace system identification. First, let us start by providing a quick synopsis on time-invariant linear subspace methods. Many algorithms have been established, some of them deterministic in nature, i.e. without considering noise in the measured data, and others stochastic, i.e. with formulations minimizing the noise uncertainty in the identification. During the 90s,

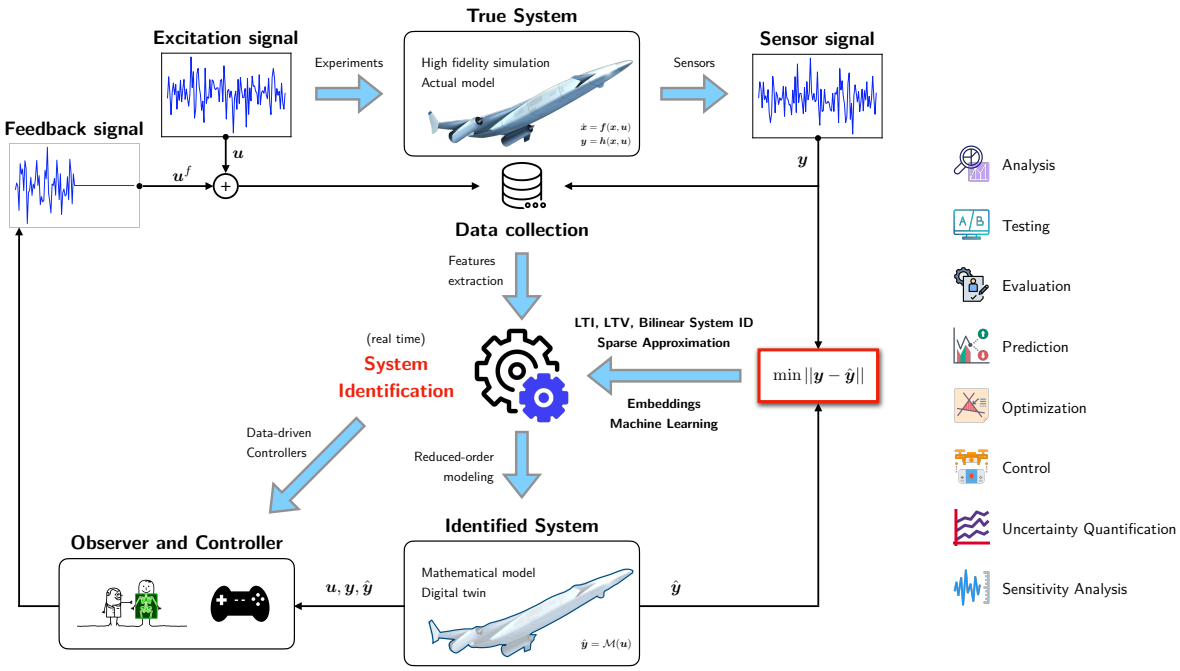


Figure 2.1: Overview of the data-driven system identification framework

building upon initial work by Gilbert and Kalman, several methods have been developed to identify most observable and controllable subspace of the system from given input-output (I/O) data [1, 6, 8, 13–19]. Under the interaction of structure and control disciplines, the Eigensystem Realization Algorithm (ERA) [13] was developed for modal parameter identification and model reduction of dynamic systems using test data. The algorithm presents a unified framework for modal parameter identification based on the Markov parameters (i.e., pulse response) making it possible to construct a Hankel matrix as the basis for the realization of a discrete-time state-space model. A few years later at NASA, Juang developed a method for simultaneously identify a linear state-space model and the associated Kalman filter from noisy input-output measurements. Known as the Observer/Kalman Identification Algorithm (OKID) and formulated entirely in the time-domain, it computes the Markov parameters of a linear system, from which the state-space model and a corresponding observer are determined simultaneously [15, 18]. The method relies on an observer equation to compress the dynamics of the system and efficiently estimate the associated system parameters (Markov parameters). In conjunction with the ERA, the method provides simultaneously both the Markov parameters and the Kalman gain, extracting all the possible information present in the data. The observer at the core of the method was proven to be the steady-state Kalman filter corresponding to the system to be identified.

2.2.1 Time-Domain State-Space Models

The equations of motion for a finite-dimensional linear-dynamic system are a set of n first-order differential equations (Eq. (2.3a)) along with an initial condition $\boldsymbol{x}(t_0)$. The n-dimensional state $\boldsymbol{x}(t)$ is most often

related to the output through the measurement equation Eq. (2.3b).

$$\dot{\boldsymbol{x}}(t) = A_c\boldsymbol{x}(t) + B_c\boldsymbol{u}(t), \tag{2.3a}$$

$$\boldsymbol{y}(t) = C\boldsymbol{x}(t) + D\boldsymbol{u}(t). \tag{2.3b}$$

The system of equations Eq. (2.3) constitutes a continuous-time state-space model of a dynamical system. Given the initial condition $\boldsymbol{x}(t_0)$ at some $t = t_0$, solving for $\boldsymbol{x}(t)$ for $t > t_0$ yields

$$\boldsymbol{x}(t) = e^{A_c(t-t_0)}\boldsymbol{x}(t_0) + \int_{t_0}^{t} e^{A_c(t-\tau)}B_c\boldsymbol{u}(\tau)\mathrm{d}\tau. \tag{2.4}$$

Without loss of generality, we will consider that $t_0 = 0$. A close approximation of a continuous-time model can be obtained by a discrete one provided that the sampling rate is sufficiently high. A linear discrete system is most commonly described by an n^{th} order difference equation, the weighting sequence, or a discrete state-space model. Let Δt be a constant time interval and $f = 1/\Delta t$ the sampling rate. Continuous versions of the A and B matrices are

$$A = e^{A_c\Delta t}, \tag{2.5a}$$

$$B = \int_0^{\Delta t} e^{A_c\tau}\mathrm{d}\tau B_c, \tag{2.5b}$$

$$\boldsymbol{x}_{k+1} = \boldsymbol{x}((k+1)\Delta t), \tag{2.5c}$$

$$\boldsymbol{u}_k = \boldsymbol{u}(k\Delta t). \tag{2.5d}$$

The discrete-time matrices A and B in Eqs (2.5a) and (2.5b) may be computed by the following series expansions:

$$A = e^{A_c\Delta t} = \sum_{i=0}^{\infty} \frac{1}{i!}\left[A_c\Delta t\right]^i, \tag{2.6a}$$

$$B = \int_0^{\Delta t} e^{A_c\tau}\mathrm{d}\tau B_c = \left[\sum_{i=0}^{\infty} \frac{1}{(i+1)!}A_c^i\left(\Delta t\right)^{i+1}\right] B_c. \tag{2.6b}$$

A sufficient condition for these series expansions to converge is that the continuous-time state matrix A_c is asymptotically stable in the sense that the real parts of all its eigenvalues are negative. If none of the eigenvalues of A_c are zero, the discrete-time matrix B may also be computed by

$$B = \left[A - I\right]A_c^{-1}B_c. \tag{2.7}$$

Therefore, a discrete-time invariant linear system can be represented by

$$\boldsymbol{x}_{k+1} = A\boldsymbol{x}_k + B\boldsymbol{u}_k \tag{2.8a}$$

$$\boldsymbol{y}_k = C\boldsymbol{x}_k + D\boldsymbol{u}_k \tag{2.8b}$$

together with an initial state vector $\boldsymbol{x}_0$, where $\boldsymbol{x}_k \in \mathbb{R}^n$, $\boldsymbol{u}_k \in \mathbb{R}^r$ and $\boldsymbol{y}_k \in \mathbb{R}^m$ are the state, control input and output vectors respectively. The constant matrices $A \in \mathbb{R}^{n\times n}$, $B \in \mathbb{R}^{r\times n}$, $C \in \mathbb{R}^{n\times m}$ and $D \in \mathbb{R}^{r\times m}$ represent the internal operation of the linear system, and are used to determine the system's

response to any input.

2.2.2 Weighting Sequence Description and Markov Parameters

Solving for the state $\boldsymbol{x}_k$ and the output $\boldsymbol{y}_k$ with arbitrary initial condition $\boldsymbol{x}_0$ in terms of the previous inputs $\boldsymbol{u}_i$, $i = 0, 1, \ldots, k$, yields

$$\boldsymbol{x}_k = A^k \boldsymbol{x}_0 + \sum_{i=1}^{k} A^{i-1} B \boldsymbol{u}_{k-i}, \tag{2.9a}$$

$$\boldsymbol{y}_k = CA^k \boldsymbol{x}_0 + \sum_{i=1}^{k} CA^{i-1} B \boldsymbol{u}_{k-i} + D\boldsymbol{u}_k. \tag{2.9b}$$

It appears naturally that the constant matrices sequence

$$h_0 = D, \quad h_1 = CB, \quad h_2 = CAB, \quad \cdots \quad h_k = CA^{k-1}B, \quad \cdots \tag{2.10}$$

plays an important role in identifying a mathematical model for linear dynamical systems. These constant matrices $\{h_i\}_{i=1,2,\cdots}$ are known as system Markov parameters (impulse response functions) or, in short, *Markov parameters*. It is obvious that the matrices A, B, C, D are embedded in the Markov parameter sequence; undeniably, the determination of Markov parameters should be tantamount to system identification. The general form of the Markov parameters is thus

$$h_i = \begin{cases} D & i = 0, \\ CA^{i-1}B & i \geq 1, \\ 0 & i < 0. \end{cases} \tag{2.11}$$

2.2.3 The Eigensystem Realization Algorithm (ERA)

The basic development of the state-space realization is attributed to Ho and Kalman [5] who introduced the important principles of minimum realization theory. The Ho-Kalman procedure uses the Hankel matrix to construct a state-space representation of a linear system from noise-free data. The methodology has been modified and substantially extended to develop the Eigensystem Realization Algorithm (ERA) [13] to identify modal parameters from noisy measurement data. System realization begins by forming the generalized Hankel matrix composed of the Markov parameters:

$$\boldsymbol{H}_k^{(p,q)} = \begin{bmatrix} h_{k+1} & h_{k+2} & \cdots & h_{k+q} \\ h_{k+2} & h_{k+3} & \cdots & h_{k+q+1} \\ \vdots & \vdots & \ddots & \vdots \\ h_{k+p} & h_{k+p+1} & \cdots & h_{k+p+q-1} \end{bmatrix} = \boldsymbol{O}^{(p)} A^k \boldsymbol{R}^{(q)}, \tag{2.12}$$

with $\boldsymbol{O}^{(p)}$ and $\boldsymbol{R}^{(q)}$ the observability and controllability matrices:

$$\boldsymbol{O}^{(p)} = \begin{bmatrix} C \\ CA \\ CA^2 \\ \vdots \\ CA^{p-1} \end{bmatrix}, \qquad \boldsymbol{R}^{(q)} = \begin{bmatrix} B & AB & A^2B & \cdots & A^{q-1}B \end{bmatrix}. \tag{2.13}$$

For the case when $k = 0$,

$$\boldsymbol{H}_0^{(p,q)} = \begin{bmatrix} h_1 & h_2 & \cdots & h_q \\ h_2 & h_3 & \cdots & h_{q+1} \\ \vdots & \vdots & \ddots & \vdots \\ h_p & h_{p+1} & \cdots & h_{p+q-1} \end{bmatrix} = \boldsymbol{O}^{(p)}\boldsymbol{R}^{(q)}. \tag{2.14}$$

If $pm \geq n$ and $qr \geq n$, matrices $\boldsymbol{R}^{(q)}$ and $\boldsymbol{O}^{(p)}$ are of rank maximum n. If the system is controllable and observable (see Appendix xx for the concepts of controllability and observability), the block matrices $\boldsymbol{R}^{(q)}$ and $\boldsymbol{O}^{(p)}$ are of rank n. Therefore,

$$\operatorname{rank}\left[\boldsymbol{H}_0^{(p,q)}\right] = \operatorname{rank}\left[\boldsymbol{O}^{(p)}\boldsymbol{R}^{(q)}\right] \leq \min\left(\operatorname{rank}\left[\boldsymbol{O}^{(p)}\right], \operatorname{rank}\left[\boldsymbol{R}^{(q)}\right]\right) = n. \tag{2.15}$$

Since $\operatorname{rank}\left[\boldsymbol{R}^{(q)}\right] = n$ ($\boldsymbol{R}^{(q)}$ is non-singular, the system is assumed to be controllable), multiplying both sides by ${\boldsymbol{R}^{(q)}}^{\dagger}$ yields

$$n = \operatorname{rank}\left[\boldsymbol{O}^{(p)}\right] = \operatorname{rank}\left[\left(\boldsymbol{O}^{(p)}\boldsymbol{R}^{(q)}\right){\boldsymbol{R}^{(q)}}^{\dagger}\right] \leq \operatorname{rank}\left[\boldsymbol{O}^{(p)}\boldsymbol{R}^{(q)}\right] = \operatorname{rank}\left[\boldsymbol{H}_0^{(p,q)}\right]. \tag{2.16}$$

Hence we have

$$\operatorname{rank}\left[\boldsymbol{H}_0^{(p,q)}\right] = n. \tag{2.17}$$

If the order is n, then the minimum dimension of the state matrix A is $n \times n$ and therefore, for any $k \geq 0$,

$$\operatorname{rank}\left[\boldsymbol{H}_k^{(p,q)}\right] = n. \tag{2.18}$$

Thus, it appears that identifying the number of dominant singular values of the Hankel matrix provides an indication about the unknown order of the reduced model to be identified. Even if more advanced methods for distinguishing true modes from noise modes exist, a simple singular value plot often allows the engineer to determine the order of the system. Thus, it is possible to observe the following

approximation

$$\begin{aligned} \boldsymbol{H}_0^{(p,q)} &= \boldsymbol{U}\boldsymbol{\Sigma}\boldsymbol{V}^\top = \begin{bmatrix} \boldsymbol{U}^{(n)} & \boldsymbol{U}^{(0)} \end{bmatrix} \begin{bmatrix} \boldsymbol{\Sigma}^{(n)} & \boldsymbol{0} \\ \boldsymbol{0} & \boldsymbol{\Sigma}^{(0)} \end{bmatrix} \begin{bmatrix} \boldsymbol{V}^{(n)\top} \\ \boldsymbol{V}^{(0)\top} \end{bmatrix} \\ &= \boldsymbol{U}^{(n)}\boldsymbol{\Sigma}^{(n)}\boldsymbol{V}^{(n)\top} + \underbrace{\boldsymbol{U}^{(0)}\boldsymbol{\Sigma}^{(0)}\boldsymbol{V}^{(0)\top}}_{\simeq \boldsymbol{0}} \\ &\simeq \boldsymbol{U}^{(n)}\boldsymbol{\Sigma}^{(n)}\boldsymbol{V}^{(n)\top} \end{aligned} \tag{2.19}$$

where $\boldsymbol{U}^{(n)}$ and $\boldsymbol{V}^{(n)}$ are orthonormal matrices:

$$\boldsymbol{U}^{(n)\top}\boldsymbol{U}^{(n)} = \boldsymbol{V}^{(n)\top}\boldsymbol{V}^{(n)} = \boldsymbol{I}^{(n)}. \tag{2.20}$$

Since $\boldsymbol{H}_0^{(p,q)}$ is primarily represented by the controllability and observability matrices, a balanced factorization leads to

$$\boldsymbol{H}_0^{(p,q)} = \boldsymbol{U}^{(n)}\boldsymbol{\Sigma}^{(n)}\boldsymbol{V}^{(n)\top} = \boldsymbol{O}^{(p)}\boldsymbol{R}^{(q)} \Rightarrow \begin{cases} \boldsymbol{O}^{(p)} = \boldsymbol{U}^{(n)}\boldsymbol{\Sigma}^{(n)1/2} \\ \boldsymbol{R}^{(q)} = \boldsymbol{\Sigma}^{(n)1/2}\boldsymbol{V}^{(n)\top} \end{cases}. \tag{2.21}$$

This choice makes both $\boldsymbol{O}^{(p)}$ and $\boldsymbol{R}^{(q)}$ balanced. Notice that $\boldsymbol{R}^{(q)}\boldsymbol{R}^{(q)\top} = \boldsymbol{O}^{(p)\top}\boldsymbol{O}^{(p)} = \boldsymbol{\Sigma}^{(n)}$. The fact that the controllability and observability matrices are equal and diagonal implies that the realized system is as controllable as it is observable. This property is called an internally balanced realization. It means that the signal transfer from the input to the state and then from the state to the output are similar and balanced.

With $k = 1$ in Eq. (2.12), one obtains that

$$\boldsymbol{H}_1^{(p,q)} = \boldsymbol{O}^{(p)}A\boldsymbol{R}^{(q)} = \boldsymbol{U}^{(n)}\boldsymbol{\Sigma}^{(n)1/2}A\boldsymbol{\Sigma}^{(n)1/2}\boldsymbol{V}^{(n)\top}, \tag{2.22}$$

and a solution for the state matrix A becomes

$$\hat{A} = \boldsymbol{O}^{(p)\dagger}\boldsymbol{H}_1^{(p,q)}\boldsymbol{R}^{(q)\dagger} = \boldsymbol{\Sigma}^{(n)-1/2}\boldsymbol{U}^{(n)\top}\boldsymbol{H}_1^{(p,q)}\boldsymbol{V}^{(n)}\boldsymbol{\Sigma}^{(n)-1/2}. \tag{2.23}$$

Moreover, from Eq. (2.13), it is clear that the first r columns of $\boldsymbol{R}^{(q)}$ form the input matrix B whereas the first m rows of $\boldsymbol{O}^{(p)}$ form the output matrix C. Defining O_i as a null matrix of order i, I_i as an identity matrix of order i and

$$\boldsymbol{E}^{(m)\top} = \begin{bmatrix} I_m & O_m & \cdots & O_m \end{bmatrix}, \tag{2.24a}$$

$$\boldsymbol{E}^{(r)\top} = \begin{bmatrix} I_r & O_r & \cdots & O_r \end{bmatrix}, \tag{2.24b}$$

a minimum realization is given by

$$\hat{A} = \boldsymbol{O}^{(p)\dagger}\boldsymbol{H}_1^{(p,q)}\boldsymbol{R}^{(q)\dagger} = \boldsymbol{\Sigma}^{(n)-1/2}\boldsymbol{U}^{(n)\top}\boldsymbol{H}_1^{(p,q)}\boldsymbol{V}^{(n)}\boldsymbol{\Sigma}^{(n)-1/2}, \tag{2.25a}$$

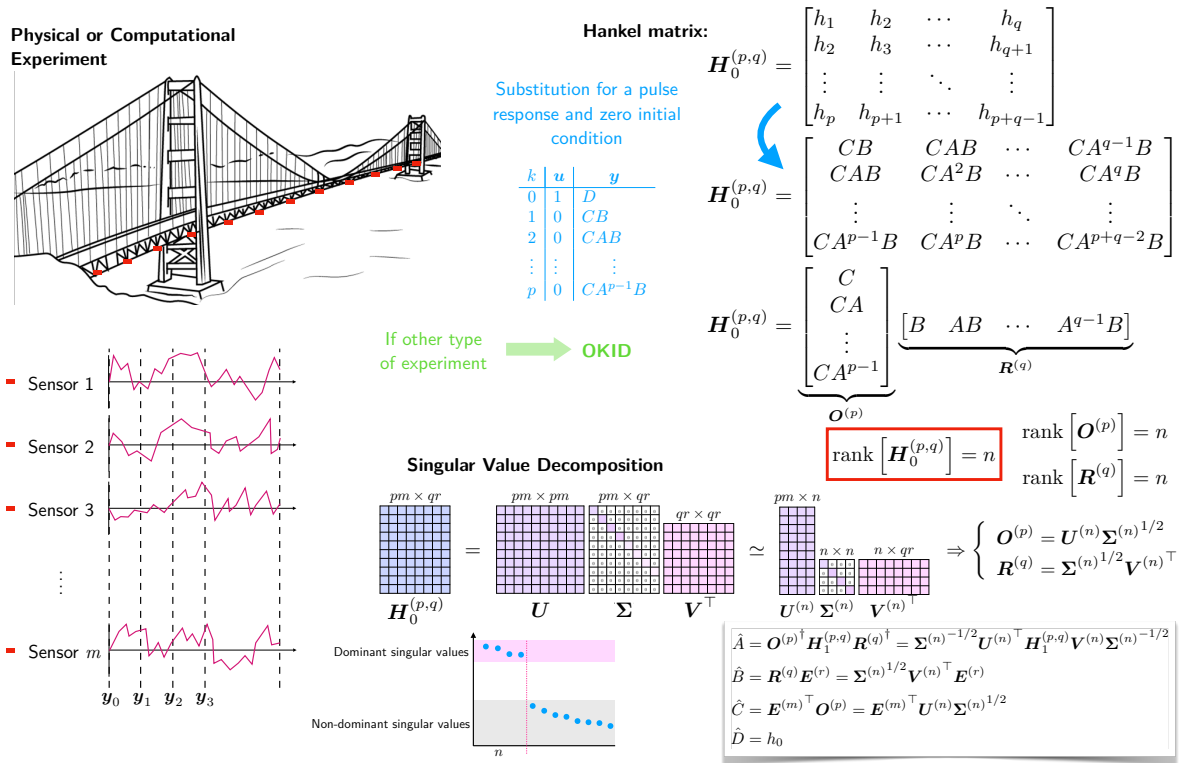


Figure 2.2: Overview of the eigensystem realization algorithm (ERA)

$$\hat{B} = \boldsymbol{R}^{(q)}\boldsymbol{E}^{(r)} = {\boldsymbol{\Sigma}^{(n)}}^{1/2}{\boldsymbol{V}^{(n)}}^{\top}\boldsymbol{E}^{(r)}, \tag{2.25b}$$

$$\hat{C} = {\boldsymbol{E}^{(m)}}^{\top}\boldsymbol{O}^{(p)} = {\boldsymbol{E}^{(m)}}^{\top}\boldsymbol{U}^{(n)}{\boldsymbol{\Sigma}^{(n)}}^{1/2}, \tag{2.25c}$$

$$\hat{D} = h_0. \tag{2.25d}$$

The realized discrete-time model represented by the matrices $\hat{A}$, $\hat{B}$, $\hat{C}$ and $\hat{D}$ can be transformed to the continuous-time model. The system frequencies and damping may then be computed from the eigenvalues of the estimated continuous-time state matrix. The eigenvectors allow a transformation of the realization to modal space and hence a determination of the complex (or damped) mode shapes and the initial modal amplitudes (or modal participation factors). Figure 2.2 displays a summary of the procedure.

Note that the Markov parameters correspond to the impulse response of the system and are thus readily available if an impulse input is applied to the system. For more generic input, one has to identify the Markov parameters through a least-squares procedure.

2.2.4 The ERA from Initial Condition Response

When there is no input to the system, the state-variable response described by Eq. (8.1) with an arbitrary set of initial conditions $\boldsymbol{x}_0$ is:

$$\boldsymbol{x}_k = A^k\boldsymbol{x}_0, \tag{2.26a}$$

$$\boldsymbol{y}_k = CA^k\boldsymbol{x}_0. \tag{2.26b}$$

In that situation, the significance of previously defined Markov parameters is gone. As they are originally defined as pulse response, there is no worthwile definition for these matrices here. Similarly, there is no meaning for controllability in this case as the input control is set to zero. However, the concept of observability is still relevant. Even though observability and controllability of a linear system are mathematical duals, the concept of observability is just a measure of how well internal states of a system can be inferred from knowledge of its external outputs. Even though controllability has no substantial meaning, it is possible to define a controllability-like matrix, named the ensemble matrix $\boldsymbol{Q}^{(q)}$, that gathers the state variable at different times:

$$\boldsymbol{Q}^{(q)} = \begin{bmatrix} \boldsymbol{x}_0 & A\boldsymbol{x}_0 & A^2\boldsymbol{x}_0 & \cdots & A^{q-1}\boldsymbol{x}_0 \end{bmatrix}. \tag{2.27}$$

Since $\boldsymbol{x}_0$ is a n-dimensional vector and $A \in \mathbb{R}^{n\times n}$, $\boldsymbol{Q}^{(q)} \in \mathbb{R}^{n\times q}$ and has rank n from the moment $q \geq n$.

Let's now define a Hankel matrix as

$$\boldsymbol{H}_k^{(p,q)} = \begin{bmatrix} \boldsymbol{y}_k & \boldsymbol{y}_{k+1} & \cdots & \boldsymbol{y}_{k+q-1} \\ \boldsymbol{y}_{k+1} & \boldsymbol{y}_{k+2} & \cdots & \boldsymbol{y}_{k+q} \\ \vdots & \vdots & \ddots & \vdots \\ \boldsymbol{y}_{k+p-1} & \boldsymbol{y}_{k+p} & \cdots & \boldsymbol{y}_{k+p+q-2} \end{bmatrix} = \boldsymbol{O}^{(p)}A^k\boldsymbol{Q}^{(q)}. \tag{2.28}$$

For the case when $k = 0$,

$$\boldsymbol{H}_0^{(p,q)} = \begin{bmatrix} \boldsymbol{y}_0 & \boldsymbol{y}_1 & \cdots & \boldsymbol{y}_{q-1} \\ \boldsymbol{y}_1 & \boldsymbol{y}_2 & \cdots & \boldsymbol{y}_q \\ \vdots & \vdots & \ddots & \vdots \\ \boldsymbol{y}_{p-1} & \boldsymbol{y}_p & \cdots & \boldsymbol{y}_{p+q-2} \end{bmatrix} = \boldsymbol{O}^{(p)}\boldsymbol{Q}^{(q)}. \tag{2.29}$$

If $pm \geq n$ and $q \geq n$, matrices $\boldsymbol{Q}^{(q)}$ and $\boldsymbol{O}^{(p)}$ are of rank maximum n. If the system is observable, the block matrix $\boldsymbol{O}_p$ is of rank n and

$$\operatorname{rank}\left[\boldsymbol{H}_k^{(p,q)}\right] = n. \tag{2.30}$$

Following the exact same steps as before, this leads to

$$\boldsymbol{H}_0^{(p,q)} = \boldsymbol{U}^{(n)}\boldsymbol{\Sigma}^{(n)}{\boldsymbol{V}^{(n)}}^\top = \boldsymbol{O}^{(p)}\boldsymbol{Q}^{(q)} \Rightarrow \begin{cases} \boldsymbol{O}^{(p)} = \boldsymbol{U}^{(n)}{\boldsymbol{\Sigma}^{(n)}}^{1/2} \\ \boldsymbol{Q}^{(q)} = {\boldsymbol{\Sigma}^{(n)}}^{1/2}{\boldsymbol{V}^{(n)}}^\top \end{cases}, \tag{2.31}$$

and a minimum realization is given by

$$\hat{A} = {\boldsymbol{O}^{(p)}}^\dagger \boldsymbol{H}_1^{(p,q)} {\boldsymbol{Q}^{(q)}}^\dagger = {\boldsymbol{\Sigma}^{(n)}}^{-1/2}{\boldsymbol{U}^{(n)}}^\top \boldsymbol{H}_1^{(p,q)} \boldsymbol{V}^{(n)}{\boldsymbol{\Sigma}^{(n)}}^{-1/2}, \tag{2.32a}$$

$$\hat{C} = {\boldsymbol{E}^{(m)}}^\top \boldsymbol{O}^{(p)} = {\boldsymbol{E}^{(m)}}^\top \boldsymbol{U}^{(n)} {\boldsymbol{\Sigma}^{(n)}}^{1/2}, \tag{2.32b}$$

$$\hat{\boldsymbol{x}}_0 = \boldsymbol{Q}^{(q)} \boldsymbol{E}^{(1)} = {\boldsymbol{\Sigma}^{(n)}}^{1/2} {\boldsymbol{V}^{(n)}}^\top \boldsymbol{E}^{(1)}. \tag{2.32c}$$

2.2.5 ERA and DMD

Similarly as ERA, the dynamic mode decomposition (DMD), first introduced in the fluid dynamics community [20–23,58,60,61], provides a practical numerical framework for Koopman mode decomposition. ERA and DMD have been useful for extracting spatial-temporal coherent structures from data, and happen to be very similar in their implementation.

ERA Algorithm	**DMD Algorithm**
1. From measurement data $\{\boldsymbol{y}_0, \boldsymbol{y}_1, \cdots\}$, build Hankel matrices $\boldsymbol{H}_0^{(p,q)}$ and $\boldsymbol{H}_1^{(p,q)}$ from $$\boldsymbol{H}_k^{(p,q)} = \begin{bmatrix} \boldsymbol{y}_k & \boldsymbol{y}_{k+1} & \cdots & \boldsymbol{y}_{k+q-1} \\ \boldsymbol{y}_{k+1} & \boldsymbol{y}_{k+2} & \cdots & \boldsymbol{y}_{k+q} \\ \vdots & \vdots & \ddots & \vdots \\ \boldsymbol{y}_{k+p-1} & \boldsymbol{y}_{k+p} & \cdots & \boldsymbol{y}_{k+p+q-2} \end{bmatrix}$$	1. From measurement data $\{\boldsymbol{y}_0, \boldsymbol{y}_1, \cdots\}$, build ensemble matrices $$\boldsymbol{\Psi}_0 = \begin{bmatrix} \boldsymbol{y}_0 & \boldsymbol{y}_1 & \cdots & \boldsymbol{y}_{N-1} \end{bmatrix},$$ $$\boldsymbol{\Psi}_1 = \begin{bmatrix} \boldsymbol{y}_1 & \boldsymbol{y}_2 & \cdots & \boldsymbol{y}_N \end{bmatrix}.$$
2. Compute SVD of $\boldsymbol{H}_0^{(p,q)}$ and select the dominant modes $$\boldsymbol{H}_0^{(p,q)} = \boldsymbol{U}\boldsymbol{\Sigma}\boldsymbol{V}^\top \simeq \boldsymbol{U}^{(n)}\boldsymbol{\Sigma}^{(n)}{\boldsymbol{V}^{(n)}}^\top.$$	2. Compute SVD of $\boldsymbol{\Psi}_0$ and select the dominant modes $$\boldsymbol{\Psi}_0 = \boldsymbol{U}\boldsymbol{\Sigma}\boldsymbol{V}^\top \simeq \boldsymbol{U}^{(n)}\boldsymbol{\Sigma}^{(n)}{\boldsymbol{V}^{(n)}}^\top.$$
3. Define the matrices $$\hat{A} = {\boldsymbol{\Sigma}^{(n)}}^{-1/2} {\boldsymbol{U}^{(n)}}^\top \boldsymbol{H}_1^{(p,q)} \boldsymbol{V}^{(n)} {\boldsymbol{\Sigma}^{(n)}}^{-1/2},$$ $$\hat{C} = \boldsymbol{U}^{(n)} {\boldsymbol{\Sigma}^{(n)}}^{1/2}[1:m,:].$$	3. Define the matrices $$\hat{\tilde{A}} = {\boldsymbol{U}^{(n)}}^\top \boldsymbol{H}_1^{(p,q)} \boldsymbol{V}^{(n)} {\boldsymbol{\Sigma}^{(n)}}^{-1},$$ $$\hat{\tilde{C}} = \boldsymbol{U}^{(n)}\boldsymbol{\Sigma}^{(n)}[1:m,:].$$
4. Dynamics on the $n-$dimensional subspace are governed by $$\boldsymbol{x}_{k+1} = \hat{A}\boldsymbol{x}_k,$$ $$\boldsymbol{y}_k = \hat{C}\boldsymbol{x}_k.$$	4. Dynamics on the $n-$dimensional subspace are governed by $$\tilde{\boldsymbol{x}}_{k+1} = \hat{\tilde{A}}\tilde{\boldsymbol{x}}_k,$$ $$\boldsymbol{y}_k = \hat{\tilde{C}}\tilde{\boldsymbol{x}}_k.$$

It is clear to see that while DMD and the ERA were developed in different contexts, they are closely related: the two realizations are related by the similarity transform

$$\hat{A} = {\boldsymbol{\Sigma}^{(n)}}^{-1/2} \hat{\tilde{A}} {\boldsymbol{\Sigma}^{(n)}}^{1/2}. \tag{2.33}$$

If $\boldsymbol{v}$ is an eigenvector of $\hat{A}$ with $\hat{A}\boldsymbol{v} = \lambda\boldsymbol{v}$, then $\boldsymbol{w} = {\boldsymbol{\Sigma}^{(n)}}^{1/2}\boldsymbol{v}$ is an eigenvector of $\hat{\tilde{A}}$ since

$$\hat{\tilde{A}}\boldsymbol{w} = \hat{\tilde{A}} {\boldsymbol{\Sigma}^{(n)}}^{1/2}\boldsymbol{v} = {\boldsymbol{\Sigma}^{(n)}}^{1/2}\hat{A}\boldsymbol{v} = \lambda {\boldsymbol{\Sigma}^{(n)}}^{1/2}\boldsymbol{v} = \lambda\boldsymbol{w}. \tag{2.34}$$

The connection suggests that strategies used in ERA computations could be leveraged for gain in DMD computations. Specifically, the Hankel matrix structure with time-shifted data used in ERA helps to overcome rank problems as long as p and q are large enough. In this dissertation, subspace identification methods are based on ERA, but it can easily be transformed into a DMD-type of algorithm using a similarity transform.

2.2.6 Conclusion

This section provided an algorithm to identify a linear subspace that governs the dominant dynamics of a dynamical process. If the process is linear, the identification is exact (up to the noise content); if the process is nonlinear, the identified linear system is a projection of the nonlinear dynamics onto a balanced linear subspace. Identifying nonlinear dynamics directly is another challenge and demands high-order expansions, as we will see in the next section.

2.3 Nonlinear System Identification: Problem Statement

The methodology for nonlinear system identification is quite different. While linear algebra techniques happen to be very powerful in identifying the system matrices of linear systems, the question on how to identify nonlinear functions $\boldsymbol{f}$ and $\boldsymbol{g}$ from Eq. (2.2a) and Eq. (2.2b) is a very challenging problem and still an open question in the field. Therefore, the majority of methods in the field adopt a major assumption to simplify the problem: the measurement equation Eq. (2.2b) is dropped and consequently we suppose to have access to the system's state $\boldsymbol{x}(t)$ at all time. The dynamics is thus expressed in a state space form from Eq. (2.2a) as

$$\frac{\mathrm{d}}{\mathrm{d}t}\boldsymbol{x}(t) = \boldsymbol{f}(t, \boldsymbol{x}, \boldsymbol{u}), \quad \boldsymbol{x}(0) = \boldsymbol{x}_0. \tag{2.35}$$

With only unknown dynamics $\boldsymbol{f}$, the main essence of popular system identification methods has been to expand this nonlinear function as a linear combination of basis functions or kernels, and their amplitude. Thus, the system identification problem is in general stated as an optimization problem as follows: given the time history of the state vector $\boldsymbol{x}(t)$, control action vector $\boldsymbol{u}(t)$ and a predetermined finite set of basis functions $\{\phi_1, \phi_2, \cdots, \phi_N\}$, $\phi_i : \mathbb{X} \times \mathbb{U} \to \mathbb{R}$, $i = 1 \cdots N$,

$$\text{find the coefficients } \boldsymbol{\alpha}_i \in \mathbb{R}^n \text{ such that } \quad \frac{\mathrm{d}}{\mathrm{d}t}\boldsymbol{x}(t) = \boldsymbol{f}(t, \boldsymbol{x}, \boldsymbol{u}) \simeq \sum_{i=1}^{N} \boldsymbol{\alpha}_i \phi_i(\boldsymbol{x}, \boldsymbol{u}), \tag{2.36}$$

with ϕ_i denoting the $i-$th basis function. As $N \to \infty$, the approximation becomes more and more exact since higher-order terms are included to describe the dynamical flow; however the representation is less parsimonious and more complicated. Thus, for practical implementation, the expansion is truncated to a finite dimension N and the choice of that number N gives way to a trade-off between accuracy and complexity. Being able to identify an optimal value for N (if it even exists) is one of the most challenging - not to say *the* most challenging - area of research in the field of system identification. It is essentially equivalent to identify the dimension of a stable subspace onto which the dynamics evolves. Even though there is no automatic way to select N optimally beforehand (this usually involves trial and error), there is an educated way to expand the dynamics using specialized basis functions ϕ_i. When ϕ_i are linear functions of $\boldsymbol{x}$ and $\boldsymbol{u}$ (decoupled), the approximation in Eq. (2.36) is a linear approximation and linear system identification techniques can be applied (equivalent of system matrices A_c and B_c). When ϕ_i are bilinear functions of $\boldsymbol{x}$ and $\boldsymbol{u}$, the approximation in Eq. (2.36) is a bilinear approximation and some specialized techniques in bilinear system identification are required to optimally solve for $\boldsymbol{\alpha}_i$. When other types of basis functions ϕ_i are involved, the challenge is immense as nonlinear system identification methods are much more arduous and there is no existence of a common framework. Nonlinear system identification problem methods [2, 30–34, 39–45, 52, 53] differ in their choice of basis functions ϕ_i and their learning methodology. Oftentimes, an automatic way of expanding the dynamics is by selecting an orthogonal basis. For example, when a polynomial basis is chosen to represent the dynamical expansion, then it is called a Carleman linearization [62, 63], which has been used extensively in nonlinear system analysis [64–67]. Additional types of basis functions such as cosine and sine waves (Fourier series) or radial basis functions can be considered. In this dissertation, all three cases (linear, bilinear and nonlinear system identification methods) are be studied. Bilinear system identification techniques will be carefully studied in chapter 7 as the introduction of a control input $\boldsymbol{u}$ drastically increases the complexity of classical system identification techniques. The next section expands on the Carleman linearization and the Koopman operator theoretic approach which offers a major step forward nonlinear system identification. In the remaining of this chapter, we consider that there is no control input. In future chapters, a control action may be introduced, depending on the maturity of the method.

2.4 Carleman Linearization

If $\boldsymbol{x}^*$ is a nominal trajectory, the departure dynamics verifies the following differential equation (Taylor expansion):

$$\delta\dot{\boldsymbol{x}} = \sum_{p=1}^{\infty} \frac{1}{p!} \frac{\partial^p \boldsymbol{f}^*}{\partial \boldsymbol{x}^p} : \delta\boldsymbol{x}^{\otimes p} = \sum_{p=1}^{\infty} \frac{1}{p!} \frac{\partial^p \boldsymbol{f}^*}{\partial \boldsymbol{x}^p} : \underbrace{\delta\boldsymbol{x} \otimes \cdots \otimes \delta\boldsymbol{x}}_{p \text{ times}} = \frac{\partial \boldsymbol{f}^*}{\partial \boldsymbol{x}} : \delta\boldsymbol{x} + \frac{1}{2} \frac{\partial^2 \boldsymbol{f}^*}{\partial \boldsymbol{x}^2} : \delta\boldsymbol{x} \otimes \delta\boldsymbol{x} + \text{HOT}, \tag{2.37}$$

with $\delta\boldsymbol{x}(0) = \boldsymbol{x}_0^* - \boldsymbol{x}_0$ and

$$\frac{\partial^p \boldsymbol{f}^*}{\partial \boldsymbol{x}^p} = \left. \frac{\partial^p \boldsymbol{f}}{\partial \boldsymbol{x}^p} \right|_{\boldsymbol{x}=\boldsymbol{x}^*} \tag{2.38}$$

denotes the higher-order sensitivity tensor evaluated along the nominal trajectory $\boldsymbol{x} = \boldsymbol{x}^*$. The product $\otimes$ is defined as the outer product and : corresponds to the tensor product. This expansion is called a

Carleman linearization and considering terms up to order p leads to a Carleman linearization of order p. With Eq. (2.37), one has a global representation of the dynamics using tensors and tensors products but index notation can become convenient to describe high-order tensor algebra. Similarly as described in [68, 69], the departure dynamics can be written as

$$\delta\dot{x}_i(t) = \sum_{p=1}^{\infty} \frac{1}{p!} f^*_{i,r_1 r_2 \cdots r_p} \delta x_{r_1} \delta x_{r_2} \cdots \delta x_{r_p}, \quad 1 \le i \le n, \quad 1 \le r_j \le n, \quad 1 \le j \le p \tag{2.39}$$

where

$$f^*_{i,r_1 r_2 \cdots r_p} = \left. \frac{\partial^p f_i}{\partial x_{r_1} \partial x_{r_2} \cdots \partial x_{r_p}} \right|_{\boldsymbol{x}=\boldsymbol{x}^*} \tag{2.40}$$

is the i^{th} row of the higher-order sensitivity tensor (i.e. first dimension of the tensor). For certain classes of dynamical systems, these sensitivity tensors can become 0 quickly or at least be very sparse. Along with these higher-order sensitivity tensors, one can compute the state transition tensors

$$\Phi^*_{i,r_1 r_2 \cdots r_p} = \left. \frac{\partial^p x_i}{\partial x_{0 r_1} \partial x_{0 r_2} \cdots \partial x_{0 r_p}} \right|_{\boldsymbol{x}=\boldsymbol{x}^*} \tag{2.41}$$

that calculates the change in the solution of the nonlinear dynamical system with changes to initial conditions. The state trajectories $\delta x_i(t)$ are mapped from their initial conditions by these state transitions tensors:

$$\delta x_i(t) = \sum_{p=1}^{\infty} \frac{1}{p!} \Phi^*_{i,r_1 r_2 \cdots r_p} \delta x_{0 r_1} \delta x_{0 r_2} \cdots \delta x_{0 r_p}. \tag{2.42}$$

The time evolution of these state transition tensors is given by the differential equations [68, 69]

$$\dot{\Phi}^*_{i,j_1} = f^*_{i,r_1} \Phi^*_{r_1,j_1} \tag{2.43a}$$

$$\dot{\Phi}^*_{i,j_1 j_2} = f^*_{i,r_1} \Phi^*_{r_1,j_1 j_2} + f^*_{i,r_1 r_2} \Phi^*_{r_1,j_1} \Phi^*_{r_2,j_2} \tag{2.43b}$$

$$\begin{aligned} \dot{\Phi}^*_{i,j_1 j_2 j_3} = {} & f^*_{i,r_1} \Phi^*_{r_1,j_1 j_2 j_3} + f^*_{i,r_1 r_2} \left(\Phi^*_{r_1,j_1 j_2} \Phi^*_{r_2,j_3} + \Phi^*_{r_1,j_1 j_3} \Phi^*_{r_2,j_2} + \Phi^*_{r_1,j_2 j_3} \Phi^*_{r_2,j_1} \right) \\ & + f^*_{i,r_1 r_2 r_3} \Phi^*_{r_1,j_1} \Phi^*_{r_2,j_2} \Phi^*_{r_3,j_3} \end{aligned} \tag{2.43c}$$

$$\begin{aligned} \dot{\Phi}^*_{i,j_1 j_2 j_3 j_4} = {} & f^*_{i,r_1} \Phi^*_{r_1,j_1 j_2 j_3 j_4} \\ & + f^*_{i,r_1 r_2} \left(\Phi^*_{r_1,j_1 j_2 j_3} \Phi^*_{r_2,j_4} + \Phi^*_{r_1,j_1 j_2 j_4} \Phi^*_{r_2,j_3} + \Phi^*_{r_2,j_2 j_3 j_4} \Phi^*_{r_1,j_1} + \Phi^*_{r_1,j_1 j_3 j_4} \Phi^*_{r_2,j_2} \right) \\ & + f^*_{i,r_1 r_2} \left(\Phi^*_{r_1,j_1 j_2} \Phi^*_{r_2,j_3 j_4} + \Phi^*_{r_1,j_1 j_3} \Phi^*_{r_2,j_2 j_4} + \Phi^*_{r_2,j_2 j_3} \Phi^*_{r_1,j_1 j_4} \right) \\ & + f^*_{i,r_1 r_2 r_3} \left(\Phi^*_{r_1,j_1 j_2} \Phi^*_{r_2,j_3} \Phi^*_{r_3,j_4} + \Phi^*_{r_1,j_1 j_3} \Phi^*_{r_2,j_2} \Phi^*_{r_3,j_4} + \Phi^*_{r_2,j_2 j_3} \Phi^*_{r_1,j_1} \Phi^*_{r_3,j_4} \right. \\ & \left. + \Phi^*_{r_1,j_1 j_4} \Phi^*_{r_2,j_2} \Phi^*_{r_3,j_3} + \Phi^*_{r_1,j_1} \Phi^*_{r_2,j_2 j_4} \Phi^*_{r_3,j_3} + \Phi^*_{r_2,j_2} \Phi^*_{r_1,j_1} \Phi^*_{r_3,j_3 j_4} \right) \\ & + f^*_{i,r_1 r_2 r_3 r_4} \Phi^*_{r_1,j_1} \Phi^*_{r_2,j_2} \Phi^*_{r_3,j_3} \Phi^*_{r_4,j_4} \\ & \vdots \end{aligned} \tag{2.43d}$$

with initial conditions

$$\dot{\Phi}^*_{i,j}(t_0,t_0)=\delta_{i,j},\qquad \dot{\Phi}^*_{i,j_1j_2\cdots j_p}(t_0,t_0)=0,\ \forall p>1. \tag{2.44}$$

These equations can be derived by comparison of coefficients and by referring to the basic definition of the state transition tensors and taking the time derivatives. Notice that these differential equations are linear and can be solved sequentially, since they are in cascade form. This special structure allows one to build a linear operator that describes the evolution of the state transition matrix and higher-order state-transition tensors. To obtain an insight of that linear operator, and without loss of generality, consider a system of dimension $n=2$ (can be easily generalized for a system of arbitrary dimension). Eq. (2.43a) can be re-written as

$$\begin{bmatrix}\dot{\Phi}^*_{1,1}\\ \dot{\Phi}^*_{1,2}\\ \dot{\Phi}^*_{2,1}\\ \dot{\Phi}^*_{2,2}\end{bmatrix}=\begin{bmatrix}f^*_{1,1}&0&f^*_{1,2}&0\\ 0&f^*_{1,1}&0&f^*_{1,2}\\ f^*_{2,1}&0&f^*_{2,2}&0\\ 0&f^*_{2,1}&0&f^*_{2,2}\end{bmatrix}\begin{bmatrix}\Phi^*_{1,1}\\ \Phi^*_{1,2}\\ \Phi^*_{2,1}\\ \Phi^*_{2,2}\end{bmatrix}\Leftrightarrow \dot{\mathbf{\Phi}}^*=\begin{bmatrix}f^*_{1,1}I_2&f^*_{1,2}I_2\\ f^*_{2,1}I_2&f^*_{2,2}I_2\end{bmatrix}\mathbf{\Phi}^*\Leftrightarrow \dot{\mathbf{\Phi}}^*=\mathcal{G}_1\mathbf{\Phi}^* \tag{2.45}$$

with $\mathbf{\Phi}^*=\begin{bmatrix}\Phi^*_{1,1}&\Phi^*_{1,2}&\Phi^*_{2,1}&\Phi^*_{2,2}\end{bmatrix}^\top$ and a first-order linear operator as

$$\mathcal{G}_1=\begin{bmatrix}f^*_{1,1}I_2&f^*_{1,2}I_2\\ f^*_{2,1}I_2&f^*_{2,2}I_2\end{bmatrix}. \tag{2.46}$$

The subscript "1" means that the operator is built considering first-order dynamics only. Now, appending Eq. (2.43b) to Eq. (2.43a), it is possible to write the linear operator $\mathcal{G}_{1+2}$ as

$$\mathcal{G}_{1+2}=\begin{bmatrix}
f^*_{1,1}I_2&f^*_{1,2}I_2&0_{2\times4}&0_{2\times4}&0_{2\times4}&0_{2\times4}&0_{2\times4}&0_{2\times4}\\
f^*_{2,1}I_2&f^*_{2,2}I_2&0_{2\times4}&0_{2\times4}&0_{2\times4}&0_{2\times4}&0_{2\times4}&0_{2\times4}\\
0_{4\times2}&0_{4\times2}&f^*_{1,1}I_4&f^*_{1,2}I_4&f^*_{1,11}I_4&f^*_{1,12}I_4&f^*_{1,21}I_4&f^*_{1,22}I_4\\
0_{4\times2}&0_{4\times2}&f^*_{2,1}I_4&f^*_{2,2}I_4&f^*_{2,11}I_4&f^*_{2,12}I_4&f^*_{2,21}I_4&f^*_{2,22}I_4\\
0_{4\times2}&0_{4\times2}&0_{4\times4}&0_{4\times4}&2f^*_{1,1}I_4&f^*_{1,2}I_4&f^*_{1,2}I_4&0_{4\times4}\\
0_{4\times2}&0_{4\times2}&0_{4\times4}&0_{4\times4}&f^*_{2,1}I_4&(f^*_{1,1}+f^*_{2,2})I_4&0_{4\times4}&f^*_{1,2}I_4\\
0_{4\times2}&0_{4\times2}&0_{4\times4}&0_{4\times4}&f^*_{2,1}I_4&0_{4\times4}&(f^*_{1,1}+f^*_{2,2})I_4&f^*_{1,2}I_4\\
0_{4\times2}&0_{4\times2}&0_{4\times4}&0_{4\times4}&0_{4\times4}&f^*_{2,2}I_4&f^*_{2,2}I_4&2f^*_{2,2}I_4
\end{bmatrix} \tag{2.47a}$$

$$=\begin{bmatrix}\mathcal{G}_1&0_{4\times24}\\ 0_{24\times4}&\mathcal{G}_2\end{bmatrix}, \tag{2.47b}$$

with the vector $\mathbf{\Phi}^*$ now defined as

$$\mathbf{\Phi}^*=\begin{bmatrix}\Phi^*_{1,1}&\Phi^*_{1,2}&\Phi^*_{2,1}&\Phi^*_{2,2}&\Phi^*_{1,11}&\Phi^*_{1,12}&\Phi^*_{1,21}&\Phi^*_{1,22}\end{bmatrix} \tag{2.48}$$

$$\Phi^*_{2,11} \quad \Phi^*_{2,12} \quad \Phi^*_{2,21} \quad \Phi^*_{2,22} \quad \cdots \quad \Phi^*_{r_1,j_2}\Phi^*_{r_2,j_2} \quad \cdots \Big]^\top, \tag{2.49}$$

with $1 \leq r_1, j_1, r_2, j_2 \leq 2$, such that

$$\dot{\mathbf{\Phi}}^* = \mathcal{G}_{1+2}\mathbf{\Phi}^*. \tag{2.50}$$

The operator $\mathcal{G}_2$ in Eq. (2.47b) is defined as

$$\mathcal{G}_2 = \begin{bmatrix} f^*_{1,1}I_4 & f^*_{1,2}I_4 & f^*_{1,11}I_4 & f^*_{1,12}I_4 & f^*_{1,21}I_4 & f^*_{1,22}I_4 \\ f^*_{2,1}I_4 & f^*_{2,2}I_4 & f^*_{2,11}I_4 & f^*_{2,12}I_4 & f^*_{2,21}I_4 & f^*_{2,22}I_4 \\ 0_{4\times4} & 0_{4\times4} & 2f^*_{1,1}I_4 & f^*_{1,2}I_4 & f^*_{1,2}I_4 & 0_{4\times4} \\ 0_{4\times4} & 0_{4\times4} & f^*_{2,1}I_4 & (f^*_{1,1}+f^*_{2,2})I_4 & 0_{4\times4} & f^*_{1,2}I_4 \\ 0_{4\times4} & 0_{4\times4} & f^*_{2,1}I_4 & 0_{4\times4} & (f^*_{1,1}+f^*_{2,2})I_4 & f^*_{1,2}I_4 \\ 0_{4\times4} & 0_{4\times4} & 0_{4\times4} & f^*_{2,1}I_4 & f^*_{2,1}I_4 & 2f^*_{2,2}I_4 \end{bmatrix}. \tag{2.51}$$

Notice that the first- and second-order dynamics of the state transition tensors are decoupled (the matrix $\mathcal{G}_{1+2}$ is block diagonal) and that the previous operator $\mathcal{G}_1$ is contained in $\mathcal{G}_{1+2}$. The dimension of the operator $\mathcal{G}_{1+2}$ is $4 + 24 = 28$, much larger than the dimension of $\mathcal{G}_1$ $(= 4)$. Further, it is imaginable to expand this operator with higher-order dynamics following Eq. (2.43c), Eq. (2.43d) and so on. Expanding the vector $\mathbf{\Phi}^*$ appropriately, it is possible to write this linear relationship for dynamical systems of arbitrary dimension up to arbitrary order as

$$\dot{\mathbf{\Phi}}^* = \mathcal{G}\mathbf{\Phi}^*. \tag{2.52}$$

The objective here is not to explicitly come up with a succession of operators $\mathcal{G}_1, \mathcal{G}_2, \mathcal{G}_3$... etc, but rather to observe that one can re-write Eq. (2.43a)-Eq. (2.43d) (and more) into a matrix form and highlight the existence of a succession of linear operators (of increasing dimension) that describe the evolution of the higher-order state-transition tensors. Hence, the knowledge of higher-order sensitivities of Eq. (2.40) leads to the knowledge of the evolution of higher-order state-transition tensors and accordingly, the knowledge of the dynamical evolution of the state of the system through Eq. (2.42). The higher the expansion, the higher the accuracy as this is equivalent to increasing the value of N from Eq. (2.36). While the developments above are valid on a theoretical level, they are of little to no use in practice because one needs to have access to values of the state-transition tensors. Methods based on Carleman linearization corresponds to identifying higher-order sensitivities (or higher-order state-transition tensors in a discrete-time domain) [62, 63]. Nonetheless, the focal point of this discussion is the fact that there exists some sort of linear operator embedded with the evolution of the state transition matrix and higher-order state transition tensors. This development is studied further in the next section.

2.5 Introduction to the Koopman Operator

From now on, and without loss of generality, we will consider the nominal trajectory $\boldsymbol{x}^*$ to be zero which leads to $\delta\boldsymbol{x} = \boldsymbol{x}$.

Let's consider the operator $\mathcal{G}_1$ and let us re-write that operator as

$$\mathcal{G}_1 = \mathcal{K}_1 \otimes I_2, \tag{2.53}$$

with

$$\mathcal{K}_1 = \begin{bmatrix} f^*_{1,1} & f^*_{1,2} \\ f^*_{2,1} & f^*_{2,2} \end{bmatrix}. \tag{2.54}$$

From Eq. (2.37) or Eq. (2.39), notice that

$$\begin{bmatrix} \dot{x}_1 \\ \dot{x}_2 \end{bmatrix} = \begin{bmatrix} f^*_{1,1} & f^*_{1,2} \\ f^*_{2,1} & f^*_{2,2} \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} \Leftrightarrow \dot{\boldsymbol{x}} = \mathcal{K}_1 \boldsymbol{x} \tag{2.55}$$

with $\boldsymbol{x} = \begin{bmatrix} x_1 & x_2 \end{bmatrix}^\top$. Considering only first-order dynamics, the operator $\mathcal{K}_1$ is a linear operator that advances the state of the dynamical system forward in time. The coefficients of that operator are directly the first-order sensitivities. Similarly, considering the operator $\mathcal{K}_2$ as

$$\mathcal{G}_2 = \mathcal{K}_2 \otimes I_4, \tag{2.56}$$

the time evolution of the state of the dynamical system considering a second-order expansion is

$$\dot{\boldsymbol{x}} = \begin{bmatrix} \dot{x}_1 \\ \dot{x}_2 \\ \dot{x^2_1} \\ \dot{x_1 x_2} \\ \dot{x_2 x_1} \\ \dot{x^2_2} \end{bmatrix} = \begin{bmatrix} f^*_{1,1} & f^*_{1,2} & f^*_{1,11} & f^*_{1,12} & f^*_{1,21} & f^*_{1,22} \\ f^*_{2,1} & f^*_{2,2} & f^*_{2,11} & f^*_{2,12} & f^*_{2,21} & f^*_{2,22} \\ 0 & 0 & 2f^*_{1,1} & f^*_{1,2} & f^*_{1,2} & 0 \\ 0 & 0 & f^*_{2,1} & f^*_{1,1} + f^*_{2,2} & 0 & f^*_{1,2} \\ 0 & 0 & f^*_{2,1} & 0 & f^*_{1,1} + f^*_{2,2} & f^*_{1,2} \\ 0 & 0 & 0 & f^*_{2,1} & f^*_{2,1} & 2f^*_{2,2} \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x^2_1 \\ x_1 x_2 \\ x_2 x_1 \\ x^2_2 \end{bmatrix}, \tag{2.57}$$

such that $\dot{\boldsymbol{x}} = \mathcal{K}_2 \boldsymbol{x}$ with

$$\mathcal{K}_2 = \begin{bmatrix} f^*_{1,1} & f^*_{1,2} & f^*_{1,11} & f^*_{1,12} & f^*_{1,21} & f^*_{1,22} \\ f^*_{2,1} & f^*_{2,2} & f^*_{2,11} & f^*_{2,12} & f^*_{2,21} & f^*_{2,22} \\ 0 & 0 & 2f^*_{1,1} & f^*_{1,2} & f^*_{1,2} & 0 \\ 0 & 0 & f^*_{2,1} & f^*_{1,1} + f^*_{2,2} & 0 & f^*_{1,2} \\ 0 & 0 & f^*_{2,1} & 0 & f^*_{1,1} + f^*_{2,2} & f^*_{1,2} \\ 0 & 0 & 0 & f^*_{2,1} & f^*_{2,1} & 2f^*_{2,2} \end{bmatrix}. \tag{2.58}$$

Notice that the symmetric basis functions $x_1 x_2$ and $x_2 x_1$ have been written separately. This time, second-order dynamics are considered and the operator $\mathcal{K}_2$ is a linear operator that advances the state

and second-order polynomial combinations of the state forward in time. This operator comprises the coefficients of the first- and second-order sensitivities, rearranged. By extension,

$$\mathcal{G}_n = \mathcal{K}_n \otimes I_{2^n}, \tag{2.59}$$

and as $n \to \infty$, both $\mathcal{G}_n$ and $\mathcal{K}_n$ become infinite dimensional operators, $\mathcal{G}$ and $\mathcal{K}$. Though the operator $\mathcal{G}$ doesn't have a name, the operator $\mathcal{K}$ is well known by the community and is called a Koopman operator. While more formal definitions will be given in the next two sections, the main idea behind the Koopman theory [47, 48] is to lift the nonlinear dynamics into a higher dimensional space where the evolution of the flow of the system can be linear. With a proper choice of basis functions (polynomial bases here), the information contained in the Koopman operator is exactly equivalent to the information contained in a Taylor series expansion (or Carleman linearization) of the nonlinear flow about a nominal trajectory with the order of expansion being equal to the order of lifting functions. The next two sections will provide a more academic definition of the Koopman operator.

2.6 Continuous-Time Koopman Operator

This section aims to provide a formal definition and discuss the basic properties of the continuous-time Koopman operator. Consider the autonomous nonlinear dynamical system

$$\frac{\mathrm{d}}{\mathrm{d}t}\boldsymbol{x}(t) = \boldsymbol{f}(\boldsymbol{x}(t)), \quad \boldsymbol{x}(0) = \boldsymbol{x}_0, \tag{2.60}$$

where $t \in \mathbb{R}$, $\boldsymbol{x} \in \mathbb{X} \subseteq \mathbb{R}^n$, $\boldsymbol{f} : \mathbb{X} \to \mathbb{R}^n$. Assuming the system has a unique solution, and consequently its inverse, existing over any time interval, Eq. (2.60) has an associated continuous-time flow map $\boldsymbol{F}_c$ defined in Eq. (2.61a) and satisfies group properties defined in Eq. (2.61b), where $\boldsymbol{F}_c^t : \mathbb{X} \to \mathbb{X}$.

$$\boldsymbol{x}(t) = \boldsymbol{F}_c^t(\boldsymbol{x}_0) = \boldsymbol{x}_0 + \int_0^t \boldsymbol{f}(\boldsymbol{x}(t))dt \tag{2.61a}$$

$$\forall t, s \in \mathbb{R}, \quad \boldsymbol{F}_c^t \circ \boldsymbol{F}_c^s(\cdot) = \boldsymbol{F}_c^{t+s}(\cdot), \quad \boldsymbol{F}_c^0(\cdot) = \mathcal{I}. \tag{2.61b}$$

The continuous-time Koopman operator $\mathcal{K}_c^t$ is defined generally as an infinite-dimensional linear operator that advances some complex-valued measurement function $\boldsymbol{\chi} : \mathbb{X} \to \mathbb{C}$ through

$$\boldsymbol{\chi}(\boldsymbol{x}(t)) = \mathcal{K}_c^t \boldsymbol{\chi}(\boldsymbol{x}_0) = \boldsymbol{\chi} \circ \boldsymbol{F}_c^t(\boldsymbol{x}_0) \tag{2.62}$$

where $\mathcal{K}_c^t : \mathcal{F} \to \mathcal{F}$, and $\chi(\boldsymbol{x})$ is a measurement function in the Hilbert space $\mathcal{F}$. Eq. (2.62) defines the Koopman operator as a unitary and possibly infinite-dimensional linear operator acting on the measurement functions. It is also important to note that the Koopman operator is a member of the continuous one-parameter unitary group of operators $\{\mathcal{K}_c^t\}_{t\in\mathbb{R}}$ generated by the *Koopman generator* $\mathcal{K}_c$ defined by Eq. (2.63a), provided the limit exists [70]. The Koopman operator in this setting is also said to satisfy the group properties defined in Eq. (2.63b),

$$\mathcal{K}_c \boldsymbol{\chi} = \lim_{\Delta t \to 0} \frac{\mathcal{K}_c^{\Delta t} \boldsymbol{\chi} - \boldsymbol{\chi}}{\Delta t} = \lim_{\Delta t \to 0} \frac{\boldsymbol{\chi} \circ \boldsymbol{F}_c^{\Delta t} - \boldsymbol{\chi}}{\Delta t}, \tag{2.63a}$$

$$\forall t, s \in \mathbb{R}, \quad \mathcal{K}_c^t \circ \mathcal{K}_c^s(\cdot) = \mathcal{K}_c^{t+s}(\cdot), \quad \mathcal{K}_c^0(\cdot) = \mathcal{I}. \tag{2.63b}$$

If the measurement functions are differentiable, Eq. (2.63a) indicates that the Koopman generator applied to a measurement function $\boldsymbol{\chi}$ is the time derivative of the measurement, $\dot{\boldsymbol{\chi}}$, and

$$\dot{\boldsymbol{\chi}} = \frac{\mathrm{d}}{\mathrm{d}t}\boldsymbol{\chi}(\boldsymbol{x}(t)) = \lim_{\Delta t \to 0} \frac{\boldsymbol{\chi}(\boldsymbol{x}(t+\Delta t)) - \boldsymbol{\chi}(\boldsymbol{x}(t))}{\Delta t} = \lim_{\Delta t \to 0} \frac{\mathcal{K}_c^{\Delta t}\boldsymbol{\chi} - \boldsymbol{\chi}}{\Delta t} = \mathcal{K}_c\boldsymbol{\chi}. \tag{2.64}$$

The central observation is that Eq. (2.60) and Eq. (2.64) are equivalent and there has been a trading between nonlinear dynamics in a finite-dimensional space and linear dynamics in a potentially infinite-dimensional lifted space. Also note that the problem of finding the operator $\mathcal{K}_c$ in Eq. (2.64) is equivalent in reformulating Eq. (2.36) as

$$\text{find the coefficients } \alpha_{i,j} \in \mathbb{R}^n \text{ such that} \quad \frac{\mathrm{d}}{\mathrm{d}t}\chi^i(\boldsymbol{x}) \simeq \sum_{i=1}^{N} \alpha_{i,j}\chi^j(\boldsymbol{x}), \tag{2.65}$$

with $\boldsymbol{\chi} = \begin{bmatrix}\chi^1 & \chi^2 & \cdots & \chi^N\end{bmatrix}^\top$ and $\alpha_{i,j}$ are the coefficients of the operator $\mathcal{K}_c$. The Koopman operator is shown to fully capture all properties of the underlying nonlinear dynamical system provided that the state vector $\boldsymbol{x}$ is observable from the lifted space of measurements $\boldsymbol{\chi}$ [54, 55]. Providing that one has access to that linear operator $\mathcal{K}_c$, this allows one to obtain precise predictions of a nonlinear dynamical system as the output of a linear dynamical system (see Table 2.1. This approach, called the Koopman operator theoretic approach, does nothing but augment the problem defined in Eq. (2.36) with an additional set of differential equations.

Table 2.1: Comparing dynamics in original coordinate system and augmented space of measurements

Dynamics in the original coordinate system	Dynamics in the lifted space of measurement $\mathcal{F}$
Nonlinear: $\dot{\boldsymbol{x}}(t) = \boldsymbol{f}(\boldsymbol{x}(t))$	Linear: $\dot{\boldsymbol{\chi}}(\boldsymbol{x}(t)) = \mathcal{K}_c\boldsymbol{\chi}(\boldsymbol{x}(t))$

As a quick and illustrative example, consider a single fixed-point nonlinear dynamical system introduced in [57]:

$$\dot{x} = \mu x \tag{2.66a}$$

$$\dot{y} = \lambda(y - x^2) \tag{2.66b}$$

Notice that one can obtain the following analytical expression for the continuous time Koopman operator by appending the state vector with x^2 as measurements, i.e., $\boldsymbol{\chi} = \begin{bmatrix}x & y & x^2\end{bmatrix}^\top$:

$$\dot{\boldsymbol{\chi}} = \begin{bmatrix}\dot{x} \\ \dot{y} \\ \dot{x^2}\end{bmatrix} = \begin{bmatrix}\mu x \\ \lambda(y - x^2) \\ 2x.\mu x\end{bmatrix} = \begin{bmatrix}\mu & 0 & 0 \\ 0 & \lambda & -\lambda \\ 0 & 0 & 2\mu\end{bmatrix}\begin{bmatrix}x \\ y \\ x^2\end{bmatrix} = \mathcal{K}_c\boldsymbol{\chi}. \tag{2.67}$$

This synthetic example highlights a case where the associated Koopman operator $\mathcal{K}_c$ is of finite dimension. As explained later in this chapter, the simple representation of the observable $\boldsymbol{\chi}$ may not

be as straightforward for every system. In other words, identifying the correct measurement functions may not be simple and may become arbitrarily complex once iterated through the dynamics. Along with the discussion of the continuous-time Koopman operator, it is natural to introduce its discrete-time counterpart. This is what is presented in the next section.

2.7 Discrete-Time Koopman Operator

First, let $\boldsymbol{F}$ be the discrete-time flow of the dynamical system that maps the state from one time to the other, with a time-step size Δt. Integrating these trajectories through the equations of motion (Eq. (2.60)) leads to the time update equation

$$\boldsymbol{F}(\boldsymbol{x}(t_0)) = \boldsymbol{x}(t_0 + \Delta t) = \boldsymbol{x}(t_0) + \int_{t_0}^{t_0+\Delta t} \boldsymbol{f}(\boldsymbol{x}(\tau))\mathrm{d}\tau, \tag{2.68}$$

or, introducing $\boldsymbol{x}_k = \boldsymbol{x}(k\Delta t)$,

$$\boldsymbol{x}_{k+1} = \boldsymbol{F}(\boldsymbol{x}_k). \tag{2.69}$$

If $\boldsymbol{\chi}_k = \boldsymbol{\chi}(\boldsymbol{x}_k)$ represents a set of measurements of the state vector $\boldsymbol{x}$ in a Hilbert space $\mathcal{F}$ of functions, then the *possibly* infinite-dimensional discrete Koopman operator $\mathcal{K}$ is a linear operator for the transition of these measurements forward in time, i.e.,

$$\boldsymbol{\chi}_{k+1} = \mathcal{K}\boldsymbol{\chi}_k, \tag{2.70}$$

where

$$\boldsymbol{\chi}_k = \begin{bmatrix} \chi_k^1 \\ \chi_k^2 \\ \vdots \end{bmatrix} = \boldsymbol{\chi}(\boldsymbol{x}_k) \tag{2.71}$$

and each $\chi_k^i = \chi^i(\boldsymbol{x}_k)$, $i = 1, 2, \ldots$, is assumed to be observable in $\mathcal{F}$. Note that Eq. (2.70) provides an infinite dimensional linear time-invariant (LTI) system version of the nonlinear flow of Eq. (2.60) in the measurement space $\mathcal{F}$. Since $\boldsymbol{\chi}_{k+1} = \boldsymbol{\chi}(\boldsymbol{x}_{k+1}) = \boldsymbol{\chi}(\boldsymbol{F}(\boldsymbol{x}_k))$, one can write

$$\mathcal{K}\boldsymbol{\chi}_k = \boldsymbol{\chi}_k \circ \boldsymbol{F}. \tag{2.72}$$

The continuous-time and discrete-time Koopman operators are connected by

$$\mathcal{K} = \mathcal{K}_c^{\Delta t}. \tag{2.73}$$

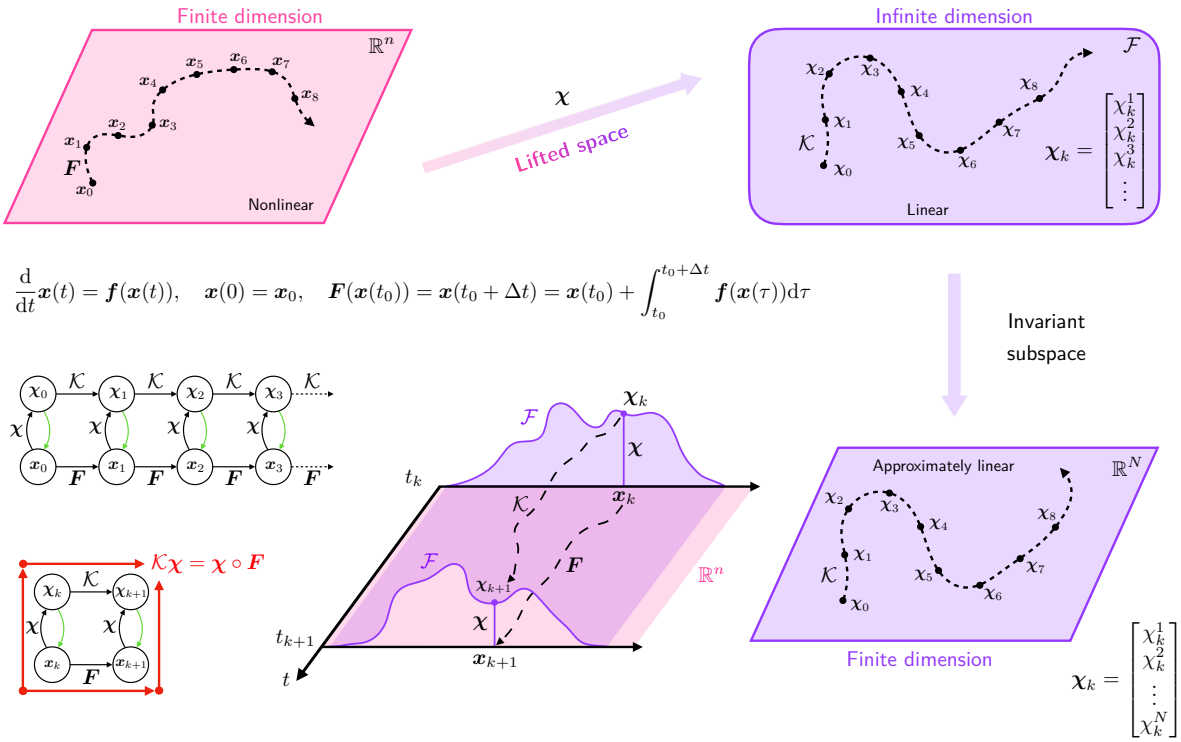


Figure 2.3: Overview of the Koopman operator theoretic framework

In practice, the measurement vector $\boldsymbol{\chi}_k$ is truncated to a finite dimension N:

$$\boldsymbol{\chi}_k = \begin{bmatrix} \chi_k^1 \\ \chi_k^2 \\ \vdots \\ \chi_k^N \end{bmatrix}. \tag{2.74}$$

Figure 2.3 offers a general view of the Koopman operator theoretic framework.

2.8 Challenges to Find a Koopman Operator

As we started to explain before, the major challenge in the field of system identification is to be able to identify a stable subspace onto which the dynamics evolves. As an extension, the core challenge of the Koopman operator theoretic approach is to specify (directly or indirectly through decompositions) the Hilbert space of measurement functions of the state of the system. A simple yet illustrative example shows a drawback associated with a naive representation of the infinite dimensional Koopman operator for a simple chaotic system. Consider the dynamical system given by:

$$\dot{x} = x^2. \tag{2.75}$$

If the state x is considered as a measurement, so does x^2 since $\dot{x} = x^2$. If x^2 is considered as a measurement, so does x^3 since $\dot{x^2} = 2\dot{x}x = 2x^3$. A cubic polynomial term is required to advance x^2 and a quartic term is required to advance x^3. And this keeps going with all monomials of arbitrary degree. Eq. (2.64) would now be written as

$$\dot{\boldsymbol{\chi}} = \begin{bmatrix} \dot{x} \\ \dot{x^2} \\ \dot{x^3} \\ \dot{x^4} \\ \vdots \end{bmatrix} = \begin{bmatrix} 0 & 1 & 0 & 0 & \cdots \\ 0 & 0 & 2 & 0 & \cdots \\ 0 & 0 & 0 & 3 & \cdots \\ 0 & 0 & 0 & 0 & \cdots \\ \vdots & \vdots & \vdots & \vdots & \ddots \end{bmatrix} \begin{bmatrix} x \\ x^2 \\ x^3 \\ x^4 \\ \vdots \end{bmatrix} = \mathcal{K}_c \boldsymbol{\chi}, \qquad \text{with } \boldsymbol{\chi} = \begin{bmatrix} x \\ x^2 \\ x^3 \\ x^4 \\ \vdots \end{bmatrix}. \tag{2.76}$$

The Koopman representation of the dynamics is now infinite-dimensional. In order to obtain a finite representation of the dynamics in the Koopman framework, one will need specialized bases. For a given dynamical system, there might be these special coordinates where the dynamics are linear, but there is no guarantee one is able to find them, especially for more complex dynamical systems. For this scalar case, this special basis actually exists and is given by $y = e^{-1/x}$ such that

$$\frac{\mathrm{d}}{\mathrm{d}t}\chi = x^{-2}e^{-1/x}\dot{x} = e^{-1/x} = \chi, \tag{2.77}$$

and $\mathcal{K}_c = 1$. Instead of capturing the evolution of all measurement functions in a Hilbert space, applied Koopman analysis attempts to identify key measurement functions that evolve linearly with the flow of the dynamics. Eigenfunctions of the Koopman operator provide just such a set of special measurements that behave linearly in time. In fact, a primary motivation to adopt the Koopman framework is the ability to simplify the dynamics through the eigen-decomposition of the operator, which is still a very active area of research. Most of the time, researchers assume a set of basis functions $\{\phi_1, \phi_2, \cdots, \phi_N\}$, $\phi_i : \mathbb{X} \to \mathbb{R}$, $i = 1 \cdots N$, as an augmented set of measurements to represent the dynamics. One could argue that given a sufficiently large amount of data, it is possible to find the operator using simple least-squares regression techniques [53]. However, long-term prediction is nearly impractical as error in propagation builds up at a fast pace. To illustrate that phenomenon, consider the Duffing equation

$$\dot{x}_1 = x_2 \tag{2.78}$$

$$\dot{x}_2 = x_1 - 4x_1^3 \tag{2.79}$$

A 6-th order expansion is considered and 300 training trajectories are generated around a nominal starting point with covariance $0.04I_2$. Exact values of the state and state-derivatives are used to formulate the least-squares problem. Figure 2.4 shows the pure propagation from one of the training points ($\boldsymbol{x}_0 = \begin{bmatrix} 0.5 & 0.45 \end{bmatrix}^\top$) and it is clearly apparent that any error introduced in higher-order basis functions ends up hurting the propagation process at future time-steps. Performing a least-squares fit on an augmented observable vector yields only poor results as there is a slow convergence to the real dynamics.

More recently, subspace realization methods have offered a way to identify a Koopman operator from data. The theory has been applied for uncontrolled [54, 55] and controlled systems [56, 57] with

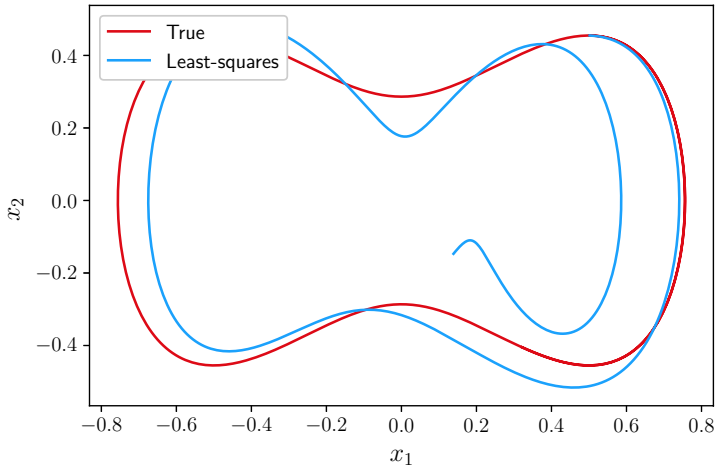


Figure 2.4: True vs. least-squares approximation of a training trajectory

promising results using popular subspace realization methods such as dynamic mode decomposition (DMD) and its extensions [58]. The resulting linear operator is a local approximator of the nonlinear dynamical system valid in the neighborhood of a nominal point and the domain of validity of this local linear approximation improves as the dimension of the lifting space is increased. However, one may need a very large dimensional lifting space to accurately capture the flow of the underlying nonlinear system.

One of the alternatives to improve the validity region of the Koopman operator and curtail the dimension of the lifting space is to consider the linearization of the nonlinear flow about a nominal trajectory of the nonlinear system rather than a nominal point. The linearization about a nominal trajectory leads to a linear time-varying (LTV) system as opposed to a linear time-invariant (LTI) system for the conventional Koopman operator. However, LTV systems exhibit distinct properties, as compared to the shift invariance exhibited by LTI systems.

2.9 Conclusion

This chapter has introduced some of the mathematical foundations useful to fully apprehend the scope of this dissertation. More than a handful of notations, this chapter aims to establish the mindset around which this research work has been conducted. From an elementary expansion of the nonlinear dynamics into a set of basis functions and their amplitude, the purpose of this dissertation is to produce methods and algorithms to accurately characterize the dynamics given some time history of observable data. The next two chapters will highlight some of the techniques developed to achieve this goal, either from a continuous or discrete-time point of view.

Chapter 3
Continuous-time Koopman Operator

3.1 Introduction

There is no doubt that the choice of basis functions significantly influences the approximation accuracy and complexity of the model. For many known physical systems, the nonlinearities can be represented by only a few terms with a judicious choice of basis functions. In this respect, many efforts have focused on adapting the architecture of the network by selecting appropriate models from a pre-defined dictionary of models [37, 38, 49–51]. However, this leads to an exhaustive search algorithm to learn the appropriate basis functions to represent the network dynamics.

More recently, advances in compressed sensing and sparse regression have been exploited to learn appropriate basis functions from an over-complete dictionary of basis functions without performing an exhaustive search [39–41]. To determine the form of the dynamics from data, these methods collect a time-history of the state and its derivative sampled at a number of instances in time. In the case where the derivative is not part of the measurement model, they construct state derivative information by finite difference methods which make derivative calculations susceptible to noise in measurements. After carefully arranging relevant basis functions in a dictionary, a linear least-squares problem is posed to find unknown coefficients of the basis functions. To enforce sparsity, an iterative least-squares problem is solved where the dictionary size is reduced by removing basis functions whose amplitudes are lower than a prescribed threshold. Although this sparse representation through the iterative least-squares problem guarantees the balance between model complexity and accuracy, the resulting algorithm is susceptible to noise in state measurement. In [42, 43], an approach named subsampling-based threshold sparse Bayesian regression (Subtsbr) is presented to accomodate high noise in the measurments for states and state derivatives. A Galerkin formulation that involves projecting the errors on a set of basis functions known as test functions is considered in [44] to avoid estimating time derivatives of the state variables. While this formulation provides better results in the presence of noise, the choice of test functions severely affects the performance of the algorithm. The formulation in [45] considers a direct integral form of the dynamics for first order systems in conjunction with a regularized ℓ_1 optimization problem to find the appropriate basis functions to approximate the unknown system dynamics.

As an extension of these recent formulations, the main objective of this chapter is to consider an integral form of the differential equation to estimate unknown amplitudes of basis functions with only state and input measurements for a first-order system [52]. Rather than a pure integral form considered in [45], a low-pass filter is designed to avoid infinite response at low frequencies or large time intervals. The secondary objective of this work is to generalize this approach for identification of second- and higher-order systems with only position-level measurement data and systems with a control input. Furthermore, the iterative least-squares problem is replaced with an iterative regularized ℓ_1 optimization problem as used in our earlier work on sparse collocation methods for optimal feedback control laws [71]. This guarantees that the sparse solution is found with high probability using convex optimization methods. The methodology is validated by considering two nonlinear oscillators with or without noisy measurements and on a second-order system involving a central force field. Comparison between a deep-learning approach and a sparse solution is presented at the end of the chapter.

The structure of this chapter is as follows: Section 3.2 provides a mathematical treatment of the system identification problem while Section 3.3 provides the derivation of the developed methodology for first and second order systems. Section 3.4 shows the efficacy of the developed approach by considering two nonlinear oscillator problems and identification of Newton's law of gravitation through satellite motion data. The chapter concludes with summary of results in Section 3.5 and generalization of the developed approach for the generic-order systems in Appendix A.

3.2 Problem Statement

This chapter aims to provide an extended, unified and automatic framework to discover the governing equations underlying a dynamical system simply from data measurements, based on the assumption that the structure of the dynamical model is governed by only a few important terms. Let us consider a general dynamical model similar to Eq. (2.35) with the addition of an affine control input

$$\dot{\boldsymbol{x}}(t) = \boldsymbol{f}(\boldsymbol{x}(t)) + G\boldsymbol{u}(t), \tag{3.1}$$

where $\boldsymbol{x}(t) \in \mathbb{R}^n$ represents the state of the system and $\boldsymbol{u}(t) \in \mathbb{R}^r$ the control action at time t and $G \in \mathbb{R}^{n\times r}$ is the constant input-influence matrix. The unknown nonlinear function $\boldsymbol{f} : \mathbb{R}^n \to \mathbb{R}^n$ represents the dynamics constraints for the system. The goal here is to find the structure of the unknown function $\boldsymbol{f}$ given the time history of $\boldsymbol{x}(t)$ and $\boldsymbol{u}(t)$ and constant control influence matrix, G. Considering a set of basis functions $\{\phi_j\}_{j=1\ldots\infty}, \phi_j : \mathbb{R}^n \to \mathbb{R}$, $\boldsymbol{f}$ can be approximated as a linear combination of these basis functions [2, 32]

$$\boldsymbol{f}(\boldsymbol{x}) = \sum_{j=1}^{\infty} \boldsymbol{\alpha}_j \phi_j(\boldsymbol{x}), \tag{3.2}$$

where $\{\boldsymbol{\alpha}_j\}_{j=1..\infty}, \boldsymbol{\alpha}_j \in \mathbb{R}^n$, is a set of unknown coefficients. There are infinitely many choices for basis functions such as polynomials, trigonometric functions, radial basis functions, etc. As we started to discuss in the previous chapter, a central difficulty in learning $\boldsymbol{f}$ lies in choosing appropriate basis

functions and the choice of basis functions unfortunately depends on the characteristics of an unknown input-output map. In an appropriate basis, the equations are often sparse in nature and the resulting model is parsimonious, i.e., a very few of $\boldsymbol{\alpha}_j$ are non-zero. It is desired to choose the basis functions that allow $\boldsymbol{f}$ to be represented with as few terms in Eq. (3.2) as possible [39, 40]. In this respect, the summation in Eq. (3.2) is taken over a finite number of N basis functions:

$$\boldsymbol{f}(\boldsymbol{x}) \approx \sum_{j=1}^{N} \boldsymbol{\alpha}_j \phi_j(\boldsymbol{x}), \tag{3.3}$$

or equivalently

$$\boldsymbol{f}(\boldsymbol{x}) \approx \boldsymbol{\alpha}\boldsymbol{\phi}(\boldsymbol{x}), \tag{3.4}$$

where $\boldsymbol{\alpha} = \begin{bmatrix} \boldsymbol{\alpha}_1 & \boldsymbol{\alpha}_2 & \cdots & \boldsymbol{\alpha}_N \end{bmatrix} \in \mathbb{R}^{n \times N}$ and $\boldsymbol{\phi}(\boldsymbol{x}) = \begin{bmatrix} \phi_1(\boldsymbol{x}) & \phi_2(\boldsymbol{x}) & \cdots & \phi_N(\boldsymbol{x}) \end{bmatrix}^T \in \mathbb{R}^N$. The objective is to search a given handbook of known functions for a set that best represents the given data. Recent advances in compressed sensing and sparse regression [39, 40, 45, 72] can be exploited to learn these few non-zero terms from an over-complete dictionary of basis functions without performing a combinatorially intractable brute-force search. The next section provides the mathematical details corresponding to finding a sparse solution for $\boldsymbol{\alpha}$.

3.3 General Methodology

As stated in the previous section, the objective is to find a sparse solution for the $\boldsymbol{\alpha}$ given the time histories of $\boldsymbol{x}(t)$ and $\boldsymbol{u}(t)$. If the time history of $\dot{\boldsymbol{x}}(t)$ is known, then one can solve for the unknown coefficients $\boldsymbol{\alpha}$ through a least-squares solution. In Refs. [39–43], different iterative algorithms are proposed to find the best set of basis functions to represent $\boldsymbol{f}$ accurately. If the time-derivative of $\boldsymbol{x}(t)$ is not available, then one needs to reconstruct this information through time history knowledge of $\boldsymbol{x}(t)$ via finite difference. Such an approach is sensitive to noise in measurements of $\boldsymbol{x}(t)$ [40]. An integral formulation is considered in [45] to avoid the finite difference to reconstruct the state time-derivative information. Though an integral formulation attenuates the high-frequency content, it provides limited attenuation at low frequencies and can lead to infinite response for a long time integration. In this section, an alternate formulation is presented to find the unknown coefficients $\boldsymbol{\alpha}_j$ without any knowledge of $\dot{\boldsymbol{x}}(t)$. A low-pass filter design is considered to provide short memory and better control over attenuation at different frequencies. First, this formulation is presented for first-order systems and then generalized for second-order systems.

3.3.1 First-Order Systems

Considering a first-order system, Eq. (3.1) can be rewritten as

$$\dot{\boldsymbol{x}}(t) = \sum_{j=1}^{N} \boldsymbol{\alpha}_j \phi_j(\boldsymbol{x}) + G\boldsymbol{u}(t) \tag{3.5}$$

and its component-wise unilateral Laplace transformation is

$$X_i^{0\mathrm{f}}(s) = sX_i(s) - x_i(0) = \sum_{j=1}^{N} \alpha_{i,j}\Phi_j(s) + \sum_{k=1}^{r} G_{i,k}U_k(s), \quad i = 1 \dots n, \tag{3.6}$$

where $\mathcal{L}\{x_i(t)\} = X_i(s), \mathcal{L}\{\phi_j(\boldsymbol{x})\} = \Phi_j(s), \mathcal{L}\{u_k(t)\} = U_k(s)$ and s is the Laplace variable. Note that $\phi_j(\boldsymbol{x})$ is an implicit function of time and hence, the Laplace transform of this time-varying signal can be considered. From now on, capital letters are used for functions in the Laplace domain. $X_i^{0\mathrm{f}}(s)$ is the original filtered signal (filtered 0-th time). For $\lambda_1 \in \mathbb{R}_+^*$, let us consider the Laplace filtering operator

$$\begin{array}{rccc} \mathcal{I}_{\lambda_1}: & \mathbb{R} & \to & \mathbb{R}, \\ & \bullet & \mapsto & \dfrac{\bullet}{s+\lambda_1}. \end{array} \tag{3.7}$$

Applying the operator to the signal $X_i^{0\mathrm{f}}$ yields

$$\begin{aligned} X_i^{1\mathrm{f}}(s) = \mathcal{I}_{\lambda_1}\left(X_i^{0\mathrm{f}}(s)\right) = \frac{sX_i(s) - x_i(0)}{s+\lambda_1} &= \sum_{j=1}^{N} \alpha_{i,j}\frac{\Phi_j(s)}{s+\lambda_1} + \sum_{k=1}^{r} G_{i,k}\frac{U_k(s)}{s+\lambda_1} \\ &= \sum_{j=1}^{N} \alpha_{i,j}\Phi_j^{1\mathrm{f}}(s) + \sum_{k=1}^{r} G_{i,k}U_k^{1\mathrm{f}}(s) \end{aligned} \tag{3.8}$$

where

$$\Phi_j^{1\mathrm{f}}(s) = \frac{\Phi_j(s)}{s+\lambda_1} \quad \text{and} \quad U_k^{1\mathrm{f}}(s) = \frac{U_k(s)}{s+\lambda_1}. \tag{3.9}$$

The superscript $^{1\mathrm{f}}$ corresponds to the filtered signal. Note that Eq. (3.8) corresponds to the integral form of Eq. (3.1) when $\lambda_1 = 0$. To this end, the filtered integral formulation is a generalization of the pure integral approach, that can have an infinite response at either very low frequency or for integration over a long time. The use of the filter allows the analyst to introduce fading memory in the approximation process, that allows specific control of the signal-to-noise ratio of the signals used in the identification process. Note that this section describes a method that uses a filter to implement a system that approximates the derivatives at low frequencies. This filter realization, however, could be generalized with larger degrees of design freedom. For instance, writing $\phi^{1f} = F(s)\phi(\boldsymbol{x}(t))$ with $F(s)$ a general strictly passive real filter of arbitrary order would allow one to extract the desirable signal properties of importance to the physics of the problem. Though the development presented in this dissertation assumes λ_1 to be a scalar quantity, one can ideally use different filters for different components of the state vector with λ_1 being a vector quantity. Now, adding and subtracting $\lambda_1 X_i(s)$ to Eq. (3.8) leads to

$$X_i^{1\mathrm{f}}(s) = \frac{(s+\lambda_1)X_i(s) - \lambda_1 X_i(s) - x_i(0)}{s+\lambda_1} = X_i(s) + \frac{-\lambda_1 X_i(s) - x_i(0)}{s+\lambda_1} = X_i(s) + Y_{i,1}(s), \tag{3.10}$$

where

$$Y_{i,1}(s) = \frac{-\lambda_1 X_i(s) - x_i(0)}{(s+\lambda_1)}. \tag{3.11}$$

Eq. (3.11) can be rewritten as

$$sY_{i,1}(s) + x_i(0) = -\lambda_1 Y_{i,1}(s) - \lambda_1 X_i(s) \tag{3.12}$$

and its inverse Laplace transform yields

$$\dot{y}_{i,1}(t) = -\lambda_1 y_{i,1}(t) - \lambda_1 x_i(t), \quad y_{i,1}(0) = -x(0). \tag{3.13}$$

Similarly, the inverse Laplace transforms for $\Phi_j^{1\mathrm{f}}$ and $U_k^{1\mathrm{f}}$ yield the corresponding first-order ODEs:

$$\dot{\phi}_j^{1\mathrm{f}}(t) = -\lambda_1 \phi_j^{1\mathrm{f}}(t) + \phi_j(\boldsymbol{x}), \quad \phi_j^{1\mathrm{f}}(0) = 0, \quad j = 1, 2, \ldots, N, \tag{3.14}$$

$$\dot{u}_k^{1\mathrm{f}}(t) = -\lambda_1 u_k^{1\mathrm{f}}(t) + u_k(t), \quad u_k^{1\mathrm{f}}(0) = 0, \quad k = 1, 2, \ldots, r. \tag{3.15}$$

For $\lambda_1 = 0$, then ϕ_j^{1f} and u_k^{1f} result in the time integration of $\phi_j(x(t))$ and $u_k(t)$. For $\lambda_1 > 0$, these equations correspond to a stable linear system of equations. By appropriately choosing the λ_1, one can control how quickly the initial condition response of these equations will go to zero. Finally, the final equation in the time domain can be written as

$$x_i^{1\mathrm{f}}(t) = x_i(t) + y_{i,1}(t) = \sum_{j=1}^{N} \alpha_{i,j} \phi_j^{1\mathrm{f}}(t) + \sum_{k=1}^{r} G_{i,k} u_k^{1\mathrm{f}}(t). \tag{3.16}$$

3.3.1.1 Least-Squares Solution

Note that the aforementioned equation provides a linear relationship between filtered signals $x_i^{1\mathrm{f}}(t)$, $u_k^{1\mathrm{f}}(t)$ and $\phi_j^{1\mathrm{f}}(t)$. Furthermore, these filtered signals can be constructed directly from the given time histories of system state and control input by integrating $N + r + 1$ equations given by Eq. (3.13), Eq. (3.14) and Eq. (3.15). Stacking time histories for $x_i^{1\mathrm{f}}(t)$, $\phi_j^{1\mathrm{f}}(t)$ and $u_k^{1\mathrm{f}}(t)$ leads to

$$\boldsymbol{x}_i^{1\mathrm{f}} = {\boldsymbol{\phi}^{1\mathrm{f}}}^{\top} \boldsymbol{\alpha}_i + \left(G_i \boldsymbol{u}^{1\mathrm{f}}\right)^{\top} \tag{3.17}$$

where $\boldsymbol{x}_i^{1\mathrm{f}} \in \mathbb{R}^{l \times 1}$, $\boldsymbol{\phi}^{1\mathrm{f}} \in \mathbb{R}^{N \times l}$, $\boldsymbol{\alpha}_i \in \mathbb{R}^{N \times 1}$, $G_i \in \mathbb{R}^{1 \times r}$ and $\boldsymbol{u}^{1\mathrm{f}} \in \mathbb{R}^{r \times l}$ with l being the number of data points. In this equation, $\boldsymbol{\alpha}_i$ is the i^{th} row of the coefficient matrix $\boldsymbol{\alpha}$ introduced in Eq. (3.4) and G_i is the i^{th} row of the coefficient matrix G introduced in Eq. (3.1). Now, one can find an optimal value of coefficient vector, $\boldsymbol{\alpha}_i$ by solving the weighted two-norm minimization:

$$\boldsymbol{\alpha}_i^{\star} = \min_{\boldsymbol{\alpha}_i} \frac{1}{2} \boldsymbol{e}^{\top} R \boldsymbol{e}, \quad \boldsymbol{e} = \boldsymbol{x}_i^{1\mathrm{f}} - {\boldsymbol{\phi}^{1\mathrm{f}}}^{T} \boldsymbol{\alpha}_i - \left(G_i \boldsymbol{u}^{1\mathrm{f}}\right)^{\top}. \tag{3.18}$$

The weight matrix R can be chosen appropriately depending upon the noise in the measurement data. Depending upon the size of l and N, the aforementioned optimization problem can be over-determined or underdetermined. In both cases, one can find the solution with an appropriate pseudo-inverse of $\boldsymbol{\phi}^{1\mathrm{f}}$, i.e.,

$${\boldsymbol{\alpha}_i^{\star}}^{\top} = \left({\boldsymbol{x}_i^{1\mathrm{f}}}^{\top} - G_i \boldsymbol{u}^{1\mathrm{f}}\right) {\boldsymbol{\phi}^{1\mathrm{f}}}^{\dagger} = \tilde{\boldsymbol{x}}_i^{1\mathrm{f}} {\boldsymbol{\phi}^{1\mathrm{f}}}^{\dagger} \tag{3.19}$$

where $\tilde{\boldsymbol{x}}_i^{1\mathrm{f}} = \boldsymbol{x}_i^{1\mathrm{f}^\top} - G_i \boldsymbol{u}^{1\mathrm{f}}$ and † stands for the pseudo-inverse. $\boldsymbol{\phi}^{1\mathrm{f}^\dagger}$ corresponds to the least-squares solution for an over-determined problem, i.e., $\boldsymbol{\phi}^{1\mathrm{f}^\dagger} = \left(\boldsymbol{\phi}^{1\mathrm{f}^\top} R \boldsymbol{\phi}^{1\mathrm{f}}\right)^{-1} \boldsymbol{\phi}^{1\mathrm{f}^\top} R$ while $\boldsymbol{\phi}^{1\mathrm{f}^\dagger}$ corresponds to the minimum-norm solution for the under-determined case, i.e., $\boldsymbol{\phi}^{1\mathrm{f}^\dagger} = \boldsymbol{\phi}^{1\mathrm{f}^\top} R \left(\boldsymbol{\phi}^{1\mathrm{f}} R \boldsymbol{\phi}^{1\mathrm{f}^\top}\right)^{-1}$. This procedure is repeated n times (for $i = 1, 2, \ldots, n$) to compute the full coefficient matrix $\boldsymbol{\alpha}$. Note that one can also compute an estimate for the control influence matrix , G through this procedure given that it also appears linearly in Eq. (3.17).

Eq. (3.19) is a minimization problem obtained by choosing certain collocation points. The choice of the collocation points typically interferes with the filter parameter choices and one should pick the collocation points judiciously, such that their spectral characteristics are not coincident. This is because if they are, the filtered states result in a null solution for the system of equations. In this work, a time-uniform distribution is chosen so that the spectral characteristics of the physics of interest are captured for each problem.

3.3.1.2 Sparse Solution

The $\boldsymbol{\alpha}_i^\star$ corresponds to the optimal solution in terms of minimizing the two-norm of state output error response. However, the two-norm solution is not guaranteed to be sparse in nature and is known to pick all the basis functions in our dictionary especially in the case of noise-corrupted measurements. To enforce sparsity, ideally the ℓ_0 norm of the coefficient vector $\boldsymbol{\alpha}_i$ needs to be minimized subject to constraints of Eq. (3.17). The ℓ_0 norm corresponds to the cardinality of the coefficient vector and its minimization leads to a non-convex problem. However, the ℓ_0 norm minimization problem can be approximated by an iterative ℓ_1-norm minimization problem, which is convex in nature with a guaranteed solution [72]:

$$\min_{\boldsymbol{\theta}_i^p} ||W^p \boldsymbol{\theta}_i^p||_1 \tag{3.20a}$$

$$\text{s.t.} \left|\left|\tilde{\boldsymbol{x}}_i^{1\mathrm{f}} - \boldsymbol{\theta}_i^{p^\top} \boldsymbol{\phi}^{1\mathrm{f}}\right|\right|_2 \leq \varepsilon \left|\left|\tilde{\boldsymbol{x}}_i^{1\mathrm{f}} - \boldsymbol{\alpha}_i^{\star^\top} \boldsymbol{\phi}^{1\mathrm{f}}\right|\right|_2, \quad \varepsilon \geq 1 \tag{3.20b}$$

where p is the iteration, $\boldsymbol{\theta}_i^p$ is the optimization variable, $\tilde{\boldsymbol{x}}_i^{1\mathrm{f}}$ is the pseudo signal, $\boldsymbol{\phi}^{1\mathrm{f}}$ is the dictionary of basis functions and $\boldsymbol{\alpha}_i^{\star^\top}$ is the optimal two-norm solution derived in the previous section. Notice that the two-norm constraint of Eq. (3.20b) corresponds to the satisfaction of Eq. (3.17). Rather than using the equality constraint of Eq. (3.17), a two-norm error is bounded by the optimal pseudo-norm solution with ε being the user-defined relaxation on two-norm error. This allows one to tradeoff sparsity with approximation error. Furthermore W^p is a diagonal matrix containing a known weight w_j for the j^{th} optimization variable. Initially, w_j can be chosen based upon any a priori knowledge about the structure of $\boldsymbol{f}$, the form of the least-squares solution or can simply be chosen to be one. In the subsequent iterations, the value of w_j is adapted according to the following formula to penalize the coefficients that are smaller than a predefined threshold δ:

$$w_j^p = \frac{1}{\left|\theta_{i,j}^{p-1}\right| + \eta}. \tag{3.21}$$

Algorithm 1 Iterative sparse algorithm for model selection

function SPARSEID($\delta, \eta, \varepsilon, \text{nb}_{\text{iterations}}, \boldsymbol{\phi}^{1\text{f}}, \tilde{\boldsymbol{x}}^{1\text{f}}, \boldsymbol{\alpha}_j^\star$)
  Initialize weighting matrix $W = I_N$
  Initialize $p = 0$
  while $p < \text{nb}_{\text{iterations}}$ **do**
    Solve for $\boldsymbol{\theta}_j^p$: $\min_{\boldsymbol{\theta}_j^p} ||W^p \boldsymbol{\theta}_j^p||_1$ such that $\left|\left|\tilde{\boldsymbol{x}}^{1\text{f}} - \boldsymbol{\theta}_j^{pT} \boldsymbol{\phi}^{1\text{f}}\right|\right|_2 \leq \varepsilon \left|\left|\tilde{\boldsymbol{x}}^{1\text{f}} - \boldsymbol{\alpha}_j^{\star T} \boldsymbol{\phi}^{1\text{f}}\right|\right|_2$
    Update weighting matrix: $W[i,i] = \dfrac{1}{\left|\theta_{i,j}^{p-1}\right| + \eta}$
  end while
  $\boldsymbol{\theta}_j = \boldsymbol{\theta}_j^p$
  for $i = 1$ **to** N **do**
    if $\boldsymbol{\theta}_j[i] < \delta$ **then**
      $\boldsymbol{\theta}_j[i] = 0$
      Remove column $\boldsymbol{\phi}_i^{1\text{f}}$ from dictionary $\boldsymbol{\phi}^{1\text{f}}$
    end if
  end for
  Compute final least-squares solution with updated dictionary: $\boldsymbol{\alpha}_j^{*T} = \tilde{\boldsymbol{x}}^{1\text{f}} \boldsymbol{\phi}^{1\text{f}^\dagger}$
end function

η is a small number to avoid division by zero. This iterative procedure is repeated unless the computed coefficients converge within a prescribed tolerance. The solution of this iterative ℓ_1 minimization problem provides us a subset of basis functions from an over-complete dictionary, which plays a dominant role in the underlying unknown dynamics. An optimal pseudo-inverse solution for the coefficients is obtained for only this subset of basis functions at the end of the procedure. Figure 3.1 illustrates these steps to obtain the sparse solution. The algorithm is given in Alg. 1. Note that the parameter δ acts as a threshold to separate the active basis function from the nonactive ones. In practice, with normalized trajectories and independently of the dynamical system considered, it is a relatively safe assumption to neglect the least dominant basis function, i.e., the basis functions with associated coefficient two to three orders of magnitude smaller than other basis function coefficients. In this respect, δ is usually chosen to be at least one order of magnitude smaller than the least dominant basis function coefficient. For noisy input-output data, the value of δ is chosen based upon the signal-to-noise ratio to avoid over-fitting the data. A good discussion on the choice of these hyper parameters on approximation accuracy has been provided in [72].

3.3.2 Second-Order Systems

Derived from the rates of generalized momenta at the acceleration level, most dynamical systems in engineering mechanics are characterized by second-order differential equations. If both position and velocity measurements are available, the second-order differential equation can be reshaped as a first order differential system and the methodology presented in the last section can be applied. In this section, the methodology presented in the last section is generalized for the identification of the second order system with time histories of position level measurements and control input vectors being available. Consider the following special class of second-order nonlinear dynamical system with nonlinearities being

Figure 3.1: Illustration of the iterative procedure to derive a sparse solution for a first order system.

a function of only position-level state variables:

$$\ddot{\boldsymbol{x}}(t) = \boldsymbol{f}(\boldsymbol{x}(t)) + G\boldsymbol{u}(t), \tag{3.22}$$

where $\boldsymbol{x}(t) \in \mathbb{R}^n$ represents the state of the system and $\boldsymbol{u}(t) \in \mathbb{R}^r$ the control action at time t and $G \in \mathbb{R}^{n\times r}$ is the constant-time input influence matrix. Once again, the function $\boldsymbol{f} : \mathbb{R}^n \to \mathbb{R}^n$ represents the dynamics constraints that define the equations of motion of the system and our objective is to identify this nonlinear function from time histories of $\boldsymbol{x}(t)$ and $\boldsymbol{u}(t)$. Following the same development as before, the unknown nonlinear function $\boldsymbol{f}$ can be expanded in terms of a dictionary of basis functions and Eq. (3.22) can be re-written as

$$\ddot{\boldsymbol{x}}(t) = \sum_{j=1}^{N} \boldsymbol{\alpha}_j \phi_j(\boldsymbol{x}) + G\boldsymbol{u}(t) \tag{3.23}$$

In a quest to determine the analog of Eq. (3.17), the component-wise Laplace transform of the aforementioned vector equation leads to

$$X_i^{0\mathrm{f}}(s) = s^2 X_i(s) - s x_i(0) - \dot{x}_i(0) = \sum_{j=1}^{N} \alpha_{i,j} \Phi_j(s) + \sum_{k=1}^{r} G_{i,k} U_k(s), \quad i = 1 \dots n, \tag{3.24}$$

Now, applying the integral operator $\mathcal{I}_{\lambda_1}$ to the original signal $X_i^{0\mathrm{f}}$ yields

$$X_i^{1\mathrm{f}}(s) = \mathcal{I}_{\lambda_1}\left(X_i^{0\mathrm{f}}(s)\right) = \frac{s^2 X_i(s) - s x_i(0) - \dot{x}_i(0)}{s + \lambda_1} = \sum_{j=1}^{N} \alpha_{i,j} \frac{\Phi_j(s)}{s + \lambda_1} + \sum_{k=1}^{r} G_{i,k} \frac{U_k(s)}{s + \lambda_1}. \tag{3.25}$$

For $\lambda_2 \in \mathbb{R}_+^*$, applying the integral operator $\mathcal{I}_{\lambda_2}$ to $X_i^{1\mathrm{f}}$ of Eq. (3.25) results in

$$X_i^{2\mathrm{f}}(s) = \mathcal{I}_{\lambda_2}\left(X_i^{1\mathrm{f}}(s)\right) = \frac{s^2 X_i(s) - s x_i(0) - \dot{x}_i(0)}{(s + \lambda_1)(s + \lambda_2)} \tag{3.26}$$

$$= \sum_{j=1}^{N} \alpha_{i,j} \frac{\Phi_j(s)}{(s + \lambda_1)(s + \lambda_2)} + \sum_{k=1}^{r} G_{i,k} \frac{U_k(s)}{(s + \lambda_1)(s + \lambda_2)} \tag{3.27}$$

Notice that $\lambda_1 = \lambda_2 = 0$ corresponds to a double integration of the state vector and non-zero values for λ_1 and λ_2 help to accommodate for initial condition errors. Finally, with

$$\Phi_j^{2\mathrm{f}}(s) = \frac{\Phi_j(s)}{(s + \lambda_1)(s + \lambda_2)} \quad \text{and} \quad U_k^{2\mathrm{f}}(s) = \frac{U_k(s)}{(s + \lambda_1)(s + \lambda_2)}, \tag{3.28}$$

the Laplace filtered (twice) version of Eq. (3.23) is

$$X_i^{2\mathrm{f}}(s) = \sum_{j=1}^{N} \alpha_{i,j} \Phi_j^{2\mathrm{f}}(s) + \sum_{k=1}^{r} G_{i,k} U_k^{2\mathrm{f}}(s). \tag{3.29}$$

Here the superscript $^{2\mathrm{f}}$ corresponds to a signal that has been filtered twice. Adding and subtracting $\lambda_1 X_i(s)$ and $\lambda_2 X_i(s)$ leads to

$$\begin{aligned} X_i^{2\mathrm{f}}(s) &= \frac{(s+\lambda_1)(s+\lambda_2)X_i(s) - (\lambda_1+\lambda_2)sX_i(s) - \lambda_1\lambda_2 X_i(s) - (s+\lambda_1)x_i(0) + \lambda_1 x_i(0) - \dot{x}_i(0)}{(s+\lambda_1)(s+\lambda_2)} \\ &= X_i(s) - \frac{(\lambda_1+\lambda_2)X_i(s) + x_i(0)}{(s+\lambda_2)} + \frac{\lambda_1^2 X_i(s) + \lambda_1 x_i(0) - \dot{x}_i(0)}{(s+\lambda_1)(s+\lambda_2)} \\ &= X_i(s) + Y_{i,1}(s) + Y_{i,2}(s), \end{aligned} \tag{3.30}$$

where

$$Y_{i,1}(s) = \frac{-(\lambda_1+\lambda_2)X_i(s) - x_i(0)}{(s+\lambda_2)}, \quad Y_{i,2}(s) = \frac{\lambda_1^2 X_i(s) + \lambda_1 x_i(0) - \dot{x}_i(0)}{(s+\lambda_2)(s+\lambda_1)}. \tag{3.31}$$

The inverse Laplace transform of Eq. (3.31) yields the linear ODEs:

$$\dot{y}_{i,1}(t) = -\lambda_2 y_{i,1}(t) - (\lambda_1+\lambda_2)x_i(t), \quad y_{i,1}(0) = -x(0), \tag{3.32}$$

$$\ddot{y}_{i,2}(t) = -(\lambda_1+\lambda_2)\dot{y}_{i,2}(t) - \lambda_1\lambda_2 y_{i,2}(t) + \lambda_1^2 x_i(t), \quad y_{i,2}(0) = 0, \quad \dot{y}_{i,2}(0) = \lambda_1 x(0) - \dot{x}(0) \tag{3.33}$$

Similarly, the inverse Laplace transforms for $\Phi_j^{2\mathrm{f}}$ and $U_k^{2\mathrm{f}}$ yield the corresponding second-order ODEs:

$$\ddot{\phi}_j^{2\mathrm{f}}(t) = -(\lambda_1+\lambda_2)\dot{\phi}_j^{2\mathrm{f}}(t) - \lambda_1\lambda_2\phi_j^{2\mathrm{f}}(t) + \phi_j(\boldsymbol{x}), \quad \phi_j^{2\mathrm{f}}(0) = 0 \quad \dot{\phi}_j^{2\mathrm{f}}(0) = 0, \quad j = 1, 2, \ldots, N, \tag{3.34}$$

$$\ddot{u}_k^{2\mathrm{f}}(t) = -(\lambda_1+\lambda_2)\dot{u}_k^{2\mathrm{f}}(t) - \lambda_1\lambda_2 u_k^{2\mathrm{f}}(t) + u_k(t), \quad u_k^{2\mathrm{f}}(0) = 0, \quad \dot{u}_k^{2\mathrm{f}}(0) = 0. \tag{3.35}$$

Finally, the inverse Laplace transformation of Eq. (3.30) results in the following linear equation in unknown coefficient vector, $\boldsymbol{\alpha}_i$:

$$x_i^{2\mathrm{f}}(t) = x_i(t) + y_{i,1}(t) + y_{i,2}(t) = \sum_{j=1}^{N} \alpha_{i,j}\phi_j^{2\mathrm{f}}(t) + \sum_{k=1}^{r} G_{i,k}u_k^{2\mathrm{f}}(t). \tag{3.36}$$

This is analogous to Eq. (3.17) for the first order systems and requires only knowledge of $x_i(t)$ and $u_k(t)$ to find unknown coefficient vector. One can now employ the iterative ℓ_1 solution in conjunction with two-norm minimization to find the appropriate basis functions and their corresponding contributions from a large dictionary of basis functions as discussed in Section 3.3.1.2. For completeness, this procedure is generalized to a generic d^{th} order system in the Appendix.

3.4 Numerical Results

The method described in the previous section to identify governing equations from measurement data is now validated on three examples of different complexity. The first example corresponds to identification of nonlinear dynamics for the Duffing oscillator while the second example corresponds to the identification of the chaotic Lorenz oscillator. The third example corresponds to identification of Newton's Law of Gravitation by considering the motion of a satellite in an orbit around the Earth. The first two examples correspond to chaotic dynamical systems which show some interesting dynamical behavior while the

third problem corresponds to the classical central-force field dynamical model. The first order system formulation presented in Section 3.3.1 is used to identify the unknown dynamics for the first two examples and the second-order formulation presented in Section 3.3.2 is used for the identification of the central force field in the third example with position-only measurements. For the central force field identification, the results are compared with a multi-layer NN as used in our prior work [73, 74].

3.4.1 Duffing Oscillator

The first example aims to legitimize the new approach of this paper with a low-pass filter based integral formulation by comparing it with a pure integral formulation for parameter estimation of the nonlinear Duffing oscillator. The Duffing oscillator represents a nonlinear spring-mass-damper system and shows dynamical behaviors of interest for many real engineering applications. The governing equations of motion for the Duffing oscillator are given as

$$\ddot{x} + \gamma\dot{x} + \alpha x + \beta x^3 = u \tag{3.37}$$

where $\gamma = 0.2$, $\alpha = 1$, $\beta = -1$, and u is a random excitation following a Gaussian distribution with mean 0 and standard deviation 1. With $x_1 = x$ and $x_2 = \dot{x}$ (x_1 is the analogous of a position, x_2 of a velocity), Eq. (3.37) can be re-written as the first-order dynamical system:

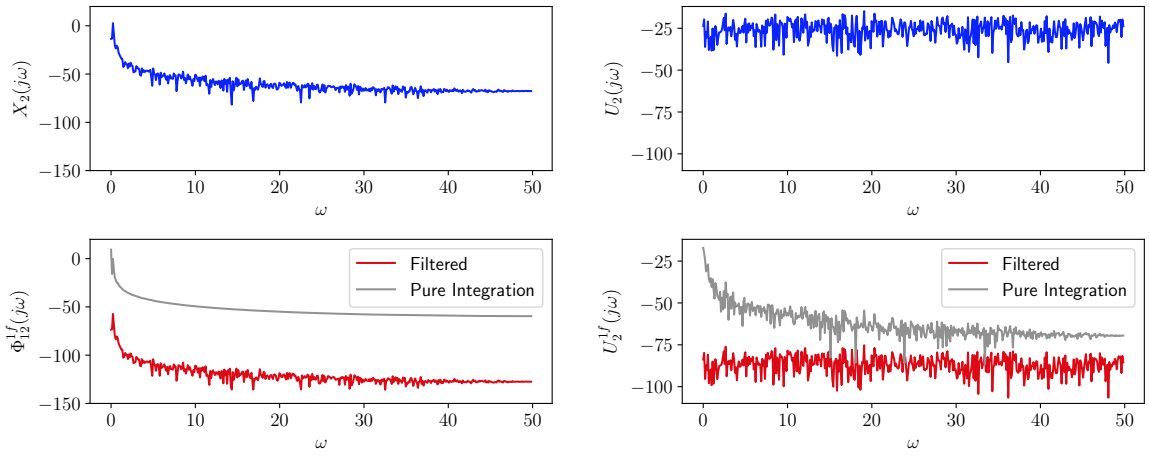
$$\dot{x}_1 = x_2 \tag{3.38a}$$

$$\dot{x}_2 = -\gamma x_2 - \alpha x_1 + \beta x_1^3 + u \tag{3.38b}$$

The training dataset for the identification purpose corresponds to response of the system for the initial condition $\boldsymbol{x}(0) = \begin{bmatrix} 1.4 & 0 \end{bmatrix}^\top$. The state and control input data are simulated at a frequency of 200Hz for 10 seconds. Two test cases are considered: the first test case corresponds to perfect measurements while the second test case corresponds to state measurements being corrupted with zero mean Gaussian white noise with variance of 10^{-4}. The initial dictionary of basis function consists of all monomials up to 10^{th} order in x_1-x_2 space, resulting in a total of 66 basis functions to approximate the unknown system dynamics:

$$\begin{aligned}\mathbb{R}_{10}[x_1, x_2] = \{&1 \;\; x_1 \;\; x_1^2 \;\; x_1^3 \;\; x_1^4 \;\; x_1^5 \;\; x_1^6 \;\; x_1^7 \;\; x_1^8 \;\; x_1^9 \;\; x_1^{10} \;\; x_2 \;\; x_1x_2 \;\; x_1^2x_2 \;\; x_1^3x_2 \;\; x_1^4x_2 \;\; x_1^5x_2 \;\; x_1^6x_2 \;\; x_1^7x_2 \;\; x_1^8x_2 \;\; x_1^9x_2 \\ &x_2^2 \;\; x_1x_2^2 \;\; x_1^2x_2^2 \;\; x_1^3x_2^2 \;\; x_1^4x_2^2 \;\; x_1^5x_2^2 \;\; x_1^6x_2^2 \;\; x_1^7x_2^2 \;\; x_1^8x_2^2 \;\; x_2^3 \;\; x_1x_2^3 \;\; x_1^2x_2^3 \;\; x_1^3x_2^3 \;\; x_1^4x_2^3 \;\; x_1^5x_2^3 \\ &x_1^6x_2^3 \;\; x_1^7x_2^3 \;\; x_2^4 \;\; x_1x_2^4 \;\; x_1^2x_2^4 \;\; x_1^3x_2^4 \;\; x_1^4x_2^4 \;\; x_1^5x_2^4 \;\; x_1^6x_2^4 \;\; x_2^5 \;\; x_1x_2^5 \;\; x_1^2x_2^5 \;\; x_1^3x_2^5 \;\; x_1^4x_2^5 \;\; x_1^5x_2^5 \\ &x_2^6 \;\; x_1x_2^6 \;\; x_1^2x_2^6 \;\; x_1^3x_2^6 \;\; x_1^4x_2^6 \;\; x_2^7 \;\; x_1x_2^7 \;\; x_1^2x_2^7 \;\; x_1^3x_2^7 \;\; x_2^8 \;\; x_1x_2^8 \;\; x_1^2x_2^8 \;\; x_1x_2^9 \;\; x_2^{10}\}\,.\end{aligned} \tag{3.39}$$

Two values of λ are investigated in order to compare the low-pass filter based integral formulation regression model ($\lambda = 100$) with a pure integration based model ($\lambda = 0$). First, a signal analysis is performed on the fundamental signals used in both regression models, specifically, their spectral signatures. Figure 3.2(a) compares the spectral content of the signal that is a direct integration of the state ($\lambda = 0$) with the filtered state ($\lambda = 100$). The signal of interest is $\Phi_{12}^{1\text{f}}$ corresponding to the filtered signal x_2 at position 12 in the dictionary. While the direct integration loses some of the spectral content as denoted by the smooth Fourier transform, $\Phi_{12}^{1\text{f}}$ seems to qualitatively share the spectral signature that



(a) State x_2

(b) Input Signal u

Figure 3.2: Duffing oscillator: Spectral signatures of the state x_2 and the input signal u and their filtered version

looks very similar in shape to $X_2(j\omega)$ of Figure 3.2(a). Similarly, the spectral content of the filtered input matches that of the original broad-band white noise excitation, shown in Figure 3.2(b). Since the amplitude of the spectral content in the integrated signals gets eroded uniformly across all frequencies because of the smoothness of the integration operation, the pure integration-based model fails to capture the input-output relationships at higher frequencies. While this model can still be used in parameter estimation, the magnitude and frequency content of the inputs should be adjusted to improve the learning process. For the low-pass filter based integral formulation regression model, prior experience of the analyst in choosing the filter time-constants enables a better control and conditioning of the approximation problem. Therefore in many signal analysis and adaptive control problems of online learning, the low-pass filter-based model is preferred to a pure integration-based model. Furthermore, the low-pass filter automatically rejects high frequency noise in practical applications of interest in systems and control.

The procedure listed in Section 3.3.1 is adopted to find the unknown coefficient vector. While the least-square solution is a combination of all the 66 basis functions, the sparse solution accurately identifies the true basis functions participating in the actual dynamics. Tables 3.1 and 3.2 present the values of the identified coefficients for both the test cases considered. These identified values appear to be very close to the true ones, which shows the efficacy of the methodology in identifying the true dynamics of the system. Relative error percentages are calculated as

$$\text{Relative Error } \% = \frac{\text{true} \ - \ \text{identified}}{\text{true}} \times 100. \tag{3.40}$$

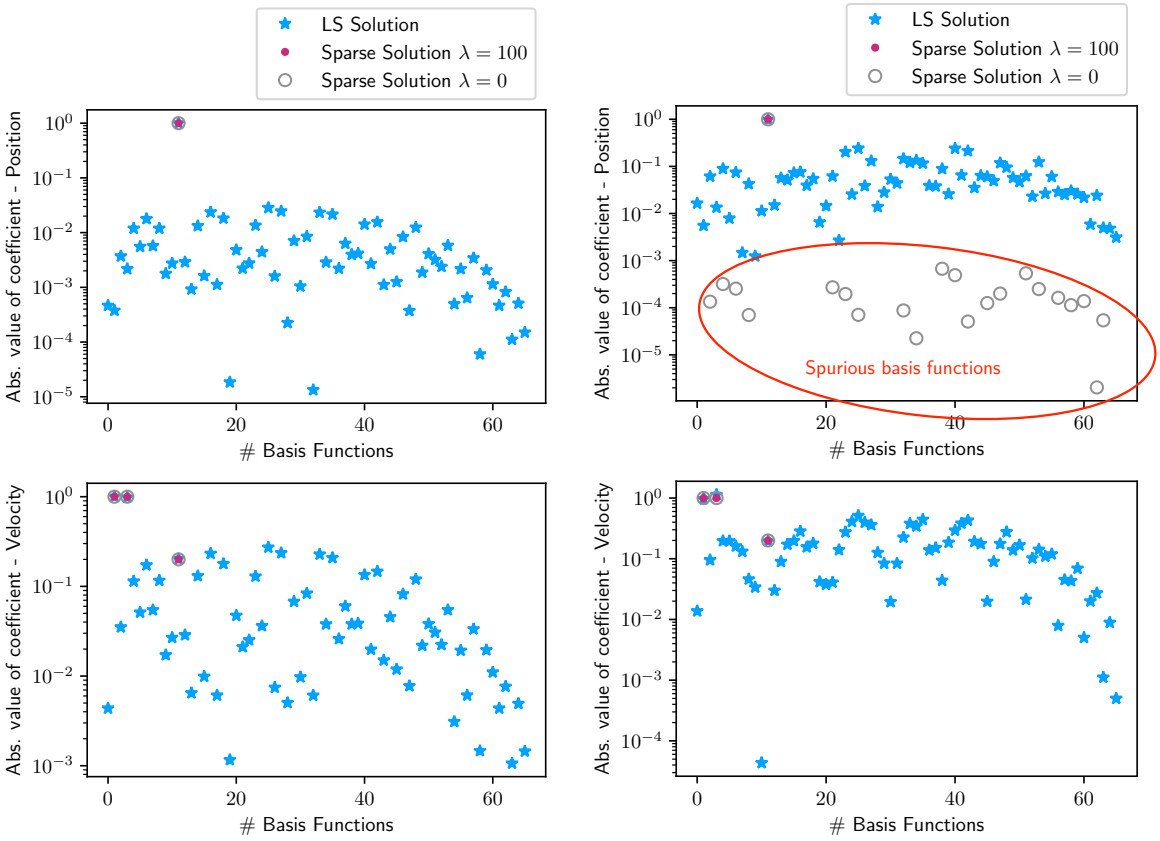
Table 3.1: Value of the coefficients for the sparse solution vs. the true coefficients - $\dot{x}_1$ equation

Value of λ	**Basis Function**	**# in the dictionary**	**True Value**	**Sparse Sol. (no Noise)**	**Sparse Sol. (with Noise)**	**Rel. Error % (no Noise)**	**Rel. Error % (with Noise)**
0	$x_1, x_2 \mapsto x_2$	12	1	0.99999964	1.000014	$3.6 \cdot 10^{-5}$	$-1.4 \cdot 10^{-3}$
100	$x_1, x_2 \mapsto x_2$	12	1	0.999999984	0.99999971	$1.6 \cdot 10^{-6}$	$2.9 \cdot 10^{-5}$

Table 3.2: Value of the coefficients for the sparse solution vs. the true coefficients - $\dot{x}_2$ equation

Value of λ	**Basis Function**	**# in the dictionary**	**True Value**	**Sparse Sol. (no Noise)**	**Sparse Sol. (with Noise)**	**Rel. Error % (no Noise)**	**Rel. Error % (with Noise)**
0	$x_1, x_2 \mapsto x_1$	2	-1	-1.00000003	-1.000011	$-3.0 \cdot 10^{-6}$	$-1.1 \cdot 10^{-3}$
0	$x_1, x_2 \mapsto x_1^3$	4	-1	-0.99999996	-0.999988	$4.0 \cdot 10^{-6}$	$1.2 \cdot 10^{-3}$
0	$x_1, x_2 \mapsto x_2$	12	-0.2	-0.20000003	-0.2000060	$-1.5 \cdot 10^{-5}$	$-3.0 \cdot 10^{-3}$
100	$x_1, x_2 \mapsto x_1$	2	-1	-1.0000004	-1.0000016	$-4.0 \cdot 10^{-5}$	$-1.6 \cdot 10^{-4}$
100	$x_1, x_2 \mapsto x_1^3$	4	-1	-0.9999995	-0.9999974	$5.0 \cdot 10^{-5}$	$2.6 \cdot 10^{-4}$
100	$x_1, x_2 \mapsto x_2$	12	-0.2	-0.200000008	-0.1999995	$4.0 \cdot 10^{-6}$	$2.5 \cdot 10^{-4}$

Figures 3.3(a) and 3.3(b) display the value of the resulting least-squares and sparse coefficients for both formulations and cases. Although the least-square solution is the best fit possible in the sense that the square of the error between the true signal and the identified signal is minimized, this results in an over-fitting with more basis functions than necessary, particularly true in the presence of noise. Both the pure integral and filtered formulations perform well when the data are not corrupted with any noise. As seen in Figure 3.3(b), the integral formulation starts picking up basis functions that do not appear in the dynamics, resulting in an over-fitting as well. This is because, in addition to the spectral content of the noise perturbation, the true spectral content of the signal has been smoothed-out making it difficult for the algorithm to distinguish the true signal from the noise. Figures 3.4(a) and 3.4(c) show the error resulting in the propagation for both test cases while using the least-square solution for the identified dynamics. Though the least squares solution provides the optimal value of the coefficients while minimizing the two-norm error of the measurement data at discrete time instants, the presence of basis functions that are not participating in the true dynamics leads to over-fitting and hence large propagation errors. Notice that the pure integration based model leads to an error in propagation of the same order of magnitude as the low-pass filter based formulation. On the other hand, the resulting sparse-identified model inherently and automatically balances model complexity with accuracy and results in small absolute error for both the test cases as seen in Figures 3.4(b) and 3.4(d). Here, the low-pass filter based formulation is able to preserve the spectral content of the original signals used in the regression process and differentiate that content from the white noise spectral signature, leading to the right selection of the basis functions. Finally, Table 3.3 presents the RMS errors for 10 random initial conditions generated from Gaussian distribution with mean $\boldsymbol{x}(0) = \{1.4,\ 0\}^T$ and covariance $P_0 = I_2$. Once again, as expected, the sparse approximation based identified system leads to minimal RMS error even in the presence of noise. This better performance of the sparse approximation method can be attributed to its ability to identify the inherent true dynamics of the system.



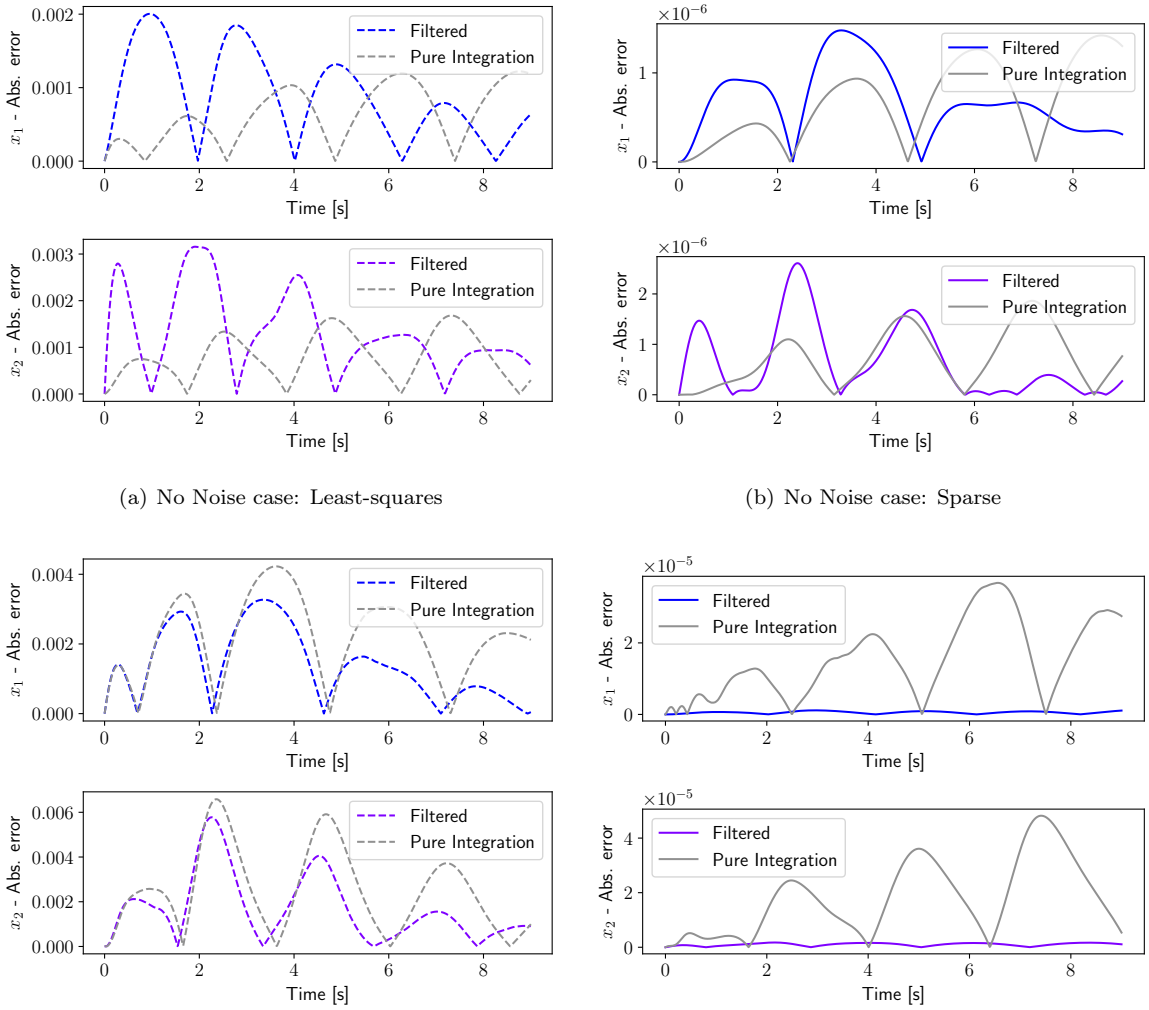
(a) No Noise case

(b) Noise case

Figure 3.3: Duffing Oscillator: Value of the coefficients for the basis functions

Conclusion on hyper-parameter λ: Setting $\lambda = 0$, the direct integration regression model is obtained in our approach. The integral equation transformation is a special case of what is implemented in this paper. The first takeaway is that the integration operation uniformly suppresses the frequency content in the signals involved in the approximation process, meaning some key spectral content at moderate frequencies will also be smoothed out by the direct integration. This mandates the direct integration process to have input signals (for training) with low frequency and larger amplitudes. Of course when the right-hand side becomes a nonlinear function of the states, this problem becomes compounded. Another byproduct of this method is that the learning process continues even when the system is turned off ($u = 0$) and the filtered integration process preserves the spectral content of the original signals used in the regression process because a fading memory is implemented in the filtered integration process. The data are thus exponentially de-weighted and the learning stops as soon as excitation stops.

Table 3.3: Duffing oscillator: RMS Error on 10 random initial conditions



(a) No Noise case: Least-squares

(b) No Noise case: Sparse

(c) Noise case: Least-squares

(d) Noise case: Sparse

Figure 3.4: Duffing oscillator: Norm of relative error for different scenarios

Solution	$\lambda = 0$		$\lambda = 100$	
	No Noise Case	**Noise Case**	**No Noise Case**	**Noise Case**
Least-squares	$7.41 \cdot 10^{-4}$	$4.02 \cdot 10^{-3}$	$8.56 \cdot 10^{-4}$	$3.41 \cdot 10^{-3}$
Sparse	$9.77 \cdot 10^{-7}$	$3.45 \cdot 10^{-5}$	$9.02 \cdot 10^{-7}$	$1.45 \cdot 10^{-6}$

3.4.2 Lorenz System

The second example corresponds to the chaotic Lorenz oscillator with governing dynamic equations given as

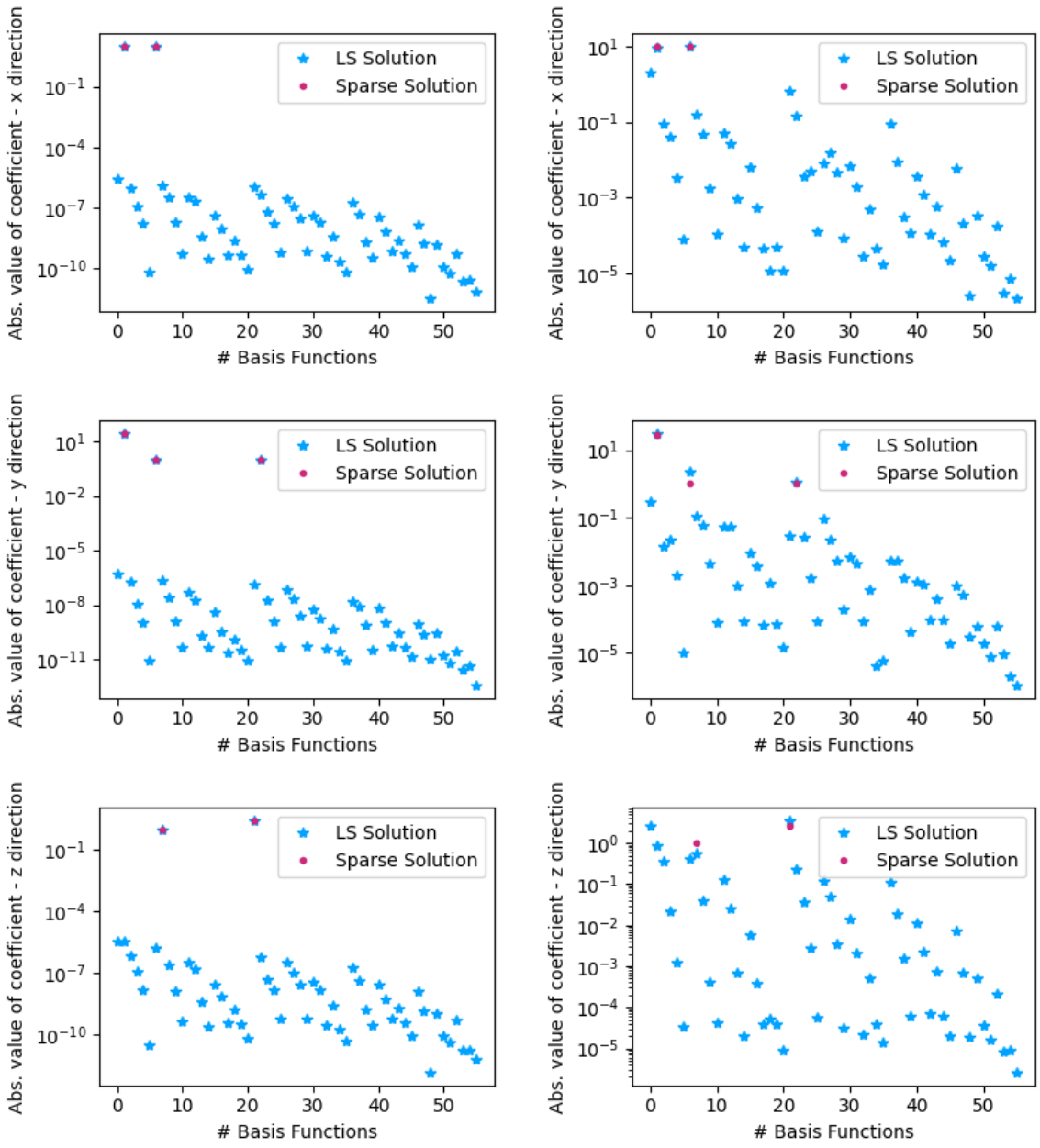
$$\begin{aligned}\dot{x} &= \sigma(y - x)\\ \dot{y} &= x(\rho - z) - y\\ \dot{z} &= xy - \beta z\end{aligned} \tag{3.41}$$

The training data set corresponds to $\sigma = 10$, $\rho = 28$, $\beta = 8/3$ and an initial condition of $\boldsymbol{x}(0) = \begin{bmatrix}-8 & 7 & 27\end{bmatrix}^\top$. The data are recorded at a frequency of 500Hz for 4 seconds. A value of $\lambda_1 = 10$ is chosen for the low-pass filter. Once again two test cases (with random initial condition, different from the ones used to generate the training data set) are considered with the first test case corresponding to perfect measurements and the second test case corresponding to measurements being corrupted by zero mean Gaussian white noise of variance 10^{-4}. The initial dictionary of basis functions consists of a total of 56 polynomial basis functions up to degree 5 in state variables.

$$\begin{aligned}\mathbb{R}_5[x,y,z] = \big\{&1 \;\; x \;\; x^2 \;\; x^3 \;\; x^4 \;\; x^5 \;\; y \;\; xy \;\; x^2y \;\; x^3y \;\; x^4y \;\; y^2 \;\; xy^2 \;\; x^2y^2 \;\; x^3y^2 \;\; y^3 \;\; xy^3 \;\; x^2y^3 \;\; y^4 \;\; xy^4 \;\; y^5\\ &z \;\; xz \;\; x^2z \;\; x^3z \;\; x^4z \;\; yz \;\; xyz \;\; x^2yz \;\; x^3yz \;\; y^2z \;\; xy^2z \;\; x^2y^2z \;\; y^3z \;\; xy^3z \;\; y^4z \;\; z^2 \;\; xz^2 \;\; x^2z^2\\ &x^3z^2 \;\; yz^2 \;\; xyz^2 \;\; x^2yz^2 \;\; y^2z^2 \;\; xy^2z^2 \;\; y^3z^2 \;\; z^3 \;\; xz^3 \;\; x^2z^3 \;\; yz^3 \;\; xyz^3 \;\; y^2z^3 \;\; z^4 \;\; xz^4 \;\; yz^4 \;\; z^5\big\}.\end{aligned} \tag{3.42}$$

Once again, the procedure listed in Section 3.3.1 is used to find the unknown coefficient vector. Figures 3.5(a) and 3.5(b) show the least squares as well as sparse solution for coefficients for both test cases. While the sparse solution correctly identifies the correct basis functions, the least-squares fits non-zero amplitude for most of the basis functions. The iterative procedure converges within 3 iterations to accurately identify the participating basis functions and drives coefficients of non-participating basis functions to zero (Figure 3.6). Tables 3.4, 3.5 and 3.6 present the values of the identified coefficients for both test cases. These identified values appear to be very close to the true ones, which shows the efficacy of the developed methodology in identifying the true dynamics of the system.

Figures 3.7(a), 3.7(b) and Figures 3.7(c), 3.7(d) shows the error resulting in the propagation for both test cases while using the least-square as well as sparse identification of inherent dynamics. Observe that the amplitude of the error is correlated to the dynamics. Regions of the phase space associated with high velocity (occurs at a lobe switching) relate with larger errors, especially in the presence of noise (see Figures 3.7(c) and 3.7(d) where there is a lobe switching at $t \simeq 2$s). Finally, Table 3.7 presents the RMS errors for 10 random initial conditions generated from Gaussian distribution with mean $\boldsymbol{x}(0) = \begin{bmatrix}-8 & 7 & 27\end{bmatrix}^\top$ and covariance $P_0 = 0.5I_3$. From these results, it is clear that the proposed sparse approximation solution leads to an order of magnitude improvement in state propagation errors. It should be noted that this approach provides 10^{-4}%-10^{-7}% error as compared to an accuracy of 0.03% as reported in [39, 40] for the same level of noise but at a higher sampling frequency of 1000Hz. This better performance of the sparse approximation method can be attributed to its ability to identify the inherent true dynamics of the system.



(a) No Noise case

(b) Noise case

Figure 3.5: Lorenz oscillator: Value of the coefficients for the basis functions

Table 3.4: Value of the coefficients for the sparse solution vs. the true coefficients - x direction

Basis Function	**# in the dictionary**	**True Value**	**Sparse Sol. (no Noise)**	**Sparse Sol. (with Noise)**	**Rel. Error % (no Noise)**	**Rel. Error % (with Noise)**
$x, y, z \mapsto x$	2	-10	-10.00000000035	-9.9999949	$-3.54 \cdot 10^{-9}$	$5.10 \cdot 10^{-7}$
$x, y, z \mapsto y$	7	10	10.00000000030	10.0000333	$-3.02 \cdot 10^{-9}$	$-3.33 \cdot 10^{-6}$

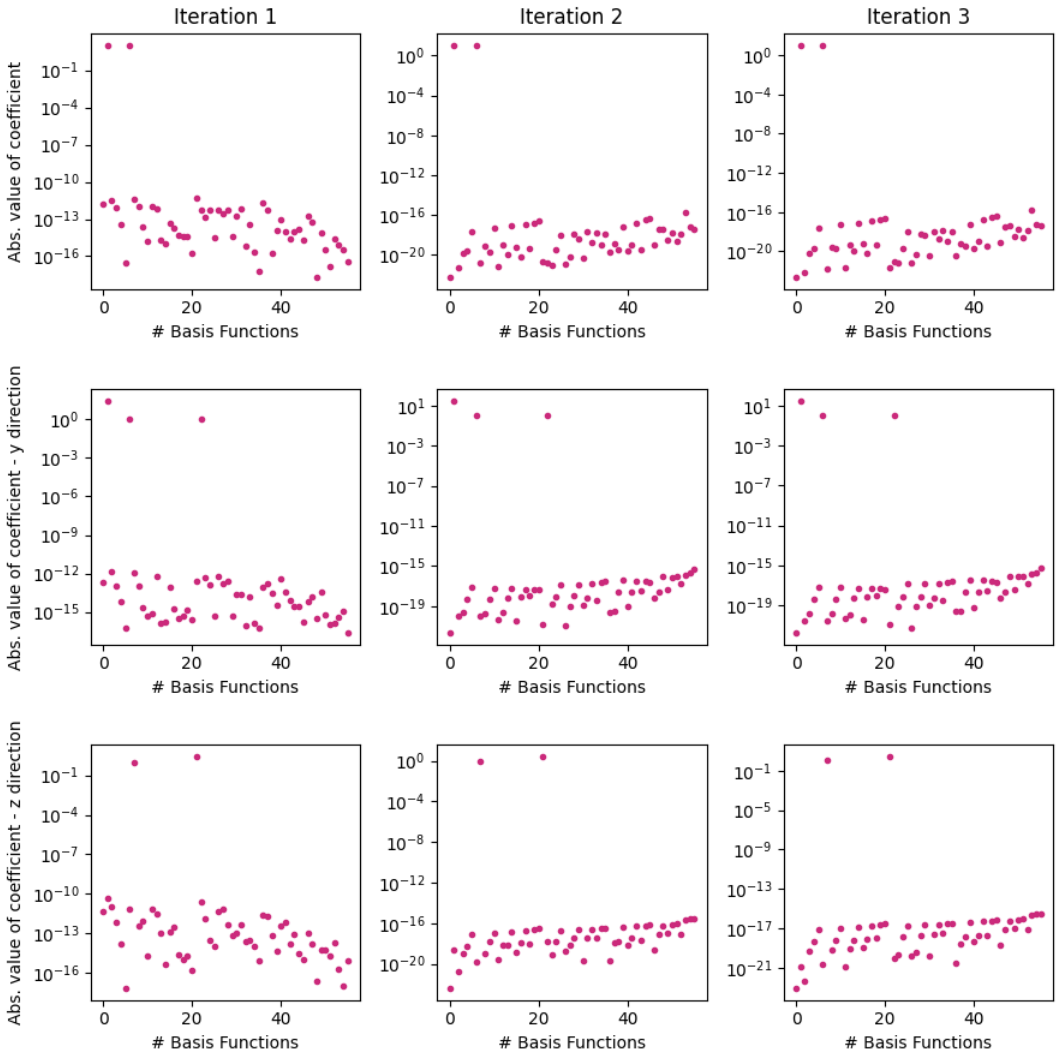


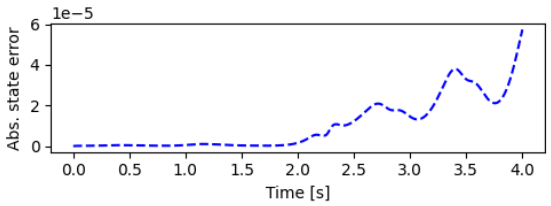
Figure 3.6: Lorenz oscillator: Evolution of the absolute value of the coefficients from the sparse approximation solution

Table 3.5: Value of the coefficients for the sparse solution vs. the true coefficients - y direction

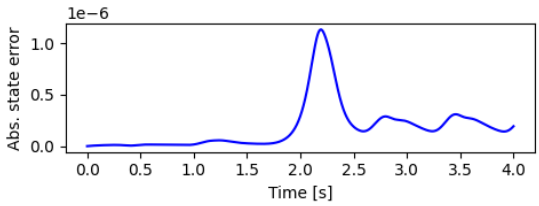
Basis Function	**# in the dictionary**	**True Value**	**Sparse Sol. (no Noise)**	**Sparse Sol. (with Noise)**	**Rel. Error % (no Noise)**	**Rel. Error % (with Noise)**
$x, y, z \mapsto x$	2	28	28.000000038	28.0008596	$-1.34 \cdot 10^{-7}$	$-3.07 \cdot 10^{-5}$
$x, y, z \mapsto y$	7	-1	1.0000000053	-1.0002722	$-5.33 \cdot 10^{-7}$	$-2.72 \cdot 10^{-4}$
$x, y, z \mapsto xz$	23	-1	-1.0000000012	-1.0000214	$-1.19 \cdot 10^{-7}$	$-2.14 \cdot 10^{-5}$

Table 3.6: Value of the coefficients for the sparse solution vs. the true coefficients - z direction

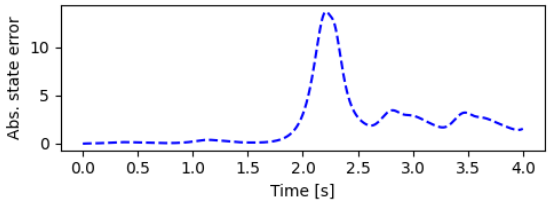
Basis Function	**# in the dictionary**	**True Value**	**Sparse Sol. (no Noise)**	**Sparse Sol. (with Noise)**	**Rel. Error % (no Noise)**	**Rel. Error % (with Noise)**
$x, y, z \mapsto xy$	8	1	1.00000000058	0.9999914	$-5.81 \cdot 10^{-8}$	$8.60 \cdot 10^{-6}$
$x, y, z \mapsto z$	22	$-8/3$	2.66666666776	-2.6666329	$-4.11 \cdot 10^{-8}$	$1.27 \cdot 10^{-5}$



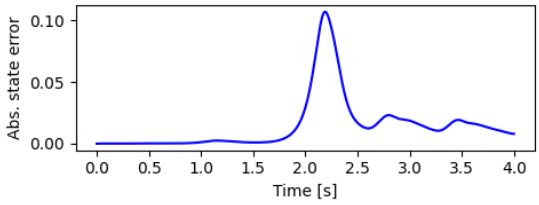
(a) No Noise case: Least-squares



(b) No Noise case: Sparse



(c) Noise case: Least-squares



(d) Noise case: Sparse

Figure 3.7: Lorenz oscillator: Norm of the relative error for different scenarios

3.4.3 Two-Body Keplerian Dynamics

The third example corresponds to the identification of the inverse square Law of Gravitation from the observation data corresponding to a satellite orbiting the Earth. Kepler's Laws, Newton's Laws of motion and Newton's Gravitational Law were developed with critical reliance on observational data. Based on Giuseppe Piazzi's observations of Ceres in 1801, Gauss calculated the orbit of Ceres from the observation data for Ceres using the theory of least squares and initiated the premises of data driven models. In this section, the classical problem of identifying the central force field from position-only observation data is considered to validate the developed approach. In addition, the results from the methods developed in this paper are compared with the same analysis performed with a multi-layered NN based approach. Previous work conducted in this respect [73] and [74] will allow us to precisely compare the machine learning based approach with the sparse approximation method.

Table 3.7: Lorenz problem: RMS Error on 10 random initial conditions

Solution	**No Noise Case**	**Noise Case**
Least-squares	$9.369 \cdot 10^{-6}$	1.418
Sparse	$1.704 \cdot 10^{-7}$	$3.117 \cdot 10^{-3}$

Table 3.8: Orbital elements for the four considered orbits of the training data set

Orbital elements	**ISS**	**Molniya**	**Polar**	**GEO**
Semi-major axis a [m]	6789500	26600000	9240000	42164000
Eccentricity e	0.0001912	0.74	0.00025	0
Inclination i [deg]	51.6414	63.4	89.8	0.01
RAAN Ω [deg]	259.0449	128	120	120
Arg. of perigee ω [deg]	182.9557	270	360	360
True anomaly ν [deg]	0	0	0	0

3.4.3.1 The Two-Body Problem in Cartesian Coordinates

Let $\boldsymbol{r}_1$ and $\boldsymbol{r}_2$ be the position vector of two bodies, and m_1 and m_2 be their respective mass. If $\boldsymbol{r} = \boldsymbol{r}_2 - \boldsymbol{r}_1$ is the relative position vector between the two bodies, the dynamics of the two-body problem are given by

$$\ddot{\boldsymbol{r}} = -\frac{\mu}{r^3}\boldsymbol{r}, \tag{3.43}$$

with $\mu = G(m_1 + m_2)$ and G is the universal gravitational constant. In an inertial reference frame and using Cartesian coordinates, with $\boldsymbol{r} = \{x,\ y,\ z\}^T$ and $r = \sqrt{x^2 + y^2 + z^2}$, Eq. (6.13) can be written as

$$\boldsymbol{f} : \boldsymbol{r} \mapsto \ddot{\boldsymbol{r}} \Leftrightarrow \boldsymbol{f} : \begin{bmatrix} x \\ y \\ z \end{bmatrix} \mapsto \begin{bmatrix} \ddot{x} \\ \ddot{y} \\ \ddot{z} \end{bmatrix} = -\frac{\mu}{r^3} \begin{bmatrix} x \\ y \\ z \end{bmatrix}. \tag{3.44}$$

The idea is to identify the governing equations of the function $\boldsymbol{f}$ introduced in Eq. (3.44) without any a priori knowledge about its structure and therefore to determinate if the resulting identified model has the ability to identify the underlying dynamics embedded in some data set.

3.4.3.2 Training Set & Dictionary of Basis Functions

Four different types of orbits are selected to build the training set: a low Earth orbit (LEO), a Molniya orbit, a polar orbit and a geo-synchronous orbit. Table 3.8 summarizes their orbital elements. The training data set corresponds to a fraction of a revolution on these orbits: range data are recorded every one second for 1.2 hours for the ISS, and 2 hours for the remaining three. Canonical units are used in this example, the length unit (LU) is chosen to be the radius of the Earth, the time unit (TU) is chosen such that the gravitational parameter $\mu = 1$.

To construct the initial dictionary, the set $\mathbb{R}_3[\boldsymbol{r}]$ consisting of 20 "monomials" up to order 3 in x, y and z is defined:

$$\mathbb{R}_3[\boldsymbol{r}] = \left\{1 \;\; x \;\; x^2 \;\; x^3 \;\; y \;\; xy \;\; x^2y \;\; y^2 \;\; xy^2 \;\; y^3 \;\; z \;\; xz \;\; x^2z \;\; yz \;\; xyz \;\; y^2z \;\; z^2 \;\; xz^2 \;\; yz^2 \;\; z^3\right\}. \tag{3.45}$$

Furthermore, sets $\mathbb{R}_{3,1}[\boldsymbol{r}, r]$, $\mathbb{R}_{3,2}[\boldsymbol{r}, r]$ and $\mathbb{R}_{3,3}[\boldsymbol{r}, r]$ are defined to consist of the aforementioned mono-

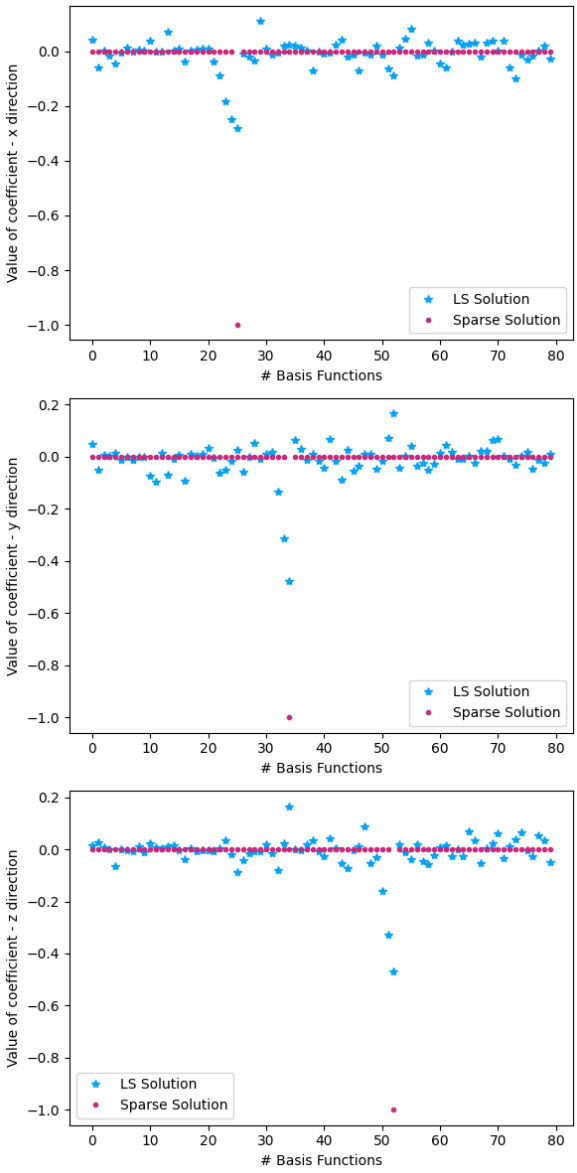


Figure 3.8: Two Body Problem: Value of the coefficients for the basis functions

mials divided by r, r^2 and r^3, i.e.,

$$\mathbb{R}_{3,1}[\boldsymbol{r}, r] = \mathbb{R}_3[\boldsymbol{r}]/r, \qquad \mathbb{R}_{3,2}[\boldsymbol{r}, r] = \mathbb{R}_3[\boldsymbol{r}]/r^2, \qquad \mathbb{R}_{3,3}[\boldsymbol{r}, r] = \mathbb{R}_3[\boldsymbol{r}]/r^3. \tag{3.46}$$

This class of basis functions is motivated by the general $1/r^n$ forms that are often seen in force interactions, especially for coservative force fields including gravity. The final dictionary consists of a total of 80 basis functions constructed as the union of $\mathbb{R}_3[\boldsymbol{r}]$, $\mathbb{R}_{3,1}[\boldsymbol{r}, r]$, $\mathbb{R}_{3,2}[\boldsymbol{r}, r]$ and $\mathbb{R}_{3,3}[\boldsymbol{r}, r]$, i.e.,

$$\mathcal{D} = \mathbb{R}_3[\boldsymbol{r}] \cup \mathbb{R}_{3,1}[\boldsymbol{r}, r] \cup \mathbb{R}_{3,2}[\boldsymbol{r}, r] \cup \mathbb{R}_{3,3}[\boldsymbol{r}, r]. \tag{3.47}$$

Table 3.9: Value of the coefficients for the sparse solution vs. the true coefficients

Basis Function	**# in the dictionary**	**True Value**	**Relative Error %**
$\boldsymbol{r} \mapsto x/r^3$	25	-1	$2.45 \cdot 10^{-11}$
$\boldsymbol{r} \mapsto y/r^3$	34	-1	$-7.21 \cdot 10^{-12}$
$\boldsymbol{r} \mapsto z/r^3$	52	-1	$-1.93 \cdot 10^{-11}$

The second-order formulation described in Section 3.3.2 is used to identify the inherent true central force field dynamics, with coefficients of the low-pass filters chosen to be $\lambda_1 = \lambda_2 = 10$. Figure 3.8 presents the coefficients found using the least-squares solution and the coefficients from the sparse approximation method. Once again the least squares method fails to clearly identify the true participating basis functions. Instead, a non-minimal combination of basis functions in the dictionary is selected to minimize the mean squared error. On the contrary, the sparse approximation approach clearly identifies the one basis function corresponding to inverse square law as reported in Table 3.9. These converged coefficients agree with the true value of μ with almost machine precision. Figure 3.9 shows the error resulting in the propagation on the training orbits using the least-square and sparse method identified dynamics. The error in the propagation is around 10^{-4} LU depending on the type of orbit for the least-squares method while an average absolute error for sparse approximation is on the order of 10^{-13} LU. The final error resulting from the least-squares solution is not satisfactory due to the excitation of basis functions that are not participating in the true dynamics.

3.4.3.3 Comparison with the Deep Learning Approach

To show the relative performance of the sparse learning methodology presented in this paper, a multi-layer NN learning-based approach is also considered. A NN can be seen as a complex nonlinear mapping between some given input and output data. Mathematically speaking, if E and F are two topological spaces, a NN is a mapping $\mathcal{M} : E \to F$ such that

$$\mathcal{M} : \boldsymbol{x} \mapsto \boldsymbol{y} = \mathcal{M}(\boldsymbol{x}), \tag{3.48}$$

where $\boldsymbol{x}$ is the input and $\boldsymbol{y}$ the output of the NN. The mapping $\mathcal{M}$ is generally nonlinear and a function of a set of parameters $\boldsymbol{\alpha}$:

$$\mathcal{M} = \mathcal{M}_{\boldsymbol{\alpha}}. \tag{3.49}$$

Along with the specific structure of the mapping $\mathcal{M}$, the set of parameters $\boldsymbol{\alpha}$ defines a NN uniquely. The notation $\boldsymbol{\alpha}$ is not chosen randomly: this set of parameters is analogous to the matrix of coefficients $\boldsymbol{\alpha}$ defined Eq. (3.5). These unknown parameters are found by minimizing the loss function

$$L(\mathcal{M}_{\boldsymbol{\alpha}}, \mathcal{S}) = \sum_{k=1}^{N} MSE(k), \quad MSE(k) = \frac{1}{m} \sum_{i=1}^{m} \left(y_k(i) - \tilde{y}_k(i)\right)^2 = \frac{1}{m} \left|\left|\boldsymbol{y}_k - \tilde{\boldsymbol{y}}_k\right|\right|_2^2 . \tag{3.50}$$

where $\tilde{\boldsymbol{y}}_k \in \mathbb{R}^m$ represents the output of the network, i.e., $\tilde{\boldsymbol{y}}_k = \mathcal{M}_{\boldsymbol{\alpha}}(\boldsymbol{x}_k)$. and the $\boldsymbol{y}_k \in \mathbb{R}^m$ represent the true measurements for the output vector. Figure 3.10 shows that how the solution from the NN and the true solution are generated to compute the Loss function to find the unknown network parameters $\boldsymbol{\alpha}$.

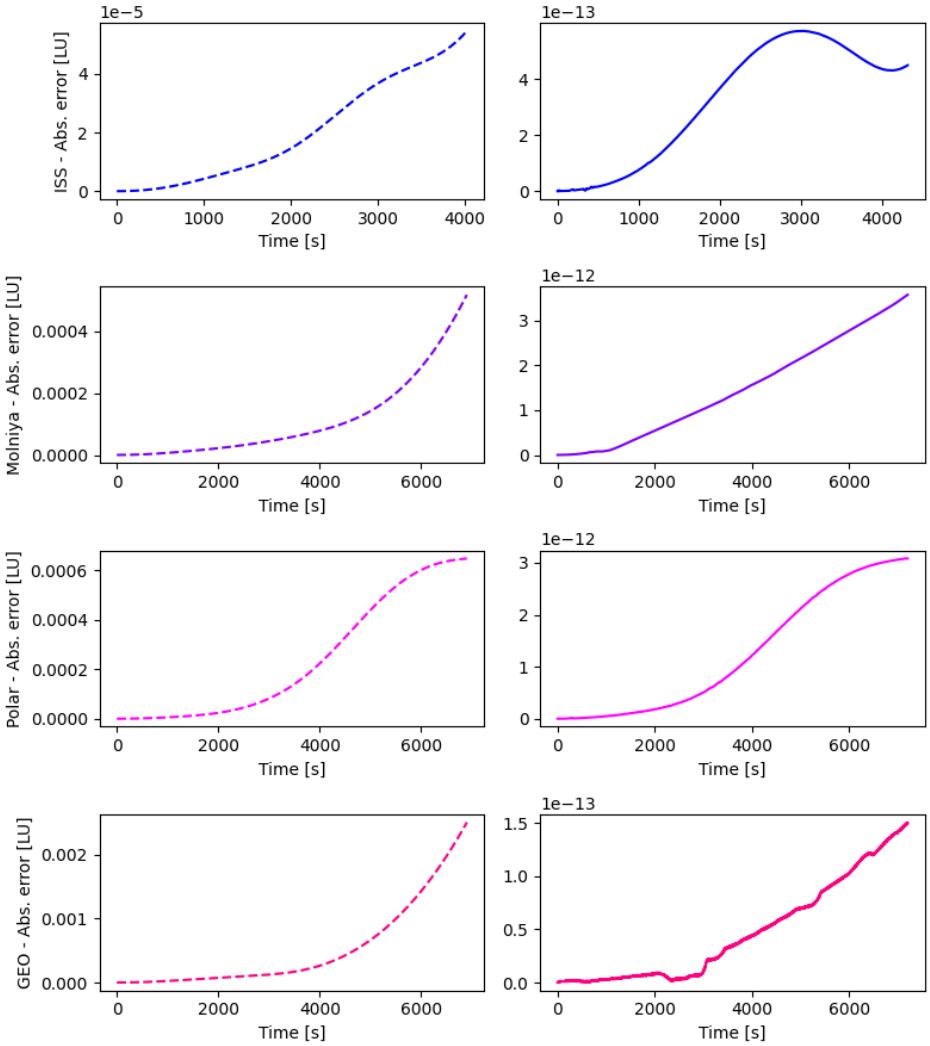


Figure 3.9: Two-body problem: Absolute relative error for radial component

Unfortunately, the resulting optimization is nonlinear in nature with multiple local minima. In our prior work [73, 74], an extensive study is conducted to understand the learning capabilities of NNs to identify the Keplerian dynamics. The goal is to examine whether the specific structure of NNs can learn the inherent dynamics of the two-body problem and examine whether a NN trained model can reproduce the well-known characteristics of Keplerian dynamics such as conservation of energy and angular momentum. Several test cases are considered to assess the learning capability of the converged NN. Three different network architectures: Feed-Forward, Residual and Deep Residual are considered in addition to studying the impact of the size of training data size on the network approximation. Table 3.10 summarizes the best results on approximating the Keplerian motion using different architectures of NN along with the results obtained with the least squares and sparse approximation in this work. It should be mentioned that the training data set for the NN is much larger than the training data set for the least-squares and sparse approximation. The training data set for the NN approximation consists of 20 random orbits with

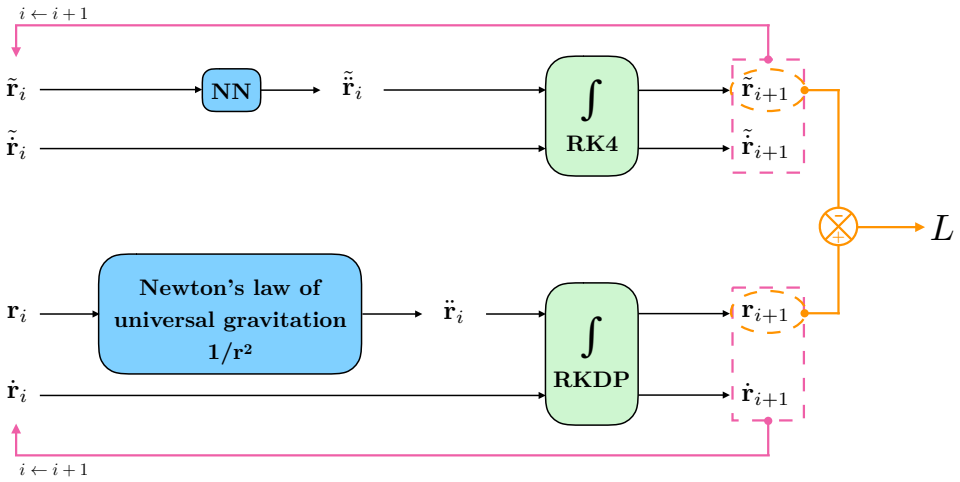


Figure 3.10: The computation of Loss Function for NN approximation: the upper part illustrates how the NN is used to approximate the dynamics along with the Runge-Kutta fixed-size step algorithm while the bottom part is the classical generation of the true solution with the known dynamics and a Dormand-Prince integration algorithm.

multiple revolutions. It has been shown that the three structures are able to provide accurate results for orbit prediction considering a large data set. Although it seems that the NN-learned model can be trained to approximate the Keplerian dynamics to good accuracy, the complexity of the learned model is an issue. The resulting NN model is a profligate model for the Keplerian dynamics as compared to Newton's Law of Gravitation. While the DeepResNet structure considered provides the most accurate results, the ResNet architecture shows very similar performance with fewer parameters. However, none of these architectures is able to identify the parsimonious structure of the governing dynamics as identified by the sparse approximation algorithm. This is because methods like NN improve the approximation accuracy by increasing the complexity of the model (defined by the parameters of the model) while fixing the basic building blocks or the basis function whereas the developed methodology along with other variants such as [45] improves the approximation capability by judiciously selecting the basis functions.

Table 3.10: Comparison of different methods to approximate the Keplerian dynamics

Method	Feed-Forward NN	Residual NN	Deep Residual NN	Least-Squares	Sparse
Average error [LU]	$7 \cdot 10^{-8}$	$4 \cdot 10^{-8}$	$3 \cdot 10^{-8}$	$2 \cdot 10^{-4}$	$4 \cdot 10^{-13}$
Number of parameters	2070	480	7720	80	3

3.5 Conclusion

This chapter has presented a convex optimization based approach for nonlinear system identification from state and control input time histories. The proposed methodology expands the unknown nonlinearities in system dynamics in terms of basis functions consisting of monomials of various orders. An integral form of the underlying nonlinear ordinary differential equations is considered to solve for the unknown coefficients for the basis functions. While conventional methods for nonlinear system identification rely

on improving the approximation accuracy by increasing the number of basis functions and hence the parameters of the model, the developed approach exploits recent advances in sparse approximation to automatically select the appropriate structure for the inherent nonlinearities. Hence, the developed methodology chooses building blocks for accurate and efficient construction of the input-output map. Though the main ideas are developed for a first order system to identify the input-output map from the time histories for full state (i.e., position and velocity) as well as control input vector, an extension is developed for the identification of second order systems where nonlinearities are a function of only position-level state variables from time histories of only position-level state variables and control input. Three numerical examples are presented to provide evidence in support of the efficacy of the proposed approach. The unique agreement in correctly identifying the true dynamics for all the three problems provides a strong basis for optimism in demonstrating the utility of the approach for identifying the inherent physics-based map from given data. The next chapter will propose a robust method for the identification of a time varying Koopman operator in the discrete-time domain based on a subspace identification method. In a similar manner to the framework presented in this chapter, this will lead to the identification of the most prevalent modes activated in the collected data.

Chapter 4
Discrete-Time Koopman Operator

4.1 Introduction

While the previous chapter proposed a method to answer the question on how to derive a representation of a continuous-time Koopman operator by selecting an optimal number of basis functions (in terms of an $\ell - 1$ norm minimization problem), the purpose of this chapter is to deliver an automatic algorithm to identify a discrete-time Koopman operator using subspace methods from data only. In the Koopman framework, given the time histories of $\boldsymbol{\chi}_k$, we are able to construct a minimal discrete-time state-space realization:

$$\boldsymbol{z}_{k+1} = A\boldsymbol{z}_k \tag{4.1}$$

$$\hat{\boldsymbol{\chi}}_k = C\boldsymbol{z}_k \tag{4.2}$$

where $\hat{\boldsymbol{\chi}}_k$ is the estimated measurement vector. Though the Koopman operator is in theory of infinite dimension, the measurement vector $\boldsymbol{\chi}_k$ is truncated to finite dimension $N >> n$ for practical implementation. In other Koopman operator related publications, $\boldsymbol{z}_k$ is assumed to be a N-dimensional hidden state vector corresponding to the Koopman dynamics, hence of dimension equal to the dimension of the lifted space. However, one can also estimate the dimension of $\boldsymbol{z}_k$ as part of the identification process. $\hat{\boldsymbol{\chi}}_k$ is the estimated measurement vector and the estimate for the state of the nonlinear system, can be extracted from the first n components of $\hat{\boldsymbol{\chi}}_k$. The observable pair of unknown system matrices (A, C) is found such that the norm of the measurement error $\|\boldsymbol{\chi}_k - \hat{\boldsymbol{\chi}}_k\|$ is minimized. As we've explained in this section, conventional subspace decomposition methods such as the Eigensystem Realization Algorithm (ERA) [8, 13, 21] or Dynamic Mode Decomposition (DMD) [20, 23, 75] provide an *observable* realizations for the system matrices (A, C), hence performing a linearization about a single point of the dynamical system in the lifted space. In earlier work [54–58], it is shown that the state prediction error improves as the dimension of the lifted space, N is increased. Indeed the resulting linear operator is a local approximator of the nonlinear dynamical system valid in the neighborhood of a nominal point and the domain of validity of this local linear approximation improves as the dimension of the lifting space is increased or equivalently the accuracy of the Koopman operator improves as the dimension of the

operator (or, equivalently, the related lifted space) is increased. Generally, N needs to be much larger than state dimension n for the Koopman operator to provide a good prediction of the system states. In other words, one may need a very large dimensional lifting space to accurately capture the flow of the underlying nonlinear system.

Moving forward, the work presented in the next section aims to extend subspace identification methods for the identification of a time varying Koopman operator. Particularly, the linearization of the nonlinear flow about a nominal trajectory of the nonlinear system rather than a nominal point is presented as an alternative to improve the validity region of the Koopman operator and curtails the dimension of the lifting space. The linearization about a nominal trajectory leads to a linear time-varying (LTV) system as opposed to a linear time-invariant (LTI) system for the conventional Koopman operator. Instead, a time-varying Koopman operator (TVKO) is developed as an alternate means to increase the prediction accuracy for a fixed dimension of the lifted space. Three numerical simulations are considered to showcase the utility of this newly developed time-varying Koopman operator in predicting the response of a nonlinear system.

4.2 Time-Varying Koopman Operator (TVKO)

The algorithm developed in this section arises from a perspective of generalizing the classical Ho-Kalman approach with OKID/ERA to the case of time-varying systems. It is shown that the generalization thus made enables the identification of time varying plant models that are in arbitrary coordinate systems at each time step. Furthermore, the coordinate systems at successive time steps are compatible with one another and makes the model sequences realized, useful in state propagation.

Earliest efforts in the development of methods for linear time-varying systems involved recursive and fast implementations of time invariant methods by exploring structural properties of the input/output realizations. The classic paper by Chu et. al, exploring the displacement structure in the Hankel matrices is representative of the efforts of this nature. Subsequently, significant results were obtained by Shokoohi and Silverman [24] and Dewilde and Van der Veen [25], that generalized several concepts in the classical linear time invariant system theory consistently. The idea of repeated experiments have been introduced [26, 27] and presented as practical methods to realize the conceptual state-space model identification strategies. However, LTV systems exhibit distinct properties, as compared to the shift invariance exhibited by LTI systems. All the subspace methods for LTI system identification exploit the fact that an infinity of system realizations exist and actually share the same Markov parameters and the eigenvalues of the state transition matrix. However, no such property exists for LTV systems and the lack of similarity transformations handicaps the application of conventional subspace methods to identify LTV systems. In our earlier work [28, 29], it is shown that there actually exists special reference frames, in which the identified models are similar to the true model, i.e., state transition matrices share the same eigenvalues, and the transformations applied allows one to compare similar state matrices. This forms the basis for spectral characterization of time-varying systems and the resulting algorithm to identify the system matrices is known as the time-varying eigensystem realization algorithm (TVERA). This section summarizes the key ideas of the TVERA algorithm and one should refer to [28] for more details on TVERA.

4.2.1 Introduction on Linear Discrete-Time Time-Varying State-Space Models

A linear discrete-time varying system is given by

$$\boldsymbol{x}_{k+1} = A_k \boldsymbol{x}_k + B_k \boldsymbol{u}_k \tag{4.3a}$$

$$\boldsymbol{y}_k = C_k \boldsymbol{x}_k + D_k \boldsymbol{u}_k \tag{4.3b}$$

together with an initial state vector $\boldsymbol{x}_0$, where $\boldsymbol{x}_k \in \mathbb{R}^n$, $\boldsymbol{u}_k \in \mathbb{R}^r$ and $\boldsymbol{y}_k \in \mathbb{R}^m$ are the state, control input and output vectors respectively, $k \geq 0$. Similarly to the time invariant case, the time-varying (non constant) matrices A_k, B_k, C_k and D_k with appropriate dimensions represent the internal operation of the linear system, and are used to determine the system's response to any input. The solution of the difference equation, given in Eq. (D.1) in the time varying case, is now given by

$$\boldsymbol{x}_k = \Phi_{k,0} \boldsymbol{x}_0 + \sum_{i=0}^{k-1} \Phi_{k,i+1} B_i \boldsymbol{u}_i, \tag{4.4a}$$

$$\boldsymbol{y}_k = C_k \Phi_{k,0} \boldsymbol{x}_0 + \sum_{i=0}^{k-1} C_k \Phi_{k,i+1} B_i \boldsymbol{u}_i + D_k \boldsymbol{u}(k), \tag{4.4b}$$

where the state transition matrix is defined in terms of its components by

$$\Phi_{k,k_0} = \begin{cases} A_{k-1} A_{k-2} \dots A_{k_0} & \text{for } k > k_0, \\ I & \text{for } k = k_0, \\ \text{undefined} & \text{for } k < k_0. \end{cases} \tag{4.5}$$

This state transition matrix is associated with the homogeneous part of Eq. (D.2) ($\boldsymbol{u}_k = \boldsymbol{0}$, initial condition response). By defining the generalized Markov parameters (or pulse response matrix) as

$$h_{k,i} = \begin{cases} C_k \Phi_{k,i+1} B_i & \text{for } i < k-1, \\ C_k B_{k-1} & \text{for } i = k-1, \\ 0 & \text{for } i > k-1, \end{cases} \tag{4.6}$$

the input-output relationship in terms of this two index coefficients is written as

$$\boldsymbol{y}(k) = C_k \Phi_{k,0} \boldsymbol{x}_0 + \sum_{i=0}^{k-1} h_{k,i} \boldsymbol{u}_i + D_k \boldsymbol{u}_k. \tag{4.7}$$

For time-invariant systems, $h_{k,i}$ depends on $k - i$ only.

4.2.2 Time-Varying Eigensystem Realization Algorithm (TVERA)

Similarly to the time-invariant case, Hankel matrices will play a major role during the identification process. The classical Hankel matrix becomes a generalized Hankel matrix dependent of time, populated

using the generalized Markov parameters:

$$\boldsymbol{H}_k^{(p,q)} = \begin{bmatrix} h_{k,k-1} & h_{k,k-2} & \cdots & h_{k,k-q} \\ h_{k+1,k-1} & h_{k+1,k-2} & \cdots & h_{k+1,k-q} \\ \vdots & \vdots & \ddots & \vdots \\ h_{k+p-1,k-1} & h_{k+p-1,k-2} & \cdots & h_{k+p-1,k-q} \end{bmatrix} = \boldsymbol{O}_k^{(p)} \boldsymbol{R}_{k-1}^{(q)}, \tag{4.8}$$

with the time varying observability and controllability matrices defined as

$$\boldsymbol{O}_k^{(p)} = \begin{bmatrix} C_k \\ C_{k+1}A_k \\ C_{k+2}A_{k+1}A_k \\ \vdots \\ C_{k+p-1}A_{k+p-2}\dots A_k \end{bmatrix}, \tag{4.9a}$$

$$\boldsymbol{R}_k^{(q)} = \begin{bmatrix} B_k & A_kB_{k-1} & A_kA_{k-1}B_{k-2} & \cdots & A_k\dots A_{k-q+2}B_{k-q+1} \end{bmatrix}. \tag{4.9b}$$

Parameters p and q are chosen such that the generalized Hankel matrix retains the rank n, the true state dimension. Indeed, if $pm \geq n$ and $qr \geq n$, matrices $\boldsymbol{R}_k^{(q)}$ and $\boldsymbol{O}_k^{(p)}$ are of rank maximum n. If the system is controllable and observable, the block matrices $\boldsymbol{R}_k^{(q)}$ and $\boldsymbol{O}_k^{(p)}$ are of rank n and so is $\boldsymbol{H}_k^{(p,q)}$. Again, identifying the number of dominant singular values of the Hankel matrix will provide an indication about the unknown order of the reduced model to be identified. Differing ranks are possible for this generalized time-varying Hankel matrix $\boldsymbol{H}_k^{(p,q)}$ at every time step for the variable state dimension problem. However, it is assumed that the state dimension does not change with the time index and it is not difficult to see that this assumption can be relaxed, given some adjustments. We retain the assumption owing to our focus on mechanical systems, in which the connection between physical degrees of freedom and the number of state variables allows us to hold the dimensionality of the state space fixed throughout the time interval of interest. Note that the definition of the generalized Hankel matrix above is only valid for $k \geq q$. In practical experiments, inputs cannot be applied at negative time index and generalized Markov parameters have no meaningful sense for negative indexes.

As for the ERA in the time-invariant case, using the singular value decomposition of $\boldsymbol{H}_k^{(p,q)}$, we can write

$$\boldsymbol{H}_k^{(p,q)} = \boldsymbol{U}_k\boldsymbol{\Sigma}_k\boldsymbol{V}_k^\top = \begin{bmatrix} \boldsymbol{U}_k^{(n)} & \boldsymbol{U}_k^{(0)} \end{bmatrix} \begin{bmatrix} \boldsymbol{\Sigma}_k^{(n)} & \boldsymbol{0} \\ \boldsymbol{0} & \boldsymbol{\Sigma}_k^{(0)} \end{bmatrix} \begin{bmatrix} {\boldsymbol{V}_k^{(n)}}^\top \\ {\boldsymbol{V}_k^{(0)}}^\top \end{bmatrix} \tag{4.10a}$$

$$= \boldsymbol{U}_k^{(n)}\boldsymbol{\Sigma}_k^{(n)}{\boldsymbol{V}_k^{(n)}}^\top + \underbrace{\boldsymbol{U}_k^{(0)}\boldsymbol{\Sigma}_k^{(0)}{\boldsymbol{V}_k^{(0)}}^\top}_{\simeq \boldsymbol{0}} \tag{4.10b}$$

$$\simeq \boldsymbol{U}_k^{(n)}\boldsymbol{\Sigma}_k^{(n)}{\boldsymbol{V}_k^{(n)}}^\top \tag{4.10c}$$

at a given time step k. In terms of the corresponding controllability and observability matrices,

$$\boldsymbol{H}_k^{(p,q)} = \boldsymbol{U}_k^{(n)} \boldsymbol{\Sigma}_k^{(n)} \boldsymbol{V}_k^{(n)\top} = \boldsymbol{O}_k^{(p)} \boldsymbol{R}_{k-1}^{(q)} \Rightarrow \begin{cases} \boldsymbol{O}_k^{(p)} & = \boldsymbol{U}_k^{(n)} \boldsymbol{\Sigma}_k^{(n)\,1/2} \\ \boldsymbol{R}_{k-1}^{(q)} & = \boldsymbol{\Sigma}_k^{(n)\,1/2} \boldsymbol{V}_k^{(n)\top} \end{cases} . \tag{4.11}$$

The same procedure at time step $k+1$ will lead to

$$\boldsymbol{H}_{k+1}^{(p,q)} = \boldsymbol{U}_{k+1}^{(n)} \boldsymbol{\Sigma}_{k+1}^{(n)} \boldsymbol{V}_{k+1}^{(n)\top} = \boldsymbol{O}_{k+1}^{(p)} \boldsymbol{R}_k^{(q)} \Rightarrow \begin{cases} \boldsymbol{O}_{k+1}^{(p)} & = \boldsymbol{U}_{k+1}^{(n)} \boldsymbol{\Sigma}_{k+1}^{(n)\,1/2} \\ \boldsymbol{R}_k^{(q)} & = \boldsymbol{\Sigma}_{k+1}^{(n)\,1/2} \boldsymbol{V}_{k+1}^{(n)\top} \end{cases} . \tag{4.12}$$

Considering the block shifted controllability matrix

$$\boldsymbol{R}_k^{(q)\leftarrow} = \begin{bmatrix} A_k B_{k-1} & A_k A_{k-1} B_{k-2} & \cdots & A_k \dots A_{k-q+1} B_{k-q} \end{bmatrix} \tag{4.13a}$$

$$= A_k \boldsymbol{R}_{k-1}^{(q)}, \tag{4.13b}$$

and block shifted observability matrix

$$\boldsymbol{O}_k^{(p)\uparrow} = \begin{bmatrix} C_{k+1} A_k \\ C_{k+2} A_{k+1} A_k \\ \vdots \\ C_{k+p-1} A_{k+p-2} \dots A_k \end{bmatrix} = \boldsymbol{O}_{k+1}^{(p)} A_k, \tag{4.14}$$

we get

$$\hat{A}_k = \boldsymbol{O}_{k+1}^{(p)\,\dagger} \boldsymbol{O}_k^{(p)\uparrow} = \boldsymbol{R}_k^{(q)\leftarrow} \boldsymbol{R}_{k-1}^{(q)\,\dagger} \tag{4.15}$$

as an estimate for the identified time-varying discrete system transition matrix. Moreover, the first r columns of $\boldsymbol{R}_k^{(q)}$ form an estimate for the identified control influence matrix,

$$\hat{B}_k = \boldsymbol{R}_k^{(q)} \boldsymbol{E}_r. \tag{4.16}$$

Similarly, the first m rows of $\boldsymbol{O}_k^{(p)}$ give an estimate for the identified output influence matrix is

$$\hat{C}_k = \boldsymbol{E}_m^\top \boldsymbol{O}_k^{(p)}. \tag{4.17}$$

4.2.3 Calculation of Discrete-Time Varying Realizations for the First Few Time Steps

As explained before, the definition of the generalized Hankel matrix is only valid for $k \geq q$. The methodology detailed in the previous section can only be employed once a full rank Hankel matrix can be populated. This section presents a method for computing the first few time step models using an additional set of experimental data, the free response experiments. The output data of N free response

experiments (also known as the zero input response) are given by

$$\tilde{\boldsymbol{H}}_k^{(p,N)} = \begin{bmatrix} \boldsymbol{y}_k^{\#1} & \boldsymbol{y}_k^{\#2} & \cdots & \boldsymbol{y}_k^{\#N} \\ \boldsymbol{y}_{k+1}^{\#1} & \boldsymbol{y}_{k+1}^{\#2} & \cdots & \boldsymbol{y}_{k+1}^{\#N} \\ \vdots & \vdots & \ddots & \vdots \\ \boldsymbol{y}_{k+p-1}^{\#1} & \boldsymbol{y}_{k+p-1}^{\#2} & \cdots & \boldsymbol{y}_{k+p-1}^{\#N} \end{bmatrix} = \boldsymbol{O}_k^{(p)} \boldsymbol{X}_k^{(N)}, \tag{4.18}$$

where $\boldsymbol{X}_k^{(N)}$ is a state variable ensemble at time k:

$$\boldsymbol{X}_k^{(N)} = \begin{bmatrix} \Phi(k,0)\boldsymbol{x}_0^{\#1} & \Phi(k,0)\boldsymbol{x}_0^{\#2} & \cdots & \Phi(k,0)\boldsymbol{x}_0^{\#N} \end{bmatrix} \in \mathbb{R}^{n \times N}. \tag{4.19}$$

As for the procedure when $k \geq q$, using the singular value decomposition of $\tilde{\boldsymbol{H}}_k^{(p,N)}$, we can write

$$\tilde{\boldsymbol{H}}_k^{(p,N)} = \boldsymbol{U}_k \boldsymbol{\Sigma}_k \boldsymbol{V}_k^\top = \begin{bmatrix} \boldsymbol{U}_k^{(n)} & \boldsymbol{U}_k^{(0)} \end{bmatrix} \begin{bmatrix} \boldsymbol{\Sigma}_k^{(n)} & \boldsymbol{0} \\ \boldsymbol{0} & \boldsymbol{\Sigma}_k^{(0)} \end{bmatrix} \begin{bmatrix} \boldsymbol{V}_k^{(n)\top} \\ \boldsymbol{V}_k^{(0)\top} \end{bmatrix} \tag{4.20a}$$

$$= \boldsymbol{U}_k^{(n)} \boldsymbol{\Sigma}_k^{(n)} \boldsymbol{V}_k^{(n)\top} + \underbrace{\boldsymbol{U}_k^{(0)} \boldsymbol{\Sigma}_k^{(0)} \boldsymbol{V}_k^{(0)\top}}_{\simeq \boldsymbol{0}} \tag{4.20b}$$

$$\simeq \boldsymbol{U}_k^{(n)} \boldsymbol{\Sigma}_k^{(n)} \boldsymbol{V}_k^{(n)\top} \tag{4.20c}$$

at a given time step k. In terms of the corresponding observability and state variable ensemble matrices,

$$\tilde{\boldsymbol{H}}_k^{(p,N)} = \boldsymbol{U}_k^{(n)} \boldsymbol{\Sigma}_k^{(n)} \boldsymbol{V}_k^{(n)\top} = \boldsymbol{O}_k^{(p)} \boldsymbol{X}_k^{(N)} \Rightarrow \begin{cases} \boldsymbol{O}_k^{(p)} = \boldsymbol{U}_k^{(n)} \boldsymbol{\Sigma}_k^{(n)1/2} \\ \boldsymbol{X}_k^{(N)} = \boldsymbol{\Sigma}_k^{(n)1/2} \boldsymbol{V}_k^{(n)\top} \end{cases}. \tag{4.21}$$

The same procedure at time step $k+1$ will lead to

$$\tilde{\boldsymbol{H}}_{k+1}^{(p,N)} = \boldsymbol{U}_{k+1}^{(n)} \boldsymbol{\Sigma}_{k+1}^{(n)} \boldsymbol{V}_{k+1}^{(n)\top} = \boldsymbol{O}_{k+1}^{(p)} \boldsymbol{X}_{k+1}^{(N)} \Rightarrow \begin{cases} \boldsymbol{O}_{k+1}^{(p)} = \boldsymbol{U}_{k+1}^{(n)} \boldsymbol{\Sigma}_{k+1}^{(n)1/2} \\ \boldsymbol{X}_{k+1}^{(N)} = \boldsymbol{\Sigma}_{k+1}^{(n)1/2} \boldsymbol{V}_{k+1}^{(n)\top} \end{cases}. \tag{4.22}$$

Note that the state variable ensemble matrix $\boldsymbol{X}_{k+1}^{(N)}$ at time $k+1$ is related to the state variable ensemble matrix $\boldsymbol{X}_k^{(N)}$ at time k by

$$\boldsymbol{X}_{k+1}^{(N)} = A_k \boldsymbol{X}_k^{(N)} \tag{4.23}$$

which leads to the estimate

$$\hat{A}_k = \boldsymbol{X}_{k+1}^{(N)} \boldsymbol{X}_k^{(N)-1} \tag{4.24}$$

for the time-varying state matrix at times $k = 0, 1, \ldots, q-1$. The calculation of the corresponding $\hat{C}_k$ is accomplished by setting

$$\hat{C}_k = \boldsymbol{E}_m^\top \boldsymbol{O}_k^{(p)}. \tag{4.25}$$

Finally, the estimate for $\hat{B}_k$ is calculated by forming the partial Hankel matrices for the first few time steps ($k = 0, 1, \ldots, q-1$):

$$\boldsymbol{H}_{k+1}^{(p,1)} = \begin{bmatrix} h_{k+1,k} \\ h_{k+2,k} \\ \vdots \\ h_{k+p,k} \end{bmatrix} = \begin{bmatrix} C_{k+1}B_k \\ C_{k+2}A_{k+1}B_k \\ \vdots \\ C_{k+p}A_{k+p-1}\ldots A_{k+1}B_k \end{bmatrix} = \boldsymbol{O}_{k+1}^{(p)} B_k, \tag{4.26}$$

leading to

$$\hat{B}_k = {\boldsymbol{O}_{k+1}^{(p)}}^\dagger \boldsymbol{H}_{k+1}^{(p,1)}. \tag{4.27}$$

This procedure is actually the one we use to identify matrices $\hat{A}_k$ and $\hat{C}_k$ in case of initial condition response experiments only (Eq. (4.18) to Eq. (4.25)).

In the presence of an input, the model matrices determined Eq. (D.32), Eq. (D.33) and Eq. (D.35) are of little use in practice. This is because of the fact that the first few models developed in this manner ($k < q$) are in a totally different coordinate system, derived from the free response singular value decomposition. Models for $k \geq q$ are in the coordinate system derived from the forced response singular value decomposition. Hence, one cannot use the models thus developed in state propagation because they have a jump discontinuity at the time step $k = q$ in their coordinate systems. The first option to alleviate this issue is to apply the coordinate transformation theory developed in the previous section. The second option is to estimate the state at time $k = q$. This approach is explained next.

4.2.4 Estimation of the State Variable at Any Time-Step

Writing the input and output from a general k^{th} time step, for p more time steps, one obtain a set of equations that can be written in a matrix form as

$$\bar{\boldsymbol{y}} = \boldsymbol{O}_k^{(p)} \boldsymbol{x}_k + \boldsymbol{\Delta}_k \bar{\boldsymbol{u}} \tag{4.28}$$

with

$$\bar{\boldsymbol{y}} = \begin{bmatrix} \boldsymbol{y}_k \\ \boldsymbol{y}_{k+1} \\ \vdots \\ \boldsymbol{y}_{k+p-1} \end{bmatrix}, \qquad \bar{\boldsymbol{u}} = \begin{bmatrix} \boldsymbol{u}_k \\ \boldsymbol{u}_{k+1} \\ \vdots \\ \boldsymbol{u}_{k+p-1} \end{bmatrix}, \tag{4.29}$$

and

$$\boldsymbol{\Delta}_k = \begin{bmatrix} D_k & & & \\ C_{k+1}B_k & D_{k+1} & & \\ \vdots & \vdots & \ddots & \\ C_{k+p-1}A_{k+p-2}\dots B_k & C_{k+p-1}A_{k+p-2}\dots B_{k+1} & \cdots & D_{k+p-1} \end{bmatrix}. \tag{4.30}$$

Eq. (D.36) can be solved using the least-squares solution:

$$\boldsymbol{x}_k = \boldsymbol{O}_k^{(p)^\dagger} \left[\bar{\boldsymbol{y}} - \boldsymbol{\Delta}_k \bar{\boldsymbol{u}}\right]. \tag{4.31}$$

This least-squares procedure can be applied at any time step k (especially at $k = 0$ to identify the initial condition) and the same formulation is valid for time-invariant systems using constant-time matrices.

4.3 Numerical Examples

This section considers three problems of varying complexity to showcase the utility of a time-varying Koopman operator (TVKO) in predicting the response of a nonlinear system. For all numerical simulations, polynomial basis functions are considered for the lifting process and a time-invariant Koopman operator (TIKO) is also identified to showcase the accuracy gained for the same degree of lifting. Depending on the dimensionality of the problem, up to six different cases are considered corresponding to different order of lifting functions to approximate the true infinite-dimensional time-invariant and time-varying Koopman operators. The goal is to compare the pure prediction capabilities of the operators and the higher-order state transition tensors with limited training data. Note that this is a pure initial condition response: testing points are not included in the training data and no true measurement of the testing trajectory is included during the prediction phase. In reality, when a true measurement is available, it is wise to reset the predicted trajectory and to depart again from that last data point. In this Chapter, we do not attempt to reinitialize the predicted trajectory; rather, we compare the capabilities of the different methods and discuss how one would use these different approaches for a pure prediction of an initial condition response.

4.3.1 Unforced Duffing

The first example corresponds to the nonlinear oscillator known as the Duffing oscillator governed by the following equations

$$\dot{x}_1 = x_2, \tag{4.32a}$$

$$\dot{x}_2 = -\delta x_2 - \alpha x_1 - \beta x_1^3 \tag{4.32b}$$

Two sets of parameters are considered corresponding to two different energy potentials, with no damping:

$$\text{Single-well potential:} \quad \alpha = 1, \beta = 1, \delta = 0, \tag{4.33a}$$

Double-well potential: $\alpha = -1, \beta = 4, \delta = 0.$ (4.33b)

Figure 4.1 presents a summary of the learning process and the identification results. Three plots show the identification capabilities of TIKO, TVKO and the propagation achieved using higher-order state transition tensors (HOSTT). The training sample set is a Gaussian distribution around the nominal with covariance $0.04I_2$. The RMSE graph at the bottom shows how the prediction errors of the different operators compare to each other. Each point on that graph corresponds to the average RMSE of 20 propagated testing trajectories with the same initial $||\delta\boldsymbol{x}_0||$, sampled uniformly around the nominal. For this particular example, the dynamics are well behaved with one equilibrium at $x = 0$ which allows for larger propagation times (20 seconds in this case) and larger initial perturbation, for training and testing. Table 4.1 explicitly gives the RMS errors for the largest initial deviation of $||\delta\boldsymbol{x}_0|| = 0.1$.
From these plots, it is clear that the accuracy of the TVKO and HOSTT operators increase with the lifting degree. While the accuracy in propagation resulting from smaller perturbations is better with HOSTT, the TVKO yields better prediction capabilities for larger initial $||\delta\boldsymbol{x}_0||$. We recall that HOSTT are calculated using true dynamical information about higher-order sensitivities, whereas TVKO only uses data from training trajectories.
The graph that shows the evolution of the RMS error for TIKO and TVKO describes how the approximation error is growing over time. It is expected that the longer the propagation time, the larger the error, which is clearly discernible from the graph. A pure prediction of 20 seconds at 50 Hz means that the operator is expected to perform 1000 prediction steps without seeing any true measurement from the testing trajectory. Depending on the desired prediction accuracy, these graphs would indicate at what time step it is relevant to include a data point from the true trajectory.

Figure 4.2 presents the same kind of summary for the double-well Duffing oscillator. Training is performed with 200 initial conditions propagated for 10 seconds at 50 Hz. In this example, the nonlinear term in the dynamics plays a more important role than in the previous case. Again, the prediction accuracy of the TVKO and HOSTT operators increase with the lifting degree and TVKO yields better accuracy for larger initial perturbations. The TIKO is challenged on this case and as prediction error builds up as time goes on, the predicted trajectory starts diverging after approximately 2 seconds of pure prediction. This is distinctly visible when looking at the evolution of the RMS error. For TIKO and TVKO, the prediction accuracy clearly increases when the order of the operator increases. Table 4.2 provides the RMS errors for the largest initial deviation of $||\delta\boldsymbol{x}_0|| = 0.1$.

Additionally, eigenvalues of the Koopman operator of order 4 are compared with the ones obtained from building a similar operator using the elements $\Phi_{i,j_1j_2j_3j_4}$ of the corresponding 5 dimensional HOSTT tensor (for a corresponding order 4). Figure 4.3 shows the evolution of these eigenvalues at different times along the nominal trajectory. Clearly these eigenvalues do not completely match; however, they happen to be close for earlier times. As time increases, more significant differences can be observed for the two operators. This is mainly explained by the fact that the higher-order state-transition tensors are a local linearization at each time step whereas the Koopman operator is calculated from using p time steps along the nominal trajectory.

Table 4.1: Single-well Duffing: RMS error for pure prediction trajectories for $||\delta\boldsymbol{x}_0|| = 0.1$

$\|\|\delta\boldsymbol{x}_0\|\| = 0.1$	Order 1	Order 2	Order 3	Order 4	Order 5	Order 6
HOSTT	$4.5 \cdot 10^{-2}$	$1.1 \cdot 10^{-2}$	$3.2 \cdot 10^{-3}$	$1.1 \cdot 10^{-3}$	N/A	N/A
TIKO	$2.0 \cdot 10^{-1}$	$2.0 \cdot 10^{-1}$	$2.0 \cdot 10^{-1}$	$1.9 \cdot 10^{-1}$	$1.9 \cdot 10^{-1}$	$1.9 \cdot 10^{-1}$
TVKO	$5.7 \cdot 10^{-2}$	$1.0 \cdot 10^{-2}$	$2.0 \cdot 10^{-3}$	$4.6 \cdot 10^{-4}$	$9.2 \cdot 10^{-5}$	$2.5 \cdot 10^{-5}$

Table 4.2: Double-well Duffing: RMS error for pure prediction trajectories for $||\delta\boldsymbol{x}_0|| = 0.1$

$\|\|\delta\boldsymbol{x}_0\|\| = 0.1$	Order 1	Order 2	Order 3	Order 4	Order 5	Order 6
HOSTT	$2.4 \cdot 10^{-1}$	$2.7 \cdot 10^{-1}$	$3.3 \cdot 10^{-1}$	$3.9 \cdot 10^{-1}$	N/A	N/A
TIKO	$3.2 \cdot 10^{-1}$	$1.8 \cdot 10^{0}$	Diverge	Diverge	Diverge	Diverge
TVKO	$2.5 \cdot 10^{-1}$	$1.3 \cdot 10^{-1}$	$8.0 \cdot 10^{-2}$	$6.9 \cdot 10^{-2}$	$5.4 \cdot 10^{-2}$	$5.2 \cdot 10^{-2}$

4.3.2 Lorenz Oscillator

The second example corresponds to the Lorenz oscillator with governing dynamic equations given as

$$\dot{x} = x\sigma(y - x), \tag{4.34a}$$

$$\dot{y} = x(\rho - z) - y, \tag{4.34b}$$

$$\dot{z} = xy - \beta z. \tag{4.34c}$$

As a more chaotic example, a pure initial condition response prediction is more challenging. For training, 240 trajectories are generated from a Gausian distribution with covariance $0.02I_3$ around the nominal, recorded at a frequency of 50 Hz for 4 seconds. Figure 4.4 shows the identification results. Three plots correspond to the identification capabilities of TIKO, TVKO and the propagation achieved using higher-order state transition tensors (HOSTT). The RMSE graph shows how the prediction errors of the different operators compare to each other with each point corresponding to the average RMSE of 16 propagated testing trajectories with the same initial $||\delta\boldsymbol{x}_0||$, sampled uniformly around the nominal. Finally, table 4.3 provides the numerical values of the RMS errors for the largest initial deviation of $||\delta\boldsymbol{x}_0|| = 0.1$.

Similarly as before, the accuracy of the TVKO and HOSTT operators increase with the lifting degree and the TVKO yields better prediction capabilities for larger initial $||\delta\boldsymbol{x}_0||$. In this example, the TIKO is not able to sustain enough prediction accuracy for the 4 seconds time span considered. The graph showing the evolution of the RMS error for TIKO shows that after 1 second (or 50 time steps), the accumulated error becomes too significant. The same graph for TVKO shows that rapid changes in dynamics around 1 and 2 seconds (switching lobes) has the effect of rapidly increasing the prediction error.

Table 4.3: Lorenz oscillator: RMS error for pure prediction trajectories for $||\delta\boldsymbol{x}_0|| = 0.1$

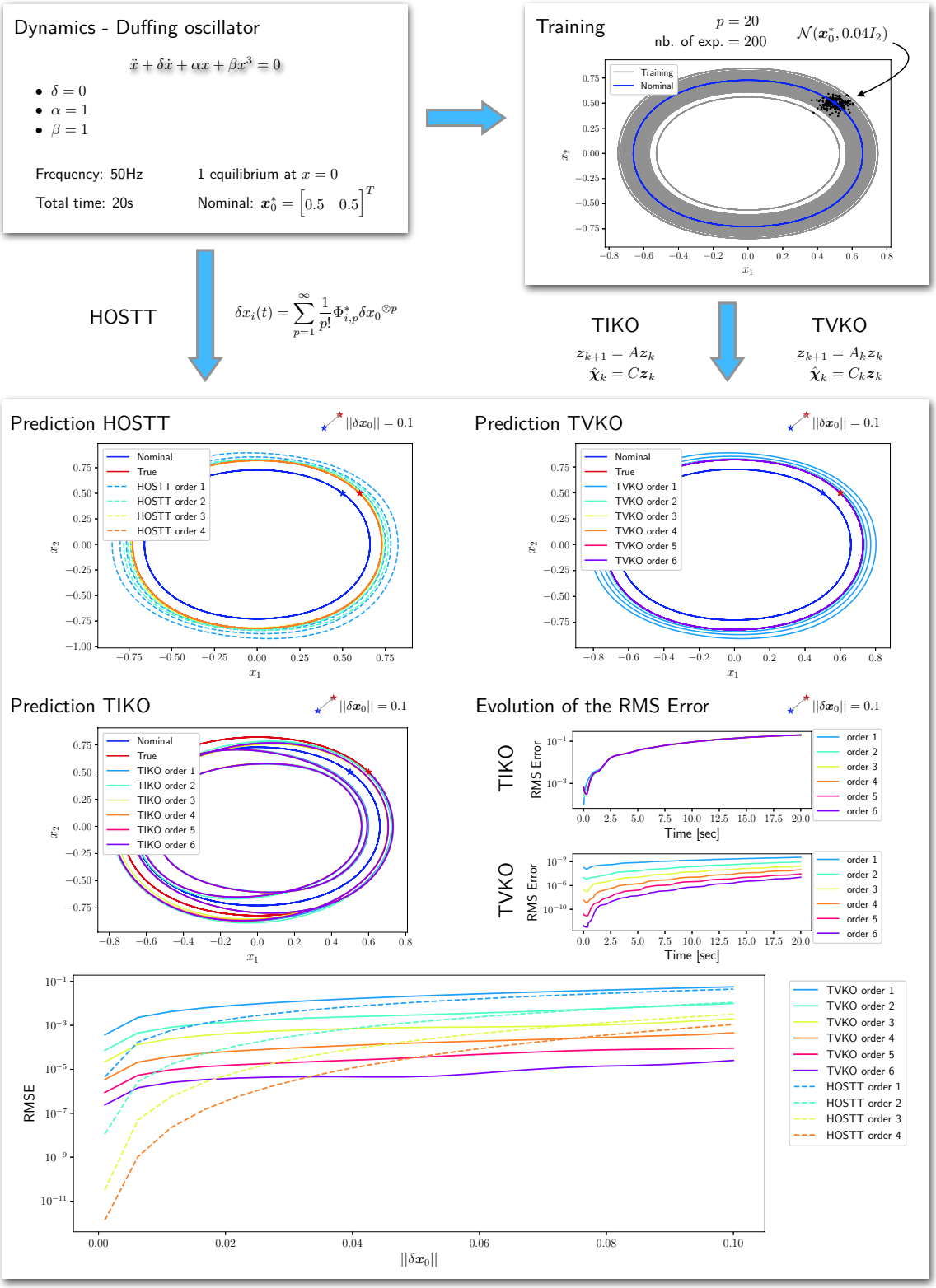


Figure 4.1: Single-well Duffing

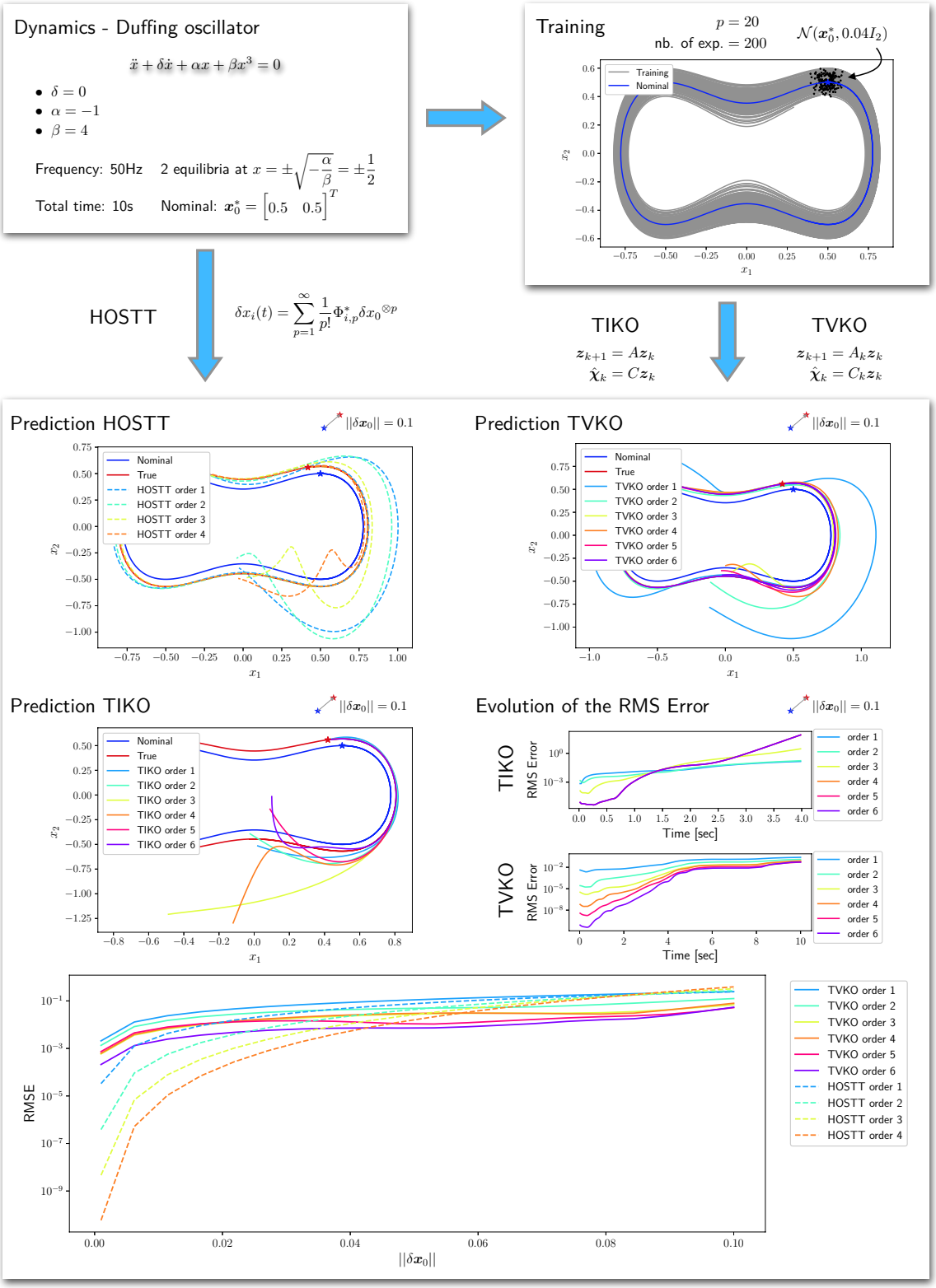


Figure 4.2: Double-well Duffing

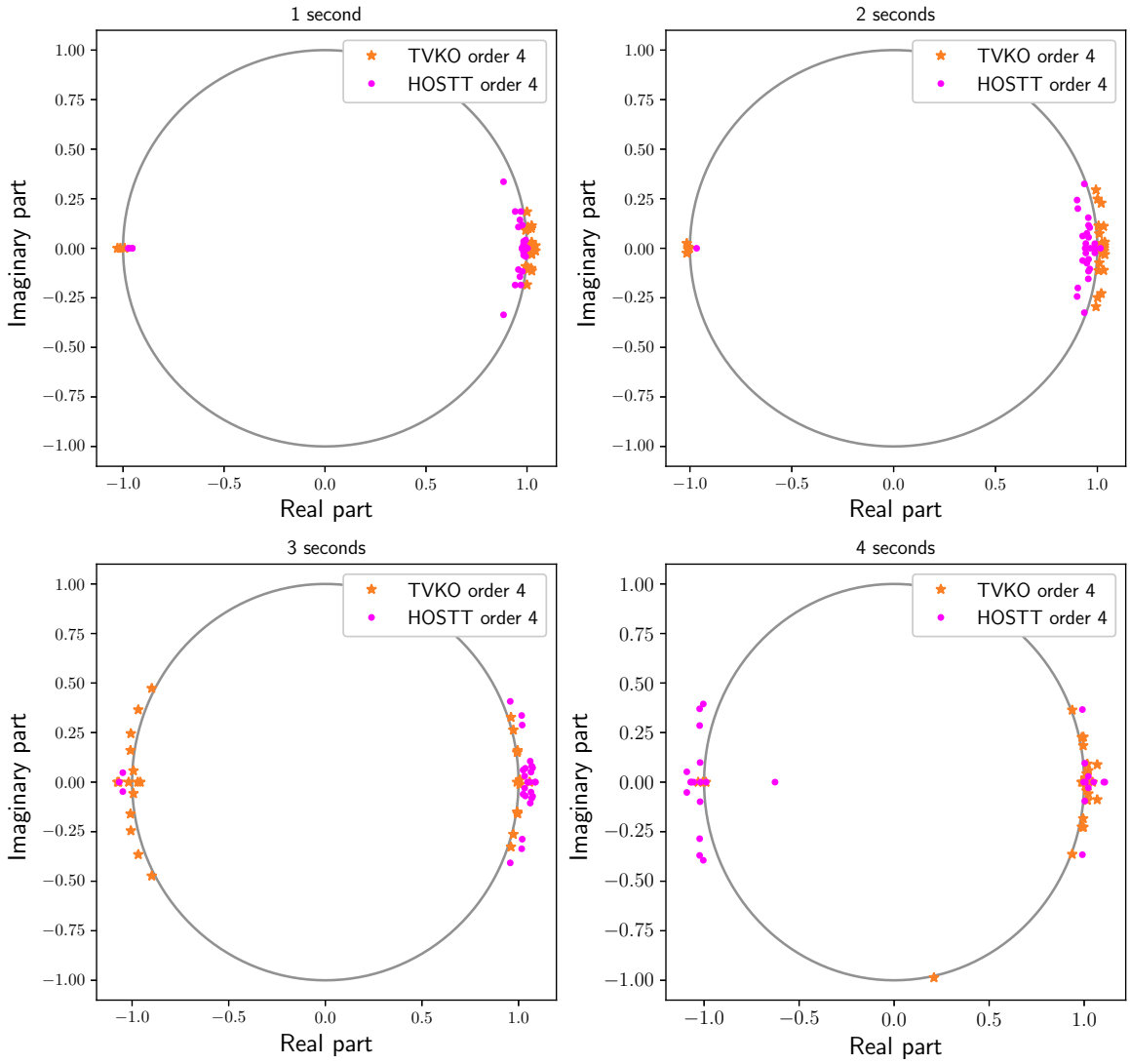


Figure 4.3: Eigenvalues of the TVKO operator and reconstructed HOSTT operator at different times

$\|\|\delta \boldsymbol{x}_0\|\| = 0.1$	Order 1	Order 2	Order 3	Order 4	Order 5	Order 6
HOSTT	$3.5 \cdot 10^{-1}$	$8.8 \cdot 10^{-2}$	$2.6 \cdot 10^{-2}$	$7.6 \cdot 10^{-3}$	N/A	N/A
TIKO	$1.2 \cdot 10^{0}$	$1.2 \cdot 10^{0}$	$1.2 \cdot 10^{0}$	Diverge	Diverge	Diverge
TVKO	$3.4 \cdot 10^{-1}$	$6.3 \cdot 10^{-2}$	$2.6 \cdot 10^{-2}$	$5.9 \cdot 10^{-3}$	$1.0 \cdot 10^{-3}$	$6.5 \cdot 10^{-4}$

These two examples of the Duffing oscillator show an additional compelling fact about the difference between a linear time-invariant Koopman operator (TIKO) and its time-varying counterpart (TVKO). For the case with one equilibrium, the linearization domain is situated around the equilibrium at $\boldsymbol{x} = \mathbf{0}$ and the Koopman approximation is valid in this region. As long as the predicted trajectory lies in this region, the approximation will remain fairly stable even though the error is slowly compounding at each time step (see Figure 4.1). Especially, the TIKO is defined uniquely and only valid in this specific region.

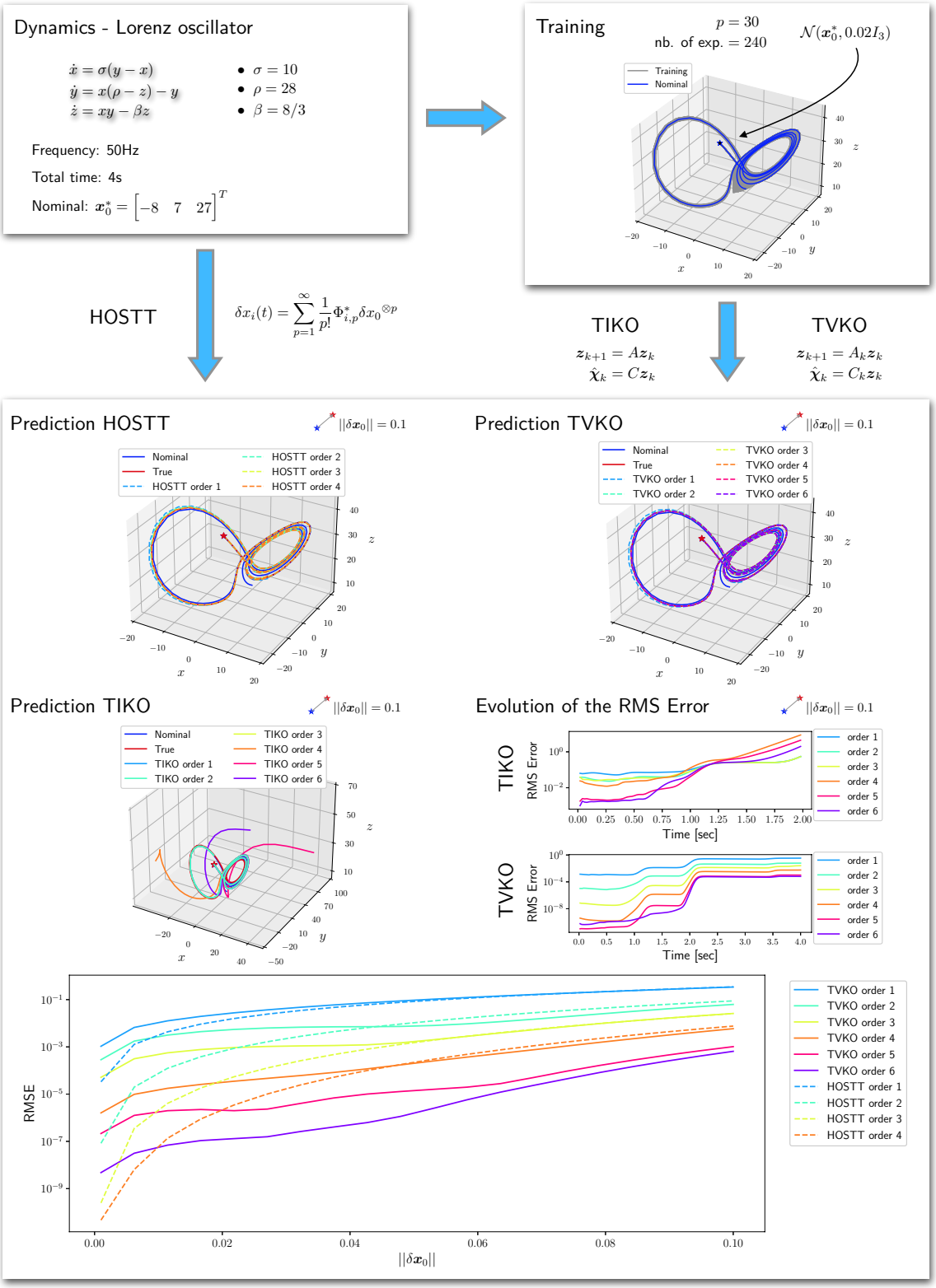


Figure 4.4: Lorenz oscillator

In the case of the double-well oscillator, it is possible to highlight two distinct (and disjoint) linearization domains around the two equilibria. As a linear operator, the TIKO cannot predict the trajectory of this two equilbria system (there is no homeomorphic coordinate transformation that captures the global dynamics of the Duffing dynamics with a linear operator, since any such linear operator has either one fixed point at the origin, or a subspace of infinitely many fixed points). Switching from one equilibrium to the other, the TIKO quickly fails in predicting the state of the system as the error grows exponentially due to reaching the limits of the linearization domain. This is also explained graphically in [76]. This is clearly seen in Figure 4.2 where the predicted trajectory starting from the right linearization domain fail to reach the left one. On the other hand, since the TVKO is a local linearization at each time-step along the nominal trajectory, it is able to seamlessly transition between the two linearization domains and offer much better prediction capabilities.

4.3.3 Flutter of an Aeroelastic Wing

This example is inspired from the work in [77] where the authors develop a structured model reference adaptive control for a wing section with structural nonlinearity, designed for active suppression of limit cycle oscillations. In this paper we are interested in the data-driven model identification of the dynamics leading to these oscillations.

4.3.3.1 Presentation of the Problem

In this example, the nonlinear aeroelastic response of a wing section, as depicted in Figures 4.5 and 4.6, is investigated. The wing is mounted vertically on a flexible support which permits two degree-of-freedom motion (see Figure 4.6). The model support system has been developed to provide direct measurements [78] of nonlinear aeroelastic response as well as to examine new control strategies of such responses [79]. A detailed description of the test apparatus is presented in [79]. The support system permits prescribed pitch and plunge motion for a mounted wing section. Plunge motion is provided by a traversing carriage. Pitch motion is provided by rotational cams that are mounted on this carriage. The model support system provides freedom in test conditions and parameters. The structural stiffness response of the system is governed by a pair of cams that are designed to provide tailored linear or nonlinear stiffness. The shape of each cam, stiffness of the springs, and pretension in the springs dictate the nature of the nonlinearity. With this approach, these cams provide a large family of prescribed stiffness configurations. Other physical properties – such as the eccentricity of the aerodynamic center, the mass of various system components, the mass eccentricity, the moment of inertia of the wing, the stiffness characteristics, and the wing shape – are easily modified for parametric investigations. This configuration also permits studies of internal resonances such as those presented in [80]. System response is measured with accelerometers and optical encoders mounted to track motion in each degree-of-freedom. The plunge and pitch displacements are measured with optical encoders attached to shafts running through the respective cams. The plunge and pitch accelerations are measured with accelerometers. Freestream velocity is determined from a pitot probe mounted in the test section.

Denoting h and α as plunge and pitch variables, the equations of motion for this aeroelastic system

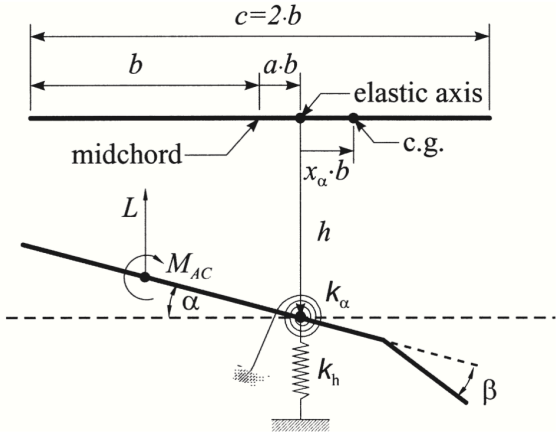


Figure 4.5: The aeroelastic model with pitch and plunge degrees of freedom

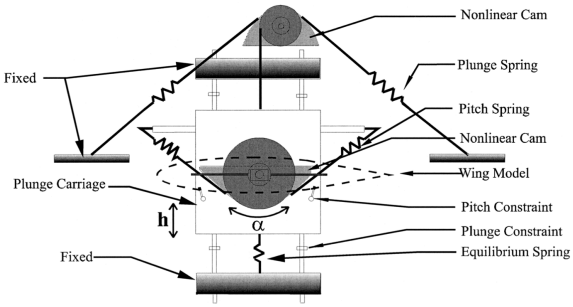


Figure 4.6: Schematic view of the experiment setup

are obtained as [81]

$$\begin{bmatrix} m_T & m_W x_\alpha b \\ m_W x_\alpha b & I_\alpha \end{bmatrix} \begin{bmatrix} \ddot{h} \\ \ddot{\alpha} \end{bmatrix} + \begin{bmatrix} c_h & 0 \\ 0 & c_\alpha \end{bmatrix} \begin{bmatrix} \dot{h} \\ \dot{\alpha} \end{bmatrix} + \begin{bmatrix} k_h & 0 \\ 0 & k_\alpha(\alpha) \end{bmatrix} \begin{bmatrix} h \\ \alpha \end{bmatrix} = \begin{bmatrix} -L \\ M \end{bmatrix}, \tag{4.35}$$

where m_T is the total mass of the wing and its support structure, m_W is the mass of the wing only, and I_a is the mass moment of inertia about the elastic axis. The elastic axis location of the model, a, may be varied and plays a significant role in the stability of the system [82]. In the above equation, $x_\alpha = r_{cg}/b$ represents the nondimensionalized distance between the center of mass and the elastic axis, and c_h and c_α are the plunge and pitch structural damping coefficients, respectively; and L and M are the aerodynamic lift and moment about the elastic axis. Structural stiffnesses are represented by k_h and k_α for plunge and pitch motions, respectively. It is possible to incorporate various nonlinear features in this design and the associated experiments. These features include aerodynamic nonlinear loads, Coulomb damping, nonlinear stiffness, and higher-order kinematics. However, in this paper, we prescribe the sole source of nonlinearity as a nonlinear torsional stiffness, which is approximated in polynomial

form as

$$k_\alpha(\alpha) = k_{\alpha 0} + k_{\alpha 1}\alpha + k_{\alpha 2}\alpha^2 + k_{\alpha 3}\alpha^3 + k_{\alpha 4}\alpha^4 + \cdots . \tag{4.36}$$

As mentioned earlier, in the experiment, the nonlinear torsional stiffness is realized by nonlinear cams, and the actual coefficients in the above polynomial representation are obtained from measured displacement and moments. Note, all polynomial terms are present in the above equation to account for the asymmetry in the measured stiffness.

There are many approaches available to represent the unsteady aerodynamic lift and moment loads. For the purpose of deriving a feedback control model for the class of systems discussed in this paper, appropriate choices may include the model in [83] or reduced order models based on frequency domain analysis. Herein, we employ a quasi-steady aerodynamic model [81]

$$L = \rho U^2 b c_{l_\alpha} \left(\alpha + \frac{\dot{h}}{U} + \left(\frac{1}{2} - a \right) b \frac{\dot{\alpha}}{U} \right) \tag{4.37a}$$

$$M = \rho U^2 b^2 c_{m_\alpha} \left(\alpha + \frac{\dot{h}}{U} + \left(\frac{1}{2} - a \right) b \frac{\dot{\alpha}}{U} \right) \tag{4.37b}$$

where U is the freestream velocity, and c_{l_α} and c_{m_α} are aerodynamic lift and moment derivatives. It is important to note that this model is proven appropriate for the low reduced frequency, subsonic flow that is observed for the experiments herein. The equations of motion in state space form are

$$\dot{\boldsymbol{x}} = \boldsymbol{f}(\boldsymbol{x}) \tag{4.38}$$

with

$$\boldsymbol{x} = \begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \end{bmatrix} = \begin{bmatrix} h \\ \alpha \\ \dot{h} \\ \dot{\alpha} \end{bmatrix} = \begin{bmatrix} \text{plunge} \\ \text{pitch} \\ \text{plunge rate} \\ \text{pitch rate} \end{bmatrix} \tag{4.39}$$

and

$$\boldsymbol{f}(\boldsymbol{x}) = \begin{bmatrix} x_3 \\ x_4 \\ -k_1 x_1 - (k_2 U^2 + p(x_2)) x_2 - c_1 x_3 - c_2 x_4 \\ -k_3 x_1 - (k_4 U^2 + q(x_2)) x_2 - c_3 x_3 - c_4 x_4 \end{bmatrix} . \tag{4.40}$$

For brevity, a set of new variables is introduced in these equations, and they are defined in Table 4.4.

As seen from the equations and Table 4.4, the nonlinear aeroelastic system is parametrically dependent on the location of the elastic axis, a (or, x_α) and the freestream velocity U. It is shown in [82, 84] that, depending on the values of these parameters, the nonlinear aeroelastic system exhibits a wide variety of bifurcation characteristics.

Table 4.4: System variables for the aeroelastic wing problem

Variable	Definition
d	$m_T I_\alpha - m_W^2 x_\alpha^2 b^2$
k_1	$I_\alpha k_h/d$
k_2	$(I_\alpha \rho b c_{l_\alpha} + m_W x_\alpha \rho b^3 c_{m_\alpha})/d$
k_3	$-m_W x_\alpha b k_h/d$
k_4	$-(m_W x_\alpha \rho b^2 c_{l_\alpha} + m_T \rho b^2 c_{m_\alpha})/d$
$p(x)$	$-m_W x_\alpha b k_\alpha(x)/d$
$q(x)$	$m_T k_\alpha(x)/d$
c_1	$(I_\alpha(c_h + \rho U b c_{l_\alpha}) + m_W x_\alpha \rho U b^3 c_{m_\alpha})/d$
c_2	$(I_\alpha \rho U b^2 c_{l_\alpha}(\frac{1}{2} - a) - m_W x_\alpha b c_\alpha + m_W x_\alpha \rho U b^4 c_{m_\alpha}(\frac{1}{2} - a))/d$
c_3	$(-m_W x_\alpha b(c_h + \rho U b c_{l_\alpha}) - m_T \rho U b^2 c_{m_\alpha})/d$
c_4	$(m_T(c_\alpha - \rho U b^3 c_{m_\alpha}(\frac{1}{2} - a)) - m_W x_\alpha \rho U b^3 c_{l_\alpha}(\frac{1}{2} - a))/d$

4.3.3.2 Results and Discussion

For the aeroelastic system discussed in this paper, the nonlinear pitch spring stiffness is measured, and the polynomial approximation of the stiffness is obtained as

$$k_\alpha(\alpha) = 6.8614(1 + 1.1438\alpha + 96.6696\alpha^2 + 9.5134\alpha^3 - 727.6641\alpha^4) \quad [N \cdot m/rad]. \tag{4.41}$$

The physical parameters of the experimental apparatus, which are used for the numerical simulation, are given in Table 4.5.

Table 4.5: System parameters for the aeroelastic wing problem

Parameter	Value
m_T	12.3870 kg
m_W	2.0490 kg
b	0.135 m
ρ	1.225 kg/m^3
r_{cg}	$0.0873 - (b + ab)$ m
I_α	$m_W r_{cg}^2 + 0.0517$ kg·m^2
c_{l_α}	6.28
c_{l_β}	3.358
c_{m_α}	$(0.5 + a)c_{l_\alpha}$
c_{m_β}	-1.94
k_h	2844.4 N/m
c_h	27.43 kg/s
c_α	0.036 kg·m^2/s
a	0.6847
U	16 m/s

The existence of limit cycle oscillations (LCO) is dependent on the initial displacement and the freestream velocity as well as parameters associated with the configuration (refer to Figure 4.5). Figure 4.7 displays the summary of the training phase and the identification results. The three propagation plots show the identification capabilities of TIKO, TVKO and the propagation achieved using higher-order state transition tensors (HOSTT). The training sample set is a Gaussian distribution around the nominal with covariance 0.01 for the plunge h and 0.001 for the pitch α, plunge rate $\dot{h}$ and pitch rate $\dot{\alpha}$. Each point on the bottom graph corresponds to the average RMSE of 20 propagated testing trajectories with the same initial $||\delta \boldsymbol{x}_0||$. For this case, the maximum order for the appended polynomial basis functions to the measurement vector is only 4 in to contain the dimension of the KOs to a reasonable size. Table 4.6 provides the RMS errors for the largest initial deviation of $||\delta \boldsymbol{x}_0|| = 0.02$. Similarly to previous examples, the accuracy of the TVKO and HOSTT operators increase with the lifting degree, and the accuracy in propagation resulting from smaller perturbations is just slightly better with HOSTT while the TVKO yields better prediction capabilities for larger initial $||\delta \boldsymbol{x}_0||$. The graph displaying the evolution of the RMS error for TIKO shows that despite good prediction accuracy during the transitional phase (before the LCO happens around 0.4 sec), the accumulated error becomes too prevalent and the reconstructed trajectory diverges. However, the TVKO yields very good accuracy for the entire time, with a clear distinction between different orders of lifting degree.

Table 4.6: Aeroelastic Wing Flutter problem: RMS error for pure prediction trajectories for $||\delta \boldsymbol{x}_0|| = 0.02$

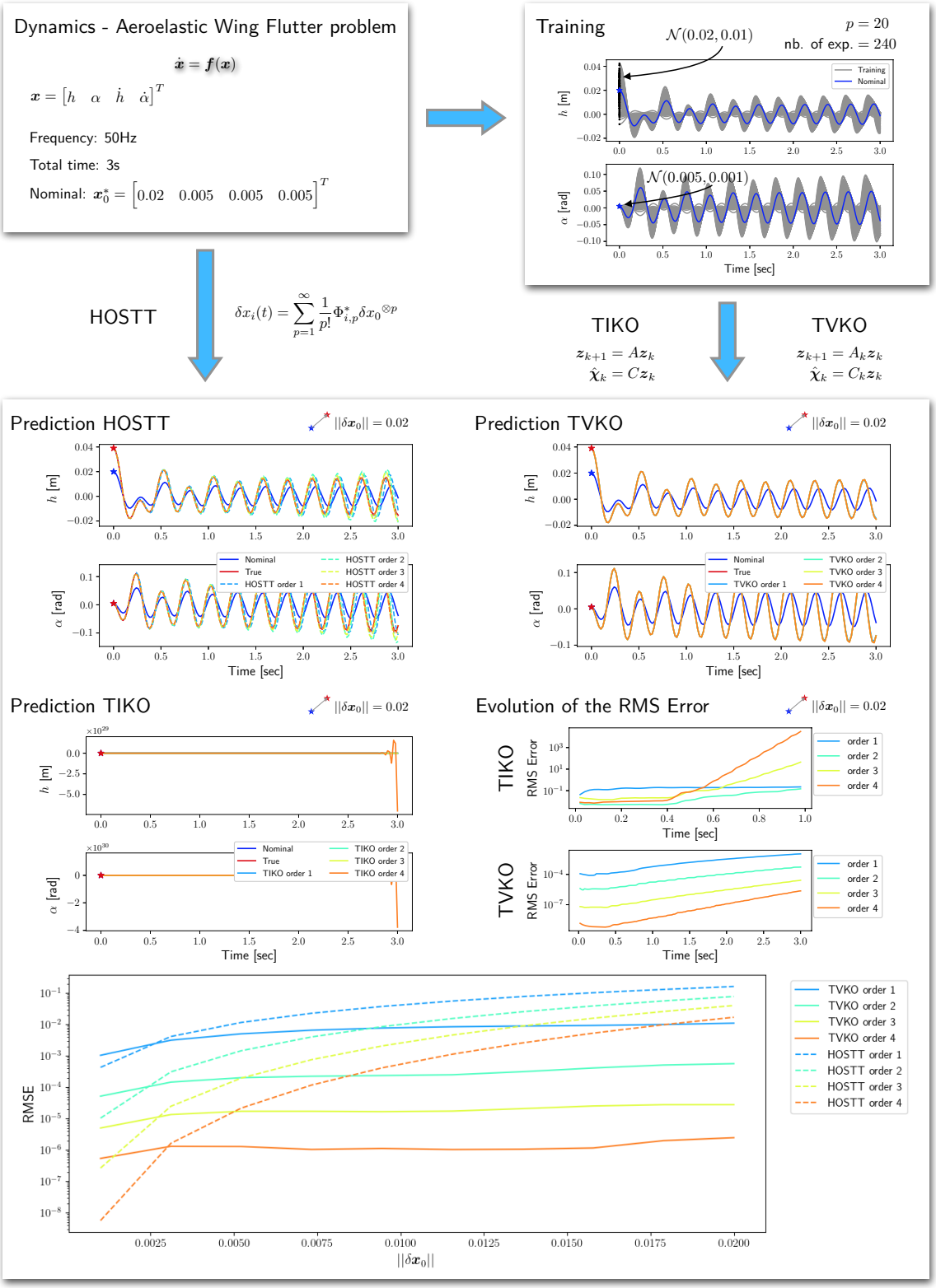


Figure 4.7: Aeroelastic wing flutter

$\|\|\delta \boldsymbol{x}_0\|\| = 0.1$	Order 1	Order 2	Order 3	Order 4
HOSTT	$1.6 \cdot 10^{-1}$	$7.9 \cdot 10^{-2}$	$4.1 \cdot 10^{-2}$	$1.7 \cdot 10^{-2}$
TIKO	$3.6 \cdot 10^{0}$	Diverge	Diverge	Diverge
TVKO	$1.1 \cdot 10^{-2}$	$5.7 \cdot 10^{-4}$	$2.8 \cdot 10^{-5}$	$2.5 \cdot 10^{-6}$

4.4 Conclusion

The work presented in this chapter allows one to extend subspace identification methods for the identification of a time varying Koopman operator. The linearization of the nonlinear flow about a nominal trajectory of the nonlinear system (instead of a nominal point for time-invariant methods) results in a substantial improvement of the validity region of the Koopman operator and curtails the dimension of the lifting space. As explained in the first section, the linearization about a nominal trajectory leads to a linear time-varying (LTV) system as opposed to a linear time-invariant (LTI) system for the conventional Koopman operator. The time-varying Koopman operator (TVKO) is developed as an alternate means to increase the prediction accuracy for a fixed dimension of the lifted space. Two versions of an academic nonlinear system, an example of a chaotic dynamical system and a more challenging aeroelastic wing flutter problem are considered and showcase the utility of this newly developed time-varying Koopman operator in predicting the response of a nonlinear system. The TVKO prediction accuracy is better in all cases, especially when the domain of initial perturbations grows. The next chapter will further highlight the application of time-varying subspace methods to high-dimensional problems and demonstrate the utility of a combined framework based on TVERA and TVKO for reduced-order modeling of high-dimensional systems.

Chapter 5
Nonlinear Model Identification from Output Data

5.1 Introduction

For many engineering applications there is a real computational challenge in developing methods and algorithms for computing invariant subspaces and their reduced dynamics. This is due to the fact that very high-dimensional nonlinear systems arising from spatial discretization of the governing partial differential equations render their computation unfeasible for realistic engineering structures (described by finite-element models for example). While previous chapters dedicated the application of the time-varying Koopman operator to low-dimensional academic examples, this chapter is devoted to show the application of time-varying subspace methods to high-dimensional problems.

The first example studies the coupled analysis between the flight dynamics, structural dynamics, heat transfer, and hypersonic aerothermodynamics, viz. AeroThermoServoElasticity (ATSE), for a hypersonic vehicles. A thorough performance analysis for ATSE is computationally intractable with high fidelity models for each discipline. Hence, there is a need to develop accurate reduced order models (ROM) for aerothermodynamics as well as thermoelasticity. The Time-Varying Eigensystem Realization Algorithm (TVERA) is used to identify a linear time varying (LTV) reduced order model from a high fidelity computational framework with guaranteed observability. The second example considers a finite element model of a geometrically nonlinear, cantilevered von Kármán beam [85, 86]. This high-fidelity simulation utilizes collocation/spectral points to describe the displacement and velocity of the beam under structural nonlinearities. The TVERA is used to first find a time-varying model on a minimal subspace that will allow for a TVKO to produce an extended model for the reduced state in a lifting space. Several orders for TVERA and TVKO are investigated and compared.

5.2 Model-Order Reduction in Hypersonic Aerothermoelasticity

Air-breathing hypersonic vehicles are under increasingly active development in recent years [87, 88]. This class of vehicles are expected to operate at high Mach number in the atmosphere for the entire mission profile that can last for 30 minutes or even longer time. Due to the high speeds, the vehicle is exposed to

the extreme aerothermodynamic environment involving combined aero-thermo-acoustical loadings. The aerothermal loads are due to the hypersonic aerodynamic pressure and heat flux. The acoustical loads are inherently stochastic and mainly due to the strong turbulent interaction present in the hypersonic boundary layer over the complex vehicle geometry. The high heating rates lead to degradation of material properties. The thermal stresses introduced by the temperature gradients and geometrical constraints affect the structural integrity and cause structural instabilities, including buckling and flutter. The thermoelastic effect further impacts the controllability of the vehicle, esp. the response effectiveness of aerodynamic control surfaces. It is clear that the coupling between the structural dynamics, heat transfer, and hypersonic aerothermodynamics, viz. aerothermoelasticity, constitutes the core subsystem governing the operation of a hypersonic vehicle. The predictive aerothermoelastic capability over extended flight time is a key ingredient for analyzing performance, stability, and reliability of hypersonic vehicles.

However, due to the current limited capability of ground tests and the lack of available flight test data, there is a significant degree of uncertainty associated with the aerothermoelastic modeling of hypersonic vehicles and limited ability to alleviate this uncertainty through experimental testing [89]. Therefore, the aerothermoelastic analysis, as a high-dimensional nonlinear multi-physics problem spanning across multiple spatial and temporal scales, involves strong stochastic dynamics as well as model uncertainties that are due to either imperfect high-fidelity models or reduced-order models. The uncertainty propagate across the coupling interfaces between the models and aggregate over time in the aerothermoelastic analysis. While there is a large body of research conducted on the uncertainty quantification (UQ) of aeroelasticity and aerothermoelasticity, the studies either focused on the calibration of models of a single discipline, or the quantification of several parametric stochastic variables in coupled analysis [90, 91]. Significant algorithmic development is required to identify, quantify, and propagate these stochastic effects and model errors through a time-dependent, high-dimensional state space, as is the case for hypersonic aerothermoelastic analysis.

Currently, the aerothermoelastic analysis is typically carried out using a computationally efficient kriging-based aerothermal surrogate coupled to nonlinear finite element models for structural dynamics and heat transfer, i.e. the thermoelastic solver [92–94]. It is relatively easy to quantify and propagate the uncertainty associated with the aerothermal loads by exploiting the mathematical formulation for kriging [95]. However, uncertainty quantification and propagation in a nonlinear dynamical system is challenging in general. As a step towards the efficient uncertainty quantification in aerothermoelastic analysis, we present a new time-varying Koopman operator (TVKO) to compliment the computational and experimental studies of hypersonic ATE. The TVKO utilizes a subspace realization method known as the time-varying Eigensystem Realization Algorithm [28] to approximate the underlying nonlinear model, such as the ATE system, as a linear time varying (LTV) model in a lifting space from time histories of input-output data from computational analysis or experiments. The linear form of the time-varying model makes it particularly amenable for the uncertainty quantification (UQ) in a computationally efficient manner.

This section aims to demonstrate the capabilities of the TVKO approach to accurately predict the ATE response of hypersonic applications. In particular, the benchmark cases include an academic problem such as the analytical flutter model of a heated panel. A high-fidelity ATE model of a hypersonic skin panel will be used to generate the input-output data. The availability of the TVKO methodology eventually will enable accurate hypersonic ATE analysis and control under uncertainty with tractable computational cost.

This section aims to demonstrate the efficacy of the proposed approach for identifying a reduced-order model of a coupled thermal-structural response in an aerothermoelastic simulation. Two cases are considered. First, a simplistic model for the flutter of a panel with prescribed increasing temperature is studied. This numerical simulation allows us to demonstrate the capability of the developed algorithm on a low order model where the measurements are of low dimension. Second, a fully-coupled nonlinear aerothermoelastic model of a hypersonic skin panel is studied. This numerical simulation demonstrates the generalizability of the TVERA algorithm to high-dimensional problems.

5.2.1 The Panel Flutter Problem

This section aims to demonstrate the efficacy of the proposed approach for identifying a reduced-order model (ROM) of a nonlinear aerothermoelastic simulation. Following previous work [96], a ROM is built for a coupled thermal-structural response for the flutter of a panel with prescribed increasing temperature. This numerical simulation allows us to demonstrate the capability of the developed algorithm on a low order model where the measurements are of low dimension.The true unknown dynamical model capturing the flutter of a heated panel is given as [97]:

$$\frac{1}{2}\pi^4 q_1(t) - 5\pi^2 R_T q_1(t) + \frac{5}{4}\pi^4 q_1^3(t) - \frac{4}{3}\lambda q_2(t) + 5\pi^4 q_1(t) q_2^2(t) + \frac{1}{2}\sqrt{\frac{\lambda\mu}{M}} q_1'(t) + \frac{1}{2} q_1''(t) = 0, \quad (5.1a)$$

$$\frac{4}{3}\lambda q_1(t) + 8\pi^4 q_2(t) - 20\pi^2 R_T q_2(t) + 5\pi^4 q_1^2(t) q_2(t) + 20\pi^4 q_2^3(t) + \frac{1}{2}\sqrt{\frac{\lambda\mu}{M}} q_2'(t) + \frac{1}{2} q_2''(t) = 0. \quad (5.1b)$$

where q_1 and q_2 are structural modal coordinates, λ is the dynamic pressure quantifying the aerodynamic loading, μ is the mass ratio quantifying the aerodynamic damping effect, R_T is the in-plane force due to the thermal stress. In general, when $R_T = 0$, there is a critical value λ_{cr}, such that the panel stays stable when $\lambda < \lambda_{cr}$, but enters limit cycle oscillation (LCO) when $\lambda > \lambda_{cr}$. When $R_T > 0$, the critical value λ_{cr} still exists. However, the panel may become statically buckled or enter chaotic response instead of being stable, when $\lambda < \lambda_{cr}$. In this example, it is assumed that $R_T = 0$ and the response of the panel is studied for a range of λ between 260 and 300. Additionally, the mass ratio and Mach number are set to $\mu = 0.01$ and $M = 5$.

For simulation purposes, the measurement data is sampled at a frequency of 100 Hz, i.e., a time step size of 0.01s, for 10 seconds. Training trajectories are simulated by random sampling of initial deviation from a zero mean Gaussian distribution with standard deviation of 0.0001 from the nominal initial condition $\overline{\boldsymbol{q}}(0) = \begin{bmatrix} 0.001 & 0 & 0 & 0 \end{bmatrix}^\intercal$. A Koopman operator is derived for different values of λ by varying λ from 260 to 300 in the increment of 1. A total of 100 experiments are performed for each value of λ to construct the TVKO operator.

Figures 5.1(a) to 5.1(f) show the identification capabilities of both the time-invariant Koopman (TI Koopman) and time-varying Koopman (TV Koopman) operators to reproduce the amplitude of the deformation modes of the panel. While the TI Koopman operator performs well before the bifurcation occurs ($\lambda < 280$), it degrades when the amplitude of the LCO increases. Particularly, the model fails to capture the transient response of the panel that transitions from initial condition to LCO. On the other hand, the TVERA procedure is able to provide a linear time-varying operator that approximates the dynamics of the nonlinear system for all values of λ.

The RMS errors are presented in Figure 5.2 for both operators for the three cases. From Fig. 5.2, it is clear that the accuracy of the time-invariant as well as time-varying Koopman operators improves when the lifting degree increases. Furthermore, the TV Koopman operator provides from one to two (for large oscillations) up to five (for small oscillations) orders of magnitude better prediction accuracy than the prediction errors corresponding to the conventional TI Koopman operator. While the accuracy of the TI Koopman operator for lifted degree 3, i.e., test case 3 is comparable to actual linearization of the nonlinear equations of motion, the prediction accuracy corresponding to the TV Koopman operator is much better than its TI counterpart for lifted degree 3.

Figure 5.3 presents the bifurcation plot corresponding to the panel flutter problem. To generate Fig. 5.3, TV Koopman approximators of order 3 are calculated for each value of λ between 260 and 300. If models for some values of λ are not available, interpolated values are calculated from the two adjacent models. This method shows a very good agreement between the identified, the interpolated and the true values because the LCO amplitude is calculated once the transient regime has vanished and the oscillatory regime has settled.

5.2.2 High-Dimensional Nonlinear Thermoelastic Simulation

Subsequently, the same type of analysis is applied to a high-dimensional nonlinear aerothermoelastic problem. The aerothermoelastic simulation is performed using the HYPATE framework, which has been extensively verified and applied to various hypersonic aerothermoelastic problems [89, 94]. In this study, the low-fidelity portion of the framework is employed. In the aerothermodynamic solver, the pressure and the heat flux are computed using full-order piston theory and Eckert's reference enthalpy method, respectively. The structural and thermal solvers are both based on the finite element formulation. The structural solver models the structural dynamics of anisotropic (i.e. composite) shallow shells with shear, geometric nonlinearity, and thermal stress. The thermal solver models heat transfer in composite shells using a layer-wise thermal lamination theory. Both solvers account for temperature-dependent material properties. The solvers of the three physical domains are solved using a second-order time-accurate loosely-coupled scheme.

In this example, a 2D skin panel configuration is considered, as shown in Figure 5.4 and Figures 5.5(a) through 5.5(j). The panel is simply supported at the leading and trailing edges. The geometrical parameters are $h = 5$mm, $a = 1$m, and $L_{le} = 1$m. The panel is made of Al7075 and the material properties are temperature dependent. The initial temperature is $T = 273K$.

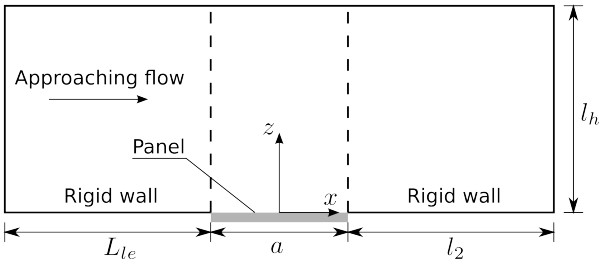
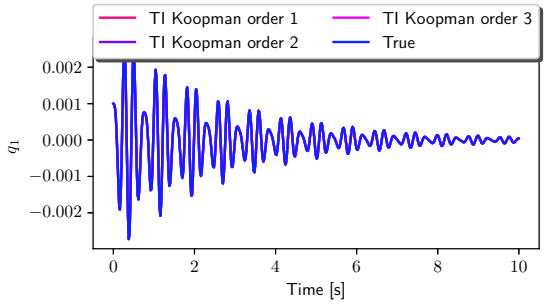
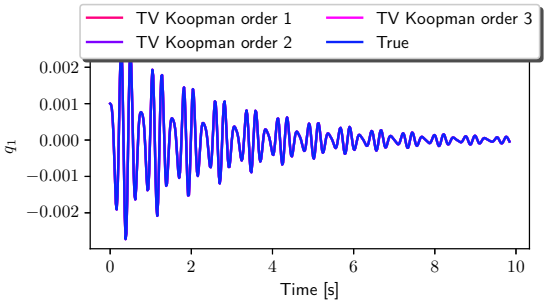


Figure 5.4: 2D skin panel configuration

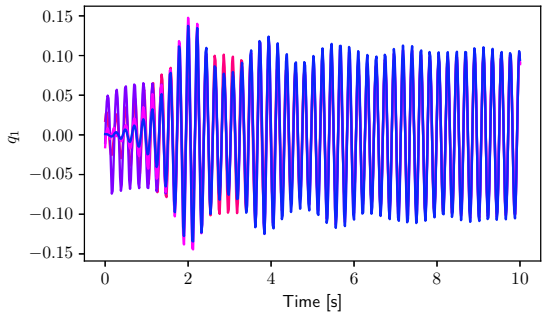
The measurement equation is now directly about the displacement of the middle line of the structure



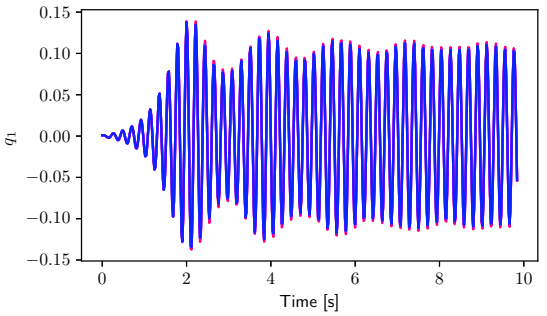
(a) LTI Koopman: q_1 for $\lambda = 260$



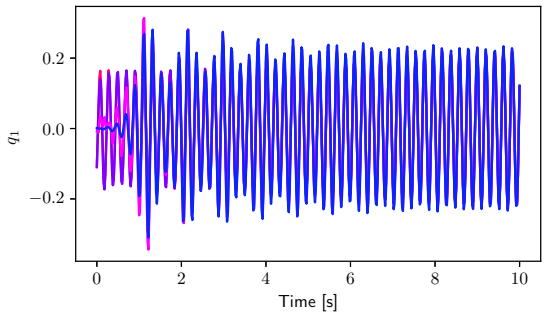
(b) LTV Koopman: q_1 for $\lambda = 260$



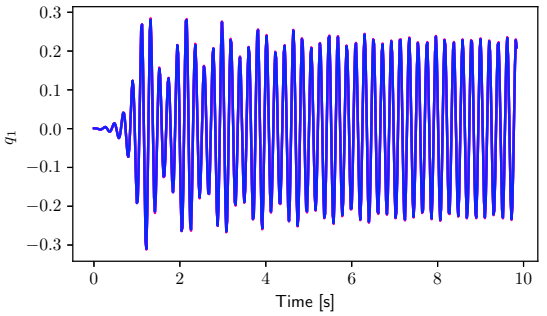
(c) LTI Koopman: q_1 for $\lambda = 280$



(d) LTV Koopman: q_1 for $\lambda = 280$



(e) LTI Koopman: q_1 for $\lambda = 300$



(f) LTV Koopman: q_1 for $\lambda = 300$

Figure 5.1: Propagation of modal state q_1 for different values of λ. LTI Koopman on the left, LTV Koopman on the right

structure and its temperature, creating a vector of dimension $49 \times 2 = 98$. The initial absolute perturbation applied to the panel is $P_{\text{inf}} = 15000$. Fifteen sets of experiment are created for the study: twelve will serve as training set for TVERA and three will be used for testing the accuracy of the derived model. The wall temperature is a parameter of the system that is being varied for each experiment; $T_w = 256$, 258, 260, 262, 264, 266, 268, 272, 274, 276, 280 and 282 K are used for training whereas $T_w = 254$, 270, 278 K are used for testing. The numerical data is acquired for 312 ms at a frequency of 5000 Hz. The procedure we employ for this numerical simulation is similar as before and a reduced model of order $n = 10$ has been found to be accurate. Although the resulting order of the reduced model is

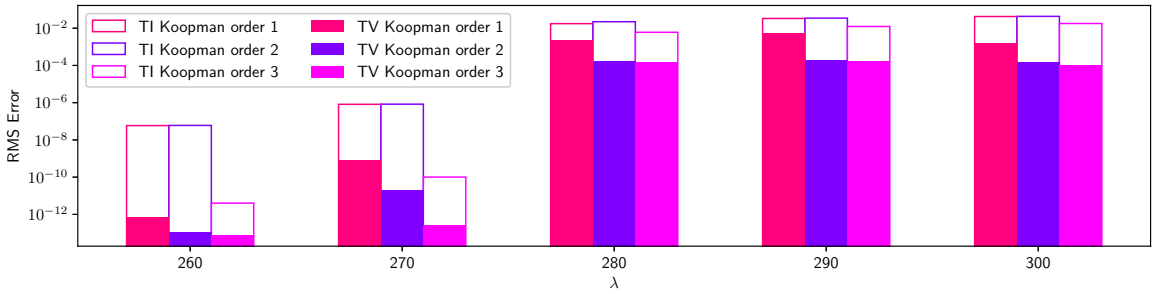


Figure 5.2: RMSE of the states q_1 and q_2 for LTI and LTV Koopman for different values of λ

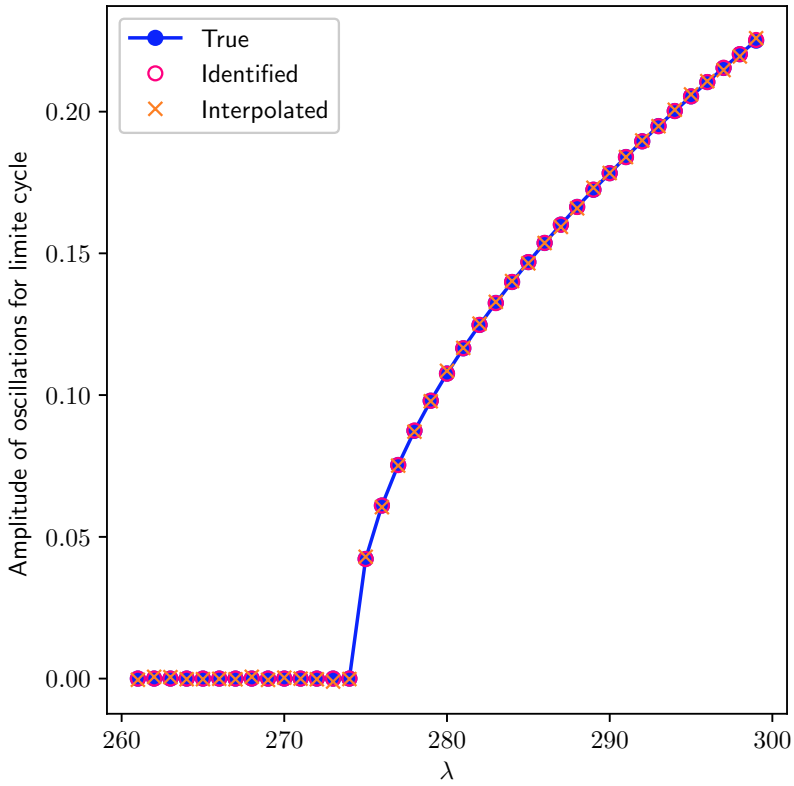


Figure 5.3: Bifurcation plot

usually found by examining the singular value decomposition plot, the one provided Figure 5.6 does not allow the analyst to gain too much insight. While it seems that 8 singular values could be found to be of greater magnitude than the rest of the sequence, it is difficult to acknowledge the presence of a clear cut criteria, especially as time increases.

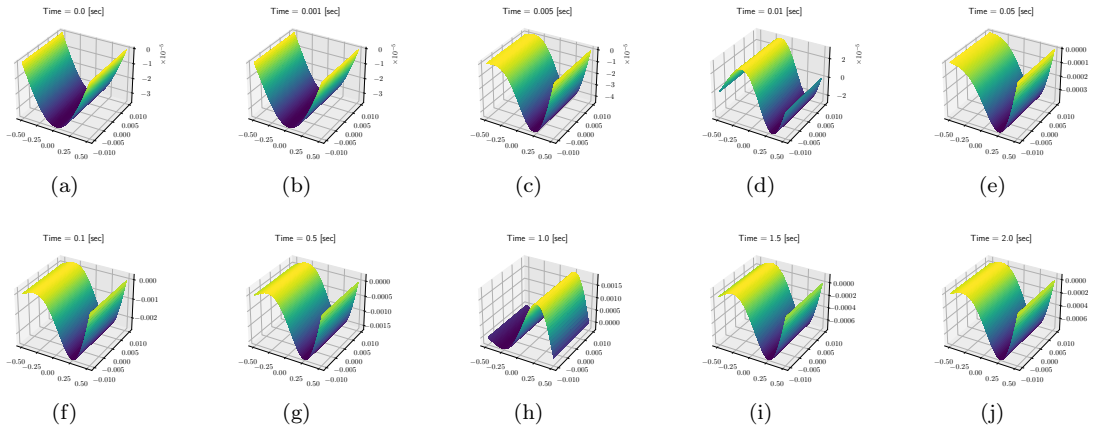


Figure 5.5: Evolution of the panel shape over the time

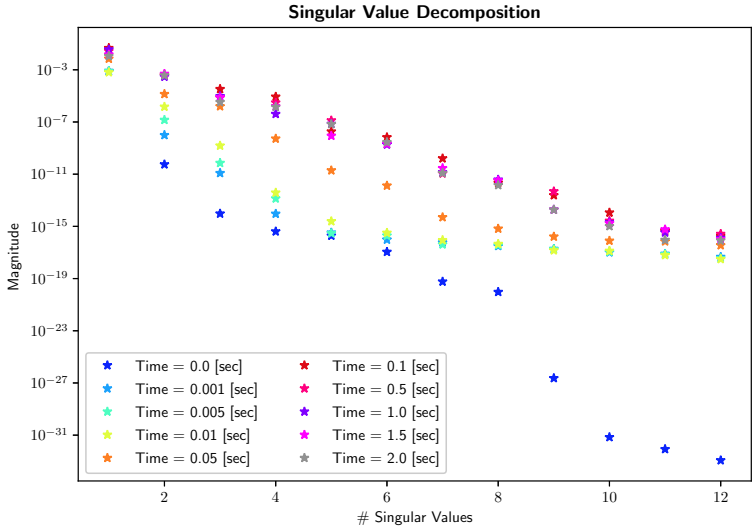


Figure 5.6: Evolution of the magnitude of the singular values from singular value decomposition

Figures 5.7 and 5.8 show the evolution of the displacement as well as the temperature increase of the panel throughout the time for the true and the identified systems. The identified linear time-varying model is able to capture the deformation of the panel correctly, even when the larger amplitude oscillations begin. As time increases, the identified model tends to slightly deviate from the true deformation, although capturing the right temperature increase. Note that the T_w of the experiments from the testing set is a parameter that is not included is the training of the reduced-order model, meaning that the model is able to reproduce the behavior of the panel under different testing conditions.

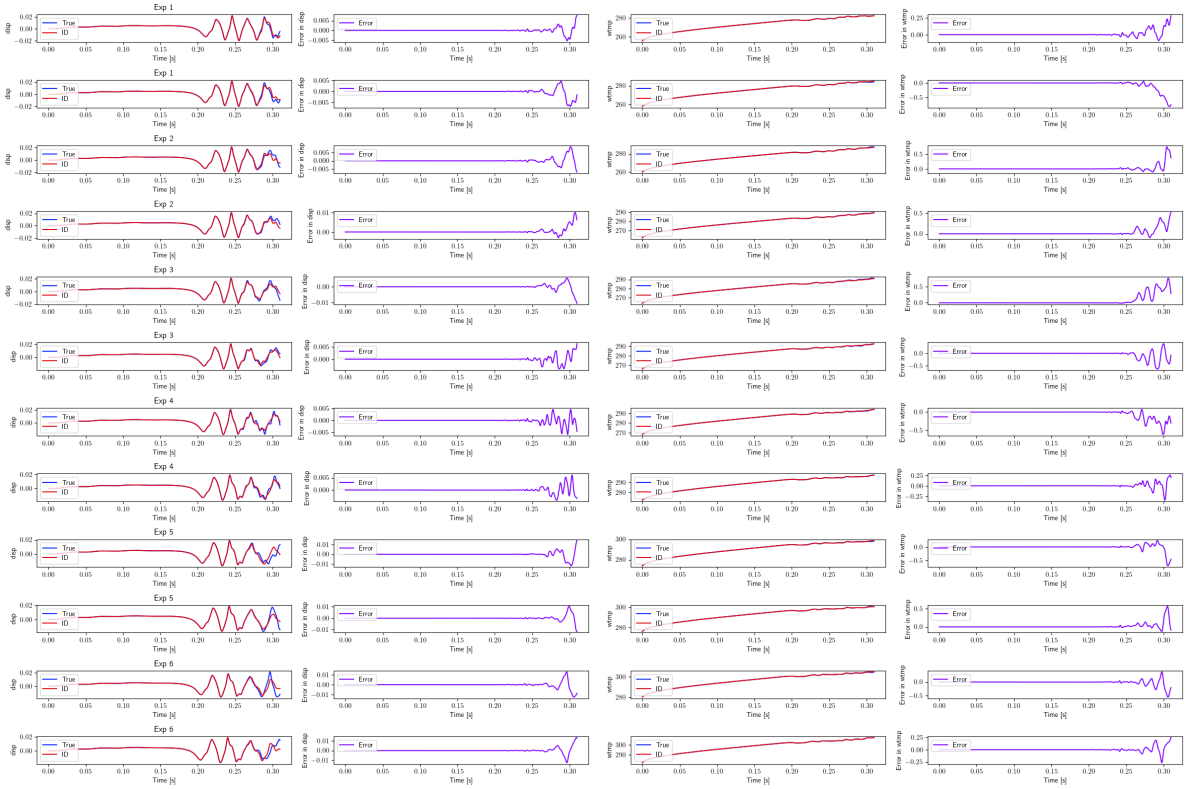


Figure 5.7: Illustration of the displacement and temperature increase from the training set

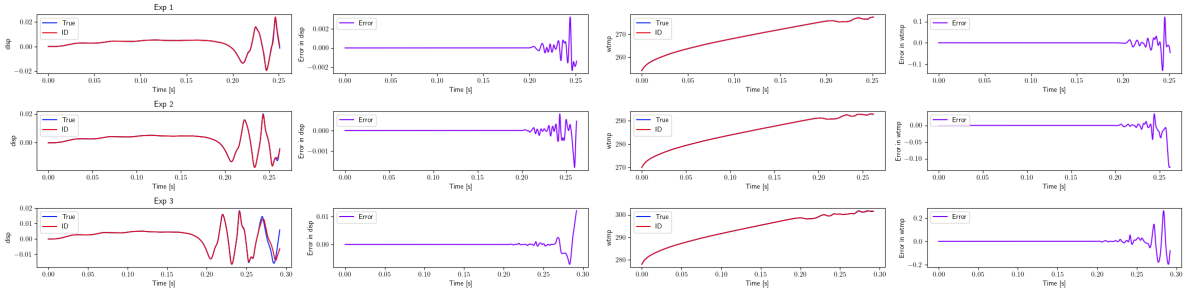


Figure 5.8: Illustration of the displacement and temperature increase from the testing set

5.3 Reduced-Order Dynamics in High-Dimensional Finite-Element Models: Von Kármán Beam

The estimation of invariant manifolds (or subspaces) allows for quantitative and qualitative understanding of nonlinear phenomena in dynamical systems and is a relatively new and rapidly evolving discipline due to advances in scientific computing. However, the computation of reduced-order models of high-dimensional mechanical systems arising from spatially discretized partial differential equations (PDEs) is still a challenging problem. Most global techniques that accurately represent very high-dimensional nonlinear systems discretize the domain of interest into a mesh e.g., via collocation or spectral points. As the dimension of the problem invariably becomes large in the case of discretized PDEs, practical numerical application or system analysis via collocation and spectral approaches becomes computationally intractable. In this section, we consider a finite element model of a geometrically nonlinear, cantilevered Von Kármán beam, illustrated in Figure 5.9 [85]. The geometric and material properties of the beam are given in Table 5.1. The equations of motion are given in the general form

$$M\ddot{\boldsymbol{x}} + \zeta\dot{\boldsymbol{x}} + K\boldsymbol{x} + \boldsymbol{f}(\boldsymbol{x}) = \boldsymbol{0}, \quad \boldsymbol{x}(t) \in \mathbb{R}^n, \tag{5.2}$$

where $\boldsymbol{x}(t)$ represents the displacement of the beam in 3 dimensions. This model is programmed in the finite element solver [98], which directly provides us the matrices M, ζ, K and the coefficients of the nonlinearity $\boldsymbol{f}$ in physical coordinates. We discretize this model using 10 elements resulting in $n = 30$ degrees of freedom. We are interested in predicting the response of the beam for initial perturbations of maximum 10^{-4} m. For illustration, Figures 5.10(a) and 5.10(b) show the beam at rest and deflected. Sixty random initial deflections in the vertical plane (x_3, see Figure 5.9) are considered and data is recorded for 2 ms. These experiments will first be used to generate a reduced-order model using TVERA. Once that model is found, the *minimal* state is estimated using the output influence matrix and lifted into a higher dimensional space. Finally, a TVKO is identified allowing for better accuracy of the state propagation. Figure 5.11 presents an overview of the procedure.

Table 5.1: Physical parameters of the Von Kármán beam model

Symbol	Meaning	Value [unit]
L	Length of beam	1 [m]
h	Height of beam	1 [mm]
b	Width of beam	0.1 [m]
E	Young's Modulus	70 [GPa]
κ	Viscous damping rate of material	10^7 [Pa·s]
ρ	Density	2700 [kg/m^3]

Figures 5.12(a), 5.12(b) and 5.12(c) show the prediction capabilities of the different models on a testing trajectory (errors are averaged over 10 random testing initial conditions). Different models using increasing orders of TVKO are displayed alongside models from ERA and TVERA, serving as references. Coupling a TVKO on top of a reduced-order model from TVERA helps decrease the prediction error

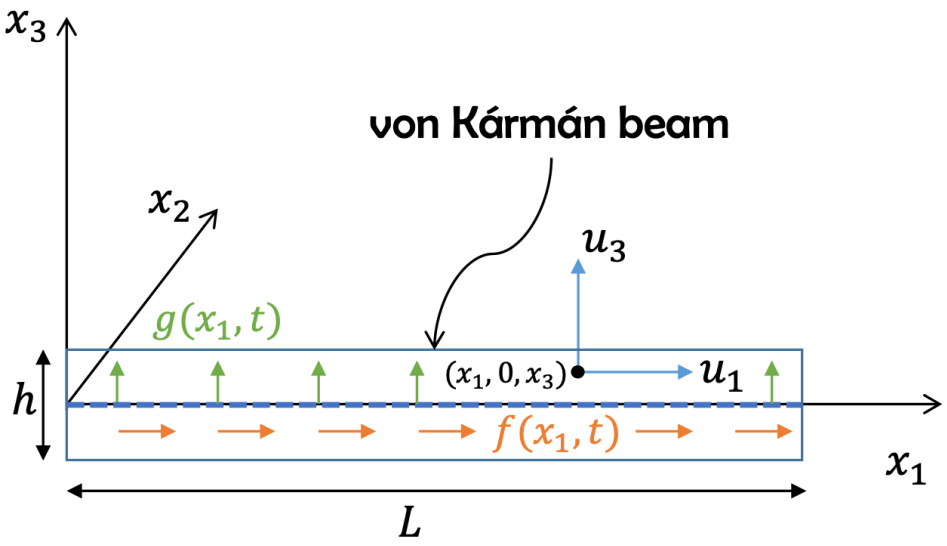
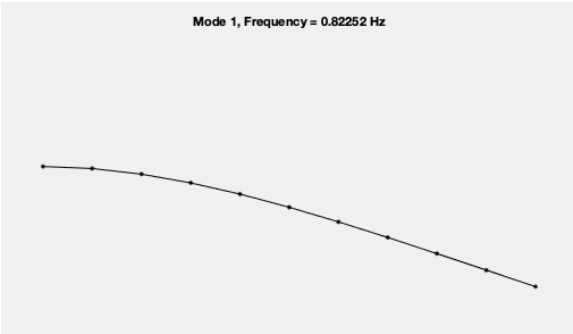


Figure 5.9: The schematic of a two-dimensional Von Kármán beam model with height h and length L, initially aligned with the x_1 axis

(a) Model of the beam, at rest



(b) Random initial perturbation

Figure 5.10: Representation of the beam using 10 elements

and increasing the order of the lifting space for TVKO also allows for better prediction capabilities. Figure 5.13 compares different models with similar dimensionality (dimension $2 \leq n \leq 9$). While higher order of TVERA can help in the prediction accuracy, a lower order TVERA coupled with TVKO can yield better accuracy in some cases. For example, a TVERA of order 2 with a TVKO of order 2 (total dimension $n = 5$) yields better prediction capabilities than TVERA of order 6 (total dimension $n = 6$).

5.4 Conclusion

This chapter has demonstrated the application of time-varying subspace methods for the analysis and reduced-order modeling of high-dimensional problems. It has started with the model-order reduction for hypersonic aerothermoelasticity, studying the coupled analysis between flight dynamics, structural dynamics, heat transfer, and hypersonic aerothermodynamics. Simulation results clearly show the improved performance of the proposed time-varying Koopman operator approach as compared to conventional

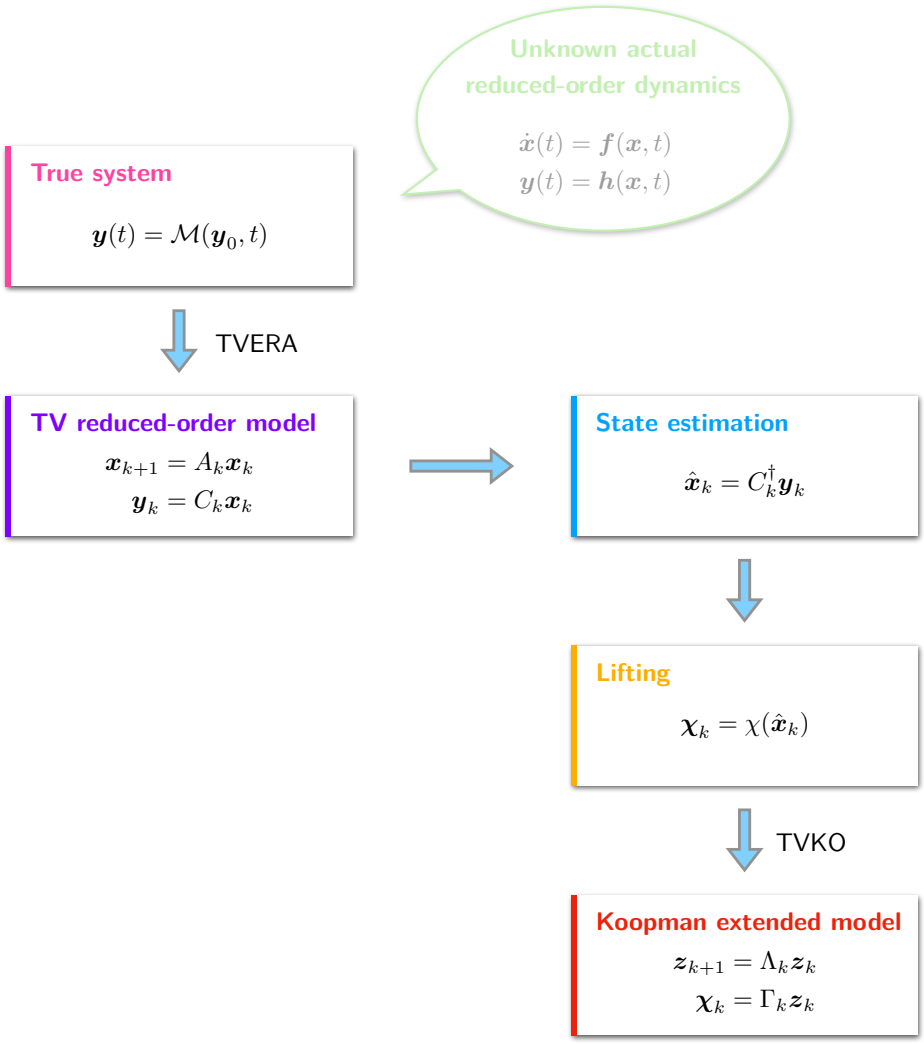
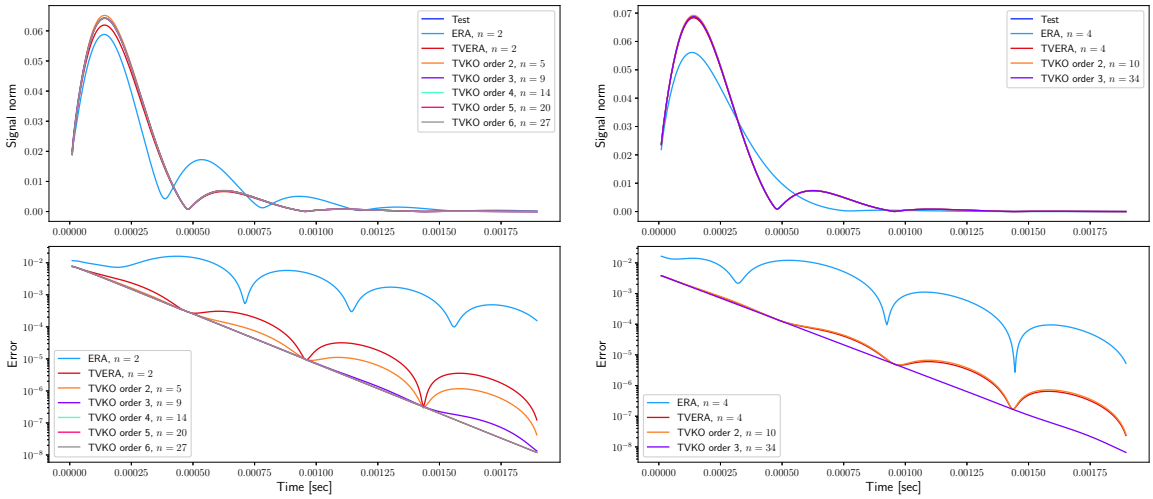


Figure 5.11: The 2-step system identification framework used for the Von Kármán beam

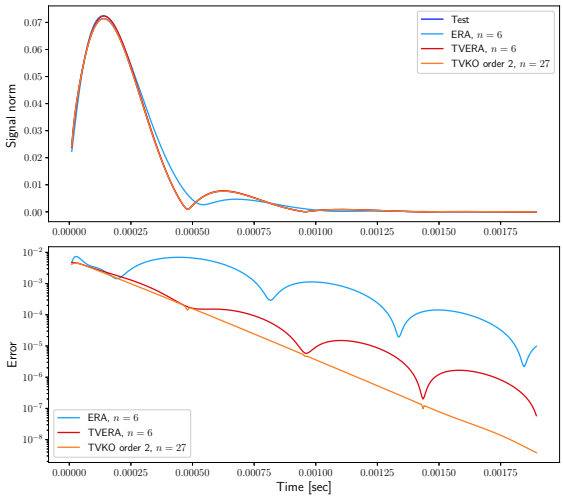
time-invariant Koopman operator for both low- and high-fidelity aerothermoelastic simulations. Through the example of a finite element model of a geometrically nonlinear, cantilevered Von Kármán beam, we also demonstrate the utility of a combined framework based on TVERA and TVKO for reduced-order modeling of high-dimensional systems. As aerothermoelastic analysis or structural analysis through finite-element models involve strong stochastic dynamics and since uncertainty quantification in nonlinear models is challenging in general, the methods developed in this chapter serve as a first step towards the analysis of high-dimensional nonlinear systems.

One key contribution of the work presented in this chapter is the combination of model reduction coupled with the Koopman framework. Usually, system identification methods based on data-driven Koopman analysis tend to build a surrogate of higher dimension from relativaly low dimensional data-sets, impacting in reality the use of the Koopman framework. What has been showed here is that selecting the right subspace to start with allows for the Koopman framework to be utilized afterwards. This directly echoes with results presented in Chapter 3. The iterative procedure presented in Chapter 3 -



(a) Prediction with ROM of dimension 2

(b) Prediction with ROM of dimension 4



(c) Prediction with ROM of dimension 6

Figure 5.12: Predicted error for different orders of initial reduced-order model

automatically selecting higher-order basis functions solving an ℓ_1-norm optimization problem to exaplain the data - is somewhat equivalent to the procedure TVERA + Koopman presented in this chapter. Further testing is needed to establish a direct correlation between the two approaches but the end goal is eventually the same. This also explains and justifies the choice for the title of Chapter 3.

Now that a clear method has been presented to derive models for - potentially high-dimensional - nonlinear systems, the next chapter will focus in depth into the calculation of uncertainties associated with a system's states. An array of different methods and approaches will be presented and compared side by side, weighing the pros and cons for each of them.

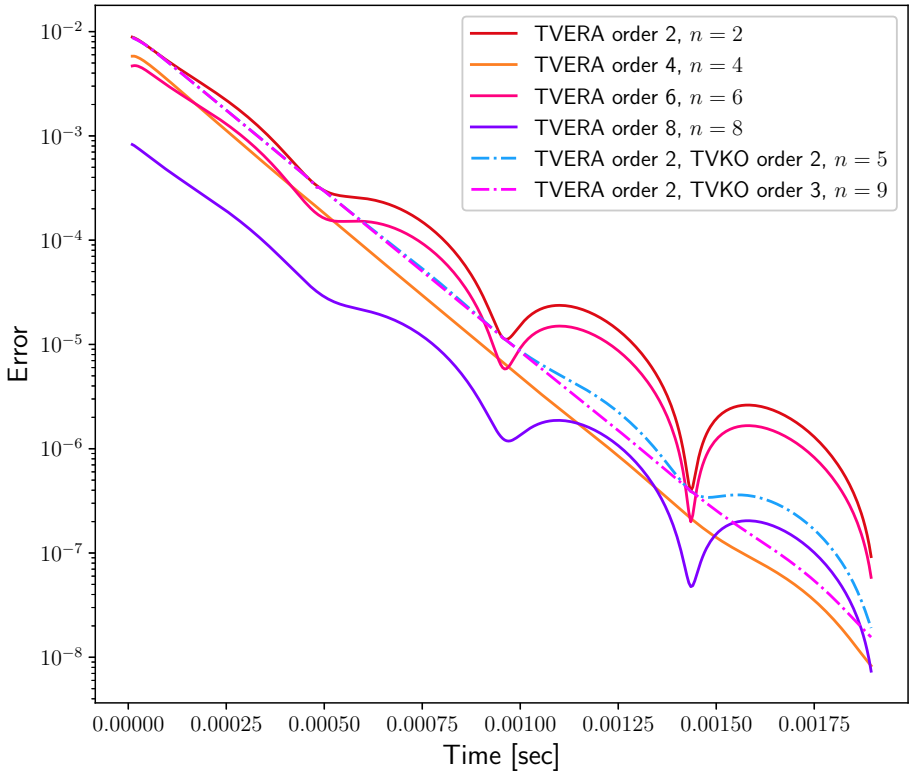


Figure 5.13: Prediction on a random testing initial perturbation for simulations with total order between 2 and 9 using a combination of TVERA and TVKO

Chapter 6
Applications to Uncertainty Quantification

6.1 Introduction

Estimation theory deals with the problem of estimating the state of a dynamical system from sensor measurements, usually corrupted by noise. Quantitative measures to accommodate uncertainties (in control as well as state variables) from a designed trajectory are rigorous and exact for the linear systems but nonlinearities in the representation of the dynamics lead to significant challenges in estimating uncertainties associated with the state vector. Approximate measures to quantify uncertainties in the real world are poor and can lead to significant compromises in the overall performance and safety. Whether one is planning a path for a robotic system or designing a maneuver for a spacecraft to avoid a collision with another spacecraft, the calculation of uncertainties associated with a trajectory is paramount in many engineering fields. Uncertainty propagation through nonlinear dynamics is computationally expensive. Conventional approaches focus on finding a reduced order model to alleviate the computational complexity associated with the uncertainty propagation algorithms. Given the significance of this problem, several approaches are developed to address the uncertainty quantification problem. This chapter exploits the fact that the moment propagation equations form a linear time-varying (LTV) system and use system theory to identify this LTV system from data only. Four different methods to estimate the moments of the probability density function associated with the dynamical states of a system are presented. Two of these methods are qualified as indirect methods as they first estimate the process as a linear time-varying dynamical system and then propagate the initial moments, or calculate the moments from quadrature; the two other methods are qualified as direct methods as they directly compute or estimate the dynamical process that governs the propagation of the moments for any probability density function. The next two sections will expand on those different methods and the last section will consider three different examples to compare these methods with each other. As a reference, statistical moments will also be calculated from CUT quadrature [99].

6.2 Moment Propagation: Direct Method

This section explains how to propagate the moments of a probability density function using the equations that govern the dynamical evolution of the moments. To gain insight into this procedure, we first consider the autonomous dynamical system

$$\frac{\mathrm{d}}{\mathrm{d}t}\boldsymbol{x}(t) = \boldsymbol{f}(\boldsymbol{x}), \quad \boldsymbol{x}(0) = \boldsymbol{x}_0 \rightsquigarrow \mathcal{N}(0). \tag{6.1}$$

with $\mathcal{N}(0)$ some initial random distribution. The work in this chapter will consider initial Gaussian uncertainty, but the type of distribution does not impact the subsequent developments. We will see that only the moments of $\mathcal{N}(0)$ are relevant. Recall from Eq. (2.42),

$$\delta x_i(t) = \sum_{p=1}^{\infty} \frac{1}{p!} \Phi^*_{i,r_1 r_2 \cdots r_p} \delta x_{0_{r_1}} \delta x_{0_{r_2}} \cdots \delta x_{0_{r_p}} \tag{6.2}$$

is the deterministic solution of Eq. (6.1) using higher-order state transition tensors evaluated along the nominal trajectory $\boldsymbol{x} = \boldsymbol{x}^*$, defined by

$$\Phi^*_{i,r_1 r_2 \cdots r_p} = \left. \frac{\partial^p x_i}{\partial x_{0_{r_1}} \partial x_{0_{r_2}} \cdots \partial x_{0_{r_p}}} \right|_{\boldsymbol{x}=\boldsymbol{x}^*}. \tag{6.3}$$

Now, by definition of the central moment of order p

$$P^{(p)}_{i_1 i_2 \cdots i_p}(t) := \mathrm{E}\left[\delta x_{i_1}(t) \delta x_{i_2}(t) \cdots \delta x_{i_p}(t)\right]. \tag{6.4}$$

In this equality, $P^{(p)}(t)$ denotes the moment of order p of the random variable $\delta\boldsymbol{x}(t)$. Eq. (6.4) provides the explicit form of the tensor $P^{(p)}(t)$ using index notation. Before diving into the core of this chapter and deriving the governing equations for moment propagation of arbitrary order, let's study an easy scalar example. Consider the scalar dynamics

$$\dot{x} = f(x) \tag{6.5}$$

with initial condition $x(0) = x_0 \rightsquigarrow \mathcal{N}(0)$. At any time t, the first moment (mean) is

$$P^{(1)}(t) = \mathrm{E}\left[x(t)\right] \tag{6.6}$$

and its time derivative

$$\begin{aligned}
\dot{P}^{(1)} &= \mathrm{E}\left[\dot{x}\right] && \text{(6.7a)}\\
&= \mathrm{E}\left[f(x)\right] && \text{(6.7b)}\\
&\simeq \mathrm{E}\left[f(P^{(1)}) + \left.\frac{\partial f}{\partial x}\right|_{x=P^{(1)}} \times (x - P^{(1)})\right] && \text{(6.7c)}\\
&\simeq f(P^{(1)}) + f^*_1(\mathrm{E}\left[x\right] - P^{(1)}) && \text{(6.7d)}\\
&\simeq f(P^{(1)}) && \text{(6.7e)}
\end{aligned}$$

This formulation is at the core of the famous paper by Kalman [4], which solved the estimation problem for a linear dynamical system with measurements corrupted with Gaussian noise, and was quickly applied to nonlinear problems via a local linearization of the dynamic system and measurement equations arriving at what is today known as the Extended Kalman Filter (EKF). The same developments with the second moment would yield

$$\dot{P}^{(2)} \simeq f_1^* P^{(2)} {f_1^*}^\top. \tag{6.8}$$

Utilizing an automated nonlinear expansion of the model about a nominal trajectory, a second order approximate solution for the moments dynamics is derived using higher-order state transition tensors as

$$\begin{aligned}
\dot{P}^{(1)} &= \mathrm{E}\left[\dot{x}\right] && \text{(6.9a)}\\
&= \mathrm{E}\left[f(x)\right] && \text{(6.9b)}\\
&\simeq \mathrm{E}\left[f(P^{(1)}) + f_1^*(x - P^{(1)}) + f_2^*(x - P^{(1)})(x - P^{(1)})^\top\right] && \text{(6.9c)}\\
&\simeq f(P^{(1)}) + f_2^* P^{(2)} && \text{(6.9d)}
\end{aligned}$$

and the time evolution of the first moment includes knowledge of the second moment. This solution can be utilized in evaluating the evolution of statistics of the departure motion as a function of the statistics of initial conditions (similarly as thus obtained in the determination of a state estimate assuming a Kalman update structure). Using Eq. (6.2) it is then possible to relate arbitrary moments of order p at time t with the initial moments [68]:

$$\begin{aligned}
P^{(1)}_{i_1}(t) &= \Phi^*_{i_1,j_1} P^{(1)}_{j_1}(0) + \frac{1}{2!}\Phi^*_{i_1,j_1j_2} P^{(2)}_{j_1j_2}(0) + \frac{1}{3!}\Phi^*_{i_1,j_1j_2j_3} P^{(3)}_{j_1j_2j_3}(0)\\
&\quad + \frac{1}{4!}\Phi^*_{i_1,j_1j_2j_3j_4} P^{(4)}_{j_1j_2j_3j_4}(0) + \mathrm{HOT}
\end{aligned} \tag{6.10a}$$

$$\begin{aligned}
P^{(2)}_{i_1i_2}(t) &= \Phi^*_{i_1,j_1}\Phi^*_{i_2,j_2} P^{(2)}_{j_1j_2}(0) + \frac{1}{2!}\left(\Phi^*_{i_1,j_1j_2}\Phi^*_{i_2,j_3} + \Phi^*_{i_1,j_1}\Phi^*_{i_2,j_2j_3}\right) P^{(3)}_{j_1j_2j_3}(0)\\
&\quad + \left[\frac{1}{3!}\left(\Phi^*_{i_1,j_1}\Phi^*_{i_2,j_2j_3j_4} + \Phi^*_{i_1,j_1j_2j_3}\Phi^*_{i_2,j_4}\right) + \frac{1}{2!2!}\Phi^*_{i_1,j_1j_2}\Phi^*_{i_2,j_3j_4}\right] P^{(4)}_{j_1j_2j_3j_4}(0) + \mathrm{HOT}
\end{aligned} \tag{6.10b}$$

$$\begin{aligned}
P^{(3)}_{i_1i_2i_3}(t) &= \Phi^*_{i_1,j_1}\Phi^*_{i_2,j_2}\Phi^*_{i_3,j_3} P^{(3)}_{j_1j_2j_3}(0)\\
&\quad + \frac{1}{2!}\left(\Phi^*_{i_1,j_1j_2}\Phi^*_{i_2,j_3}\Phi^*_{i_3,j_4} + \Phi^*_{i_1,j_1}\Phi^*_{i_2,j_2j_3}\Phi^*_{i_3,j_4} + \Phi^*_{i_1,j_1}\Phi^*_{i_2,j_2}\Phi^*_{i_3,j_3j_4}\right) P^{(4)}_{j_1j_2j_3j_4}(0) + \mathrm{HOT}
\end{aligned} \tag{6.10c}$$

$$P^{(4)}_{i_1i_2i_3i_4}(t) = \Phi^*_{i_1,j_1}\Phi^*_{i_2,j_2}\Phi^*_{i_3,j_3}\Phi^*_{i_4,j_4} P^{(4)}_{j_1j_2j_3j_4}(0) + \mathrm{HOT} \tag{6.10d}$$

$$\vdots$$

Here, Eq. (6.10) describes the evolution of the first four moments of a probability density function with respect to its first four moments. The quantities on the left hand side refer to the value of the corresponding statistical moment at time t propagated from a certain initial time and a corresponding initial moment value.

The unknowns in the above equations are the time-varying higher-order tensors $\Phi^*_{i,r_1r_2\cdots r_p}$. If one

has access to an analytic version of the dynamical flow $\boldsymbol{f}$ then it is possible to compute them numerically and thus propagate the moments forward in time. This will be referred later as *analytical higher-order moments* or *analytical HOM* in short. However, since the time-varying coefficients of the state transition tensors appear linearly, classical data-driven approaches can be used to obtain an estimate of the dynamical equations Eq. (6.10) that govern the evolution of a probability density function's moments. A TVKO-based approach can effectively be exploited to identify a model for Eq. (6.10) from data only and be used to propagate initial moments to arbitrary future times. This is qualified as a direct method as no reduced-order dynamical model is generated for the actual system; rather it is the process that governs the evolution of the moments of an arbitrary probability density function that is being identified. These two direct approaches are labeled 2 and 3 in Figure 6.1, with 2 being the analytical solution and 3 the data-driven approach based on TVKO.

6.3 Moment Propagation: Indirect Methods

An other way to approach the moment propagation problem is to first build a model of the dynamical system from data and then use that model to propagate the initial moments to any arbitrary time step, hence the categorization as an indirect method. Modeling based on a TVKO can be used to find a time-varying linear representation of a dynamical system up to any arbitrary order as defined in chapter 4. Given a discrete LTV system given as $\boldsymbol{z}_{k+1} = A_k \boldsymbol{z}_k$, the moment propagation equations are

$$P^{(1)}_{i_1}(t_{k+1}) = \mathrm{E}\left[z_{i_1,k+1}\right] = \mathrm{E}\left[A_{i_1 j_1,k} z_{j_1,k}\right] = A_{i_1 j_1,k} P^{(1)}_{j_1}(t_k) \tag{6.11a}$$

$$P^{(2)}_{i_1 i_2}(t_{k+1}) = \mathrm{E}\left[z_{i_1,k} z_{i_2,k+1}\right] = \mathrm{E}\left[A_{i_1 j_1,k} A_{i_2 j_2,k} z_{j_1,k} z_{j_2,k}\right] = A_{i_1 j_1,k} A_{i_2 j_2,k} P^{(2)}_{j_1 j_2}(t_k) \tag{6.11b}$$

$$\vdots \qquad \vdots$$

and can be written for any arbitrary moment. This method is labeled 4 in Figure 6.1. A final alternate method would consist in using initial CUT samples from the initial probability density function and propagate these points through $\boldsymbol{z}_{k+1} = A_k \boldsymbol{z}_k$. Statistical moments can then be inferred at each time step. This last two method is labeled 5 in Figure 6.1

6.4 Summary

Figure 6.1 offers a compact overview of the methods. The first method that uses CUT quadrature points to estimate statistical moments over time will be considered the reference method, i.e. relative errors will be calculated from the results of this method. As a brief outline, the five methods are summarized below.

Method	Details
1	Quadrature method. Statistical moments are calculated over time from an initial probability density function sampled with the CUT method.
2	Exact method using true propagated higher-order tensors. Higher-order tensors are computed along a nominal trajectory and used to calculate moments with Eq. (6.10).
3	Similar as 2 but uses a TVKO-based approach to estimate higher-order tensors with time. Since the moments appear linearly in Eq. (6.10), a LTV model can be identified from several propagated distributions from which statistical moments are calculated over time. These statistical moments form the training set to identify the LTV model.
4	Indirect approach. First a model of the dynamics is identified using a TVKO-based approach. Moments are then propagated over time using Eq. (6.11).
5	Indirect approach. Similar as method 1, it is a quadrature method. The model from method 4 is used to replace the true dynamics in method 1.

6.5 Numerical Simulations

This section considers three examples to illustrate how different methods for uncertainty quantification presented in this chapter compare with respect to each other. Note that all simulations generate a unique set of training points and testing points that are used to train and test the five methods.

6.5.1 Double-Well Duffing Oscillator

The first example corresponds to the classical nonlinear Duffing oscillator studied before governed by the following equations

$$\dot{x} = y, \tag{6.12a}$$

$$\dot{y} = -\delta y - \alpha x - \beta x^3 \tag{6.12b}$$

with parameters $\alpha = -1, \beta = 4, \delta = 0$, corresponding to a double-well potential. In order to fairly and accurately compare the four methods discussed previously, a unique set of data will be used for training. In this case, 32 points (initial conditions) are generated from 4 normal distributions around a nominal point $\boldsymbol{x}_0^* = \begin{bmatrix} 0.5 & 0.5 \end{bmatrix}^\top$. Full covariance matrices associated with these distributions range from $3 \times 10^{-4} I_2$ to $9 \times 10^{-4} I_2$. Individual trajectories are used for the CUT quadrature method to reconstruct the moments over time as well as to learn the Duffing dynamics, useful to build an identified system with TVKO. Statistical moments are also calculated along the trajectories in order to provide data for learning the dynamics of the moment equations. Figure 6.2 presents an overall summary of the procedure. The upper portion of Figure 6.2succinctly explains the difference between the direct and indirect methods. While the direct method directly identifies the dynamics that govern the moments evolution, the indirect method first identifies the Duffing dynamics and then propagates the moments of

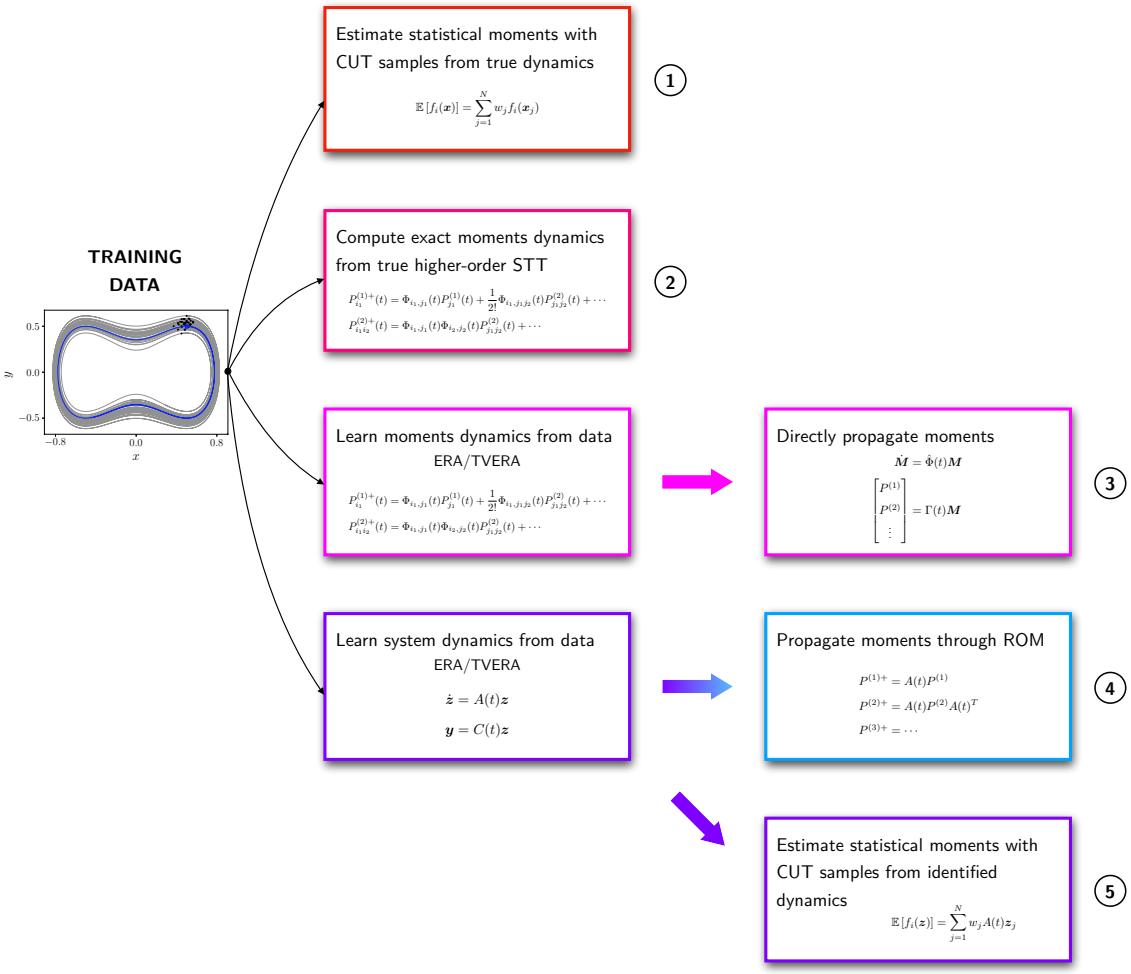


Figure 6.1: Summary of the methods used for moment propagation/estimation

the initial probability density function over time. In addition, analytical values of higher-order tensors are calculated along the nominal trajectories to reconstruct the actual true dynamics of Eq. (6.10). Finally, CUT points are sampled from the testing distribution to be used to calculate both *true* statistical moments as well as statistical moments from the identified model. Monte Carlo points are superimposed to the phase space to visualize the evolution of the distribution. The four methods considered in this chapter are able to match the distribution and mostly agree with the CUT method, with relative errors for the first two moments are plotted on the left column. Moments calculated from the CUT samples using the identified dynamics (TVKO) yield the smallest relative error while the errors induced from the three other methods are of the same order of magnitude. Notice that propagating moments through the identified model yields the largest error overall. The two direct methods (analytical and data-driven) are of the same level of accuracy. The overall better performance of using CUT samples in conjunction with a time-varying identified model is explained by the optimality of the CUT sampling method with the higher accuracy of the TVKO to represent the true dynamics (here, the TVKO is of order 6).

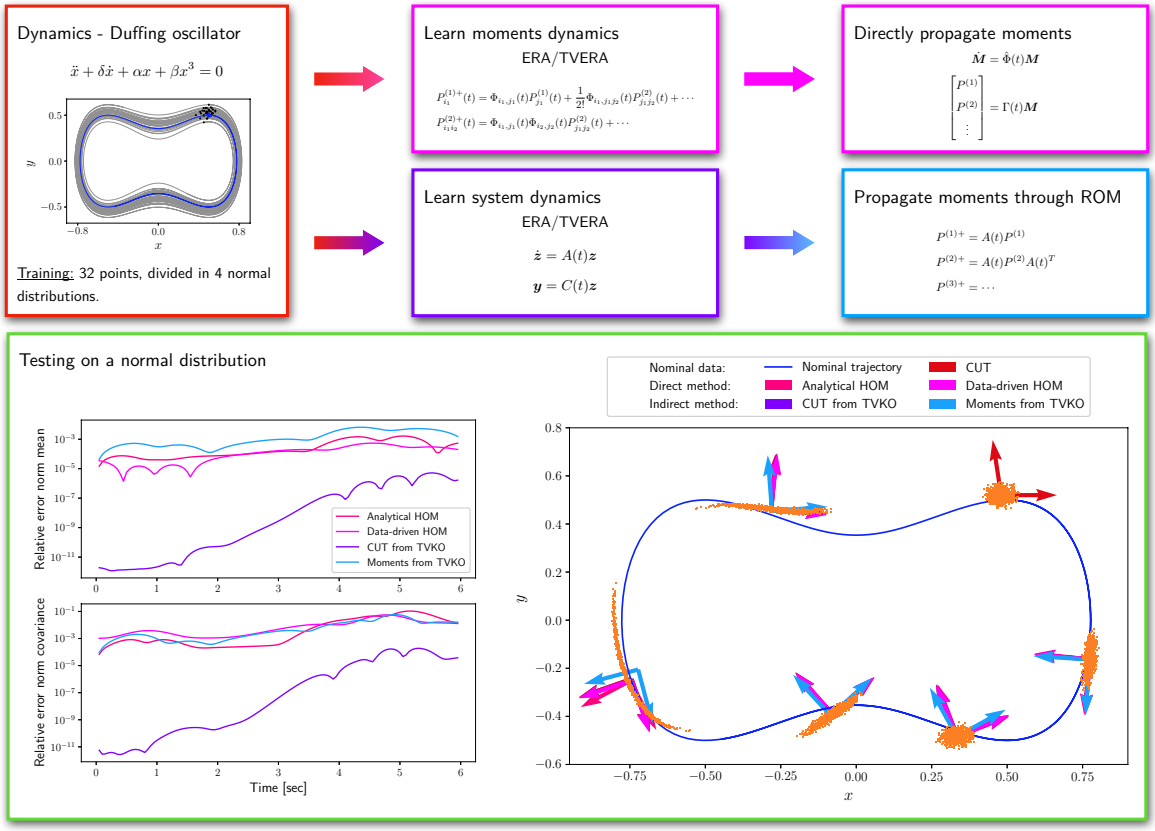


Figure 6.2: Uncertainty quantification analysis for the double-well Duffing oscillator

6.5.2 Flutter of an Aeroelastic Wing

The same example as in chapter 4, inspired from the work in [77], is considered here. Originally developed as a structured model for reference adaptive control of a wing section with structural nonlinearity, we are interested in quantifying the uncertainties associated with the oscillations of the wing. For a detailed presentation of the system, refer to chapter 4. 92 initial points, arranged in 4 initial normal distributions are considered for training with data recorded at a frequency of 50 Hz for 3 seconds. Those 92 points are sampled around the nominal $\boldsymbol{x}_0^* = \begin{bmatrix} 0.02 & 0.005 & 0.005 & 0.005 \end{bmatrix}^\top$ with covariances matrices ranging from $1 \times 10^{-5} I_4$ to $3 \times 10^{-5} I_4$. The procedure employed is the same as the previous example and results are reported on 6.3 with $3 - \sigma$ bounds plotted in addition to a set of Monte Carlo points generated from a testing normal distribution. The results from the four methods presented in Figure 6.3, direct or indirect, agree with the CUT quadrature method to compute the evolution of the moments of the testing distribution. CUT samples in conjunction with the time-varying identified model using TVKO yields the best results overall thanks to the optimality of the CUT sampling method with the higher accuracy of the TVKO (of order 4 here).

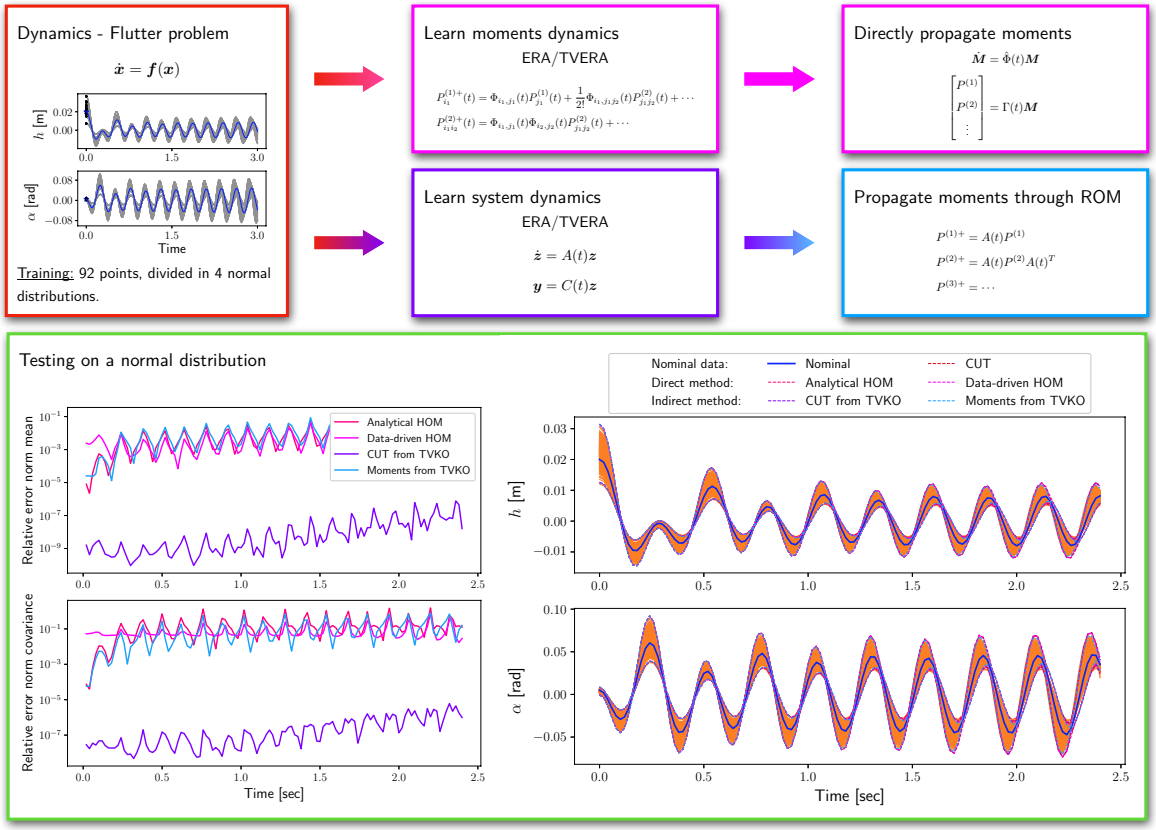


Figure 6.3: Uncertainty quantification analysis for the flutter of an aeroelastic wing

6.5.3 Two-Body Problem

The last example in this chapter is the famous two-body problem dynamics, given by

$$\ddot{\boldsymbol{r}} = -\frac{\mu}{r^3}\boldsymbol{r}, \tag{6.13}$$

in an inertial reference frame and using Cartesian coordinates. $\mu = G(m_1 + m_2)$ and G is the universal gravitational constant, $\boldsymbol{r}_1$ and $\boldsymbol{r}_2$ are the position vector of the two bodies, m_1 and m_2 their respective mass and $\boldsymbol{r} = \boldsymbol{r}_2 - \boldsymbol{r}_1$ is the relative position vector between them. 300 trajectories are generated around the nominal $\boldsymbol{x}_0^* = \begin{bmatrix} 7000 & 0 & 0 & 0 & -1.0374090357 & 7.477128835 \end{bmatrix}^\top$ with covariances matrices at $1 \times I_3$ for position and 1×10^{-6} for velocity. Five hours of data is recorded every 100 seconds. The methodology to estimate the moments over time of an initial normal distribution is exactly similar as before and results are reported on Figure 6.4. Two thousands Monte Carlo points are generated from a testing normal distribution to visualize the overall shape of the distribution over time. Once again, the four methods agree with the CUT quadrature method to compute the evolution of the moments of the testing distribution. CUT samples in conjunction with the time-varying identified model using TVKO offer the best accuracy (TVKO of order 4 here) although the data-driven method to estimate a dynamical model for the moments provides similar accuracy for the second moment.

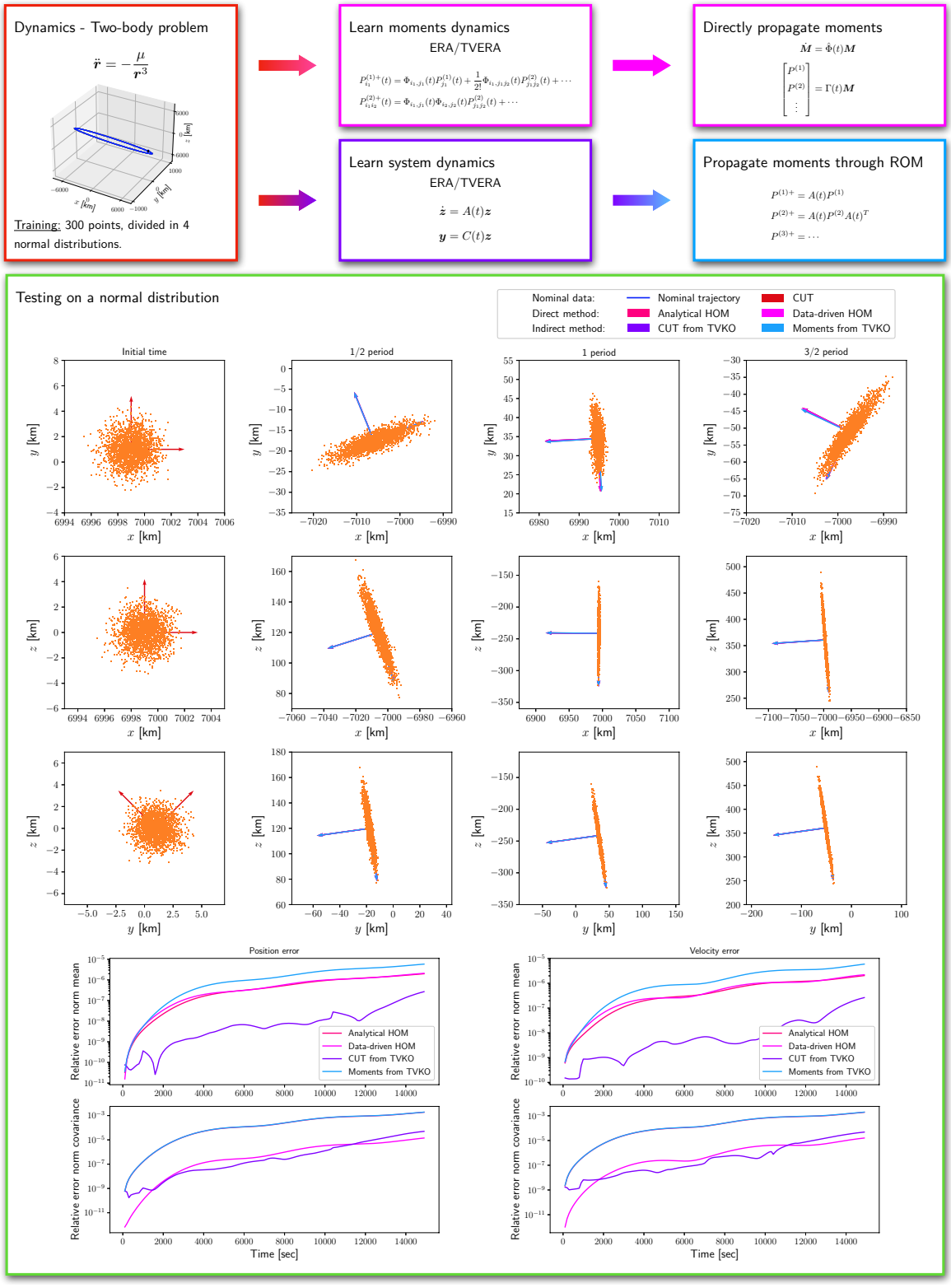


Figure 6.4: Uncertainty quantification analysis for the two-body problem

6.6 Conclusion

This chapter has presented several approaches to address the uncertainty quantification problem. Four different methods to estimate the moments of the probability density function associated with the dynamical states of a system are presented. Two of these methods are indirect methods as they first estimate the process as a linear time-varying dynamical system and then propagate the initial moments, or calculate the moments from quadrature; two other methods are qualified as direct methods as they directly compute or estimate the dynamical process that governs the propagation of the moments for any probability density function. Numerical simulations have been conducted to evaluate the error resulting in the propagation of the moments associated with an initial Gaussian uncertainty. A data-driven method where the dynamical system is approximated using TVKO combined with a quadrature method perform the best overall. A direct data-driven method to estimate the dynamics governing the moments propagation also performs strong.

Chapter 7
Bilinear Koopman Operator

7.1 Introduction

Until this chapter only have we considered unforced dynamical processes. As presented in chapter 4, the theory to identify a Koopman operator of any order given some data from an unforced dynamical system is well developed. For majority of the problems, the exact Koopman operator that would describe the evolution of the dynamical states in a lifting space linearly is of infinite dimension. In practice, analysts consider a lifting space of finite dimension (adequately with given requirements and computing capabilities) giving rise to a truncated Koopman operator. However, as presented in chapter 4, increasing the order of the operator offers a guarantee to gain accuracy on the state prediction. When ERA/DMD or TVERA are used to find a TIKO or TVKO there is no difficulties as the dynamics that govern the evolution of lifting functions of the state are expressed with respect to these lifting functions themselves. Selecting a basis of a function space as lifting functions provides the guarantee of the closure of the lifting space, under the dynamics considered. This was well illustrated with the example

$$\dot{x} = x^2, \tag{7.1}$$

and the infinite dimensional Koopman operator

$$\dot{\boldsymbol{\chi}} = \begin{bmatrix} \dot{x} \\ \dot{x^2} \\ \dot{x^3} \\ \dot{x^4} \\ \vdots \end{bmatrix} = \begin{bmatrix} 0 & 1 & 0 & 0 & \cdots \\ 0 & 0 & 2 & 0 & \cdots \\ 0 & 0 & 0 & 3 & \cdots \\ 0 & 0 & 0 & 0 & \cdots \\ \vdots & \vdots & \vdots & \vdots & \ddots \end{bmatrix} \begin{bmatrix} x \\ x^2 \\ x^3 \\ x^4 \\ \vdots \end{bmatrix} = \mathcal{K}_c \boldsymbol{\chi}, \qquad \text{with } \boldsymbol{\chi} = \begin{bmatrix} x \\ x^2 \\ x^3 \\ x^4 \\ \vdots \end{bmatrix}. \tag{7.2}$$

Although the Koopman operator is of infinite dimension, increasing the order of the polynomial basis as lifting functions guarantees a better accuracy in the prediction of the system's states.

Now, let us consider the controlled version of Eq. (7.1), such that

$$\dot{x} = x^2 + u. \tag{7.3}$$

Note that the controlled action appears linearly in Eq. (7.3). If the state x is considered as a measurement, so does x^2 and u since $\dot{x} = x^2 + u$. If x^2 is considered as a measurement, so does x^3 and xu since $\dot{x^2} = 2\dot{x}x = 2x^3 + 2xu$. This new term, xu, presents two difficulties. The first major difficulty is that the dynamics in a lifted space is *not* linear anymore, nor it is control affine and approximating the controlled nonlinear system by a linear system with affine control would yield poor results. Some previous attempts in this direction showed very mixed results [56, 57] and the theory around the Koopman operator for controlled system is not mature enough. Secondly, one could argue that a new type of lifting functions could be introduced, function of both the state and the control vectors. This would lead to identify a Koopman operator that would not only predict future values of the state but also future values of the control input which is not desirable.

Instead, this chapter introduces the concept of bilinear Koopman operator. Bilinear state-space model identification is of interest for two main reasons. Some physical systems are inherently bilinear and bilinear models of high dimension can approximate a broad class of nonlinear systems. Nevertheless, no well-established technique for bilinear system identification is available yet, even less in the context of Koopman. The aim of this chapter is to offer a global overview of bilinear system identification and to offer some perspectives and advances for bilinear system identification, working towards a bilinear Koopman operator.

7.2 Bilinear OKID

The concept of deterministic system identification in context of the Observer/Kalman Identification algorithm (OKID) has been introduced in [100] rewriting the bilinear model as an equivalent linear model (ELM). The ELM can be identified with any linear identification method, from which the original bilinear model is then recovered. In contrast with previous attempts for state-space bilinear system identification, a benefit of the method in [100] is the freedom in choosing the form of input excitation as long as it is sufficiently rich. The same advantage and the fact that data from a single experiment are sufficient could potentially make the approach more appealing than the one adopted by other authors, based on multiple pulses over multiple experiments. However, a proper version of the Observer/Kalman Identification algorithm (OKID) adapted for bilinear systems is *not* formally outlined. The work presented in this section aims to fill that gap and is a step forward in the identification of a bilinear Koopman operator.

Consider a discrete time-invariant bilinear system represented by

$$\boldsymbol{x}_{k+1} = A\boldsymbol{x}_k + \sum_{i=1}^{r} N_i \boldsymbol{x}_k u_{i,k} + B\boldsymbol{u}_k \tag{7.4a}$$

$$\boldsymbol{y}_k = C\boldsymbol{x}_k + D\boldsymbol{u}_k \tag{7.4b}$$

together with an initial state vector $\boldsymbol{x}_0$, where $\boldsymbol{x}_k \in \mathbb{R}^n$, $\boldsymbol{u}_k \in \mathbb{R}^r$ and $\boldsymbol{y}_k \in \mathbb{R}^m$ are the state, control input and output vectors respectively. In the above experssion, $u_{i,k}$ represents the channel i of the control input $\boldsymbol{u}_k$. The constant matrices $A \in \mathbb{R}^{n\times n}$, $N_i \in \mathbb{R}^{n\times n}$, $B \in \mathbb{R}^{r\times n}$, $C \in \mathbb{R}^{n\times m}$ and $D \in \mathbb{R}^{r\times m}$ represent the internal operation of the linear system, and are used to determine the system's response to

any input. Let us write the first few terms from Eq. (7.4a):

$$\boldsymbol{x}_1 = A\boldsymbol{x}_0 + N_{i_1}\boldsymbol{x}_0 u_{i_1,0} + B\boldsymbol{u}_0 \tag{7.5a}$$

$$\begin{aligned}\boldsymbol{x}_2 &= A\boldsymbol{x}_1 + N_{i_2}\boldsymbol{x}_1 u_{i_2,1} + B\boldsymbol{u}_1 \\ &= A^2\boldsymbol{x}_0 + AN_{i_1}\boldsymbol{x}_0 u_{i_1,0} + AB\boldsymbol{u}_0 + N_{i_2}\left(A\boldsymbol{x}_0 + N_{i_1}\boldsymbol{x}_0 u_{i_1,0} + B\boldsymbol{u}_0\right)u_{i_2,1} + B\boldsymbol{u}_1 \\ &= \left(A^2 + AN_{i_1}u_{i_1,0} + N_{i_2}Au_{i_2,1} + N_{i_2}N_{i_1}u_{i_1,0}u_{i_2,1}\right)\boldsymbol{x}_0 + AB\boldsymbol{u}_0 + N_{i_2}Bu_{i_2,1}\boldsymbol{u}_0 + B\boldsymbol{u}_1\end{aligned} \tag{7.5b}$$

$$\begin{aligned}\boldsymbol{x}_3 &= A\boldsymbol{x}_2 + N_{i_3}\boldsymbol{x}_2 u_{i_3,2} + B\boldsymbol{u}_2 \\ &= \left(A^3 + A^2N_{i_1}u_{i_1,0} + AN_{i_2}Au_{i_2,1} + AN_{i_2}N_{i_1}u_{i_1,0}u_{i_2,1}\right)\boldsymbol{x}_0 + A^2B\boldsymbol{u}_0 + AN_{i_2}B\boldsymbol{u}_0u_{i_2,1} + AB\boldsymbol{u}_1 \\ &\quad + \left(N_{i_3}A^2u_{i_3,2} + N_{i_3}AN_{i_1}u_{i_1,0}u_{i_3,2} + N_{i_3}N_{i_2}Au_{i_2,1}u_{i_3,2} + N_{i_3}N_{i_2}N_{i_1}u_{i_1,0}u_{i_2,1}u_{i_3,2}\right)\boldsymbol{x}_0 \\ &\quad + N_{i_3}ABu_{i_3,2}\boldsymbol{u}_0 + N_{i_3}N_{i_2}Bu_{i_2,1}u_{i_3,2}\boldsymbol{u}_0 + N_{i_3}Bu_{i_3,2}\boldsymbol{u}_1 + B\boldsymbol{u}_2\end{aligned} \tag{7.5c}$$

and observe how the number of parameters increase exponentially due to the fact that bilinear matrices $N_{i_1}, N_{i_2}, N_{i_3}, \cdots$ appear in different matrix products. Similarly to the classical OKID algorithm, let's assume that the initial condition is zero: $\boldsymbol{x}_0 = \boldsymbol{0}$. We define a sequence of constant matrices, $\tilde{h}_i$, called bilinear system Markov parameters or, in short, *bilinear Markov parameters* as

$$h_0 = D \qquad \tilde{h}_0 = D \tag{7.6a}$$

$$h_1 = CB \qquad \tilde{h}_1 = CB \tag{7.6b}$$

$$h_2 = CAB \qquad \tilde{h}_2 = CAB,\ CN_{i_1}B \tag{7.6c}$$

$$h_3 = CA^2B \qquad \tilde{h}_3 = CA^2B,\ CAN_{i_1}B,\ CN_{i_1}AB,\ CN_{i_1}N_{i_2}B \tag{7.6d}$$

$$\begin{aligned}h_4 = CA^3B \qquad \tilde{h}_4 = {} & CA^3B,\ CA^2N_{i_1}B,\ CN_{i_1}A^2B,\ CAN_{i_1}AB,\ CAN_{i_1}N_{i_2}B, \\ & CN_{i_1}N_{i_2}AB,\ CN_{i_1}AN_{i_2}B,\ CN_{i_1}N_{i_2}N_{i_3}B\end{aligned} \tag{7.6e}$$

Remember the sequence on the left is the sequence of Markov parameters for a linear time-invariant system. While the structure of the parameters is the same for both sequences, bilinear Markov parameters are more numerous than classical Markov parameters and their number increases exponentially as time increases. And not only the number of parameters to describe the I/O relationship of a bilinear system increases exponentially but a multi-dimensional input exacerbates the situation by increasing the counts of matrices N_i. Figure 7.1 compares the exponential growth of bilinear Markov parameters versus the linear growth of Markov parameters. After just a few time steps, the number of bilinear Markov parameters is so important that any attempt at identifying all those parameters is not really an option. In a sense, this is what the authors in [100] have attempted to say without really pinpointing where this exponential growth comes from.

Naturally, and very similarly to the classical OKID algorithm, rather than identifying the bilinear Markov parameters which may exhibit very slow decay, one can use an asymptotically stable bilinear observer to form a stable discrete state-space model for the system to be identified. With such a procedure, bilinear Markov parameters (and thus the state-space model) and a corresponding observer are determined simultaneously, similarly as in [18,19] for the linear case. One could attempt to implement such an algorithm by following the steps of the OKID algorithm but in practice, in order to keep the size

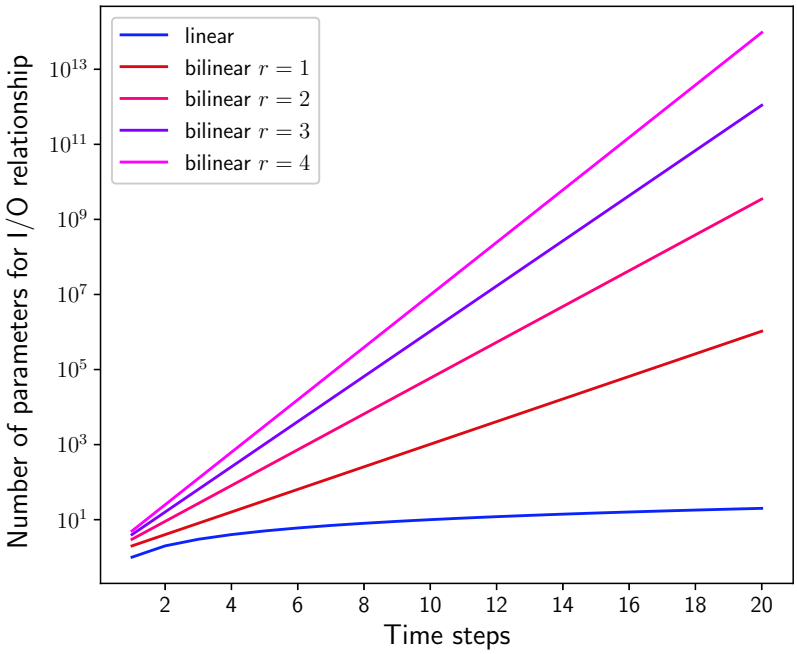


Figure 7.1: Exponential growth of bilinear Markov parameters vs. Linear growth of Markov parameters

of the problem tractable, only very low values for the observer order are admissible. Authors in [100] realized this by only selecting observer orders of at most $p = 8$ in numerical simulations.

While actual implementation of a bilinear OKID (or bilinear time-varying OKID) with a random input signal could be delivered without trouble by following the steps from the classical algorithms in Appendix C and E the practical use of such algorithms is extremely limited without further analysis. Even though several pathways for implementing such procedures with practical uses are being studied, no satisfactory algorithm has been implemented at the time this dissertation is being written. However, past work utilizing specialized input has been shown to be useful in the practical identification of bilinear system matrices. In [11], the authors use a set of semi-pulses (pulses that last only for a few time-steps) with non-zero initial condition for continuous bilinear system identification. In [101, 102], a single experiment consisting of several pulses with non-zero initial condition is used for continuous bilinear system identification. A few years later, in [103], authors use a set of semi-pulses with zero initial condition for continuous bilinear system identification. In [100], a single experiment with pulses and nonzero initial condition is used for discrete bilinear system identification. The major common denominator with all these methods is that it uses a specific type of input: pulses. Either as a set of short experiments or as a single longer experiment comprised of several pulses, all of these methods exploit the fact that pulses or semi-pulses make the control action vanish after a few time-steps, subsequently making the terms $N_i \boldsymbol{x}_k$ vanish. This helps in keeping the number of parameters to identify low and helps all of these algorithms to be compelling for practical use. Note that all of these methods identify the bilinear dynamics exactly, up to noise levels (i.e. no approximation is made). In this dissertation we wish to add one more brick to the existing bilinear system identification framework. The next section

describes a procedure where a set of step inputs is being used with arbitrary initial condition for the identification of a continuous bilinear system of equations. This method relies on the method presented in [103] with some adjustments.

7.3 Continuous Bilinear System Identification with Specialized Input

The algorithm presented in [103] relies on the central observation that the bilinear system of equations becomes a linear time invariant system upon the application of constant forcing functions. The authors exhibited the solution of the bilinear system of equations and showed that while the general input output behavior is indeed nonlinear, one can generate an analytical solution for a set of specified inputs. The generic algorithm is presented in details in [103] and the reader can refer to it for more details. The first example in the numerical simulation section is a direct application of this algorithm. In this section, it is desired to present a different version of this algorithm by identifying matrices A_c, N_{c_i}, C and the initial condition $\boldsymbol{x}_0$ for a bilinear dynamical system of the form

$$\dot{\boldsymbol{x}} = A_c\boldsymbol{x} + \sum_{i=1}^{r} N_{c_i}\boldsymbol{x}u_i, \qquad \boldsymbol{x}_0 \neq \boldsymbol{0}, \tag{7.7a}$$

$$\boldsymbol{y} = C\boldsymbol{x}. \tag{7.7b}$$

Additionally, the algorithm outlined in this section uses a set of step inputs when other methods in the literature are using pulses, with the only requirement that the step inputs have to go to zero at some point in time and be zero for a few time steps. This weaker condition on the input provides more flexibility to the the analyst when adjust the control action (also pulses are very difficult to apply to real mechanical systems for example, with instances where pulse inputs can impair the system). The step by step algorithm is presented below.

1. We suppose we perform a set of $N_1 + N_1 \times N_2$ experiments. This set is comprised of N_1 random initial condition response experiments from arbitrary $\boldsymbol{x}_0$, and for each of them, an additional set of N_2 forced response experiments is performed with step inputs. The requirement is that the step inputs have to go to zero at some point in time and be zero for a few time steps, but can have any profile before or after. We will assume that the input is nonzero at time step k_0 and is zero for p time steps after that. Throughout the description of the procedure we will give conditions on N_1 and N_2.

2. Perform ERA (see chapter 2) on the first set of N_1 experiments. This allows to obtain a realization of the pair $(\hat{A}, \hat{C})$ (and hence, $\hat{A}_c$) as well as the observability matrix $\hat{\boldsymbol{O}}^{(p)}$. For the identification to capture the full dynamics, it is required that $N_1 \geq n$.

3. For each group of N_2 experiments, build matrices

$$Y_{k_0} = \begin{bmatrix} \boldsymbol{y}_{k_0}^{\#1} & \boldsymbol{y}_{k_0}^{\#2} & \cdots & \boldsymbol{y}_{k_0}^{\#N_2} \end{bmatrix}, \tag{7.8a}$$

$$Y_{k_0}^{(N_2)} = \begin{bmatrix} \boldsymbol{y}_{k_0}^{\#1} & \boldsymbol{y}_{k_0}^{\#2} & \cdots & \boldsymbol{y}_{k_0}^{\#N_2} \\ \boldsymbol{y}_{k_0+1}^{\#1} & \boldsymbol{y}_{k_0+1}^{\#2} & \cdots & \boldsymbol{y}_{k_0+1}^{\#N_2} \\ \vdots & \vdots & \ddots & \vdots \\ \boldsymbol{y}_{k_0+p-1}^{\#1} & \boldsymbol{y}_{k_0+p-1}^{\#2} & \cdots & \boldsymbol{y}_{k_0+p-1}^{\#N_2} \end{bmatrix}. \tag{7.8b}$$

Calculate the identified state at time k_0

$$\hat{\boldsymbol{x}}_{k_0} = \hat{C} Y_{k_0} \tag{7.9}$$

and the matrix

$$F = \frac{1}{\Delta t} \log \left(\hat{\boldsymbol{O}}^{(p)^\dagger} Y_{k_0}^{(N_2)} \hat{\boldsymbol{x}}_{k_0}^\dagger \right) - \hat{A}_c. \tag{7.10}$$

4. Repeat the procedure N_1 times and populate the matrix

$$\boldsymbol{N}_c = \begin{bmatrix} F^{\#1} & F^{\#2} & \cdots & F^{\#N_1} \end{bmatrix}. \tag{7.11}$$

In parallel, build the matrix

$$V_{k_0}^{(N_2)} = \begin{bmatrix} Iu_{1,k_0}^{\#1} & Iu_{1,k_0}^{\#2} & \cdots & Iu_{1,k_0}^{\#N_2} \\ Iu_{2,k_0}^{\#1} & Iu_{2,k_0}^{\#2} & \cdots & Iu_{2,k_0}^{\#N_2} \\ \vdots & \vdots & \ddots & \vdots \\ Iu_{r,k_0}^{\#1} & Iu_{r,k_0}^{\#2} & \cdots & Iu_{r,k_0}^{\#N_2} \end{bmatrix}. \tag{7.12}$$

The identified bilinear matrices $\hat{N}_{c_i}$ are

$$\begin{bmatrix} \hat{N}_{c_1} & \hat{N}_{c_2} & \cdots & \hat{N}_{c_r} \end{bmatrix} = \boldsymbol{N}_c V_{k_0}^{(N_2)^\dagger}. \tag{7.13}$$

The matrix $V_{k_0}^{(N_2)}$ is invertible if full rank hence $N_2 \geq r$ and a rich input.

5. Initial condition $\hat{\boldsymbol{x}}_0$ can then be identified similarly as in the ERA procedure solving a least-squares problem.

That procedure will be used in subsequent sections for bilinear system identification and sensitivity analysis.

7.4 Numerical Simulations

7.4.1 Example with Unstable Linear Part

The procedure for bilinear system identification highlighted in [103] is applied to an example where the linear part of the bilinear system matrix is unstable. The synthetic continuous bilinear dynamical

system given in its state-space form by

$$\dot{\boldsymbol{x}}(t) = A_c\boldsymbol{x}(t) + \sum_{i=1}^{2} N_{c_i}\boldsymbol{x}(t)u_i(t) + B_c\boldsymbol{u}(t) \tag{7.14a}$$

$$\boldsymbol{y}(t) = C\boldsymbol{x}(t) + D\boldsymbol{u}(t) \tag{7.14b}$$

with

$$A_c = \begin{bmatrix} 0 & 1 & 0 \\ -1 & 0 & 0 \\ 0 & 0 & 0.3 \end{bmatrix}, \quad N_{c_1} = \begin{bmatrix} 1 & -1 & 0 \\ 0 & 2 & 1 \\ 1 & 3 & 4 \end{bmatrix}, \quad N_{c_2} = \begin{bmatrix} 0 & 0 & 1 \\ 1 & 0 & 1 \\ 4 & 2 & 1 \end{bmatrix}, \tag{7.15}$$

$$B_c = \begin{bmatrix} 1 & 0 \\ 0 & 2 \\ 1 & 1 \end{bmatrix}, \quad C = \begin{bmatrix} 1 & 0 & 1 \\ -1 & 1 & 2 \end{bmatrix}, \quad D = \begin{bmatrix} 0 & 0 \\ 0 & 0 \end{bmatrix}. \tag{7.16}$$

The procedure is applied with $N_1 = N_2 = 10$ and a frequency of acquisition $f = 20$ Hz for a total time of 5 seconds. As is the case for linear systems, the realized system matrices are not unique, because the state space description is not unique. However, the input/output mapping should be unique and the linear part of the identified system matrix should have the same eigenvalues as the true system matrix. The errors in the system matrix eigenvalues (between true and identified) are

$$\left|\left|\lambda\left(A_c\right) - \lambda\left(\hat{A}_c\right)\right|\right| \simeq 10^{-12} \tag{7.17a}$$

$$\left|\left|\lambda\left(N_{c_1}\right) - \lambda\left(\hat{N}_{c_1}\right)\right|\right| \simeq 10^{-12} \tag{7.17b}$$

$$\left|\left|\lambda\left(N_{c_2}\right) - \lambda\left(\hat{N}_{c_2}\right)\right|\right| \simeq 10^{-12} \tag{7.17c}$$

The identified system was subject to some test inputs and the response from the true system to the same test inputs was performed. The test inputs applied to the plants are

$$\boldsymbol{u}(t) = \begin{bmatrix} \sin(7t) \\ \cos(10t) \end{bmatrix}. \tag{7.18}$$

Output profiles obtained from the true and identified systems are visualized in Figure 7.2 and the SVD plot is displayed in Figure 7.3.

7.4.2 Mass-Spring-Damper System with Varying Stiffness

Consider a mass-spring-damper system with one degree of freedom with no forcing input and a time-varying stiffness $k(t)$:

$$m\ddot{x} + c\dot{x} + k(t)x = 0. \tag{7.19}$$

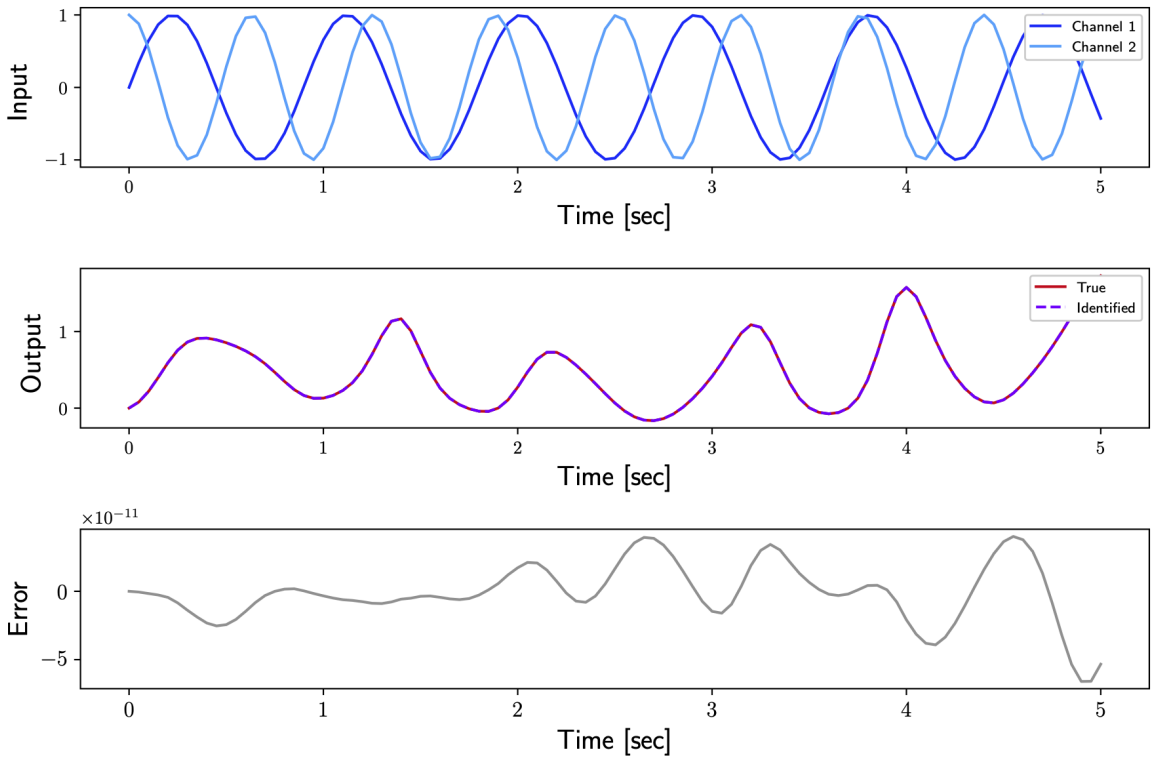


Figure 7.2: Error on the predicted output from test input

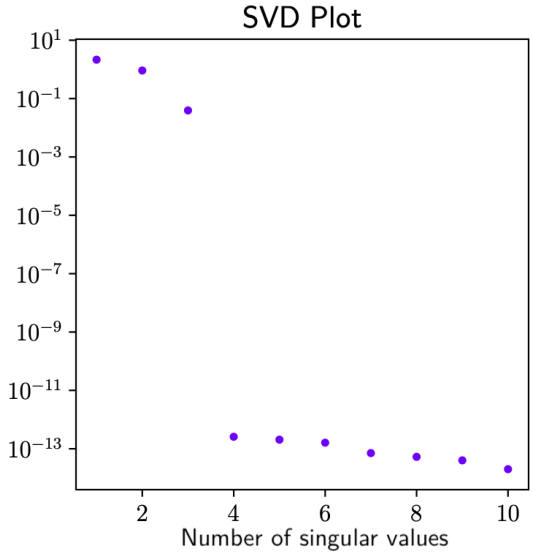


Figure 7.3: SVD plot

This system can be seen as a continuous bilinear dynamical system, given in its state-space form by

$$\dot{\boldsymbol{x}}(t) = A_c\boldsymbol{x}(t) + N_c\boldsymbol{x}(t)u(t) \tag{7.20a}$$

$$\boldsymbol{y}(t) = C\boldsymbol{x}(t) \tag{7.20b}$$

$$u(t) = k(t) \tag{7.20c}$$

with

$$A_c = \begin{bmatrix} 0 & 1 \\ 0 & -c/m \end{bmatrix}, \quad N_c = \begin{bmatrix} 0 & 0 \\ -1/m & 0 \end{bmatrix}, \quad C = \begin{bmatrix} 1 & 0 \end{bmatrix}. \tag{7.21}$$

The parameters are given by

$$m = 2, \quad c = 0.5, \quad \boldsymbol{x}_0 = \begin{bmatrix} 2 & -1 \end{bmatrix}^\top. \tag{7.22}$$

The procedure described earlier is applied with $N_1 = N_2 = 10$ and a frequency of acquisition $f = 10$ Hz for a total time of 20 seconds. The errors in the system matrix eigenvalues (between true and identified) are

$$\left|\left|\lambda\left(A_c\right) - \lambda\left(\hat{A}_c\right)\right|\right| \simeq 10^{-11}, \tag{7.23}$$

$$\left|\left|\lambda\left(N_c\right) - \lambda\left(\hat{N}_c\right)\right|\right| \simeq 10^{-11}. \tag{7.24}$$

As is evident, the eigenvalues of the linear part of the identified system and the full nonlinear response were again captured with high precision in example. The identified system was subject to some test input $k(t) = 5 + 4\cos(2t)$ and the response from the true system to the same test input was performed. Output profiles obtained from the true and identified systems are visualized in Figure 7.4 and SVD plot in Figure 7.5.

7.4.3 Chain Oscillator

The schematic of an n-degree-of-freedom, nonlinear oscillator chain is given Figure 7.6, where each spring has linear stiffness k_i [N/m] and cubic stiffness κ_i [N/m^3], each damper has linear damping coefficient c_i [N.s/m] and each mass a mass m_i ([kg]). The equations of motion for the n-mass oscillator chain, are given by

$$M\ddot{\boldsymbol{x}} + \zeta\dot{\boldsymbol{x}} + K\boldsymbol{x} + \boldsymbol{f}(\boldsymbol{x}) = \boldsymbol{0}, \quad \boldsymbol{x}(t) \in \mathbb{R}^n \tag{7.25}$$

where

$$M = \text{diag}(m_1, m_2, \cdots, m_n) \tag{7.26}$$

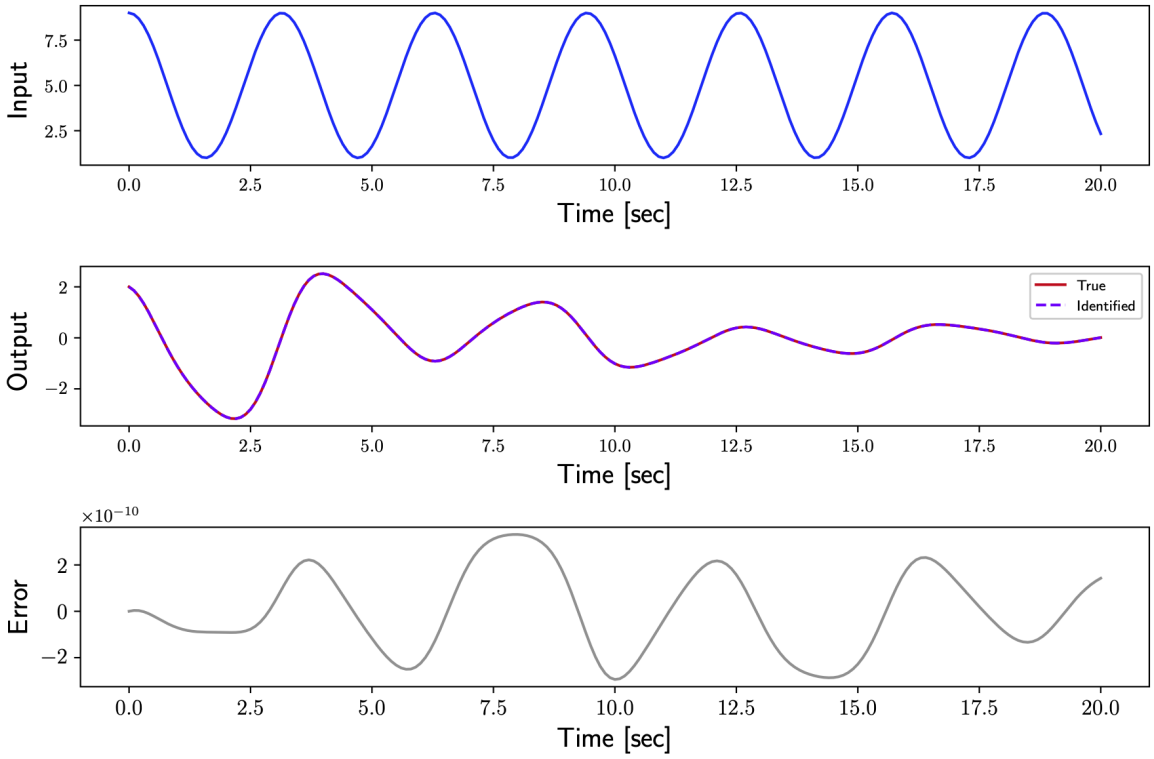


Figure 7.4: Error on the predicted output from test input

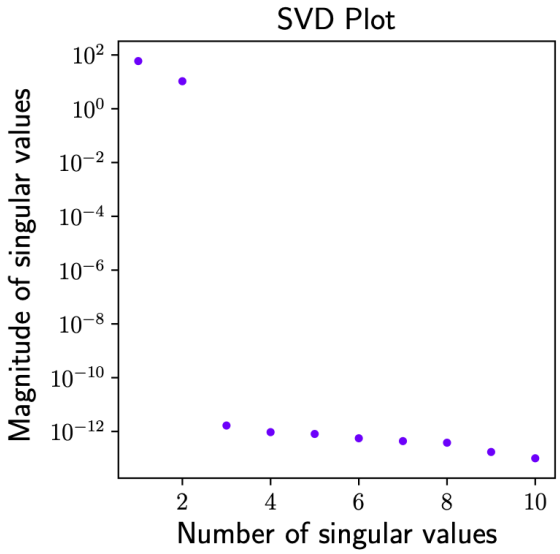


Figure 7.5: SVD plot

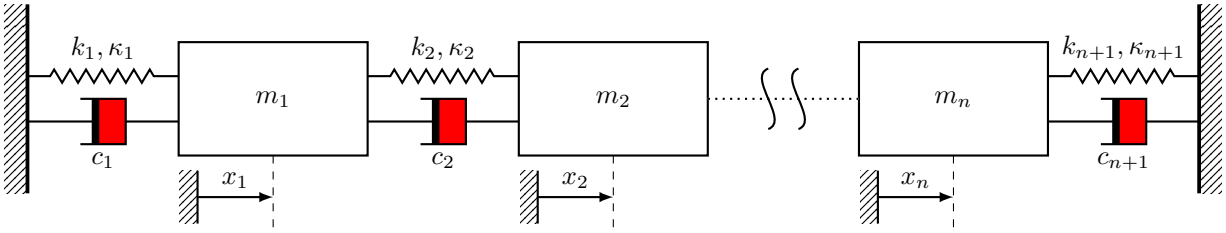


Figure 7.6: n-degree-of-freedom, nonlinear oscillator chain

$$\zeta = \begin{bmatrix} c_1 + c_2 & -c_2 & & & \\ -c_2 & c_2 + c_3 & -c_3 & & \\ & \ddots & \ddots & \ddots & \\ & & -c_{n-1} & c_{n-1} + c_n & -c_n \\ & & & -c_n & c_{n+1} \end{bmatrix} \tag{7.27}$$

$$K = \begin{bmatrix} k_1 + k_2 & -k_2 & & & \\ -k_2 & k_2 + k_3 & -k_3 & & \\ & \ddots & \ddots & \ddots & \\ & & -k_{n-1} & k_{n-1} + k_n & -k_n \\ & & & -k_n & k_{n+1} \end{bmatrix} \tag{7.28}$$

and $\boldsymbol{f}(\boldsymbol{x}) = \kappa \boldsymbol{f}_3 \boldsymbol{x}^{\otimes 3}$ where $\boldsymbol{f}_3 \in \mathbb{R}^{n \times n^3}$ is a sparse cubic-coefficients array such that

$$\boldsymbol{f}_3 \boldsymbol{x}^{\otimes 3} = \begin{bmatrix} x_1^3 - (x_2 - x_1)^3 \\ (x_2 - x_1)^3 - (x_3 - x_2)^3 \\ \vdots \\ (x_n - x_{n-1})^3 - x_n^3 \end{bmatrix} \in \mathbb{R}^n \tag{7.29}$$

For time being, it is assumed that $\kappa = 0$ and that the stiffness matrix is deviated from a nominal $\boldsymbol{K}_0$:

$$K = K_0 + \Delta K. \tag{7.30}$$

Thus, the equations of motion are

$$\dot{\mathbf{x}} = \begin{bmatrix} \dot{\boldsymbol{x}} \\ \ddot{\boldsymbol{x}} \end{bmatrix} = A_c \mathbf{x} + \sum_{i=1}^{n+1} N_{c_i} \mathbf{x} \Delta k_i \tag{7.31}$$

with

$$A_c = \begin{bmatrix} \mathbf{0}_{n\times n} & I_{n\times n} \\ -M^{-1}K_0 & -M^{-1}\zeta \end{bmatrix} \in \mathbb{R}^{2n\times 2n} \tag{7.32}$$

$$\boldsymbol{N}_{c_i} = \begin{bmatrix} \mathbf{0}_{n\times n} & \mathbf{0}_{n\times n} \\ \tilde{N}_{c_i} & \mathbf{0}_{n\times n} \end{bmatrix} \tag{7.33}$$

$$\tilde{N}_{c_1} = \begin{bmatrix} -\frac{1}{m_1} & 0 & 0 & \cdots & 0 \\ 0 & 0 & 0 & \cdots & 0 \\ 0 & 0 & 0 & \cdots & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & 0 & \cdots & 0 \end{bmatrix} \tag{7.34}$$

$$\tilde{N}_{c_2} = \begin{bmatrix} -\frac{1}{m_1} & \frac{1}{m_1} & 0 & \cdots & 0 \\ \frac{1}{m_2} & -\frac{1}{m_2} & 0 & \cdots & 0 \\ 0 & 0 & 0 & \cdots & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & 0 & \cdots & 0 \end{bmatrix} \tag{7.35}$$

$$\tilde{N}_{c_3} = \begin{bmatrix} 0 & 0 & 0 & \cdots & 0 \\ 0 & -\frac{1}{m_2} & \frac{1}{m_2} & \cdots & 0 \\ 0 & \frac{1}{m_2} & -\frac{1}{m_3} & \cdots & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & 0 & \cdots & 0 \end{bmatrix} \tag{7.36}$$

$$\tilde{N}_{c_{n+1}} = \begin{bmatrix} 0 & 0 & 0 & \cdots & 0 \\ 0 & 0 & 0 & \cdots & 0 \\ 0 & 0 & 0 & \cdots & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & 0 & \cdots & -\frac{1}{m_n} \end{bmatrix} \tag{7.37}$$

In addition, we suppose we have access to the position of the n masses through the measurement equation

$$\mathbf{y} = C\mathbf{x}, \quad C = I \tag{7.38}$$

In this example we suppose $n = 5$. After identification, we have a good match on the eigenvalues of A_c and bilinear matrices N_c:

$$\left|\left|\lambda\left(A_c\right) - \lambda\left(\hat{A}_c\right)\right|\right| \simeq 10^{-13} \tag{7.39}$$

$$\left|\left|\lambda\left(N_c\right) - \lambda\left(\hat{N}_c\right)\right|\right| \simeq 10^{-7} \tag{7.40}$$

as well as very good identification results (see Figure 7.7). The testing is performed by randomly switching stiffnesses by intervals of 5 seconds around a nominal of 20.

For this example, it is also desired to analyze the sensitivity of the measured outputs with respect to the stiffnesses $k_i(t)$. First, the general equations for sensitivity analysis from a bilinear system identification approach will be outlined; the method will then be applied to the chain oscillator identified model.

For a generic nonlinear dynamical system,

$$\dot{\boldsymbol{x}}(t) = \boldsymbol{f}(\boldsymbol{x}, \boldsymbol{\theta}) \tag{7.41}$$

we define the sensitivity of the state with respect to the parameter $\boldsymbol{\theta} \in \mathbb{R}^r$ by

$$\Psi(t) = \frac{\partial \boldsymbol{x}}{\partial \boldsymbol{\theta}}(t). \tag{7.42}$$

The differential equation that governs the evolution of this sensitivity matrix Ψ is thus

$$\dot{\Psi}(t) = \frac{\partial \boldsymbol{f}}{\partial \boldsymbol{\theta}} + \frac{\partial \boldsymbol{f}}{\partial \boldsymbol{x}}\frac{\partial \boldsymbol{x}}{\partial \boldsymbol{\theta}}(t) = \frac{\partial \boldsymbol{f}}{\partial \boldsymbol{\theta}} + \frac{\partial \boldsymbol{f}}{\partial \boldsymbol{x}}\Psi(t). \tag{7.43}$$

For a bilinear dynamical system as represented in Eq. (7.7), we simply have

$$\frac{\partial \boldsymbol{f}}{\partial \boldsymbol{\theta}} = \begin{bmatrix} N_{c_1}\boldsymbol{x} & N_{c_2}\boldsymbol{x} & \cdots & N_{c_r}\boldsymbol{x} \end{bmatrix} \tag{7.44a}$$

$$\frac{\partial \boldsymbol{f}}{\partial \boldsymbol{x}} = A_c + \sum_{i=1}^{r} N_{c_i}\theta_i. \tag{7.44b}$$

Note that if $\boldsymbol{f}(\boldsymbol{x}, \boldsymbol{\theta}) = \theta \boldsymbol{x}$, $\dot{\Psi}(t) = \boldsymbol{x} + \theta\Psi(t)$.

Leverage bilinear system identification techniques by getting an estimate for matrices A_c and N_{c_i}, it is possible to obtain an estimate of Ψ over time thus getting an estimate of the sensitivity of the state with respect to the parameter vector. The sensitivity of the state vector with respect to the first input $k_1(t)$ is displayed Figure 7.8. The accurate results confirm the validity of the bilinear-based approach.

7.4.4 Hovering Helicopter

The example of a hovering helicopter under wind disturbance as well as model parameter uncertainties [24,28] is considered. The dynamics of the system are given by

$$\dot{\boldsymbol{x}} = A\boldsymbol{x} + B\delta + B_w u_w \tag{7.45}$$

where

$$\boldsymbol{x} = \begin{bmatrix} u_h & q_h & \theta_h & y \end{bmatrix}^T, \tag{7.46}$$

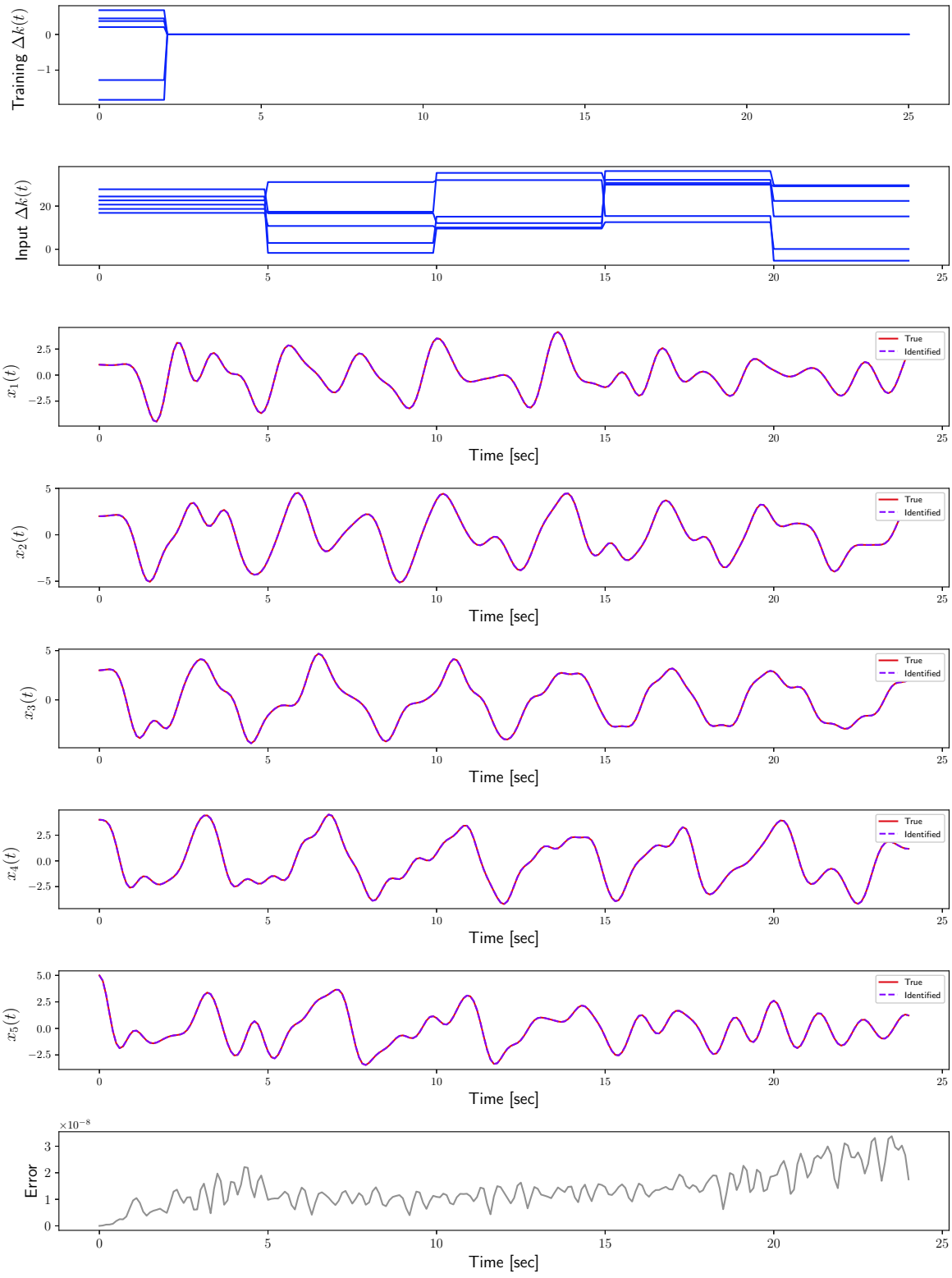


Figure 7.7: Prediction accuracy for the chain oscillator

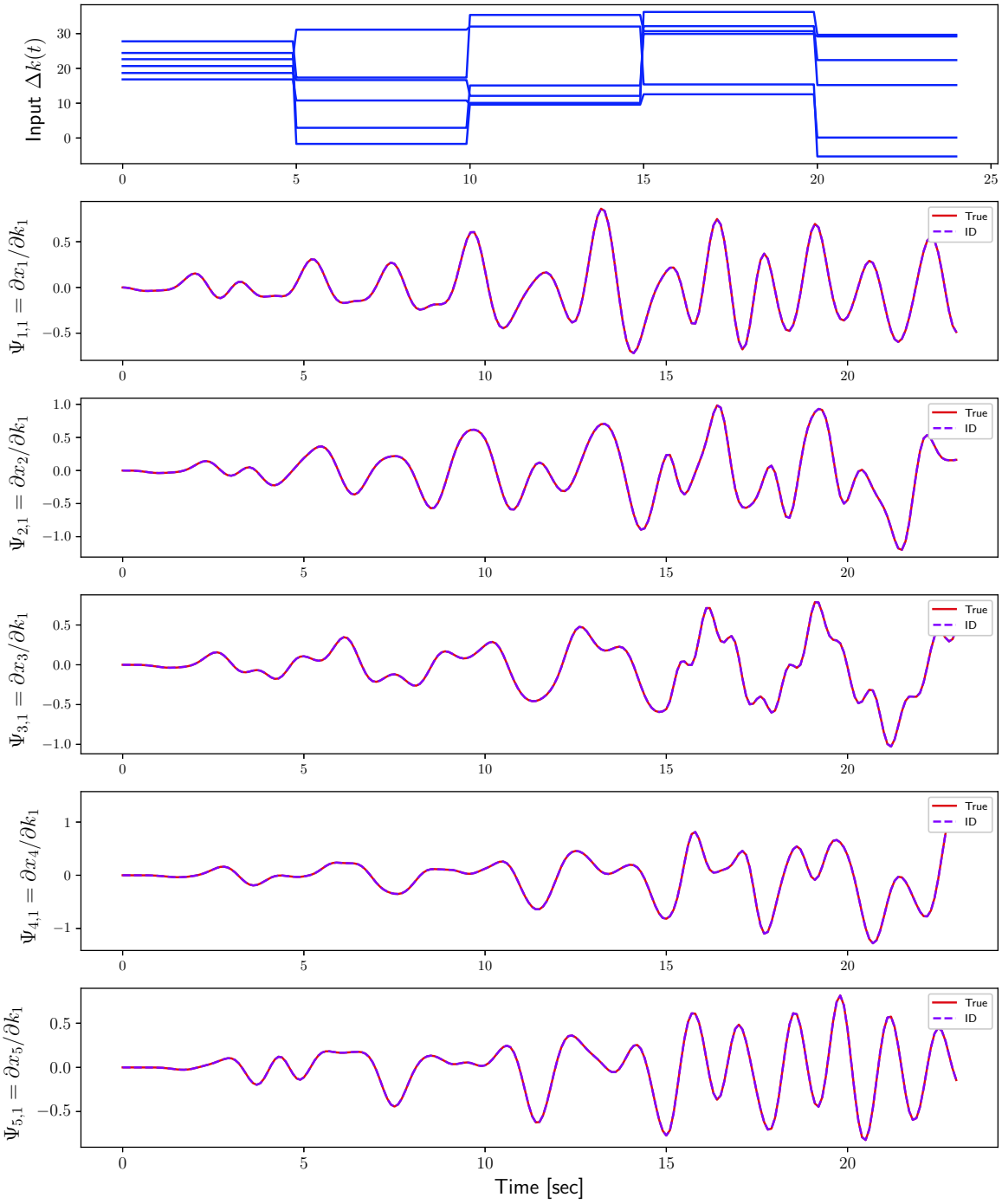


Figure 7.8: Sensitivity of the state vector with respect to the first input $k_1(t)$

$$A = \begin{bmatrix} p_1 & p_2 & -g & 0 \\ p_3 & p_4 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 1 & 0 & 0 & 0 \end{bmatrix}, \tag{7.47}$$

$$B = \begin{bmatrix} p_5 \\ p_6 \\ 0 \\ 0 \end{bmatrix}, \tag{7.48}$$

$$B_w = \begin{bmatrix} -p_1 \\ -p_3 \\ 0 \\ 0 \end{bmatrix}, \tag{7.49}$$

u_h [ft/s] represents the horizontal velocity of the helicopter, θ_h [$\times 10^{-2}$ rad] represents the pitch angle, q_h [$\times 10^{-2}$ rad/s] represents the pitch angular velocity and y [ft] represents the horizontal perturbation from a ground point reference. g corresponds to the acceleration due to gravity and is equal to 0.322. δ represents the control input to the system. u_w represents the wind disturbance on the helicopter and is modeled as a zero mean Gaussian white noise with variance $\sigma_w^2 = 18$.
The model comprises of six model parameters p_1 to p_6. The first four parameters $p_1 \breve{} p_4$ represent the aerodynamic stability derivatives while the parameters p_5 and p_6 represent the aerodynamic control derivatives. For identification purposes, initial conditions to the system are assumed to be zero: $\boldsymbol{x}_0 = \mathbf{0}$. The control law implemented is that of a full state feedback [24] where

$$\delta = -K\boldsymbol{x}, \tag{7.50}$$

and $K = \begin{bmatrix} 1.9890 & 0.2560 & 0.7589 & 1 \end{bmatrix}^T$. On substituting the control law in the original system, one obtains the closed-loop stochastic system

$$\dot{\boldsymbol{x}} = A_c\boldsymbol{x} + B_w u_w \tag{7.51}$$

where $A_c = A - BK$. Similar to Ref. [24], it is assumed that parameters $\boldsymbol{p} = \begin{bmatrix} p_1 & p_2 & p_3 & p_4 \end{bmatrix}^T$ are uncertain. Eq. 7.51 can be re-written in the form of a bilinear system

$$\dot{\boldsymbol{x}} = \tilde{A}_c\boldsymbol{x} + \sum_{i=1}^{4} N_{c_i}\boldsymbol{x}u_i + B_c\boldsymbol{u}, \tag{7.52}$$

with the augmented input vector being $\boldsymbol{u} = \begin{bmatrix} p_1 & p_2 & p_3 & p_4 & p_1 u_w & p_3 u_w \end{bmatrix}^T$ and the continuous time-invariant system matrices defines as

$$\tilde{A}_c = \begin{bmatrix} -p_5 k_1 & -p_5 k_2 & -g - p_5 k_3 & -p_5 k_4 \\ -p_6 k_1 & -p_6 k_2 & -p_6 k_3 & -p_6 k_4 \\ 0 & 1 & 0 & 0 \\ 1 & 0 & 0 & 0 \end{bmatrix}, \tag{7.53}$$

$$N_{c_1} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix}, \quad N_{c_2} = \begin{bmatrix} 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix}, \tag{7.54}$$

$$N_{c_3} = \begin{bmatrix} 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix}, \quad N_{c_4} = \begin{bmatrix} 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix}, \tag{7.55}$$

$$N_{c_5} = \begin{bmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix}, \quad N_{c_6} = \begin{bmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix}, \tag{7.56}$$

$$B_c = \begin{bmatrix} 0 & 0 & 0 & 0 & -1 & 0 \\ 0 & 0 & 0 & 0 & 0 & -1 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix}. \tag{7.57}$$

For testing, the parameters $p_1 \breve{} p_4$ are monotonically varied between the lower and upper bounds

$$\boldsymbol{p}_{lb} = \begin{bmatrix} -0.0488 & 0.0013 & 0.126 & -3.3535 \end{bmatrix}^T, \tag{7.58}$$

$$\boldsymbol{p}_{ub} = \begin{bmatrix} -0.0026 & 0.0247 & 2.394 & -0.1765 \end{bmatrix}^T. \tag{7.59}$$

Results of the identification are presented Figure 7.9. With eigenvalues of A_c and bilinear matrices N_c matching up to machine precision,

$$\left|\left| \lambda\left(A_c\right) - \lambda\left(\hat{A}_c\right) \right|\right| \simeq 10^{-12} \tag{7.60}$$

$$\left|\left| \lambda\left(N_c\right) - \lambda\left(\hat{N}_c\right) \right|\right| \simeq 10^{-12} \tag{7.61}$$

the identified bilinear model is able to reproduce the dynamics of the true model. Figure 7.10 shows the sensitivity of the state vector with respect to the first parameter p_1.

7.4.5 Controlled Duffing Oscillator

This example corresponds to the controlled nonlinear Duffing oscillator governed by the following equations

$$\dot{x} = y + g_1 u_1, \tag{7.62a}$$

$$\dot{y} = -\delta y - \alpha x - \beta x^3 + g_2 u_2 \tag{7.62b}$$

with parameters $\alpha = 1$, $\delta = -0.1$, $g_1 = 0$ and $g_2 = 1$. For this example, we want to study the capabilities of a bilinear system to approximate nonlinear dynamics in presence of an external input. First, it is desired to visualize the effect of the nonlinearity coefficient β on the identification capabilities of bilinear system identification algorithms. Figures 7.11(a) to 7.11(c) show the predicted trajectories for different values of β, with a testing input of $u_2(t) = 0.05\cos(2\pi t + 2\pi/3)$. Even though the trajectories are very similar (the domain of interest is centered at 0 with radius of 0.02, making the nonlinear coefficient almost inconsequential), the nonlinear term has a huge impact on the approximation capabilities. A classic bilinear approach is thus valid for small nonlinearities in contained domains, but has the domain of interest grows and the nonlinear term becomes more and more significant, an other approach is desired.

One could augment the measurement vector with additional lifting functions, giving rise to a bilinear Koopman operator. Figures 7.12(a) to 7.12(f) present the approximation capabilities with increasing order of the bilinear Koopman operator. Reaching order 6, or a dimension of the operator of 27, the approximation is excellent, confirming that a bilinear system identification approach in conjunction with the Koopman framework is a valid method to approximate controlled nonlinear systems.

7.4.6 Controlled Van Der Pol Oscillator

This example corresponds to the controlled Van Der Pol oscillator governed by the following equations

$$\dot{x} = y + g_1 u_1, \tag{7.63a}$$

$$\dot{y} = \mu(1 - x^2)y - x + g_2 u_2 \tag{7.63b}$$

with $\mu = 0.8$, $g_1 = 0$ and $g_2 = 1$. Similarly as the previous example, Figures 7.13(a) to 7.13(f) present the approximation capabilities with increasing order of the bilinear Koopman operator, with a testing input of $u_2(t) = 0.5\cos(2\pi t\ 5 + 2\pi/3)$.

7.5 Conclusion

This chapter has introduced the concept of bilinear Koopman operator. Controlled systems in the context of Koopman yield bilinear dynamics in a lifted space. Since some physical systems are inherently bilinear and bilinear models of high dimension can approximate a broad class of nonlinear systems, this chapter has offered some perspectives and advances for bilinear system identification, working

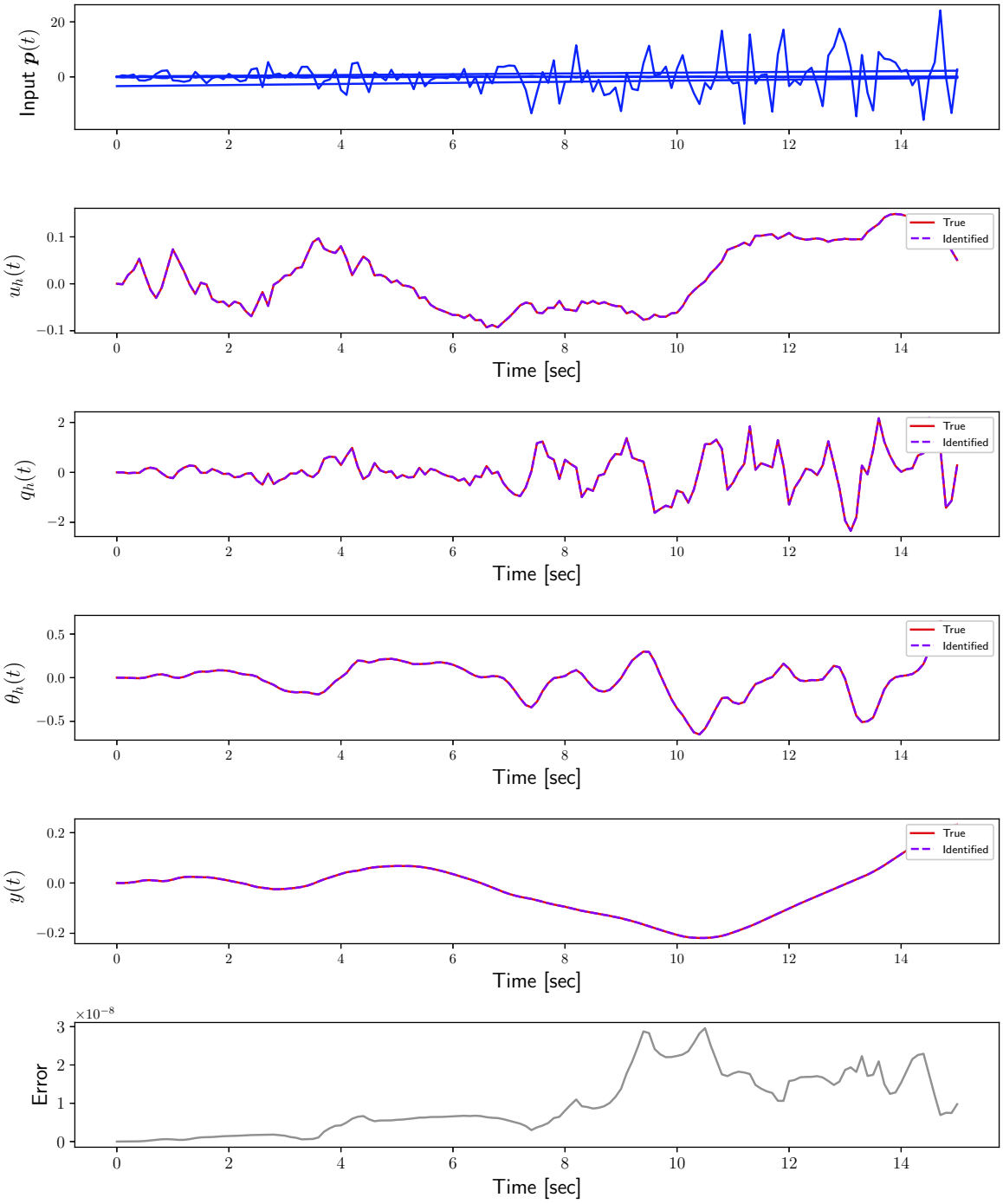


Figure 7.9: Prediction accuracy for the hovering helicopter

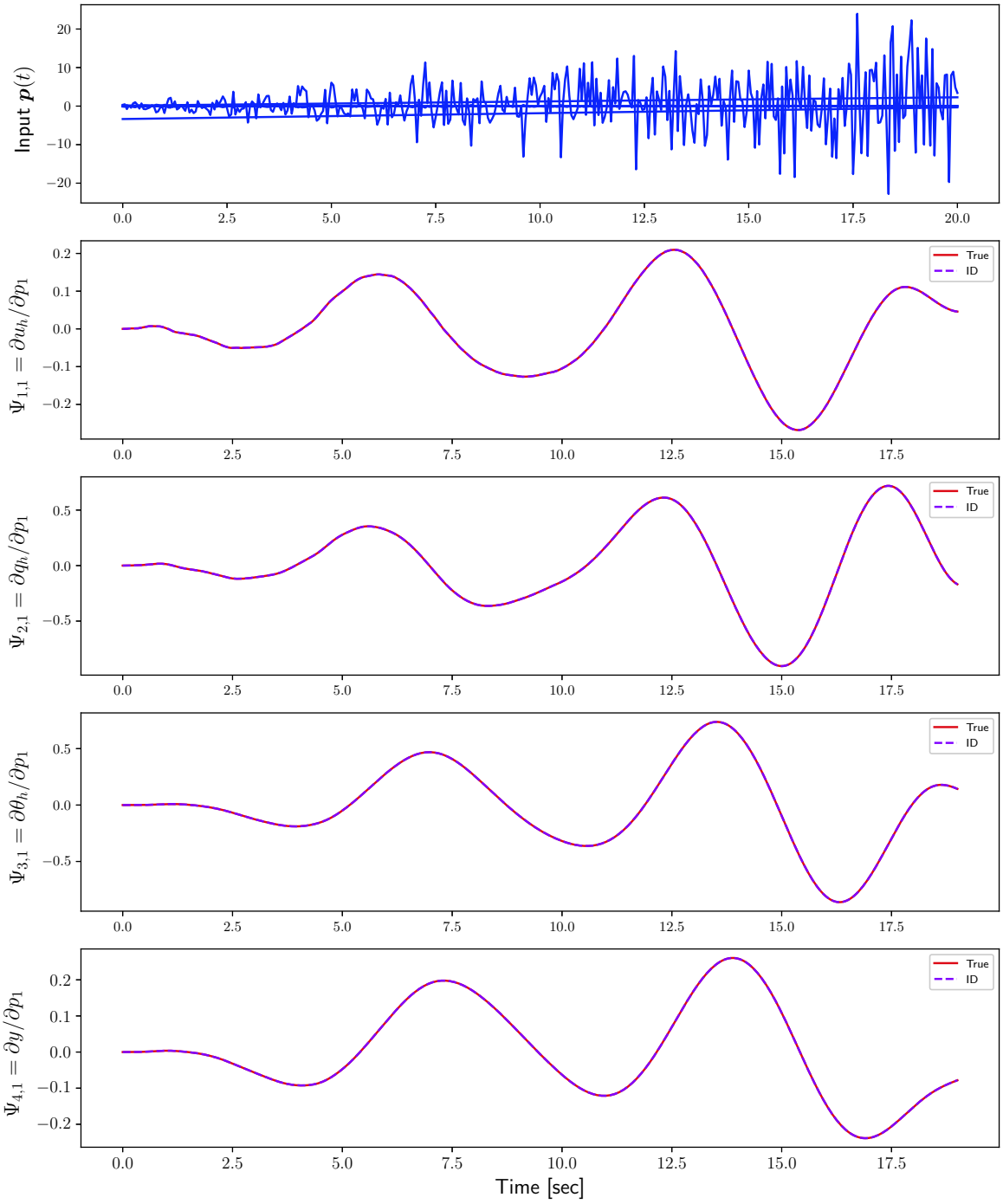
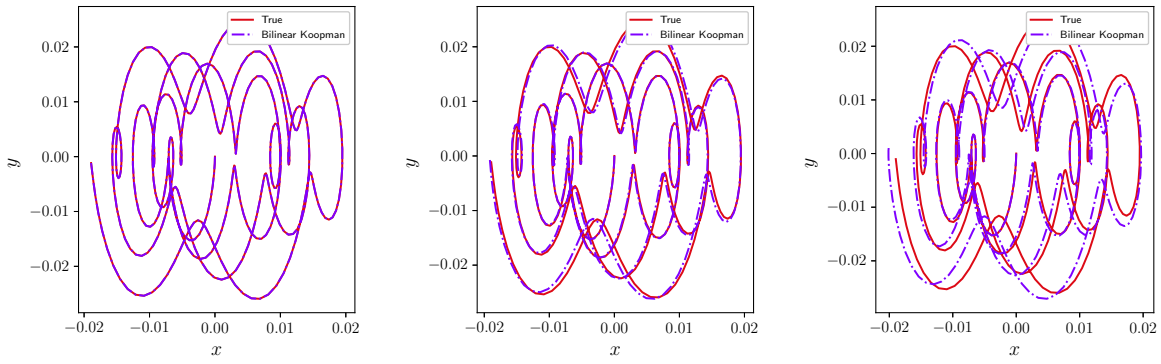
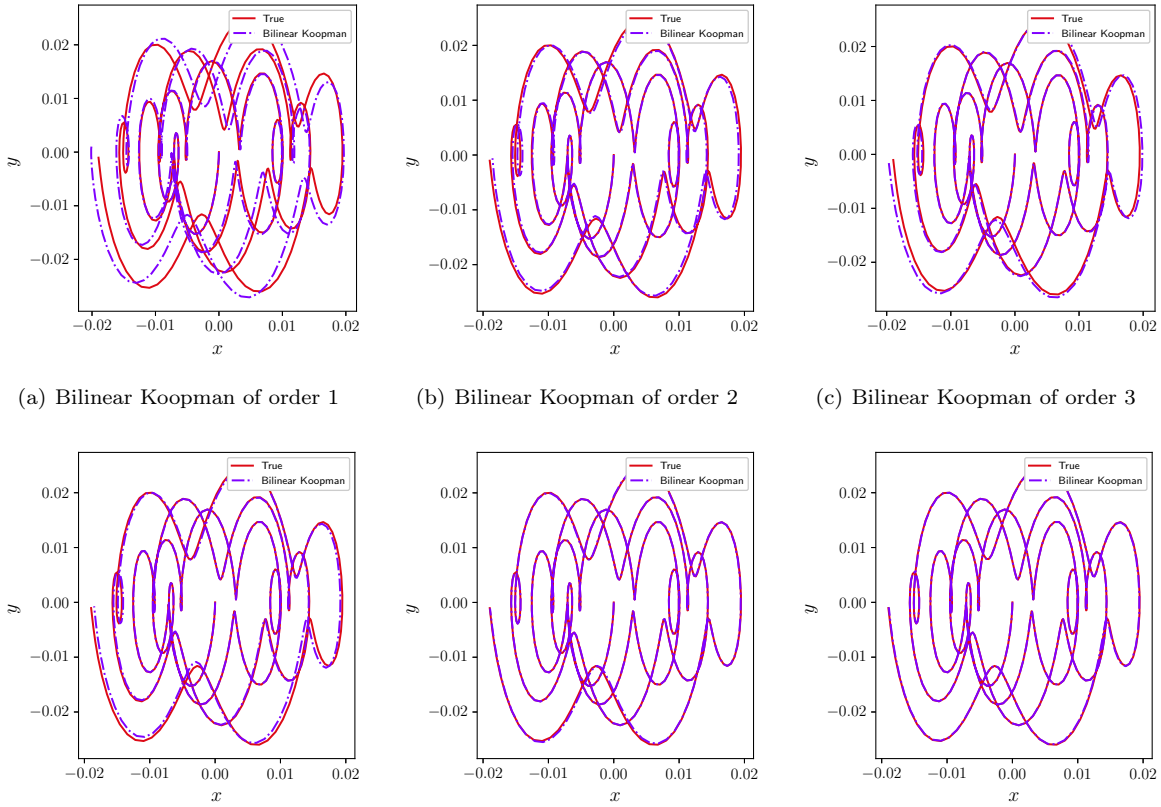


Figure 7.10: Sensitivity accuracy for the hovering helicopter



(a) Linear case: $\beta = 0$ (b) Slightly nonlinear case: $\beta = 0.6$ (c) Full nonlinear case: $\beta = 5$

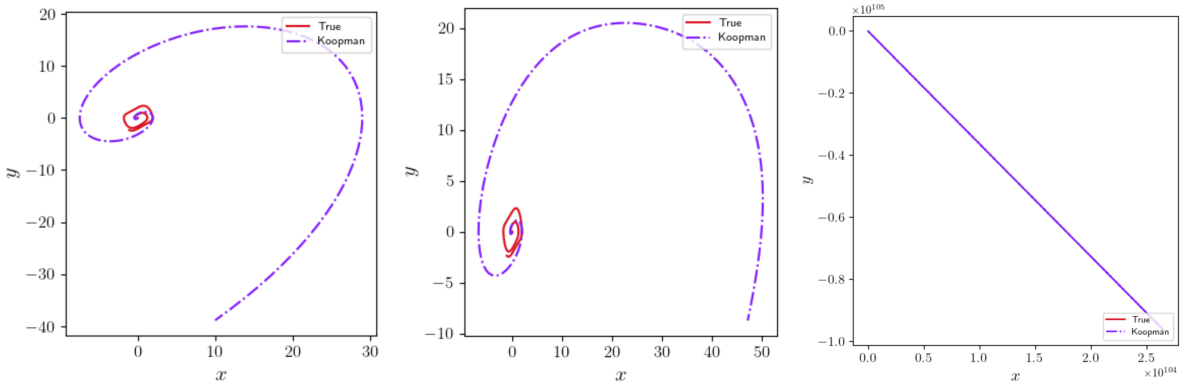
Figure 7.11: Classical bilinear system identification of the nonlinear Duffing oscillator with increasing degree of nonlinearity



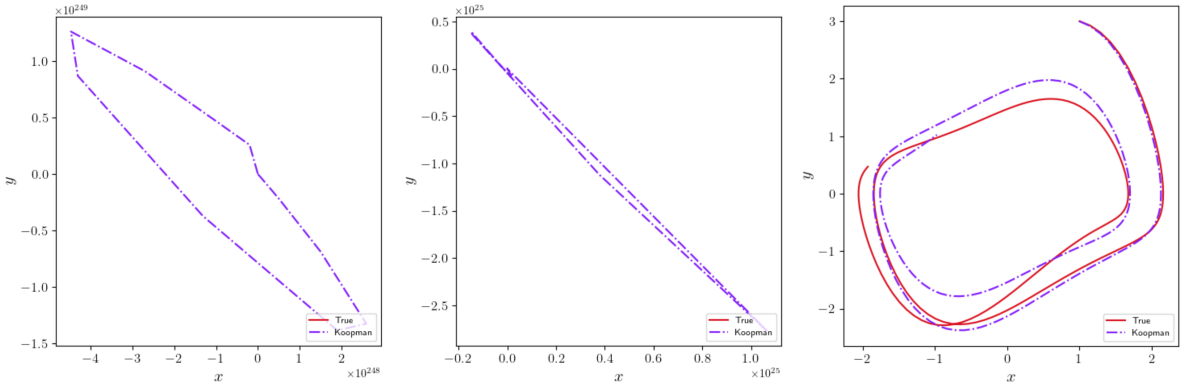
(a) Bilinear Koopman of order 1 (b) Bilinear Koopman of order 2 (c) Bilinear Koopman of order 3

(d) Bilinear Koopman of order 4 (e) Bilinear Koopman of order 5 (f) Bilinear Koopman of order 6

Figure 7.12: Varying orders of Koopman bilinear system identification of the nonlinear Duffing oscillator with $\beta = 5$



(a) Bilinear Koopman of order 1 (b) Bilinear Koopman of order 2 (c) Bilinear Koopman of order 3



(d) Bilinear Koopman of order 4 (e) Bilinear Koopman of order 5 (f) Bilinear Koopman of order 6

Figure 7.13: Varying orders of Koopman bilinear system identification of the nonlinear Van Der Pol oscillator with $\mu = 0.8$

towards a bilinear Koopman operator. Several numerical simulations confirm the growing interest in bilinear system identification and validate the methods and algorithms presented in this chapter. The same framework is employed for sensitivity analysis of nonlinear systems where it is desired to estimate the variation of a measured output given the deviation of a constitutive parameter of the system.

The methods developed in this chapter and more generally in this dissertation assume good quality data - or at least *good enough* quality - so that the identified model is reliable. While it is not the objective here to define *good enough* data, the next chapter is aimed at opening the discussion to mitigate noise in the data. More specifically, two robust and reliable methods are presented and validated on several numerical simulations to enhance system identification algorithms in the presence of high noise levels.

Chapter 8
Mitigating Noise in the Data

8.1 Introduction

This dissertation has introduced several data-driven identification methods and algorithms. However, the main difficulty in linear and nonlinear system identification applications stems from the interplay of noise and unmodeled dynamics. Noise, finite length of data, and parameters variation are some of the issues that limit the application of such methods and there are many instances when this limitation is significant enough that it becomes necessary to deal with situations where no model in the model set can adequately describe the real system behavior. This chapter introduces a data-correlation approach to the time-varying eigensystem realization algorithm (TVERA/DC) has potential way to temper the effect of noise.

Secondly, we recall that the main role of subspace identification methods is to find a smaller space in which the dynamics of a true system are evolving and the key step in this subspace identification method is a singular value decomposition (SVD) of the Hankel matrix used to estimate the order of the system. Due to measurement noise, nonlinearity, and computer round-off, the Hankel matrix will usually be of full rank which does not, in general, equal the true order of the system under test. The reliance on the SVD for the low-rank approximation makes it difficult to seamlessly apply the subspace methods to problems with important levels of noise. This chapter also investigates a method to minimize the nuclear norm of the Hankel matrix as a way to offer an interesting alternative as a heuristic for low-rank approximation problems where a clear-cut criterion for model selection cannot be deduced from the SVD. Once the dimension of this subspace is found, a projected model is derived based on a factorization of a structured Hankel matrix constructed from input-output data. To obtain this low rank model, a nuclear norm minimization method for estimating the dimension of the system matrices of a linear time invariant (LTI) state-space model in the presence of measurement noise is presented. In the second part of this chapter, we will consider two numerical examples of different dimension with various levels of noise and show the efficacy of the proposed approach.

8.2 Subspace System Identification Including Data Correlations

While classical state-space realization techniques are, in essence, a least-squares fit to the pulse response measurements, introducing output auto-correlation and cross-correlations over a defined number of lag values has the potential to temper the effect of noise. This section introduces a data-correlation approach to the time-varying eigensystem realization algorithm (TVERA/DC) and we will see that the bias terms affecting the TVERA when "white" measurement noise is present can be omitted in the TVERA/DC by properly tuning some of the parameters. First, this section will explain the basic concepts of data-correlation for time-invariant system realization (ERA/DC) and will provide a description of this procedure adapted for time-varying system identification, introducing the time-varying eigensystem realization algorithm with data-correlation (TVERA/DC).

Recall that a discrete-time invariant linear system can be represented by

$$\boldsymbol{x}_{k+1} = A\boldsymbol{x}_k + B\boldsymbol{u}_k \tag{8.1a}$$

$$\boldsymbol{y}_k = C\boldsymbol{x}_k + D\boldsymbol{u}_k \tag{8.1b}$$

together with an initial state vector $\boldsymbol{x}_0$, where $\boldsymbol{x}_k \in \mathbb{R}^n$, $\boldsymbol{u}_k \in \mathbb{R}^r$ and $\boldsymbol{y}_k \in \mathbb{R}^m$ are the state, control input and output vectors respectively. The constant matrices $A \in \mathbb{R}^{n\times n}$, $B \in \mathbb{R}^{r\times n}$, $C \in \mathbb{R}^{n\times m}$ and $D \in \mathbb{R}^{r\times m}$ represent the internal operation of the linear system, and are used to determine the system's response to any input.

8.2.1 Time-Invariant Linear System Identification: the Eigensystem Realization Algorithm with Data Correlation (ERA/DC)

The eigensystem realization algorithm with data correlations (ERA/DC) includes an additional fit to output correlations whereas the ERA is basically a least-square fit to the pulse response measurements only. The bias terms affecting the ERA when noise is present can, in principle, be omitted in the ERA/DC by properly tuning some of the parameters. The computational steps of the ERA/DC are outlined in this section.

8.2.1.1 Block Correlation Hankel Matrices

The ERA method with data correlation requires the definition of a square matrix of order $\gamma = pm$,

$$\boldsymbol{\mathcal{R}}_{\boldsymbol{HH}}(k) = \boldsymbol{H}(k)\boldsymbol{H}(0)^\top \tag{8.2}$$

$$= \begin{bmatrix} h_{k+1} & h_{k+2} & \cdots & h_{k+q} \\ h_{k+2} & h_{k+3} & \cdots & h_{k+q+1} \\ \vdots & \vdots & \ddots & \vdots \\ h_{k+p} & h_{k+p+1} & \cdots & h_{k+p+q-1} \end{bmatrix} \begin{bmatrix} h_1 & h_2 & \cdots & h_q \\ h_2 & h_3 & \cdots & h_{q+1} \\ \vdots & \vdots & \ddots & \vdots \\ h_p & h_{p+1} & \cdots & h_{p+q-1} \end{bmatrix}^\top \tag{8.3}$$

$$= \begin{bmatrix} \sum_{i=1}^{q} h_{k+i}h_i^\top & \sum_{i=1}^{q} h_{k+i}h_{i+1}^\top & \cdots & \sum_{i=1}^{q} h_{k+i}h_{p+i-1}^\top \\ \sum_{i=1}^{q} h_{k+i+1}h_i^\top & \sum_{i=1}^{q} h_{k+i+1}h_{i+1}^\top & \cdots & \sum_{i=1}^{q} h_{k+i+1}h_{p+i-1}^\top \\ \vdots & \vdots & \ddots & \vdots \\ \sum_{i=1}^{q} h_{k+p+i-1}h_i^\top & \sum_{i=1}^{q} h_{k+p+i-1}h_{i+1}^\top & \cdots & \sum_{i=1}^{q} h_{k+p+i-1}h_{p+i-1}^\top \end{bmatrix}. \tag{8.4}$$

where h_k are Markov parameters in the case of forced systems or directly outputs of the system in the case of autonomous systems. Hankel matrices $\boldsymbol{H}(k)$ are defined the same way as in the ERA. Note that the data correlation matrix $\boldsymbol{\mathcal{R}_{HH}}(k)$ can be smaller in size than the Hankel matrix $\boldsymbol{H}(k)$ if $qr \leq pm = \gamma$. For the case when $k = 0$, the correlation matrix $\boldsymbol{\mathcal{R}_{HH}}(0)$ becomes

$$\boldsymbol{\mathcal{R}_{HH}}(0) = \boldsymbol{H}(0)\boldsymbol{H}(0)^\top \tag{8.5}$$

$$= \begin{bmatrix} h_1 & h_2 & \cdots & h_q \\ h_2 & h_3 & \cdots & h_{q+1} \\ \vdots & \vdots & \ddots & \vdots \\ h_p & h_{p+1} & \cdots & h_{p+q-1} \end{bmatrix} \begin{bmatrix} h_1 & h_2 & \cdots & h_q \\ h_2 & h_3 & \cdots & h_{q+1} \\ \vdots & \vdots & \ddots & \vdots \\ h_p & h_{p+1} & \cdots & h_{p+q-1} \end{bmatrix}^\top \tag{8.6}$$

$$= \begin{bmatrix} \sum_{i=1}^{q} h_i h_i^\top & \sum_{i=1}^{q} h_i h_{i+1}^\top & \cdots & \sum_{i=1}^{q} h_i h_{p+i-1}^\top \\ \sum_{i=1}^{q} h_{i+1}h_i^\top & \sum_{i=1}^{q} h_{i+1}h_{i+1}^\top & \cdots & \sum_{i=1}^{q} h_{i+1}h_{p+i-1}^\top \\ \vdots & \vdots & \ddots & \vdots \\ \sum_{i=1}^{q} h_{p+i-1}h_i^\top & \sum_{i=1}^{q} h_{p+i-1}h_{i+1}^\top & \cdots & \sum_{i=1}^{q} h_{p+i-1}h_{p+i-1}^\top \end{bmatrix}. \tag{8.7}$$

The matrix $\boldsymbol{\mathcal{R}_{HH}}(0)$ consists of auto-correlations of Markov parameters such as $\sum_{i=1}^{q} h_i h_i^\top$ and cross-correlations such as $\sum_{i=1}^{q} h_i h_{i+1}^\top$ at lag time values in the range $\pm p$, summed over q values. If noises in the Markov parameters (or outputs) are not correlated, the correlation matrix $\boldsymbol{\mathcal{R}_{HH}}(0)$ will contain less noise than the Hankel matrix $\boldsymbol{H}(0)$.

In terms of controllability and observability matrices, $\boldsymbol{\mathcal{R}_{HH}}(k)$ can be written as

$$\boldsymbol{\mathcal{R}_{HH}}(k) = \boldsymbol{O}^{(p)} A^k \boldsymbol{R}^{(q)} {\boldsymbol{R}^{(q)}}^\top {\boldsymbol{O}^{(p)}}^\top = \boldsymbol{O}^{(p)} A^k \boldsymbol{R}^{(\gamma)}, \tag{8.8}$$

where $\boldsymbol{R}^{(\gamma)} = \boldsymbol{R}^{(q)} {\boldsymbol{R}^{(q)}}^\top {\boldsymbol{O}^{(p)}}^\top$.

The data correlation matrix $\boldsymbol{\mathcal{R}_{HH}}(k)$ can be used to form a block correlation Hankel matrix

$$\begin{aligned}\boldsymbol{\mathcal{H}}(k) &= \begin{bmatrix} \boldsymbol{\mathcal{R}_{HH}}(k) & \boldsymbol{\mathcal{R}_{HH}}(k+\tau) & \cdots & \boldsymbol{\mathcal{R}_{HH}}(k+\zeta\tau) \\ \boldsymbol{\mathcal{R}_{HH}}(k+\tau) & \boldsymbol{\mathcal{R}_{HH}}(k+2\tau) & \cdots & \boldsymbol{\mathcal{R}_{HH}}(k+(\zeta+1)\tau) \\ \vdots & \vdots & \ddots & \vdots \\ \boldsymbol{\mathcal{R}_{HH}}(k+\xi\tau) & \boldsymbol{\mathcal{R}_{HH}}(k+(\xi+1)\tau) & \cdots & \boldsymbol{\mathcal{R}_{HH}}(k+(\xi+\zeta)\tau) \end{bmatrix} \\ &= \begin{bmatrix} \boldsymbol{O}^{(p)} \\ \boldsymbol{O}^{(p)}A^{\tau} \\ \vdots \\ \boldsymbol{O}^{(p)}A^{\xi\tau} \end{bmatrix} A^{k} \begin{bmatrix} \boldsymbol{R}^{(\gamma)} & A^{\tau}\boldsymbol{R}^{(\gamma)} & \cdots & A^{\zeta\tau}\boldsymbol{R}^{(\gamma)} \end{bmatrix} \\ &= \boldsymbol{O}^{(\xi)}A^{k}\boldsymbol{R}^{(\zeta)}. \end{aligned} \tag{8.9}$$

For the case when $k = 0$,

$$\begin{aligned}\boldsymbol{\mathcal{H}}(0) &= \begin{bmatrix} \boldsymbol{\mathcal{R}_{HH}}(0) & \boldsymbol{\mathcal{R}_{HH}}(\tau) & \cdots & \boldsymbol{\mathcal{R}_{HH}}(\zeta\tau) \\ \boldsymbol{\mathcal{R}_{HH}}(\tau) & \boldsymbol{\mathcal{R}_{HH}}(2\tau) & \cdots & \boldsymbol{\mathcal{R}_{HH}}((\zeta+1)\tau) \\ \vdots & \vdots & \ddots & \vdots \\ \boldsymbol{\mathcal{R}_{HH}}(\xi\tau) & \boldsymbol{\mathcal{R}_{HH}}((\xi+1)\tau) & \cdots & \boldsymbol{\mathcal{R}_{HH}}((\xi+\zeta)\tau) \end{bmatrix} \\ &= \begin{bmatrix} \boldsymbol{O}^{(p)} \\ \boldsymbol{O}^{(p)}A^{\tau} \\ \vdots \\ \boldsymbol{O}^{(p)}A^{\xi\tau} \end{bmatrix} \begin{bmatrix} \boldsymbol{R}^{(\gamma)} & A^{\tau}\boldsymbol{R}^{(\gamma)} & \cdots & A^{\zeta\tau}\boldsymbol{R}^{(\gamma)} \end{bmatrix} \\ &= \boldsymbol{O}^{(\xi)}\boldsymbol{R}^{(\zeta)}. \end{aligned} \tag{8.10}$$

τ is an integer chosen to chosen to prevent significant overlap of adjacent correlation blocks. The matrices $\boldsymbol{R}^{(\zeta)}$ and $\boldsymbol{O}^{(\xi)}$ are called the block correlation controllability and observability matrices.

8.2.1.2 Hankel Norm Approximation

Similarly to the ERA, the ERA/DC process continues with the factorization of the block correlation matrix $\boldsymbol{\mathcal{H}}(0)$ using singular value decomposition so that

$$\boldsymbol{\mathcal{H}}(0) = \boldsymbol{\mathcal{U}}\boldsymbol{\Sigma}\boldsymbol{\mathcal{V}}^{\top} = \begin{bmatrix} \boldsymbol{\mathcal{U}}^{(n)} & \boldsymbol{\mathcal{U}}^{(0)} \end{bmatrix} \begin{bmatrix} \boldsymbol{\Sigma}^{(n)} & \boldsymbol{0} \\ \boldsymbol{0} & \boldsymbol{\Sigma}^{(0)} \end{bmatrix} \begin{bmatrix} {\boldsymbol{\mathcal{V}}^{(n)}}^{\top} \\ {\boldsymbol{\mathcal{V}}^{(0)}}^{\top} \end{bmatrix} = \boldsymbol{\mathcal{U}}^{(n)}\boldsymbol{\Sigma}^{(n)}{\boldsymbol{\mathcal{V}}^{(n)}}^{\top} + \underbrace{\boldsymbol{\mathcal{U}}^{(0)}\boldsymbol{\Sigma}^{(0)}{\boldsymbol{\mathcal{V}}^{(0)}}^{\top}}_{\simeq \boldsymbol{0}} \simeq \boldsymbol{\mathcal{U}}^{(n)}\boldsymbol{\Sigma}^{(n)}{\boldsymbol{\mathcal{V}}^{(n)}}^{\top}, \tag{8.11}$$

and

$$\boldsymbol{\mathcal{H}}(0) = \boldsymbol{\mathcal{U}}^{(n)}\boldsymbol{\Sigma}^{(n)}{\boldsymbol{\mathcal{V}}^{(n)}}^{\top} = \boldsymbol{O}^{(\xi)}\boldsymbol{R}^{(\zeta)} \Rightarrow \begin{cases} \boldsymbol{O}^{(\xi)} = \boldsymbol{\mathcal{U}}^{(n)}{\boldsymbol{\Sigma}^{(n)}}^{1/2} \\ \boldsymbol{R}^{(\zeta)} = {\boldsymbol{\Sigma}^{(n)}}^{1/2}{\boldsymbol{\mathcal{V}}^{(n)}}^{\top} \end{cases} . \tag{8.12}$$

Again, this choice makes both $\boldsymbol{O}^{(\xi)}$ and $\boldsymbol{R}^{(\zeta)}$ balanced.

8.2.1.3 Minimum Realization

From Eq. (8.10) we have directly

$$\boldsymbol{O}^{(p)} = {\boldsymbol{E}^{(\gamma)}}^{\intercal}\boldsymbol{O}^{(\xi)} = {\boldsymbol{E}^{(\gamma)}}^{\top}\boldsymbol{\mathcal{U}}^{(n)}{\boldsymbol{\Sigma}^{(n)}}^{1/2}. \tag{8.13}$$

From Eq. (8.10), an expression of $\boldsymbol{R}^{(q)}$ can be found

$$\boldsymbol{R}^{(q)} = {\boldsymbol{O}^{(p)}}^{\dagger}\boldsymbol{H}(0) = \left[{\boldsymbol{E}^{(\gamma)}}^{\top}\boldsymbol{\mathcal{U}}^{(n)}{\boldsymbol{\Sigma}^{(n)}}^{1/2}\right]^{\dagger}\boldsymbol{H}(0), \tag{8.14}$$

and a realization is shown to be

$$\hat{A} = {\boldsymbol{O}^{(\xi)}}^{\dagger}\boldsymbol{\mathcal{H}}(1){\boldsymbol{R}^{(\zeta)}}^{\dagger} = {\boldsymbol{\Sigma}^{(n)}}{}^{-1/2}{\boldsymbol{\mathcal{U}}^{(n)}}^{\top}\boldsymbol{\mathcal{H}}(1){\boldsymbol{\mathcal{V}}^{(n)}}{\boldsymbol{\Sigma}^{(n)}}^{-1/2}, \tag{8.15a}$$

$$\hat{B} = \boldsymbol{R}^{(q)}\boldsymbol{E}^{(r)} = \left[{\boldsymbol{E}^{(\gamma)}}^{\top}\boldsymbol{\mathcal{U}}^{(n)}{\boldsymbol{\Sigma}^{(n)}}^{1/2}\right]^{\dagger}\boldsymbol{H}(0)\boldsymbol{E}^{(r)}, \tag{8.15b}$$

$$\hat{C} = {\boldsymbol{E}^{(m)}}^{\top}\boldsymbol{O}^{(p)} = {\boldsymbol{E}^{(m)}}^{\top}{\boldsymbol{E}^{(\gamma)}}^{\top}\boldsymbol{\mathcal{U}}^{(n)}{\boldsymbol{\Sigma}^{(n)}}^{1/2}, \tag{8.15c}$$

$$\hat{D} = h_0. \tag{8.15d}$$

8.2.2 Time-Varying Linear System Identification: the Time-Varying Eigensystem Realization Algorithm with Data-Correlation (TVERA/DC)

From a perspective of generalizing the classical Ho-Kalman approach with ERA and ERA/DC, this paper develops an extension of the time-varying eigensystem realization algorithm (TVERA) by including data correlations in the process (TVERA/DC). With the assumption that noises in data are not correlated, the resulting procedure takes advantage of the fact that auto-correlation and cross-correlation between outputs will contain less noise than original outputs.

Consider that we obtained $M \times N$ experiments, arranged in M batches of N experiments. Similarly as the TVERA procedure [28, 29], form the initial condition response experiments Hankel matrix

$$\tilde{\boldsymbol{H}}_k^{(p,N),\#j} = \begin{bmatrix} \boldsymbol{y}_k^{\#1,j} & \boldsymbol{y}_k^{\#2,j} & \cdots & \boldsymbol{y}_k^{\#N,j} \\ \boldsymbol{y}_{k+1}^{\#1,j} & \boldsymbol{y}_{k+1}^{\#2,j} & \cdots & \boldsymbol{y}_{k+1}^{\#N,j} \\ \vdots & \vdots & \ddots & \vdots \\ \boldsymbol{y}_{k+p-1}^{\#1,j} & \boldsymbol{y}_{k+p-1}^{\#2,j} & \cdots & \boldsymbol{y}_{k+p-1}^{\#N,j} \end{bmatrix} = \boldsymbol{O}_k^{(p)}\boldsymbol{X}_k^{(N),\#j}, \quad j = 1 \ldots M. \tag{8.16}$$

$\boldsymbol{O}_k^{(p)} \in \mathbb{R}^{pm\times n}$ is the observability matrix at time k and $\boldsymbol{X}_k^{(N),j}$ is the state variable ensemble matrix of batch j at time k, i.e. the state initial condition responses of experiments of batch j at time k. Whereas the standard TVERA method proceeds using $\tilde{\boldsymbol{H}}_k^{(p,N),j}$ to derive state-space matrices A_k and C_k, the TVERA method with data correlations requires the definition of a square matrix $\tilde{\boldsymbol{\mathcal{R}}}_k^{\#j} \in \mathbb{R}^{pm\times pm}$ such that for $j = 1 \dots M$ we have

$$\tilde{\boldsymbol{\mathcal{R}}}_k^{\#j} = \tilde{\boldsymbol{H}}_k^{(p,N),j}\, \tilde{\boldsymbol{H}}_0^{(p,N),j^\top} \tag{8.17a}$$

$$= \begin{bmatrix} \boldsymbol{y}_k^{\#1,j} & \boldsymbol{y}_k^{\#2,j} & \cdots & \boldsymbol{y}_k^{\#N,j} \\ \boldsymbol{y}_{k+1}^{\#1,j} & \boldsymbol{y}_{k+1}^{\#2,j} & \cdots & \boldsymbol{y}_{k+1}^{\#N,j} \\ \vdots & \vdots & \ddots & \vdots \\ \boldsymbol{y}_{k+p-1}^{\#1,j} & \boldsymbol{y}_{k+p-1}^{\#2,j} & \cdots & \boldsymbol{y}_{k+p-1}^{\#N,j} \end{bmatrix} \begin{bmatrix} \boldsymbol{y}_0^{\#1,j} & \boldsymbol{y}_0^{\#2,j} & \cdots & \boldsymbol{y}_0^{\#N,j} \\ \boldsymbol{y}_1^{\#1,j} & \boldsymbol{y}_1^{\#2,j} & \cdots & \boldsymbol{y}_1^{\#N,j} \\ \vdots & \vdots & \ddots & \vdots \\ \boldsymbol{y}_{p-1}^{\#1,j} & \boldsymbol{y}_{p-1}^{\#2,j} & \cdots & \boldsymbol{y}_{p-1}^{\#N,j} \end{bmatrix}^\top \tag{8.17b}$$

$$= \begin{bmatrix} \sum_{i=1}^{N} \boldsymbol{y}_k^{\#i,j} \boldsymbol{y}_0^{\#i,j^\top} & \sum_{i=1}^{N} \boldsymbol{y}_k^{\#i,j} \boldsymbol{y}_1^{\#i,j^\top} & \cdots & \sum_{i=1}^{N} \boldsymbol{y}_k^{\#i,j} \boldsymbol{y}_{p-1}^{\#i,j^\top} \\ \sum_{i=1}^{N} \boldsymbol{y}_{k+1}^{\#i,j} \boldsymbol{y}_0^{\#i,j^\top} & \sum_{i=1}^{N} \boldsymbol{y}_{k+1}^{\#i,j} \boldsymbol{y}_1^{\#i,j^\top} & \cdots & \sum_{i=1}^{N} \boldsymbol{y}_{k+1}^{\#i,j} \boldsymbol{y}_{p-1}^{\#i,j^\top} \\ \vdots & \vdots & \ddots & \vdots \\ \sum_{i=1}^{N} \boldsymbol{y}_{k+p-1}^{\#i,j} \boldsymbol{y}_0^{\#i,j^\top} & \sum_{i=1}^{N} \boldsymbol{y}_{k+p-1}^{\#i,j} \boldsymbol{y}_1^{\#i,j^\top} & \cdots & \sum_{i=1}^{N} \boldsymbol{y}_{k+p-1}^{\#i,j} \boldsymbol{y}_{p-1}^{\#i,j^\top} \end{bmatrix}. \tag{8.17c}$$

Let a $\xi \times M$ block correlation Hankel matrix be formed as

$$\tilde{\boldsymbol{\mathcal{H}}}_k^{(\xi,M)} = \begin{bmatrix} \tilde{\boldsymbol{\mathcal{R}}}_k^{\#1} & \tilde{\boldsymbol{\mathcal{R}}}_k^{\#2} & \cdots & \tilde{\boldsymbol{\mathcal{R}}}_k^{\#M} \\ \tilde{\boldsymbol{\mathcal{R}}}_{k+\tau}^{\#1} & \tilde{\boldsymbol{\mathcal{R}}}_{k+\tau}^{\#2} & \cdots & \tilde{\boldsymbol{\mathcal{R}}}_{k+\tau}^{\#M} \\ \vdots & \vdots & \ddots & \vdots \\ \tilde{\boldsymbol{\mathcal{R}}}_{k+(\xi-1)\tau}^{\#1} & \tilde{\boldsymbol{\mathcal{R}}}_{k+(\xi-1)\tau}^{\#2} & \cdots & \tilde{\boldsymbol{\mathcal{R}}}_{k+(\xi-1)\tau}^{\#M} \end{bmatrix} = \begin{bmatrix} \boldsymbol{O}_k^{(p)} \\ \boldsymbol{O}_{k+\tau}^{(p)} \\ \vdots \\ \boldsymbol{O}_{k+(\xi-1)\tau}^{(p)} \end{bmatrix} \begin{bmatrix} \tilde{\boldsymbol{X}}_k^{(p),\#1} & \tilde{\boldsymbol{X}}_k^{(p),\#2} & \cdots & \tilde{\boldsymbol{X}}_{k\tau}^{(p),\#M} \end{bmatrix} \tag{8.18}$$

$$= \boldsymbol{O}_k^{(\xi)}\, \tilde{\boldsymbol{X}}_k^{(M)} \tag{8.19}$$

with $\tilde{\boldsymbol{X}}_k^{(p),\#j} = \boldsymbol{X}_k^{(N),\#j} \boldsymbol{X}_0^{(N),\#j^\intercal} \boldsymbol{O}_0^{(p)^\intercal} \in \mathbb{R}^{n\times pm}$ for $j = 1 \dots M$. Now, note that

$$\tilde{\boldsymbol{X}}_{k+1}^{(p),\#j} = \boldsymbol{X}_{k+1}^{(N),\#j} \boldsymbol{X}_0^{(N),\#j^\intercal} \boldsymbol{O}_0^{(p)^\intercal} = A_k \boldsymbol{X}_k^{(N),\#j} \boldsymbol{X}_0^{(N),\#j^\intercal} \boldsymbol{O}_0^{(p)^\intercal} = A_k \tilde{\boldsymbol{X}}_k^{(p),\#j}, \quad j = 1 \dots M. \tag{8.20}$$

The matrix A_k at time k can be derived as

$$\hat{A}_k = \tilde{\boldsymbol{X}}_{k+1}^{(M)} \tilde{\boldsymbol{X}}_k^{(M)\dagger}, \tag{8.21}$$

and the output matrix C_k as

$$\hat{C}_k = \boldsymbol{O}_k^{(\xi)}[1:m,:]. \tag{8.22}$$

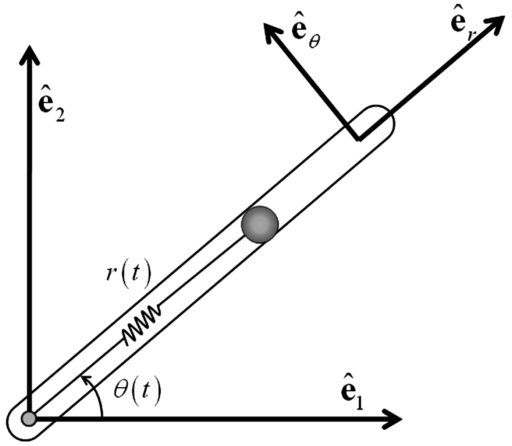


Figure 8.1: Point mass in a rotating tube setup

Next section will present the approximation capabilities of these two algorithms with respect to their original versions where data-correlation is not included.

8.2.3 Dynamics of a Point Mass in a Rotating Tube

8.2.3.1 Model Description

As an illustrative example for the TVERA/DC algorithm, let's consider the dynamics of a point mass in a rotating tube governed by a second order differential equation given by

$$\delta\ddot{r}(t) = \left(\dot{\theta}^2(t) - \frac{k}{m}\right)\delta r(t) + u(t) + l\dot{\theta}^2(t) \tag{8.23}$$

where the new variable $\delta r(t) = r(t) - l$ has been introduced, together with the definition of l, as the free length of the spring (when no force is applied on it, i.e., Hooke's Law applies as $F_s = -k\delta r$). The function $u(t)$ is the radial control force applied on the point mass, and the parameters k and m are the spring stiffness and the mass of the point mass of interest. The time variation in this linear system is brought about by the profile of the angular velocity of the rotating tube $\dot{\theta}(t)$. Choosing the origin of the coordinate system at the position $r_0 = l$ (with no loss of generality), the second order differential equation is given by

$$\delta\ddot{r}(t) = \left(\dot{\theta}^2(t) - \frac{k}{m}\right)\delta r(t) + u(t) \tag{8.24}$$

where the redefinition of the origin renders the system linear time varying without any extra forcing functions. In the first order state space form ($x_1(t) = \delta r(t), x_2(t) = \delta\dot{r}(t)$), the equations can be written as

$$\begin{bmatrix}\dot{x}_1(t)\\ \dot{x}_2(t)\end{bmatrix} = \begin{bmatrix}0 & 1\\ \dot{\theta}^2(t) - \dfrac{k}{m} & 0\end{bmatrix}\begin{bmatrix}x_1(t)\\ x_2(t)\end{bmatrix} + \begin{bmatrix}0\\ 1\end{bmatrix}u(t) = A(t)\boldsymbol{x}(t) + B(t)u(t) \tag{8.25}$$

together with the measurement equations

$$\begin{bmatrix} y_1(t) \\ y_2(t) \end{bmatrix} = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} x_1(t) \\ x_2(t) \end{bmatrix} + \begin{bmatrix} 0 \\ 1 \end{bmatrix} u(t) \tag{8.26}$$

To compare with the identified models, analytical discrete-time models were also generated by computing the state transition matrix (equivalent A_k) and the convolution integrals (equivalent B_k with a zero order hold assumption on the inputs). Because the system matrices are time varying, matrix differential equations are given by

$$\dot{\Phi}(t, t_k) = A(t)\Phi(t, t_k), \qquad \dot{\Psi}(t, t_k) = A(t)\Psi(t, t_k) + I, \tag{8.27}$$

$\forall\ t \in [t_k, t_{k+1}]$, with initial conditions

$$\Phi(t_k, t_k) = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}, \qquad \Psi(t_k, t_k) = \begin{bmatrix} 0 & 0 \\ 0 & 0 \end{bmatrix} \tag{8.28}$$

such that

$$A_k = \Phi(t_{k+1}, t_k), \qquad B_k = \Psi(t_{k+1}, t_k)B, \tag{8.29}$$

would represent the equivalent discrete-time varying system (true model). For the current investigation, the time variation profile of $\dot{\theta}(t) = 3\sin(\frac{1}{2}t)$, with the mass and stiffness of the system were chosen to be $m = 1$ and $k = 10$, respectively.

8.2.3.2 Linear Time-Varying Reduced-Order Model using TVERA/DC

The time interval of interest was held to be 40 seconds, with the discretization sampling frequency set to be 10 Hz. The training is performed on 20 trajectories generated from 20 random initial conditions. A white noise of mean zero and covariance $0.02I_m$ is added to the measurements. Testing of the identified models from both algorithms TVERA and TVERA/DC is performed on 10 random trajectories, not included in the training set. Figure 8.2 shows the comparison in identification performance. The approximation from TVERA degrades as time increases and while it is able to preserve the frequency content, the amplitude of the identified signal is off after just a few seconds. On the other hand, the signal reconstructed from the TVERA/DC procedure is able to match the true trajectory, up to the noise content. Tables 8.1 and 8.2 present the RMS and absolute errors in prediction for models from the two algorithms.

Table 8.1: RMS error for 10 trials

TVERA	**TVERA/DC**
$1.4 \cdot 10^{-1}$	$5.5 \cdot 10^{-2}$

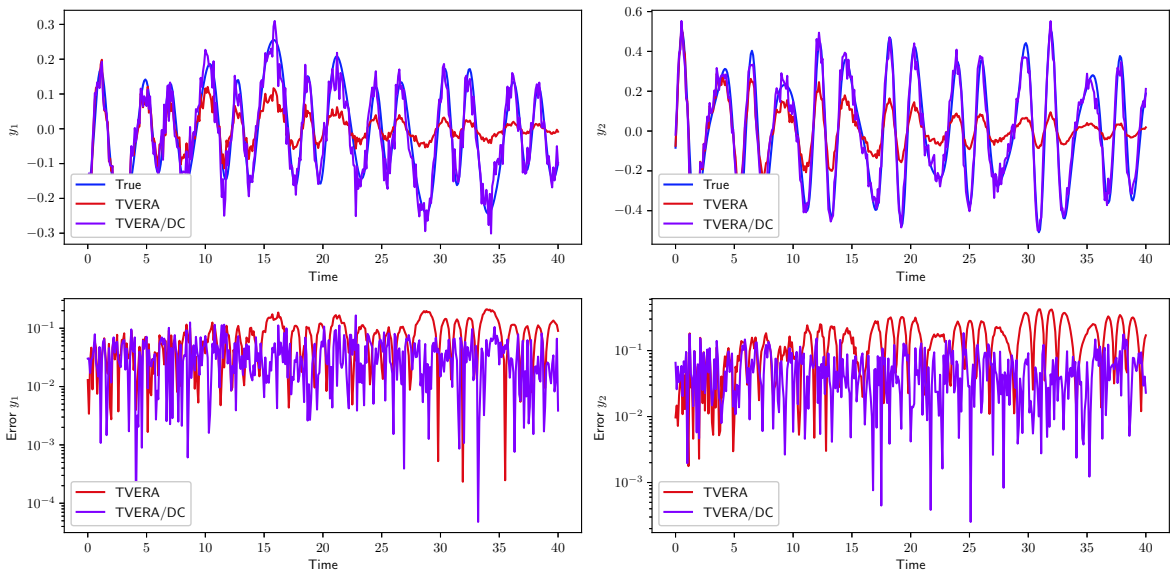


Figure 8.2: Linear time-varying model identification performance on a testing trajectory

Table 8.2: Absolute error for 10 trials

TVERA	**TVERA/DC**
$1.1 \cdot 10^{-1}$	$4.4 \cdot 10^{-2}$

As a second motivational case to support this new method, the identification of the vibrational characteristics of a space structure is considered.

8.2.4 Model of a Flexible Space Structure

8.2.4.1 Coupled Rigid and Flexible Body Model

For the purpose of obtaining output data for the various algorithms used in this paper, we consider the vibrations of the spacecraft depicted in Figure 8.3. This spacecraft is a large, flexible structure whose vibrational modes we wish to study. The structure is modeled as a rigid frame with a flexible membrane clamped within. It is readily apparent that the dynamics of both a rigid and flexible body are well defined; however, this model is simply used as a basis of comparison for the different data-driven algorithms introduced later in this paper. In reality, a similar spacecraft would not have such simple dynamics.

The coupled equations of motion for this model are derived based on the Lagrangian formulation presented in [104], [105] and are omitted from this paper for brevity's sake. Only the resulting equation of motion related to the motion of the membrane is necessary for this analysis. The reference frames and relevant vectors are depicted in Figure 8.3. The black $\{\hat{\boldsymbol{i}}, \hat{\boldsymbol{j}}, \hat{\boldsymbol{k}}\}$ frame is the inertial-reference frame and the red $\{\hat{\boldsymbol{b}}_1, \hat{\boldsymbol{b}}_2, \hat{\boldsymbol{b}}_3\}$ frame is the body-fixed frame. Now that the basis vectors are defined, we move on to deriving the dynamics of the system.

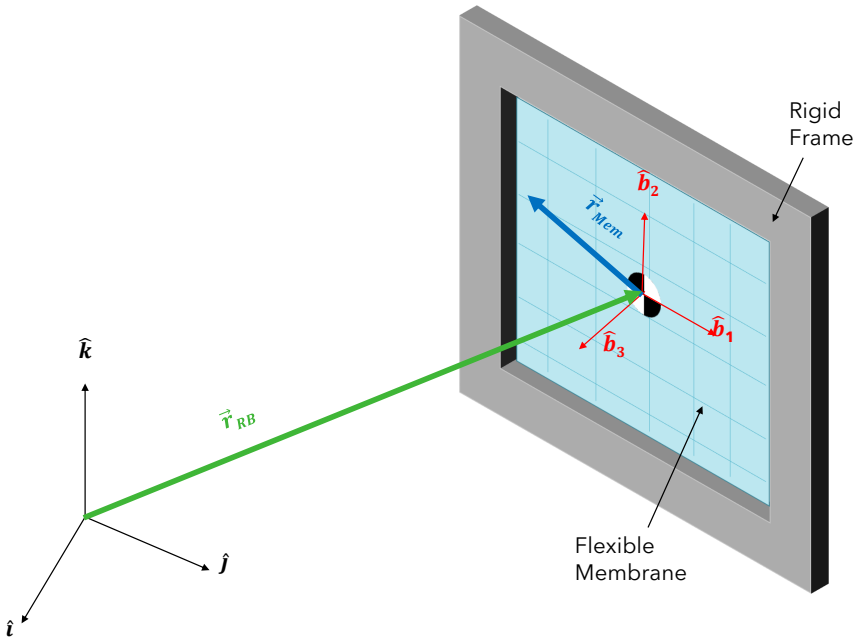


Figure 8.3: Spacecraft model and frame/vector definitions

Dynamics

The position vectors shown in Figure 8.3 are defined as

$$\boldsymbol{r}_{RB} = X\hat{\boldsymbol{b}}_1 + Y\hat{\boldsymbol{b}}_2 + Z\hat{\boldsymbol{b}}_3 \tag{8.30}$$

$$\boldsymbol{r}_{Flex} = \boldsymbol{r}_{RB} + \boldsymbol{r}_{Mem} \tag{8.31}$$

$$\begin{aligned}\boldsymbol{r}_{Flex} &= \boldsymbol{r}_{RB} + x\hat{\boldsymbol{b}}_1 + y\hat{\boldsymbol{b}}_2 + \eta\hat{\boldsymbol{b}}_3 \\ &= (X+x)\hat{\boldsymbol{b}}_1 + (Y+y)\hat{\boldsymbol{b}}_2 + (Z+\eta)\hat{\boldsymbol{b}}_3\end{aligned} \tag{8.32}$$

where $\{x, y, \eta\}$ is the position of any membrane element relative to the rigid body's center of mass written in body-fixed coordinates, and $\{X, Y, Z\}$ are the inertial components of the center of mass of the rigid frame also written in body-fixed coordinates.

The full set of differential equations of motion for the spacecraft are omitted from this paper for brevity's sake. For the purposes of this paper, the only necessary equation of motion is

$$\int_{b_1}^{b_2}\int_{a_1}^{a_2} [\rho(\ddot{\boldsymbol{r}}_{Flex})\cdot\boldsymbol{b}_3 - P\nabla^2\eta]dxdy = \int_{b_1}^{b_2}\int_{a_1}^{a_2} \hat{f}dxdy \tag{8.33}$$

since our goal is to identify the dynamics of the membrane, not the rigid body dynamics. Note that P is the tension in the membrane, ρ is the two-dimensional density of the membrane, and $\hat{f}$ is an arbitrary distributed load. The integral bounds a_1/a_2 and b_1/b_2 are the x and y bounds of the membrane relative to its geometric center, respectively. Note that we are only considering displacement normal to the membrane's surface, corresponding to the variable η. This meaning there are no transverse vibrations

in the $\hat{\boldsymbol{b}}_1$ and $\hat{\boldsymbol{b}}_2$ directions. In the next section, an approach to solving the above partial differential equation is shown.

Method of Weighted Residuals

The partial-differential equation shown in equation (8.33) details the motion of the model's membrane and can be solved numerically by converting it to a system of ordinary differential equations. This is achieved via the Galerkin method of weighted residuals [106]. The displacement of the membrane, $\eta(x, y, t)$, is written as the double sum

$$\eta(x, y, t) = \sum_{i=1}^{n} \sum_{j=1}^{n} \phi_{ij}(x, y) q_{ij}(t) \tag{8.34}$$

where n^2 is the total number of assumed modes, $\phi_{ij}(x, y)$ are basis functions to be chosen by the user, and $q_{ij}(t)$ are the corresponding modal amplitudes of $\phi_{ij}(x, y)$. The basis functions for $\phi_{ij}(x, y)$ are chosen to be the known basis functions of a fully clamped membrane,

$$\phi_{ij}(x, y) = \sin\left[\frac{i\pi}{a}(x - \frac{a}{2})\right] \sin\left[\frac{j\pi}{b}(y - \frac{b}{2})\right] \qquad i, j = 1, 2, ..., n \tag{8.35}$$

where a and b are the membrane's width and height, respectively. For the rest of this paper, the double indices will be dropped. Writing in terms of a single index avoids confusion. In the case of $n = 2$, we refer to the modes [(1,1) (1,2) (2,1) (2,2)] instead as modes [1 2 3 4], respectively. This meaning,

$$\phi_i(x, y) := \phi_{jk}(x, y) \qquad \text{where } i = n(j - 1) + k$$

where now $\phi_i(x, y)$ goes from 1 to n^2. For the rest of this paper, n^2 is denoted as N and represents the total number of assumed modes.

Inserting equation (8.35) into the expanded form of equation (8.33) leads to,

$$\begin{aligned} \int_{b_1}^{b_2} \int_{a_1}^{a_2} \Big[& \rho \dot{\nu}_3 \phi_j + \rho \phi_i \ddot{q}_i \phi_j + \rho \dot{\omega}_1 y \phi_j - \rho \dot{\omega}_2 x \phi_j - \rho \nu_1 \omega_2 \phi_j + \rho(-\omega_2^2 - \omega_1^2)(\phi_i q_i \phi_j) \\ & + \rho \omega_3 \omega_2 y \phi_j + \rho \omega_1 \nu_2 \phi_j + \rho \omega_1 \omega_3 x \phi_j - P \phi_{xx_i} q_i \phi_j - P \phi_{yy_i} q_i \phi_j - \hat{f} \phi_j \Big] dx dy = 0 \\ & \qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad i = 1, 2, ..., N \qquad j = 1, 2, ..., N \end{aligned} \tag{8.36}$$

The derivation of the expression for $\ddot{\boldsymbol{r}}_{Flex}$ is omitted from this paper. In equation (8.36), $\{\nu_1, \nu_2, \nu_3\}$ are the inertial velocities and $\{\omega_1, \omega_2, \omega_3\}$ are the angular velocities. Both velocities are written in the body-fixed coordinate system.

Note that in equation (8.36), repeated indices, such as $\phi_i \ddot{q}_i$, represent the full summation of the function we are approximating with our chosen basis functions, whereas ϕ_j represents the basis functions being used as weighting functions in the Galerkin method of weighted residuals. As a result of this, we now have a system of N ordinary differential equations instead of one partial differential equation. The

system of equations defined in (8.36) can be rearranged into the following form:

$$M\ddot{\boldsymbol{q}} + K\boldsymbol{q} = F \tag{8.37}$$

where $\boldsymbol{q} \in \mathbb{R}^{N\times 1}$ is a vector of the modal amplitudes. M, K, and F are defined as

$$M_{ij} = \int_{b_1}^{b_2}\int_{a_1}^{a_2}\Big[\rho\phi_i\phi_j\Big]dxdy \tag{8.38}$$

$$K_{ij} = \int_{b_1}^{b_2}\int_{a_1}^{a_2}\Big[-\rho(\omega_1^2+\omega_2^2)\phi_j - P(\phi_{xx_i}+\phi_{yy_i})\phi_j\Big]dxdy \tag{8.39}$$

$$F_j = \int_{b_1}^{b_2}\int_{a_1}^{a_2}\Big[\hat{f} + \rho(-\dot{\nu}_3 - \dot{\omega}_1 y + \dot{\omega}_2 x + \nu_1\omega_2 - \omega_3\omega_2 y - \omega_1\nu_2 - \omega_1\omega_3 x)\Big]\phi_j dxdy \tag{8.40}$$

where $M \in \mathbb{R}^{N\times N}$, $K \in \mathbb{R}^{N\times N}$, and $F \in \mathbb{R}^{N\times 1}$. Here, i and j represent the (i^{th}, j^{th}) element of each matrix and both go from 1 to N. Note that there is no repeated index here so there is no summation in these definitions. In the next section, we will utilize the mass-spring form of equation (8.37) to obtain a state-space model for our system.

State-Space Model

By assuming free vibrations ($F_j = 0$), equation (8.37) can be written as a first-order, continuous, time-varying, linear system of the form:

$$\dot{\boldsymbol{x}} = A_c(t)\boldsymbol{x} \tag{8.41}$$

$$\boldsymbol{y} = C\boldsymbol{x} \tag{8.42}$$

where $\boldsymbol{x} \in \mathbb{R}^{2N\times 1}$ is the state vector of time-varying modal amplitudes and their time derivatives and $\boldsymbol{y} \in \mathbb{R}^{m\times 1}$ is the acceleration output vector. The system matrices are written as,

$$A_c = \begin{bmatrix} O^{N\times N} & I^{N\times N} \\ -M^{-1}K(t) & O^{N\times N} \end{bmatrix} \tag{8.43}$$

$$C = \begin{bmatrix} -C_1 M^{-1}K(t) & 0^{mxN} \end{bmatrix} \quad , \quad C_1 = \begin{bmatrix} \phi_1(x_1,y_1) & \dots & \phi_N(x_1,y_1) \\ \vdots & \ddots & \vdots \\ \phi_1(x_m,y_m) & \dots & \phi_N(x_m,y_m) \end{bmatrix} \tag{8.44}$$

where I and O are the identity and zero matrices, respectively. M and K are the mass and stiffness matrices defined in equations (8.38) and (8.39), respectively. Note that the only time-varying aspect of the system comes from the $\omega_1^2 + \omega_2^2$ term in K (assuming membrane parameters are constant). The output for this system is the acceleration of the membrane at m locations; representing m sensors distributed across the membrane. This is visualized in Figure 8.4 with a top-down view of the membrane. The black dashed line represents the border of the membrane, the black circle the geometric center, and the red dots the accelerometers.

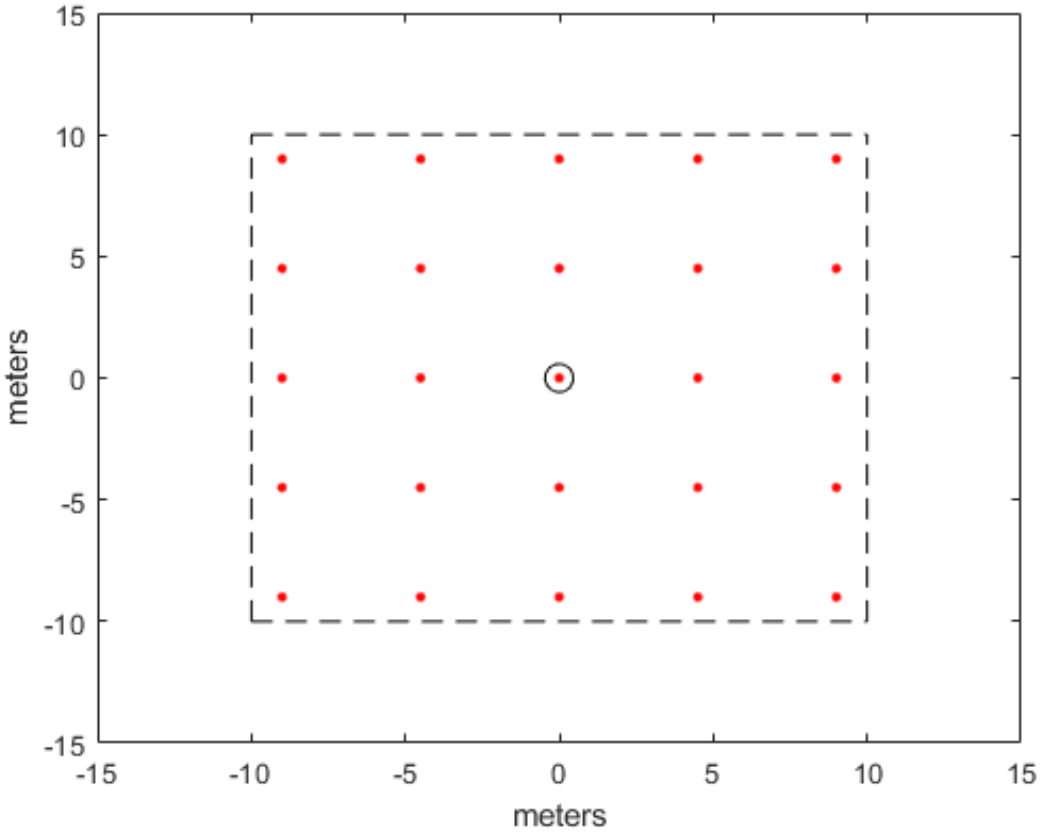


Figure 8.4: Example of sensor placement

For the results in this paper, the first four functions of $\phi_{ij}(x, y)$, defined in equation (8.35), were chosen as the basis functions (i.e. $i = j = 2$). This implies $A_c \in \mathbb{R}^{8\times 8}$. We can then discretize equations Eq. (8.41) and Eq. (8.42) as

$$\boldsymbol{x}_{k+1} = A_k \boldsymbol{x}_k \tag{8.45}$$

$$\boldsymbol{y}_k = C_k \boldsymbol{x}_k \tag{8.46}$$

where A_k is the state-transition matrix between time steps, $\Phi(k+1, k)$, and $C_k = C$. We now have our reference discrete state-space model we wish to identify via data-driven modeling. Now, we must utilize this model to obtain output data.

Initial Conditions

In order to correctly identify all the dynamics of our derived model, we need to ensure that all modes are excited in our output data. To do this, we choose the initial conditions of our state to be linear combinations of all four assumed flexible body modes, which are the columns of the eigenvector matrix $\Phi = \left[\boldsymbol{\phi}_1, \boldsymbol{\phi}_2, \ldots, \boldsymbol{\phi}_i, \ldots, \boldsymbol{\phi}_{2N}\right]$ obtained from the characteristic equation $|A - \lambda_i| \, \phi_i$. The initial

conditions are chosen to be

$$\boldsymbol{x}_{0_i} = r_1(\boldsymbol{\phi}_1 + \boldsymbol{\phi}_2) + r_2(\boldsymbol{\phi}_3 + \boldsymbol{\phi}_4) + r_3(\boldsymbol{\phi}_5 + \boldsymbol{\phi}_6) + r_4(\boldsymbol{\phi}_7 + \boldsymbol{\phi}_8) \quad i = 1, 2, \ldots, K \tag{8.47}$$

where r_1 through r_4 are random scalars drawn from a normal distribution and K is chosen by the user to ensure enough output data is obtained to capture the order of the system. In the absence of a real input, the initial conditions act as a sort of input instead.

8.2.4.2 Data Acquisition and Setup Description

To obtain the output data for the system, the continuous state-space model was numerically integrated with MATLAB's ode45 function with the initial conditions defined in the previous section. The following table details the parameter values used in simulation. Note that the two different values of ω_1/ω_2 correspond to two separate cases of a time-invariant and time-varying system. All other parameters remain the same across the two.

N	a	b	ρ	P	ω_1/ω_2	ω_1/ω_2
4	20 m	20 m	5 $\frac{kg}{m^2}$	200 N	0.5 $\frac{rad}{s}$	0.1sin(t) + 0.3 $\frac{rad}{s}$

As for the sensor error, the noise was chosen to have a normal distribution with a standard deviation equal to 5% of the max acceleration the true-system experienced (different for each initial condition case). The different levels of noise will be used for comparison of the different algorithms.

8.2.4.3 Linear Time-Invariant Reduced-Order Model using ERA/DC

First, linear models are derived from data using the algorithms outlined in the previous section. A hundred random trajectories are considered for the identification, with testing performed on an additional set of 20 trajectories, different from training. Figure 8.5 shows the performance of the two algorithms on those testing trajectories (average on the 20 trajectories of the testing set). The noise level is indicated as a reference and serves as a lower bound for the identification error. By carefully selecting identification parameters, the eigensystem realization algorithm with data-correlation (ERA/DC) performs up to one order of magnitude better than its ERA counterpart. While non-linearities and noise (rogue sensor measurements here) degrade the approximation capabilities of the ERA, the ERA/DC is less sensitive to noise in the data thanks to correlation calculations. While the relative error induced by the identified model from ERA grows over time and is impacted by measurement noise, the identified model from ERA/DC yields significantly less approximation error and sometimes correctly approximates the response down to - a;most - the non-compressible noise level. The average relative error for the 20 test trials for ERA is at $1.1 \cdot 10^0$ while the error from ERA/DC is $7.4 \cdot 10^{-2}$, using the same data set. Models for different of parameter τ have been generated and average errors for the 20 test trajectories are presented in Table 8.3. This confirms that the bias terms affecting the ERA when measurement noise is present can, in principle, be omitted in the ERA/DC by properly choosing the integer τ.

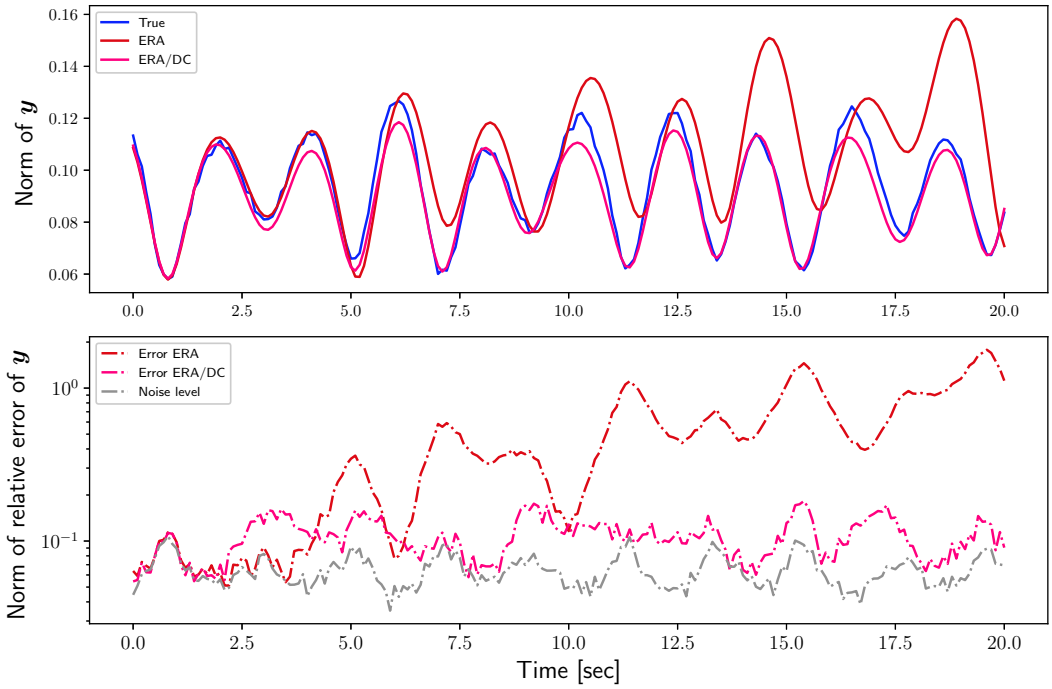


Figure 8.5: Average linear time-invariant models identification performances on 20 testing trajectories

Table 8.3: Average relative error for the 20 test trials

Value of parameter τ	**Error in approximation from ERA/DC**
$\tau = 1$	$1.3 \cdot 10^{0}$
$\tau = 2$	$9.8 \cdot 10^{-1}$
$\tau = 3$	$8.1 \cdot 10^{-1}$
$\tau = 4$	$5.8 \cdot 10^{-1}$
$\tau = 5$	$2.5 \cdot 10^{-1}$
$\tau = 6$	$9.3 \cdot 10^{-2}$
$\tau = 7$	$7.4 \cdot 10^{-2}$

As mentioned earlier, the main difficulty in linear system identification applications stems from the interplay of noise and unmodeled dynamics and most systems are only linear to a first approximation. Since this immediately limits the application of the results obtained by linear system identification algorithms, the time-varying Eigensystem Realization Algorithm with data-correlation is applied to the same data set. Figure 8.6 shows the performance of the TVERA/DC algorithm to reproduce the system response in presence of noise, where the relative identification error is up to the noise level.

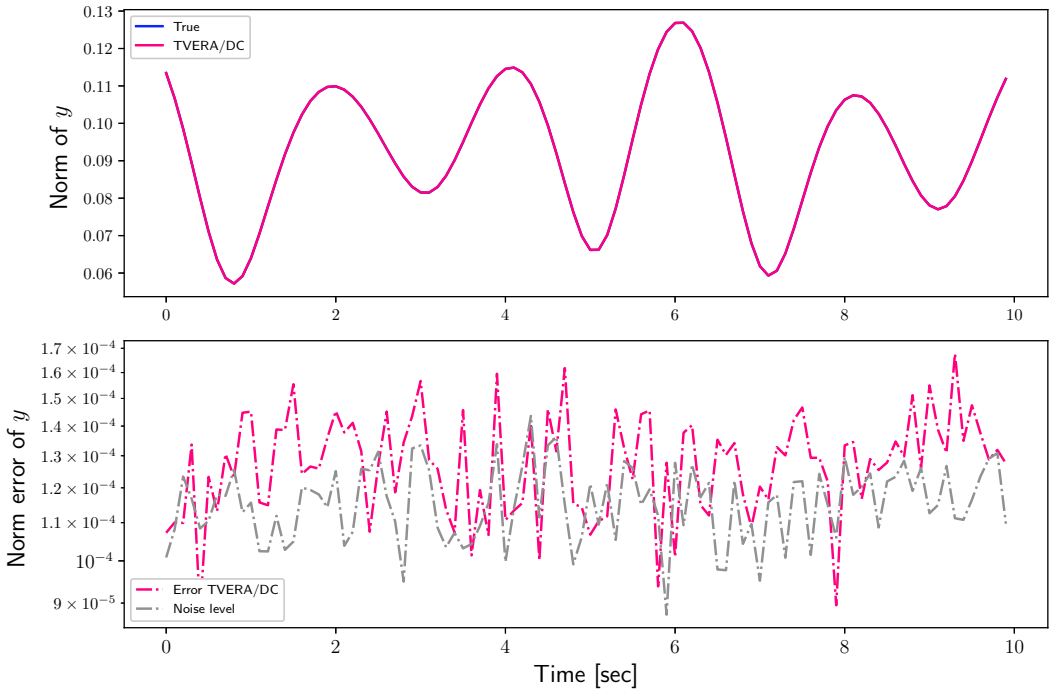


Figure 8.6: Average linear time-varying models identification performances on 20 testing trajectories

8.3 Nuclear Norm Optimization

8.3.1 Introduction

The matrix rank minimization problem, or minimizing the rank of a matrix subject to convex constraints, has recently attracted much interest. The rank minimization problem arises in a diverse set of fields, where notions of order, complexity, or dimensionality are expressed by means of the rank of an appropriate matrix. The nuclear norm (sum of singular values, also known as the Schatten 1-norm or trace norm) and its variants have lately received a lot of consideration in convex heuristics in control, signal processing, and statistics. Applications include system identification, low-order controller design, collaborative filtering in machine learning, and Euclidean embedding problems [107]. Since rank minimization is NP-hard in general, the nuclear norm of a matrix is often used as a relaxation for rank minimization problems the same way ℓ_1-norm minimization techniques are used for cardinality minimization and sparse signal estimation. In this paper, the problem of identifying a reduced-order linear time-invariant (LTI) dynamical system from observations of its inputs and outputs (I/O data) only is considered. Many algorithms have been established for identification purposes, some of them deterministic in nature, i.e. without considering noise in the measured data, and others stochastic, i.e. with formulations minimizing the noise uncertainty in the identification. During the 90s, building upon initial work by Gilbert and Kalman, several methods have been developed to identify most observable and controllable subspace of the system from given input-output (I/O) data. Under the interaction of structure and control disciplines, the Eigensystem Realization Algorithm (ERA) was developed for modal parameter identification and model reduction of dynamic systems using test data. The algorithm presents a unified framework for

modal parameter identification based on the Markov parameters (i.e., pulse response) making it possible to construct a Hankel matrix as the basis for the realization of a discrete-time state-space model. The key step in this subspace identification method is a singular value decomposition (SVD) of the Hankel matrix used to estimate the order of the system. Due to measurement noise, nonlinearity, and computer round-off, the Hankel matrix will usually be of full rank which does not, in general, equal the true order of the system under test. The reliance on the SVD for the low-rank approximation makes it difficult to seamlessly apply the subspace methods to problems with important levels of noise. Minimizing the nuclear norm of the Hankel matrix provides an interesting alternative as a heuristic for low-rank approximation problems where a clear-cut criterion for model selection cannot be deduced from the SVD.

From the classical SVD decomposition, some singular values of $\boldsymbol{\Sigma}$ may be relatively small and negligible in the sense that they contain more noise information than system information. Hence, the approximation $\boldsymbol{U}^{(0)}\boldsymbol{\Sigma}^{(0)}\boldsymbol{V}^{(0)\mathsf{T}} \simeq \mathbf{0}$ (truncation of nonzero small singular values) is to account for noise in the data and for quantitatively partitioning the realized model into principal and perturbation (noise) portions so that the noise portion can be disregarded. In other words, the directions determined by these singular values have less significant degrees of controllability and observability relative to noise. The reduced model of order n after deleting the singular values $\sigma_{n+1}, \ldots$ is then considered as the robustly controllable and observable part of the realized system. As an analyst, it should not be the aim to obtain a system realization which exactly reproduces the noisy sequence of data: a realization which reproduced a smoothed version of the sequence, and which closely represents the underlying linear dynamics of the system, is more desirable. Some work has been conducted to provide a mathematical framework for establishing the relationship between the singular values and the characteristics of the noise [14]. Besides the singular values, several other accuracy indicators have been developed for quantitatively partitioning the realized model into principal and perturbational (noise) portions so that the noise portion is disregarded.

8.3.2 Candidate Methods for Distinguishing True Modes from Noise Modes

In [1], two approaches to distinguish true modes from noise modes are considered. These include the Modal Amplitude Coherence (MAC) and the Mode Singular Value (MSV).
Consider an identified discrete-time model in modal coordinates

$$\boldsymbol{x}_{mk+1} = \hat{A}_m \boldsymbol{x}_{mk} + \hat{B}_m \boldsymbol{u}_k \tag{8.48a}$$

$$\boldsymbol{y}_k = \hat{C}_m \boldsymbol{x}_{mk} + D\boldsymbol{u}_k \tag{8.48b}$$

where $\hat{A}_m$ is a diagonal matrix containing the identified eigenvalues $\hat{\lambda}_i, i = 1, 2, \ldots, n$ of the system, and $\hat{B}_m$ and $\hat{C}_m$ are the input and output matrices in modal coordinates, respectively. Let these two

matrices be partitioned as

$$\hat{B}_m = \begin{bmatrix} \hat{b}_1 \\ \hat{b}_2 \\ \vdots \\ \hat{b}_n \end{bmatrix}, \hat{C}_m = \begin{bmatrix} \hat{c}_1 & \hat{c}_2 & \cdots & \hat{c}_n \end{bmatrix}. \tag{8.49}$$

Each Markov parameter can be written as a combination of n components contributed from different modal coordinates:

$$h_k = \hat{C}_m \hat{A}_m^k \hat{B}_m = \sum_{i=1}^{n} \hat{c}_i \hat{\lambda}_i^k \hat{b}_i. \tag{8.50}$$

Now, define the sequence

$$\hat{r}_i = \begin{bmatrix} \hat{b}_i & \hat{\lambda}_i \hat{b}_i & \hat{\lambda}_i^2 \hat{b}_i & \cdots & \hat{\lambda}_i^{l-2} \hat{b}_i \end{bmatrix} \tag{8.51}$$

where l is the length of the data. This sequence represents the time history reconstructed from the identified eigenvalue, $\hat{\lambda}_i$, and the row vector $\hat{b}_i$ and is called the identified modal amplitude time history for the i-th mode, because it represents the temporal contribution of the i-th mode associated with the output vector $\hat{c}_i$ to the Markov parameters sequence. The output vector $\hat{c}_i$ in modal coordinates coordinates is nothing more than the i-th mode shape information at the sensor points.
Since Markov parameters are independent through similarity transformation, the Hankel matrix $\boldsymbol{H}_0^{(p,q)}$ is also

$$\boldsymbol{H}_0^{(p,q)} = \begin{bmatrix} C_m B_m & C_m A_m B_m & \cdots & C_m A_m^{q-1} B_m \\ C_m A_m B_m & C_m A_m^2 B_m & \cdots & C_m A_m^q B_m \\ \vdots & \vdots & \ddots & \vdots \\ C_m A_m^{p-1} B_m & C_m A_m^p B_m & \cdots & C_m A_m^{p+q-2} B_m \end{bmatrix} \tag{8.52}$$

$$= \tilde{\boldsymbol{O}}^{(p)} \tilde{\boldsymbol{R}}^{(q)}, \tag{8.53}$$

with

$$\tilde{\boldsymbol{O}}^{(p)} = \begin{bmatrix} C_m \\ C_m A_m \\ \vdots \\ C_m A_m^{p-1} \end{bmatrix}, \tag{8.54}$$

$$\tilde{\boldsymbol{R}}^{(q)} = \begin{bmatrix} B_m & A_m B_m & \cdots & A_m^{q-1} B_m \end{bmatrix}. \tag{8.55}$$

A_m, B_m and C_m represent the real system matrices in modal coordinates. $\tilde{\boldsymbol{R}}^{(q)}$ can also be partitioned as

$$\tilde{\boldsymbol{R}}^{(q)} = \begin{bmatrix} \tilde{r}_1 \\ \tilde{r}_2 \\ \vdots \\ \tilde{r}_n \end{bmatrix} = \begin{bmatrix} \tilde{b}_1 & \tilde{\lambda}_1\tilde{b}_1 & \cdots & \tilde{\lambda}_1^{q-1}\tilde{b}_1 \\ \tilde{b}_2 & \tilde{\lambda}_2\tilde{b}_2 & \cdots & \tilde{\lambda}_2^{q-1}\tilde{b}_2 \\ \vdots & \vdots & \cdots & \vdots \\ \tilde{b}_n & \tilde{\lambda}_n\tilde{b}_n & \cdots & \tilde{\lambda}_n^{q-1}\tilde{b}_n \end{bmatrix} \tag{8.56}$$

A dot product of $\hat{r}_i$ and $\tilde{r}_i$ results in the Modal Amplitude Coherence while analyzing the contribution of each identified mode results in the Mode Singular Value.

8.3.2.1 Modal Amplitude Coherence (MAC)

The MAC developed can be thought of as a dot product between

1. $\tilde{r}_i$: the vector of the unit pulse response history asociated with a mode of the identified model, and
2. $\hat{r}_i$: the vector of the unit pulse response history used in the identification:

$$\text{MAC}_i = \frac{|\tilde{r}_i\hat{r}_i^*|}{\sqrt{|\tilde{r}_i\tilde{r}_i^*||\hat{r}_i\hat{r}_i^*|}}, \tag{8.57}$$

where $i = 1, 2, \ldots, n$ and * means transpose and complex conjugate. If the two vectors coincide, then the model reproduces the pulse response data.

8.3.2.2 Mode Singular Value (MSV)

The MSV is a method of characterizing the contribution of each identified mode to the identified model pulse response history. The ERA attempts to identify a model to match the pulse response history: it is thus reasonable that a mode that has a large contribution to the identified model's pulse response history has a large contribution to the system's pulse response data and is then well identified by the algorithm. Each modal coordinate contributes to the pulse response by the individual modal sequence, which can be quantified by taking its maximum singular value:

$$\text{MSV}_i = \sqrt{|\hat{c}_i|(1 + |\hat{\lambda}_i| + |\hat{\lambda}_i^2| + \ldots + |\hat{\lambda}_i^{l-2}|)|\hat{b}_i|}. \tag{8.58}$$

8.3.2.3 Conclusion

MAC and MSV are the two methods traditionally used but usually tend to be very noisy and unreliable, especially for high frequency mode. Nuclear norm minimization offers an alternative method for computing low rank matrix approximations. Compared to other candidate methods for distinguishing true modes from noise modes, the minimum nuclear norm approximation typically provides a more clear-cut criterion for model selection than the SVD low-rank approximation.

8.3.3 Matrix Rank Minimization

The rank minimization problem arises in a diverse set of fields, where notions of order, complexity, or dimension are expressed by means of the rank of an appropriate matrix. The nuclear norm (sum of singular values) of a matrix is often used as a relaxation for rank minimization problems the same way ℓ_1-norm minimization techniques is used for cardinality minimization and sparse signal estimation. The nuclear norm (also known as the Schatten 1-norm or trace norm) and its variants have lately received a lot of interest in convex heuristics in control, signal processing, and statistics. Applications include system identification, low-order controller design, collaborative filtering in machine learning, and Euclidean embedding problems [107].

The matrix rank minimization problem consists of finding a matrix of minimum rank that satisfies a given set of convex constraints

$$\min_{X} \operatorname{rank}(X) \tag{8.59a}$$

$$\text{s.t. } X \in C \tag{8.59b}$$

where X is the optimization variable and C is a convex set of constraints. In this Chapter, a linear system identification method utilizing nuclear norm minimization is presented.

8.3.4 Application to System Identification

Recall from the singular value decomposition from the Eigensystem Realization Algorithm:

$$\boldsymbol{H}_0^{(p,q)} = \boldsymbol{U}\boldsymbol{\Sigma}\boldsymbol{V}^\intercal = \begin{bmatrix}\boldsymbol{U}^{(n)} & \boldsymbol{U}^{(0)}\end{bmatrix} \begin{bmatrix}\boldsymbol{\Sigma}^{(n)} & \mathbf{0} \\ \mathbf{0} & \boldsymbol{\Sigma}^{(0)}\end{bmatrix} \begin{bmatrix}\boldsymbol{V}^{(n)\intercal} \\ \boldsymbol{V}^{(0)\intercal}\end{bmatrix} \tag{8.60a}$$

$$= \boldsymbol{U}^{(n)}\boldsymbol{\Sigma}^{(n)}\boldsymbol{V}^{(n)\intercal} + \underbrace{\boldsymbol{U}^{(0)}\boldsymbol{\Sigma}^{(0)}\boldsymbol{V}^{(0)\intercal}}_{\simeq \mathbf{0}} \tag{8.60b}$$

$$\simeq \boldsymbol{U}^{(n)}\boldsymbol{\Sigma}^{(n)}\boldsymbol{V}^{(n)\intercal}. \tag{8.60c}$$

Some singular values of $\boldsymbol{\Sigma}$ may be relatively small and negligible in the sense that they contain more noise information than system information. Hence, the approximation $\boldsymbol{U}^{(0)}\boldsymbol{\Sigma}^{(0)}\boldsymbol{V}^{(0)\intercal} \simeq \mathbf{0}$ (truncation of nonzero small singular values) is to account for noise in the data and for quantitatively partitioning the realized model into principal and perturbation (noise) portions so that the noise portion can be disregarded. In other words, the directions determined by these singular values have less significant degrees of controllability and observability relative to noise. The reduced model of order n after deleting the singular values $\sigma_{n+1}, \ldots$ is then considered as the robustly controllable and observable part of the realized system. Some work has been conducted to provide a mathematical framework for establishing the relationship between the singular values and the characteristics of the noise [14]. Nuclear norm minimization offers an alternative method for computing low rank matrix approximations. Compared to other candidate methods for distinguishing true modes from noise modes, the minimum nuclear norm approximation typically provides a more clear-cut criterion for model selection than the SVD low-rank approximation.

The formulation for minimizing the nuclear norm is

$$\min_{X} \ ||X||_* \tag{8.61a}$$

$$\text{s.t.} \ \left|\left|\boldsymbol{H}_0^{(p,q)} - X\right|\right|_F \leq \varepsilon |SNR| \tag{8.61b}$$

where SNR is the signal-to-nose ratio defined as

$$SNR = \frac{E\left[\left(\boldsymbol{y} - E\left[\boldsymbol{y}\right]\right)\left(\boldsymbol{y} - E\left[\boldsymbol{y}\right]\right)^{\mathsf{T}}\right]}{V}. \tag{8.62}$$

The measurement noise is a Gaussian white noise with zero mean and variance V. The principal SVD for system order identification is performed on X. It is expected that the main benefit of the nuclear norm approach is its ability to handle data sets with a high percentage of noise. Thus, numerical simulations with relatively high signal-to-noise ratios will be conducted.

8.3.5 Numerical Examples

This section aims to demonstrate the validity of the proposed approach on two different numerical simulations. The first example is an automobile system with a state of order $n = 4$. The second example, more sensitive to noise, is a classical free three mass-spring-damper system or order $n = 6$.

8.3.5.1 Simulated Automobile System

Figure 8.7 shows an automobile simulated by a simplified two-degree-of-freedom system including a rigid bar supported by two springs with spring constants k_1 and k_2, and two dashpots with damping coefficients c_1 and c_2. The displacement $x(t)$ and the rotation angle $\theta(t)$ define the motions of the system. Note that $x(t)$ is the displacement of the center of mass. The force $F(t)$ and the torque $M(t)$ provide the excitation sources of the system to the system for the purpose of the structural tests.

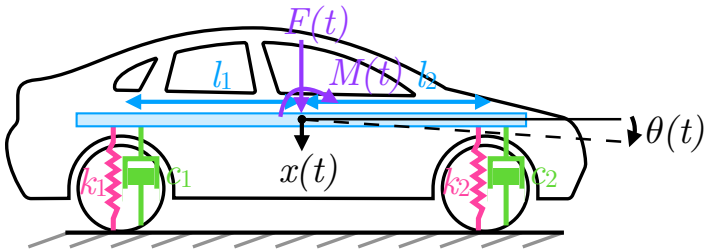


Figure 8.7: A simulated automobile system

The second-order differential equations for this system are

$$M\ddot{\boldsymbol{x}} + \zeta\dot{\boldsymbol{x}} + K\boldsymbol{x} = B_2\boldsymbol{u} \tag{8.63a}$$

$$\boldsymbol{y} = C\boldsymbol{x} \tag{8.63b}$$

where

$$M = \begin{bmatrix} m & 0 \\ 0 & \mathcal{I} \end{bmatrix}, K = \begin{bmatrix} k_1 + k_2 & -(k_1 l_1 - k_2 l_2) \\ -(k_1 l_1 - k_2 l_2) & k1 l_1^2 + k_2 l_2^2 \end{bmatrix}, \zeta = \begin{bmatrix} c_1 + c_2 & -(c_1 l_1 - c_2 l_2) \\ -(c_1 l_1 - c_2 l_2) & c1 l_1^2 + c_2 l_2^2 \end{bmatrix}, \quad (8.64)$$

with m and $\mathcal{I}$ the mass and the moment of inertia of the vehicle and the partial constant-time input influence matrix is $B_2 = I_2$. For the discrete-time system, the constant-time output influence matrix C is considered to be $C = I_4$ (all states are measured) and the direct transmission matrix is zero.

In order to study the utility of nuclear norm optimization on the Hankel matrix, five different cases are considered: one free-noise case where perfect input-output data are collected for the OKID/TVERA procedure and four other cases with increasing amount of noise in the output data. In all cases, data is recorded at a frequency of 5 Hz for 50 sec. The size of the Hankel matrix $\boldsymbol{H}_0^{(p,q)}$ is $pm \times qr$ with $m = 4, r = 2$ and $p = q = 29$.
For the first case, as clearly indicated by the singular value decomposition plot Figure 8.8, it is straightforward to see that the dimension of the system is $n = 4$. The clear cut after the four first singular values is serving as a sign to identify the order of the system. Figure 8.9 displays the SVD for different level of noise. For each plot, two SVD are shown: the original SVD with no optimization and the SVD resulting from the Hankel matrix after nuclear norm optimization (labeled "Optimized SVD").

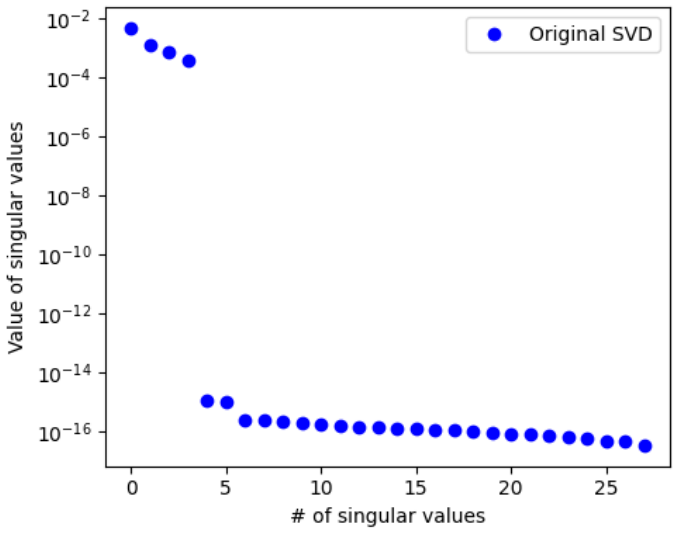
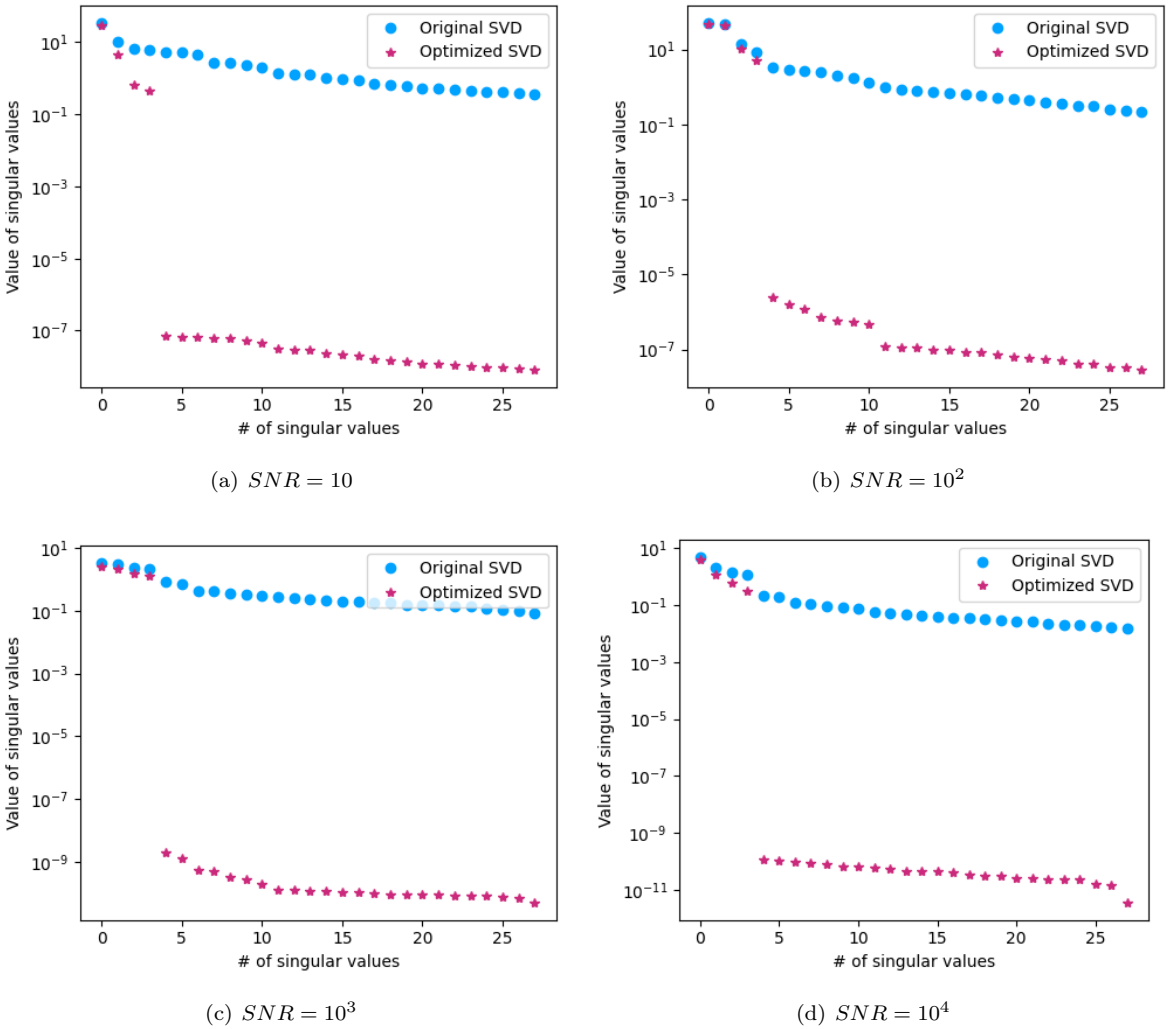


Figure 8.8: Simulated automobile system: SVD with no noise

8.3.5.2 Free Three Mass-Spring-Dashpot System

Figure 8.10 shows a free three mass-spring-dashpot system where m_1, c_i and k_i $(i = 1, 2, 3)$ are masses, damping coefficients and spring constants, $x_i(t)$ are absolute displacements for respective masses and $F_i(t)$ are input forces.



(a) $SNR = 10$

(b) $SNR = 10^2$

(c) $SNR = 10^3$

(d) $SNR = 10^4$

Figure 8.9: Simulated automobile system: SVD for different levels of noise

The second-order differential equations for this system are

$$M\ddot{\boldsymbol{x}} + \zeta\dot{\boldsymbol{x}} + K\boldsymbol{x} = B_2\boldsymbol{u} \tag{8.65a}$$

$$\boldsymbol{y} = C\boldsymbol{x} \tag{8.65b}$$

where

$$M = \begin{bmatrix} m_1 & 0 & 0 \\ 0 & m_2 & 0 \\ 0 & 0 & m_3 \end{bmatrix}, K = \begin{bmatrix} k_1 + k_3 & -k_1 & -k_3 \\ -k_1 & k_1 + k_2 & -k_2 \\ -k_3 & -k_2 & k_2 + k_3 \end{bmatrix}, \zeta = \begin{bmatrix} c_1 + c_3 & -c_1 & -c_3 \\ -c_1 & c_1 + c_2 & -c_2 \\ -c_3 & -c_2 & c_2 + c_3 \end{bmatrix}, \tag{8.66}$$

The partial constant-time input influence matrix is $B_2 = I_3$ and, for the discrete-time system, the constant-time output influence matrix C is considered to be $C = I_6$ (again, all states are measured) and

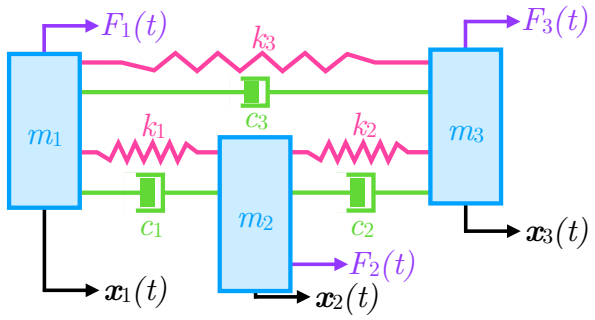


Figure 8.10: A free mass-spring-dashpot system

the direct transmission matrix is zero. Similarly to the Simulated Automobile System, five different cases are considered, comprised of one free-noise case and four other cases with increasing amount of noise in the output data. In all cases, data is recorded at a frequency of 5 Hz for 50 sec. The size of the Hankel matrix $\boldsymbol{H}_0^{(p,q)}$ is $pm \times qr$ with $m = 6, r = 3$ and $p = q = 17$.

Shown by Figure 8.11, the dimension of the system is $n = 6$. Figure 8.12 displays the SVD for different level of noise. For each plot, two SVD are shown: the original SVD with no optimization and the SVD resulting from the Hankel matrix after nuclear norm optimization (labeled "Optimized SVD").

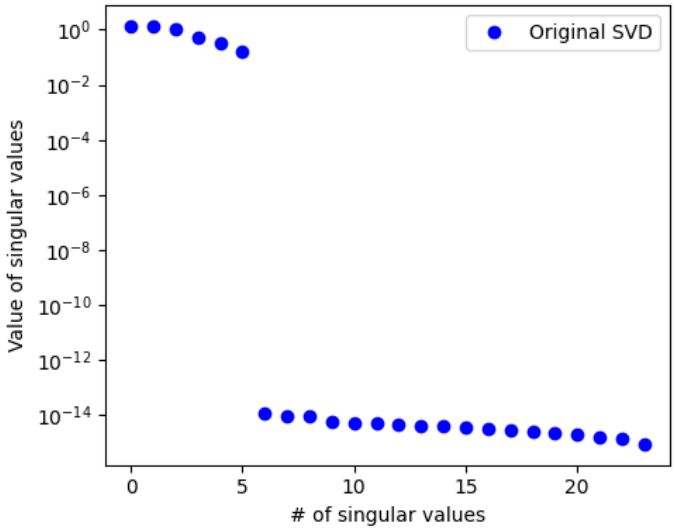
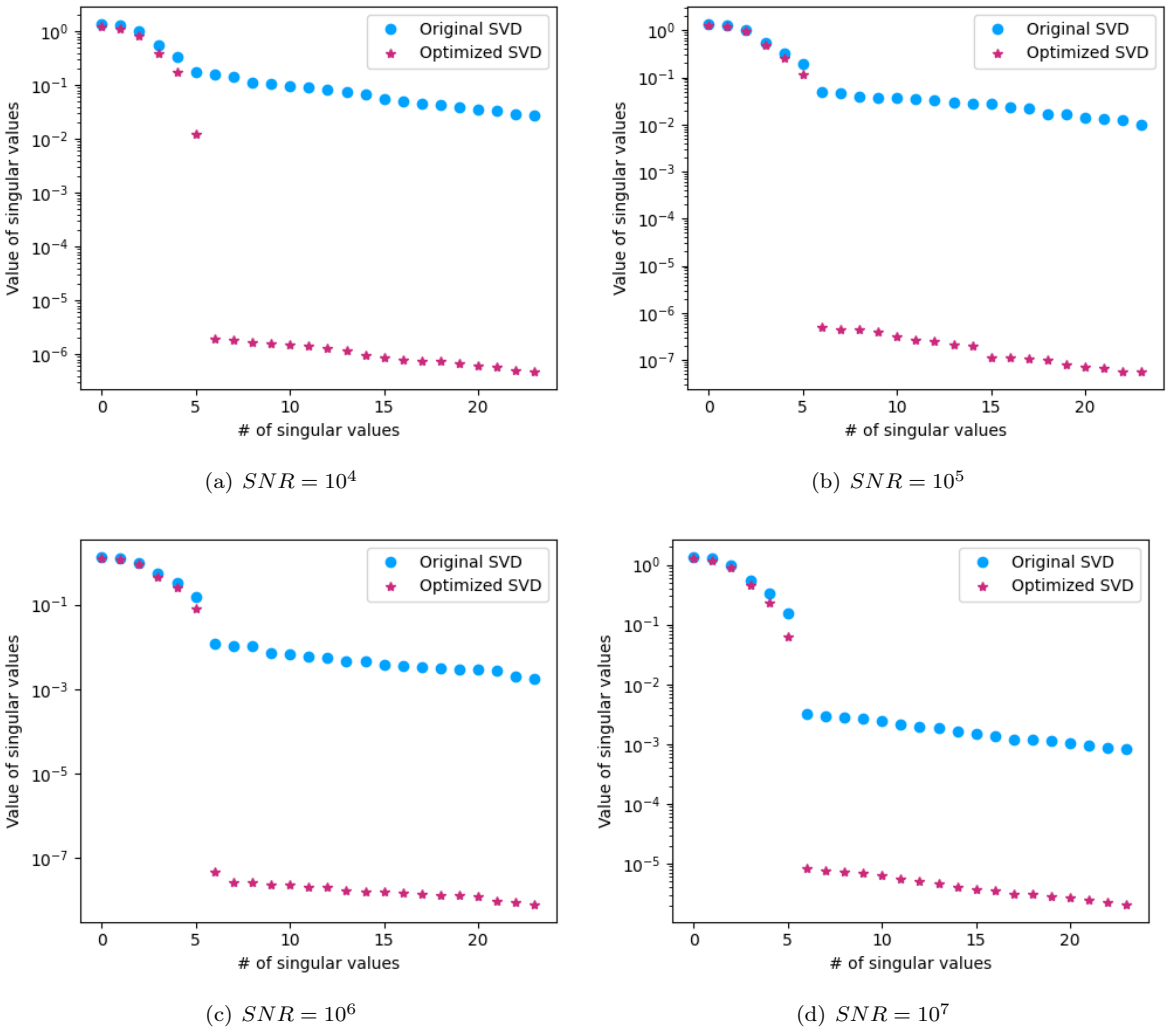


Figure 8.11: Three mass-spring-dashpot system: SVD with no noise



(a) $SNR = 10^4$

(b) $SNR = 10^5$

(c) $SNR = 10^6$

(d) $SNR = 10^7$

Figure 8.12: Three mass-spring-dashpot system: SVD for different levels of noise

Chapter 9
Conclusion

This dissertation develops a novel, unified and robust data-driven framework to address unsolved problems in the area of nonlinear dynamical system identification and reduced-order modeling. The main contribution of this dissertation is a common foundation to develop algorithms that combines the latest techniques in time-varying subspace realization methods, sparse representation and embeddings. More precisely, it offers a structure for data-driven linear time-invariant and time-varying system identification, sparse system identification, data-driven uncertainty quantification, and bilinear system identification as well as offering tools to mitigate noise in the data for engineering applications. This framework uses state-of-the-art linear and nonlinear system identification techniques for state prediction but also for reduced-order modeling of high-dimensional systems. This framework has further unique applications in uncertainty quantification and sensitivity analysis.

9.1 Contributions

The first contribution of this dissertation is to structure the mathematical foundations around the system identification problem. While the theory for the identification of linear systems is mature and well understood, nonlinear system identification does not benefit from a common framework. Chapter 2 of outlined a universal frame that connects classical formulations with higher-order sensitivity tensors, Carleman linearization and the Koopman operator theoretic framework. In essence, the majority of nonlinear identification methods and algorithms rely on one of these.

Based on this common framework, the second contribution of this dissertation is the development of a convex optimization-based approach for nonlinear system identification from state and control input time histories. The proposed methodology in Chapter 3 expands the unknown nonlinearities in system dynamics in terms of basis functions consisting of monomials of various orders, building an analogous version of a continuous-time Koopman operator. The algorithm exploits recent advances in sparse approximation to automatically select the appropriate structure for the inherent nonlinearities. The excellent agreement in correctly identifying the true dynamics for different problems of various complexity provides a strong basis for optimism in demonstrating the utility of the approach for identifying the inherent physics-based map from given data.

The third contribution of this dissertation is to extend and merge time-varying subspace system identification methods with the identification of a time-varying Koopman operator of arbitrary order. By linearization of the nonlinear flow about a nominal trajectory rather than a nominal point, Chapter 4 has presented an alternative to improve the validity region of the Koopman operator and curtail the dimension of the lifting space in the case of time-invariant operators. The linearization about a nominal trajectory leads to a linear time-varying (LTV) system and a time-varying Koopman operator (TVKO) is developed as an alternate means to increase the prediction accuracy for a fixed dimension of the lifted space. Numerical simulations have demonstrated the utility of this newly developed time-varying Koopman operator in predicting the response of a nonlinear system. In complement, Chapter 5 has exhibited the performance of this framework through and simulations of complex and high-dimensional nonlinear systems, where traditional methods either fail or perform poorly. Time-varying reduced-order models in conjunction with a higher-order time-varying Koopman representation of the dynamics allows for far better prediction accuracy than traditional time-invariant subspace methods. Additionally, the methods and algorithms introduced from Chapter 2 to Chapter 5 are in stark contrast with machine learning a neural network-based methods. In this dissertation, the objective is to derive parsimonious and tractable surrogates of dynamical systems in order to utilize them for optimization, state prediction, uncertainty quantification, sensitivity analysis, control; the model creation is only the first step. The inherent heavy structure of surrogates based on machine learning models make them difficult to be used for any application other than pure prediction. Even though some work is being conducted to extract physical quantities from ML-based models, physics-informed AI and machine learning is still an immense field to be explored by the community.

The fourth contribution of this dissertation is to enhance and propose a unified framework for the uncertainty quantification of nonlinear systems. Several direct and indirect methods are proposed to accurately propagate the moments of an initial probability density function through nonlinear dynamics. A combined time-varying reduced-order model with CUT quadrature is able to achieve minimal error in moment propagation for a wide class of nonlinear systems. Once again, the framework proposed in Chapter 6 unifies several techniques and serves as a reference for the field of uncertainty quantification.

The fifth contribution of this dissertation is to properly and accurately introduce the Koopman operator for controlled systems. Chapter 7 clearly relates the identification of a Koopman operator for controlled system to bilinear systems and defines for the first time the bilinear Observer/Kalman Identification Algorithm (bilinear OKID). Moreover, it precisely pinpoints the difficulty associated with bilinear system identification methods by exhibiting the growth in parameters in the I/O representation for bilinear systems. Nevertheless, a continuous-time Koopman operator for controlled system is presented with successful applications to nonlinear system identification. Additionally, it is shown that bilinear system identification techniques offer a compelling framework in the sensitivity analysis of nonlinear systems.

The sixth contribution of this dissertation is to introduce a few approaches to deal with imperfect data. Noise, finite length of data, parameters variation, unmodeled dynamics are some of the issues that limit the application of classical identification algorithms. Following Chapter 8 introduces an approach with data-correlation to the time-varying eigensystem realization algorithm and demonstrate the efficacy of the new algorithm to temper the effect of noise. By properly tuning some of the parameters, the bias

terms affecting the TVERA when measurement noise is present can be omitted in the TVERA/DC, leading to better prediction accuracy. Additionally, due to measurement noise, the Hankel matrix will usually be of full rank which does not, in general, equal the true order of the system and the reliance on the SVD for the low-rank approximation makes it difficult to effectively with important levels of noise. Chapter 8 proposes a method to minimize the nuclear norm of the Hankel matrix as a way to offer an alternative for low-rank approximation problems where a clear-cut criterion for model selection cannot be deduced from the SVD. Numerical examples of different dimension with various levels of noise prove the efficacy of the proposed heuristic approach.

9.2 Future Work

The contributions developed in this dissertation extend the envelope of modern data-driven system identification; however, several notable avenues exist for further enhancing the developed framework. With the aspiration to build a computationally fast, robust and accurate data-driven framework to execute and operate real-time and on-board, we highlight several research directions for future consideration:

1. The first research direction would be to obtain a better understanding of data acquisition, specifically for system identification oriented applications. This involves what type of input signal to be used, how to condition and measure the signal, and how to acquire the output data. How to collect the data (from potentially different sources), fuse them, extract relevant features, pre-process and de-noise them involve several decisions that are crucial for a system identification framework that relies on collected data.

2. The second direction would be to properly establish an estimation framework based on observability conditions and connected to the Kalman framework. Preliminary results show that the observer gain in OKID or TVOKID is *the* Kalman gain when the observer order tends to infinity. Observability and state estimation is also a crucial component of the Koopman framework when one wants to extract the original physical state from a higher-dimensional lifting space.

3. Another research direction would be to extend bilinear system identification techniques using the TVERA framework. At present, exact bilinear system identification algorithms suffer from the growth of parameters and make use of specialized inputs to counter this effect. Alternate formulations using TVERA could have the potential to alleviate this issue.

4. Time-varying system identification algorithms provide a set of system matrices that are explicitly function of time whereas they are in fact a function of the nominal state. Initial work building libraries of system matrices over time and assigning each of them with the corresponding estimated state exhibits a lot of interest and is one solution to extend the validity of the system matrices beyond the last time-step.

5. In the same spirit, time-varying identification techniques can be used for cases where some parameter of the system is time-varying. This would lead to some sort of tensor decomposition in lieu of a classic matrix SVD decomposition.

6. The simultaneous identification of an observer and controller has been performed for time-invariant linear systems with full-state feedback. This can be extended to time-varying observer/controller

and potentially to other types of controller (other than full-state feedback).

7. A Q-Markov cover is a state-space realization that not only matches the Markov parameters of a linear system but also its covariance parameters. A time-varying version of a Q-Markov algorithm would allow one to have access to all possible equivalent time-varying realizations of a linear time-varying system.

Appendix A
Laplace Filtering Operation for Systems of Arbitrary Order

A.1 Introduction

The developments presented in this Appendix aim to extend even further the mathematical development developed in Chapter and show how to identify the dynamics of a general d^{th}-order physical system where only position measurements are accessible and the governing dynamics rely only upon position knowledge. For $d \in \mathbb{N}$, this system is represented as a general d^{th}-order nonlinear dynamical system

$$\boldsymbol{x}^{(d)}(t) = \boldsymbol{f}(\boldsymbol{x}(t)) + G\boldsymbol{u}(t), \tag{A.1}$$

where $\boldsymbol{x}(t) \in \mathbb{R}^n$ represents the state of the system and $\boldsymbol{u}(t) \in \mathbb{R}^n$ the control action at time t and $G \in \mathbb{R}^{n\times r}$ is the constant-time input influence matrix. The nonlinear function $\mathbf{f} : \mathbb{R}^n \to \mathbb{R}^n$ represents the dynamics constraints that define the equations of motion of the system and is unknown. Some important results useful for further developments will be presented first. In the following, $\lambda_1, \lambda_2, \cdots$ are real positive numbers.

A.2 Mathematical Preliminaries

For $(i, j) \in \mathbb{N}^2$, let us define the function ϕ such that $\phi(i, j) = 1$ if $i = 0$ or $j = 0$ and

$$\phi(i,j) = (-1)^j \sum_{k=1}^{i} \sum_{\substack{(l_1,l_2,\cdots,l_k)\in[\![1,i]\!] \\ n_{l_1}+n_{l_2}+\cdots+n_{l_k}=j \\ l_1<l_2<\cdots<l_k}} \lambda_{l_1}^{n_{l_1}} \lambda_{l_2}^{n_{l_2}} \dots \lambda_{l_k}^{n_{l_k}} \quad \text{otherwise.} \tag{A.2}$$

For $(i, j) \in \mathbb{N}^2$, it can be shown that the number of terms in the series $\phi(i, j)$ is 1 if i or j is 0 and

$$\sum_{k=1}^{j} \binom{i}{k} \binom{j-1}{k-1} \quad \text{otherwise.} \tag{A.3}$$

Similarly, one have the following relationship for $(i,j) \in \mathbb{N}^2$

$$\phi(i,j) - \lambda_{i+1}\phi(i+1,j-1) = \phi(i+1,j). \tag{A.4}$$

For any $d \in \mathbb{N}^*$ and $x \in \mathbb{R}$, it is possible to write

$$\begin{aligned} x^d &= a_0 \prod_{m=1}^{d} (x+\lambda_m) + a_1 \prod_{m=1}^{d-1} (x+\lambda_m) + \cdots + a_{d-1}(x+\lambda_1) + a_d \\ &= \sum_{i=0}^{d-1} \left[a_i \prod_{m=1}^{d-i} (s+\lambda_m) \right] + a_d. \end{aligned} \tag{A.5}$$

The coefficients are

$$a_i = \phi(d-i+1,i). \tag{A.6}$$

This last assumption is a central result and can be proved by induction.
For $d = 1$, one have $x^1 = a_0(x-\lambda_1) + a_1$ and can obtain $a_0 = 1$ and $a_1 = -\lambda_1$ by matching degree. Also, by definition of the function ϕ, $\phi(0,1) = 1 = a_0$ and $\phi(1,1) = -\lambda_1 = a_1$.
Suppose that the assumption holds true up to a certain integer $d \geq 1$. There exists coefficients $b_0, b_1, \cdots, b_{d+1}$ such that

$$x^{d+1} = b_0 \prod_{m=1}^{d+1} (x+\lambda_m) + b_1 \prod_{m=1}^{d} (x+\lambda_m) + \cdots + b_d(x+\lambda_1) + b_{d+1}. \tag{A.7}$$

On the other hand,

$$\begin{aligned} x^{d+1} &= x^d x \\ &= \left[a_0 \prod_{m=1}^{d} (x+\lambda_m) + a_1 \prod_{m=1}^{d-1} (x+\lambda_m) + \cdots + a_{d-1}(x+\lambda_1) + a_d \right] x \\ &= a_0 x \prod_{m=1}^{d} (x+\lambda_m) + a_1 x \prod_{m=1}^{d-1} (x+\lambda_m) + \cdots + a_{d-1}x(x+\lambda_1) + a_d x \\ &= a_0 \prod_{m=1}^{d+1} (x+\lambda_m) + a_1 \prod_{m=1}^{d} (x+\lambda_m) + \cdots + a_{d-1}(x+\lambda_1)(x+\lambda_2) + a_d(x+\lambda_1) \\ &\quad - \left[a_0\lambda_{d+1} \prod_{m=1}^{d} (x+\lambda_m) + a_1\lambda_d \prod_{m=1}^{d-1} (x+\lambda_m) + \cdots + a_{d-1}\lambda_2(x+\lambda_1) + a_d\lambda_1 \right] \\ &= a_0 \prod_{m=1}^{d+1} (x+\lambda_m) + (a_1 - a_0\lambda_{d+1}) \prod_{m=1}^{d} (x+\lambda_m) + \cdots + (a_d - a_{d-1}\lambda_2)(x+\lambda_1) - a_d\lambda_1. \end{aligned} \tag{A.8}$$

It is clear that $b_0 = a_0 = 1 = \phi(n+2,0)$ and $b_{d+1} = -a_d\lambda_1 = -(-1)^d\lambda_1^d \cdot \lambda_1 = (-1)^{d+1}\lambda_1^{d+1} = \phi(1,d+1)$.
For the other coefficients,

$$b_i = a_i - a_{i-1}\lambda_{d+2-i} = \phi(d-i+1,i) - \phi(d-i+1+1,i-1)\lambda_{d+2-i} = \phi(d+2-i,i). \tag{A.9}$$

This finishes the proof.

A.3 Laplace Filtering Operation

Before proceeding, consider a set of basis functions $\{\phi_i\}_{i=1\ldots\infty}, \phi_i : \mathbb{R}^n \to \mathbb{R}$, so that one can approximate $\boldsymbol{f}$ as a linear combination of finite number of N basis functions such that $\forall\ \boldsymbol{x} \in \mathbb{R}^n$, the following is true

$$\boldsymbol{f}(\boldsymbol{x}) = \sum_{i=1}^{\infty} \boldsymbol{\alpha}_i \phi_i(\boldsymbol{x}), \tag{A.10}$$

where $\{\boldsymbol{\alpha}_i\}_{i=1..\infty}, \boldsymbol{\alpha}_i \in \mathbb{R}^n$, is a set of unknown coefficients. In practice, a specific integer N is selected such that the N first basis functions will provide an acceptable approximation of the actual function $\boldsymbol{f}$:

$$\boldsymbol{f}(\boldsymbol{x}) \approx \sum_{i=1}^{N} \boldsymbol{\alpha}_i \phi_i(\boldsymbol{x}) = \boldsymbol{\alpha}\boldsymbol{\phi}(\boldsymbol{x}), \tag{A.11}$$

or equivalently

$$\boldsymbol{f}(\boldsymbol{x}) \approx \boldsymbol{\alpha}\boldsymbol{\phi}(\boldsymbol{x}), \tag{A.12}$$

where $\boldsymbol{\alpha} = \begin{bmatrix} \boldsymbol{\alpha}_1 & \boldsymbol{\alpha}_2 & \cdots & \boldsymbol{\alpha}_N \end{bmatrix}^T \in \mathbb{R}^{N\times n}$ and $\boldsymbol{\phi}(\boldsymbol{x}) = \begin{bmatrix} \phi_1(\boldsymbol{x}) & \phi_2(\boldsymbol{x}) & \cdots & \phi_N(\boldsymbol{x}) \end{bmatrix}^T \in \mathbb{R}^{N\times 1}$. Hence Eq. (A.1) can be rewritten as

$$\boldsymbol{x}^{(d)}(t) = \sum_{i=1}^{N} \boldsymbol{\alpha}_i \phi_i(\boldsymbol{x}) + G\boldsymbol{u}(t). \tag{A.13}$$

From now on, consider a one-dimensional system, that is $n = 1$ and $\boldsymbol{x} = x$ is a scalar variable. If $n > 1$, it is sufficient to work along the dimension of $\boldsymbol{x}$ and repeat the steps along each dimension. The Laplace transform of the left-hand side of Eq. (A.13) is

$$\mathcal{L}\left\{x^{(d)}(t)\right\} = s^d X(s) - \sum_{k=1}^{d} s^{d-k} x^{(k-1)}(0). \tag{A.14}$$

For $\lambda_i \in \mathbb{R}_+^*$, let us consider the Laplace filtering operator

$$\begin{array}{rccc} \mathcal{I}_{\lambda_i} : & \mathbb{R} & \to & \mathbb{R}, \\ & \bullet & \mapsto & \dfrac{\bullet}{s+\lambda_i}. \end{array} \tag{A.15}$$

Applying successively $\mathcal{I}_{\lambda_1}, \mathcal{I}_{\lambda_2}, \cdots, \mathcal{I}_{\lambda_d}$ to Eq. (A.14) gives

$$X^{\mathrm{df}}(s) = \mathcal{I}_{\lambda_d} \circ \mathcal{I}_{\lambda_{d-1}} \circ \cdots \circ \mathcal{I}_{\lambda_1} \left[\mathcal{L} \left\{ x^{(d)}(t) \right\} \right] = \frac{s^d X(s) - \sum_{k=1}^{d} s^{d-k} x^{(k-1)}(0)}{\prod_{m=1}^{d} (s + \lambda_m)} = X_1^{\mathrm{df}}(s) - X_2^{\mathrm{df}}(s), \tag{A.16}$$

where

$$X_1^{\mathrm{df}}(s) = \frac{s^d X(s)}{\prod_{m=1}^{d} (s + \lambda_m)} \qquad \text{and} \qquad X_2^{\mathrm{df}}(s) = \frac{\sum_{k=1}^{d} s^{d-k} x^{(k-1)}(0)}{\prod_{m=1}^{d} (s + \lambda_m)}. \tag{A.17}$$

Using previous developments, it is possible to write $X_1^{\mathrm{df}}(s)$ as

$$X_1^{\mathrm{df}}(s) = \frac{s^d X(s)}{\prod_{m=1}^{d} (s + \lambda_m)} = X(s) + \sum_{i=1}^{d} \frac{\phi(d-i+1, i)}{\prod_{m=0}^{i-1} (s + \lambda_{d-m})} X(s). \tag{A.18}$$

Similarly, the second term $X_2^{\mathrm{df}}(s)$ is

$$\begin{aligned}
X_2^{\mathrm{df}}(s) &= \frac{\sum_{k=1}^{d} s^{d-k} x^{(k-1)}(0)}{\prod_{m=1}^{d} (s + \lambda_m)} \\
&= \frac{\sum_{k=1}^{d-1} \left(\sum_{i=0}^{d-k-1} \phi(d-k-i+1, i) \prod_{m=1}^{d-k-i} (s + \lambda_m) + \phi(1, d-k) \right) x^{(k-1)}(0)}{\prod_{m=1}^{d} (s + \lambda_m)} + \frac{x^{(d-1)}(0)}{\prod_{m=1}^{d} (s + \lambda_m)} \\
&= \sum_{k=1}^{d-1} \sum_{j=k}^{d-1} \frac{\phi(d-j+1, j-k)}{\prod_{m=0}^{j-1} (s + \lambda_{d-m})} x^{(k-1)}(0) + \sum_{k=1}^{d} \frac{\phi(1, d-k)}{\prod_{m=0}^{d-1} (s + \lambda_{d-m})} x^{(k-1)}(0) \\
&= \sum_{j=1}^{d} \sum_{k=1}^{j} \frac{\phi(d-j+1, j-k)}{\prod_{m=0}^{j-1} (s + \lambda_{d-m})} x^{(k-1)}(0)
\end{aligned} \tag{A.19}$$

Finally, the compact expression for X^{df} is:

$$X^{\mathrm{df}}(s) = X(s) + \sum_{j=1}^{d} \frac{\phi(d-j+1,j)X(s) - \sum_{k=1}^{j} \phi(d-j+1,j-k)x^{(k-1)}(0)}{\prod_{m=0}^{j-1}(s+\lambda_{d-m})}. \tag{A.20}$$

This expression of X^{df} is easily expressed in the time-domain. Performing the same operations on the right-hand side of Eq. (A.13) leads to

$$\Phi_i^{\mathrm{df}}(s) = \mathcal{I}_{\lambda_d} \circ \mathcal{I}_{\lambda_{d-1}} \circ \cdots \circ \mathcal{I}_{\lambda_1} \left[\mathcal{L}\left\{\phi_i(x)\right\}\right] = \frac{\Phi_i(s)}{\prod_{m=1}^{d}(s+\lambda_m)}, \quad i = 1, 2, \ldots, N, \tag{A.21}$$

and

$$U^{\mathrm{df}}(s) = \mathcal{I}_{\lambda_d} \circ \mathcal{I}_{\lambda_{d-1}} \circ \cdots \circ \mathcal{I}_{\lambda_1} \left[\mathcal{L}\left\{u(t)\right\}\right] = \frac{U(s)}{\prod_{m=1}^{d}(s+\lambda_m)}, \tag{A.22}$$

where $\mathcal{L}\{\phi_i(\boldsymbol{x}(t))\} = \Phi_i(s)$ and $\mathcal{L}\{u(t)\} = U(s)$. Finally, the Laplace filtered version of Eq. (A.13) is

$$X^{\mathrm{df}}(s) = \sum_{i=1}^{N} \alpha_i \Phi_i^{\mathrm{df}}(s) + G U^{\mathrm{df}}(s). \tag{A.23}$$

with X^{df}, Φ_i^{df} and U^{df} defined in Eq. (A.20), Eq. (A.21) and Eq. (A.22). Going back in the time domain, one can write the corresponding differential equations for Φ^{df} and U^{df} immediately. For Φ_i^{df}, first rewrite Eq. (A.21) as

$$\begin{aligned} \Phi_i^{\mathrm{df}}(s) = \frac{\Phi_i(s)}{\prod_{m=1}^{d}(s+\lambda_m)} &\Leftrightarrow \Phi_i^{\mathrm{df}}(s) \prod_{m=1}^{d}(s+\lambda_m) = \Phi_i(s) \\ &\Leftrightarrow s^d \Phi_i^{\mathrm{df}}(s) = -\sum_{k=0}^{d-1} a_k s^k \Phi_i^{\mathrm{df}}(s) + \Phi_i(s). \end{aligned} \tag{A.24}$$

with

$$a_{m-k} = \sum_{1 \le l_1 < l_2 < \cdots < l_k \le m} \prod_{j=1}^{k} \lambda_{l_j}. \tag{A.25}$$

Considering zero initial conditions, $\phi_i^{\mathrm{df}(k)}(0) = 0$ for $k = 0, 1, \ldots, d$, one obtains the differential equation

$$\phi_i^{\mathrm{df}(d)}(t) = -\sum_{k=0}^{d-1} a_k \phi_i^{\mathrm{f}(k)}(t) + \phi_i(x), \quad i = 1, 2, \ldots, N. \tag{A.26}$$

The same procedure for the input signal leads to the differential equation

$$u^{\mathrm{df}(d)}(t) = -\sum_{k=0}^{d-1} a_k u^{\mathrm{df}(k)}(t) + u(t), \tag{A.27}$$

with similar zero initial conditions. Notice that Eq. (A.20), Eq. (A.26) and Eq. (A.27) lead to Eq. (3.11), Eq. (3.14) and Eq. (3.15) for $d = 1$ and Eq. (3.30), Eq. (3.34) and Eq. (3.35) for $d = 2$.

Appendix B
Controllability and Observability of linear systems

Before solving for the Markov parameters, it is of great importance to know whether all the states of a system can be controlled and/or observed since a solvable system of linear algebraic equations has a solution if and only if the rank of the system matrix is full. While controllability is concerned with whether one can design control input to steer the state to arbitrarily values, observability is concerned with whether without knowing the initial state, one can determine the state of a system given the input and the output.

B.1 Controllability in the Discrete-Time Domain

A state $\boldsymbol{x}(q)$ is said to be *controllable* or *state-controllable* if this state can be reached from any initial state of the system in a finite time interval by some control action. If all states are controllable, the system is called *completely controllable* or simply *controllable*. Given A, B and $\boldsymbol{x}(0)$, the idea is to find the sufficient and necessary condition to determine how to reach $\boldsymbol{x}(q)$ without ambiguity. It is clear that since A and $\boldsymbol{x}(0)$ are given, it is therefore equivalent to determine $\boldsymbol{x}(q)$ or $\tilde{\boldsymbol{x}}(q) = \boldsymbol{x}(q) - A^q\boldsymbol{x}(0)$: to determine complete controllability, it is sufficient and necessary to determine whether the zero state (instead of all initial states) can be transferred to all final states.

The solution to the discrete representation at time $t = q\Delta t$ where Δt is the sampling period is

$$\boldsymbol{x}(q) = A^q\boldsymbol{x}(0) + \sum_{i=1}^{q} A^{i-1}B\boldsymbol{u}(q-i) \tag{B.1}$$

or in a compact matrix form

$$\boldsymbol{x}(q) = A^q \boldsymbol{x}(0) + \begin{bmatrix} B & AB & A^2B & \cdots & A^{q-1}B \end{bmatrix} \begin{bmatrix} \boldsymbol{u}(q-1) \\ \boldsymbol{u}(q-2) \\ \boldsymbol{u}(q-3) \\ \vdots \\ \boldsymbol{u}(0) \end{bmatrix}. \tag{B.2}$$

The expression of $\tilde{\boldsymbol{x}}(q)$ can be written as

$$\tilde{\boldsymbol{x}}(q) = \boldsymbol{R}^{(q)} \overline{\boldsymbol{u}} \tag{B.3}$$

where

$$\boldsymbol{R}^{(q)} = \begin{bmatrix} B & AB & A^2B & \cdots & A^{q-1}B \end{bmatrix}, \tag{B.4}$$

and

$$\overline{\boldsymbol{u}} = \begin{bmatrix} \boldsymbol{u}(q-1) \\ \boldsymbol{u}(q-2) \\ \boldsymbol{u}(q-3) \\ \vdots \\ \boldsymbol{u}(0) \end{bmatrix}. \tag{B.5}$$

$\boldsymbol{R}^{(q)}$ is called the controllability matrix. Reaching $\boldsymbol{x}(q)$ without ambiguity is thus equivalent to find a solution of Eq. (B.3) for $\overline{\boldsymbol{u}}$. Therefore, the discrete time-invariant linear system, Eq. (8.1), of order n is controllable if and only if the $n \times qr$ block controllability matrix $\boldsymbol{R}^{(q)}$ has rank n (assuming $qr > n$).

B.2 Observability in the Discrete-Time Domain

A state $\boldsymbol{x}(p)$ is *observable* if the knowledge of the input $\boldsymbol{u}(k)$ and output $\boldsymbol{y}(k)$ over a finite time interval $0 \leq k \leq p-1$ completely determines the state $\boldsymbol{x}(p)$:

$$\boldsymbol{x}(p) = A^p \boldsymbol{x}(0) + \sum_{i=1}^{p} A^{i-1} B \boldsymbol{u}(p-i). \tag{B.6}$$

With knowledge of the system matrices A and B and the control input $\boldsymbol{u}(k)$, $0 \leq k \leq p-1$, it is necessary and sufficient to see whether the initial state $\boldsymbol{x}(0)$ can be completely determined from the

output sequence $\boldsymbol{y}(k)$, $0 \leq k \leq p-1$. In a compact matrix form, we can write

$$\overline{\boldsymbol{y}} = \begin{bmatrix} \boldsymbol{y}(0) \\ \boldsymbol{y}(1) \\ \boldsymbol{y}(2) \\ \vdots \\ \boldsymbol{y}(p-1) \end{bmatrix} = \begin{bmatrix} C \\ CA \\ CA^2 \\ \vdots \\ CA^{p-1} \end{bmatrix} \boldsymbol{x}(0) = \boldsymbol{O}^{(p)} \boldsymbol{x}(0), \tag{B.7}$$

where a unique solution for $\boldsymbol{x}(0)$ exists if and only if $\boldsymbol{O}^{(p)}$ has rank n (full rank, assuming $pm > n$). Thus, the discrete time-invariant linear system, Eq. (8.1), of order n is observable if and only if the $pm \times n$ block observability matrix $\boldsymbol{O}^{(p)}$ has rank n, where

$$\boldsymbol{O}^{(p)} = \begin{bmatrix} C \\ CA \\ CA^2 \\ \vdots \\ CA^{p-1} \end{bmatrix}. \tag{B.8}$$

These two notions of controllability and observability will be of central attention in the next section for the development of the Eigensystem Realization Algorithm.

B.3 Coordinate Transformation

After having introduced the basic formulations of discrete-time invariant linear systems and before going in depth in the narrative of the Eigensystem Realization Algorithm, some results concerning coordinate transformation are introduced in this section.

In many problems, analysts need to use different coordinate systems in order to describe different quantities. A well-chosen coordinate system allows position and direction in space to be represented in a very convenient manner and allow sometimes to have some insights about the system itself. After all, two independent observers might well choose coordinate systems with different origins, and different orientations of the coordinate axes. The dynamic behavior of a mechanical system strongly depends upon its mathematical representation. This is due to the fact that nonlinearity is not an inherent attribute of a physical system, but rather dependent upon the mathematical description of the system's behavior. Ideally, one has an infinity of coordinate choices to represent the same physical system. In the study of celestial mechanics, the quest to find a judicious coordinate system led to the development of various canonical transformations. An extended discussion will be conducted in section concerning coordinate systems and transformations. This section only presents a few important results for linear discrete-time invariant systems.

Let a new state vector be defined such that

$$\tilde{\boldsymbol{x}}(k) = T\boldsymbol{x}(k), \tag{B.9}$$

where T is a nonsingular square matrix. Substitution of Eq. (B.9) into Eq. (8.1) yields

$$\begin{cases} \tilde{\boldsymbol{x}}(k+1) = TAT^{-1}\tilde{\boldsymbol{x}}(k) + TB\boldsymbol{u}(k) \\ \boldsymbol{y}(k) \qquad = CT^{-1}\tilde{\boldsymbol{x}}(k) + \tilde{D}\boldsymbol{u}(k) \end{cases} \tag{B.10}$$

The effect of the input $\boldsymbol{u}(k)$ on the output $\boldsymbol{y}(k)$ is unchanged for the system. Thus the matrices $\left\{TAT^{-1}, TB, CT^{-1}, D\right\}$ describe the same input-output relationship as the matrices $\{A, B, C, D\}$:

$$\begin{cases} \boldsymbol{x}(k+1) = A\boldsymbol{x}(k) + B\boldsymbol{u}(k) \\ \boldsymbol{y}(k) \qquad = C\boldsymbol{x}(k) + D\boldsymbol{u}(k) \end{cases} \Leftrightarrow \begin{cases} \tilde{\boldsymbol{x}}(k+1) = \tilde{A}\tilde{\boldsymbol{x}}(k) + \tilde{B}\boldsymbol{u}(k) \\ \boldsymbol{y}(k) \qquad = \tilde{C}\tilde{\boldsymbol{x}}(k) + \tilde{D}\boldsymbol{u}(k) \end{cases} \tag{B.11}$$

with

$$\tilde{\boldsymbol{x}}(k) = T\boldsymbol{x}(k), \tag{B.12a}$$

$$\tilde{A} = TAT^{-1}, \tag{B.12b}$$

$$\tilde{B} = TB, \tag{B.12c}$$

$$\tilde{C} = CT^{-1}, \tag{B.12d}$$

$$\tilde{D} = D. \tag{B.12e}$$

This transformation is called a similarity transformation. The fact that the input-output relationship remains unchanged should also indicate that the pulse sequence, or Markov parameters, is invariant through coordinate transformation. Indeed, for $i \geq 1$,

$$\tilde{h}_i = \tilde{C}\tilde{A}^{i-1}\tilde{B} = CT^{-1}\left(TAT^{-1}\right)^{i-1}TB = CT^{-1}TA^{i-1}T^{-1}TB = CA^{i-1}B = h_i. \tag{B.13}$$

As a result, there are an infinite number of state-space representations that produce the same input-output description. Additionally, because matrices are similar if and only if they represent the same linear operator with respect to (possibly) different bases, similar matrices share all properties of their shared underlying operator (their rank in particular).

Appendix C
Observer Kalman Filter Identification Algorithm (OKID)

Most techniques to identify the Markov parameters sequence Eq. (2.11) are based on the Fast Fourier Transform (FFT) of the inputs and measured outputs to compute the Frequency Response Functions (FRFs), and then use the Inverse Discrete Fourier Transform (IDFT) to compute the sampled pulse response histories. The discrete nature of the FFT causes one to obtain pulse response rather than impulse response, and a somewhat rich input is required to prevent numerical ill-conditioning. Indeed, the FRF is a ratio between the output and input DFT transform coefficients which requires the input signal to be rich in frequencies so that the corresponding quantity is invertible. However, considerable information can be deduced simply by observing frequency response functions, justifying why FRFs are still generated so often. Another approach is to solve directly in the time domain for the system Markov parameters from the input and output data. In [15], a method has been developed to compute the Markov parameters of a linear system in the time-domain. A drawback of this direct time-domain method is the need to invert an input matrix which necessarily becomes particularly large for lightly damped systems. Rather than identifying the system Markov parameters which may exhibit very slow decay, one can use an asymptotically stable observer to form a stable discrete state-space model for the system to be identified. The method is referred as the Observer/Kalman filter Identification algorithm (OKID) and is a procedure where the state-space model and a corresponding observer are determined simultaneously [15, 19].

C.1 Classical Formulation

Considering a sequence of l elements, assuming zero initial condition $\boldsymbol{x}_0 = \mathbf{0}$:

$$\boldsymbol{y}_{l-1} = \sum_{i=1}^{l-1} CA^{i-1}B\boldsymbol{u}_{l-1-i} + D\boldsymbol{u}_{l-1}. \tag{C.1}$$

In a matrix form, Eq. (C.1) is written as

$$\bar{\boldsymbol{y}} = \boldsymbol{Y}\boldsymbol{U} \tag{C.2}$$

with

$$\bar{\boldsymbol{y}} = \begin{bmatrix} \boldsymbol{y}_0 & \boldsymbol{y}_1 & \boldsymbol{y}_2 & \cdots & \boldsymbol{y}_{l-1} \end{bmatrix}, \tag{C.3a}$$

$$\boldsymbol{Y} = \begin{bmatrix} D & CB & CAB & \cdots & CA^{l-2}B \end{bmatrix}, \tag{C.3b}$$

$$\boldsymbol{U} = \begin{bmatrix} \boldsymbol{u}_0 & \boldsymbol{u}_1 & \boldsymbol{u}_2 & \cdots & \boldsymbol{u}_{l-1} \\ & \boldsymbol{u}_0 & \boldsymbol{u}_1 & \cdots & \boldsymbol{u}_{l-2} \\ & & \boldsymbol{u}_0 & \cdots & \boldsymbol{u}_{l-3} \\ & & & \ddots & \vdots \\ & & & & \boldsymbol{u}_0 \end{bmatrix}. \tag{C.3c}$$

The matrix $\bar{\boldsymbol{y}}$ is an $m \times l$ output data matrix and the matrix $\boldsymbol{Y}$, of dimension $m \times rl$, contains all the Markov parameters to be determined. As summarized in Table C.1, Eq. (E.2) is insolvable in the multi-input multi-output case in general: the solution $\boldsymbol{Y}$ is not unique whereas it should be (Markov parameters must be unique for a finite-dimensional linear system). The matrix $\boldsymbol{Y}$ can only be uniquely determined from this set of equations for $r = 1$. However, even in this case, if the input has zero initial value or is not rich enough in frequency content or if anything makes the matrix $\boldsymbol{U}$ ill-conditioned, the matrix $\boldsymbol{Y} = \bar{\boldsymbol{y}}\boldsymbol{U}^{-1}$ cannot be accurately computed.

Table C.1: Equations vs. unknowns for Eq. (C.2)

# Equations	# Unknowns
$m \times l$	$m \times rl$

However, if A is asymptotically stable so that for some l_0, $CA^kB \simeq 0$ for all time steps $k \geq l_0$, Eq. (E.2) can be approximated by

$$\bar{\boldsymbol{y}} \simeq \widetilde{\boldsymbol{Y}}\widetilde{\boldsymbol{U}}, \tag{C.4}$$

with

$$\bar{\boldsymbol{y}} = \begin{bmatrix} \boldsymbol{y}_0 & \boldsymbol{y}_1 & \boldsymbol{y}_2 & \cdots & \boldsymbol{y}_{l-1} \end{bmatrix}, \tag{C.5a}$$

$$\widetilde{\boldsymbol{Y}} = \begin{bmatrix} D & CB & CAB & \cdots & CA^{l_0-1}B \end{bmatrix}, \tag{C.5b}$$

$$\widetilde{\boldsymbol{U}} = \begin{bmatrix} \boldsymbol{u}_0 & \boldsymbol{u}_1 & \boldsymbol{u}_2 & \cdots & \boldsymbol{u}_{l_0} & \cdots & \boldsymbol{u}_{l-1} \\ & \boldsymbol{u}_0 & \boldsymbol{u}_1 & \cdots & \boldsymbol{u}_{l_0-1} & \cdots & \boldsymbol{u}_{l-2} \\ & & \boldsymbol{u}_0 & \cdots & \boldsymbol{u}_{l_0-2} & \cdots & \boldsymbol{u}_{l-3} \\ & & & \ddots & \vdots & \cdots & \vdots \\ & & & & \boldsymbol{u}_0 & \cdots & \boldsymbol{u}_{l-l_0-1} \end{bmatrix}. \tag{C.5c}$$

Choose the data record length l greater than $r(l_0 + 1)$ and Eq. (C.4) indicates that there are more equations than constraints, as summarized in Table C.2. It follows that the first l_0+1 Markov parameters

approximately satisfy

$$\widetilde{\boldsymbol{Y}} = \bar{\boldsymbol{y}}\widetilde{\boldsymbol{U}}^{\dagger}, \tag{C.6}$$

and the approximation error decreases as l_0 increases.

Table C.2: Equations vs. unknowns for Eq. (C.4)

# Equations	# Unknowns
$m \times l$	$m \times r(l_0 + 1)$

Unfortunately, for lightly damped structures, the integer l_0 and thus the the data length l required to make the approximation in Eq. (C.4) valid becomes impractically large in the sense that the size of the matrix $\widetilde{\boldsymbol{U}}$ is too large to solve for its pseudo-inverse numerically. A solution to artificially increase the damping of the system in order to allow a decent numerical solution is to add a feedback loop to make the system as stable as desired.

C.2 State-Space Observer Model

In practice, the primary purpose of introducing an observer is an artifice to compress the data and improve system identification results. A state estimator, also known as an observer, can be used to provide an estimate of the system state from input and output measurements. Add and subtract the term $G\boldsymbol{y}_k$ to the right-hand side of the state equation in Eq. (8.1a) to yield

$$\begin{aligned} \boldsymbol{x}_{k+1} &= A\boldsymbol{x}_k + B\boldsymbol{u}_k + G\boldsymbol{y}_k - G\boldsymbol{y}_k && \text{(C.7a)} \\ &= (A + GC)\boldsymbol{x}_k + (B + GD)\boldsymbol{u}_k - G\boldsymbol{y}_k && \text{(C.7b)} \\ &= \bar{A}\boldsymbol{x}_k + \bar{B}\boldsymbol{v}_k && \text{(C.7c)} \end{aligned}$$

where

$$\bar{A} = A + GC, \tag{C.8a}$$

$$\bar{B} = \begin{bmatrix} B + GD & -G \end{bmatrix}, \tag{C.8b}$$

$$\boldsymbol{v}_k = \begin{bmatrix} \boldsymbol{u}_k \\ \boldsymbol{y}_k \end{bmatrix}, \tag{C.8c}$$

and G is an arbitrary matrix chosen to make the matrix $\bar{A}$ as stable as desired. The Markov parameters of this observer system are referred as *observer Markov parameters*. The new input-output in matrix form is therefore

$$\bar{\boldsymbol{y}} = \bar{\boldsymbol{Y}}\boldsymbol{V} \tag{C.9}$$

with

$$\bar{\boldsymbol{y}} = \begin{bmatrix} \boldsymbol{y}_0 & \boldsymbol{y}_1 & \boldsymbol{y}_2 & \cdots & \boldsymbol{y}_{l-1} \end{bmatrix}, \tag{C.10}$$

$$\bar{\boldsymbol{Y}} = \begin{bmatrix} D & C\bar{B} & C\bar{A}\bar{B} & \cdots & C\bar{A}^{l-2}\bar{B} \end{bmatrix}, \tag{C.11}$$

$$\boldsymbol{V} = \begin{bmatrix} \boldsymbol{u}_0 & \boldsymbol{u}_1 & \boldsymbol{u}_2 & \cdots & \boldsymbol{u}_{l-1} \\ & \boldsymbol{v}_0 & \boldsymbol{v}_1 & \cdots & \boldsymbol{v}_{l-2} \\ & & \boldsymbol{v}_0 & \cdots & \boldsymbol{v}_{l-3} \\ & & & \ddots & \vdots \\ & & & & \boldsymbol{v}_0 \end{bmatrix}. \tag{C.12}$$

Table C.3: Equations vs. unknowns for Eq. (C.9)

# Equations	# Unknowns
$m \times l$	$m \times ((r+m)(l-1)+r)$

Similarly as before, when $C\bar{A}^k\bar{B} \simeq 0$ for all time steps $k \geq l_0$ for some l_0, Eq. (C.9) can be approximated by

$$\bar{\boldsymbol{y}} \simeq \widetilde{\bar{\boldsymbol{Y}}}\widetilde{\boldsymbol{V}}, \tag{C.13}$$

with

$$\bar{\boldsymbol{y}} = \begin{bmatrix} \boldsymbol{y}_0 & \boldsymbol{y}_1 & \boldsymbol{y}_2 & \cdots & \boldsymbol{y}_{l-1} \end{bmatrix}, \tag{C.14}$$

$$\widetilde{\bar{\boldsymbol{Y}}} = \begin{bmatrix} D & C\bar{B} & C\bar{A}\bar{B} & \cdots & C\bar{A}^{l_0-1}\bar{B} \end{bmatrix}, \tag{C.15}$$

$$\widetilde{\boldsymbol{V}} = \begin{bmatrix} \boldsymbol{u}_0 & \boldsymbol{u}_1 & \boldsymbol{u}_2 & \cdots & \boldsymbol{u}_{l_0} & \cdots & \boldsymbol{u}_{l-1} \\ & \boldsymbol{v}_0 & \boldsymbol{v}_1 & \cdots & \boldsymbol{v}_{l_0-1} & \cdots & \boldsymbol{v}_{l-2} \\ & & \boldsymbol{v}_0 & \cdots & \boldsymbol{v}_{l_0-2} & \cdots & \boldsymbol{v}_{l-3} \\ & & & \ddots & \vdots & \cdots & \vdots \\ & & & & \boldsymbol{v}_0 & \cdots & \boldsymbol{v}_{l-l_0-1} \end{bmatrix}. \tag{C.16}$$

Table C.4: Equations vs. unknowns for Eq. (C.13)

# Equations	# Unknowns
$m \times l$	$m \times ((r+m)l_0+r)$

The first l_0+1 observer Markov parameters approximately satisfy

$$\widetilde{\bar{\boldsymbol{Y}}} = \bar{\boldsymbol{y}}\widetilde{\boldsymbol{V}}^{\dagger}, \tag{C.17}$$

and the approximation error decreases as l_0 increases. To solve for $\widetilde{\bar{\boldsymbol{Y}}}$ uniquely, all the rows of $\widetilde{\boldsymbol{V}}$ must be linearly independent. Furthermore, to minimize any numerical error due to the computation of the pseudo-inverse, the rows of $\widetilde{\boldsymbol{V}}$ should be chosen as independently as possible. As a result, the maximum value of l_0 is the number that maximizes the quantity $(r+m)l_0 + r \leq l$ of independent rows of $\widetilde{\boldsymbol{V}}$. The maximum l_0 means the upper bound of the order of the deadbeat observer. Furthermore, it is known that the rank of a sufficiently large Hankel matrix $\boldsymbol{H}_0^{(p,q)}$ is the order of the controllable and observable part of the system (the identified state matrix $\hat{A}$ represents only the controllable and observable part of the system). The size of the Hankel matrix is $pm \times qr$ comprised of $p+q-1$ Markov parameters; with $p=q$, this means $2p-1$ Markov parameters. If l_0 is the number of Markov parameters calculated through OKID, it means that $l_0 = 2p-1$. Assuming $qr > n$, the maximum rank of $\boldsymbol{H}_0^{(p,q)}$ is thus mp. If p is chosen such that $mp \geq n$, then a realized state matrix $\hat{A}$ with order n should exist. Therefore, the number of Markov parameters computed, l_0, must be chosen such that

$$mp \geq n \Leftrightarrow m\frac{l_0+1}{2} \geq n, \tag{C.18}$$

where m is the number of outputs and n is the order of the system. To conclude, the lower and upper bounds on l_0 are

$$\frac{2n}{m} - 1 \leq l_0 \leq \frac{l-r}{r+m} \tag{C.19}$$

with l being the length of the input signal considered.

The observer Markov parameters can then be used to compute the Markov parameters and the matrices A, B, C and D.

C.3 Computation of Markov Parameters from Observer Markov Parameters

To recover the system Markov parameters from the observer Markov parameters, write

$$\bar{h}_0 = D, \tag{C.20a}$$

$$\bar{h}_k = C\bar{A}^{k-1}\bar{B} \tag{C.20b}$$

$$= \begin{bmatrix} C(A+GC)^{k-1}(B+GD) & -C(A+GC)^{k-1}G \end{bmatrix} \tag{C.20c}$$

$$= \begin{bmatrix} \bar{h}_k^{(1)} & -\bar{h}_k^{(2)} \end{bmatrix} \tag{C.20d}$$

Thus, the Markov parameter h_1 of the system is simply

$$h_1 = CB = C(B+GD) - (CG)D = \bar{h}_1^{(1)} - \bar{h}_1^{(2)}D = \bar{h}_1^{(1)} - \bar{h}_1^{(2)}h_0. \tag{C.21}$$

Considering the product

$$\bar{h}_2^{(1)} = C(A+GC)(B+GD) = CAB + CGCB + C(A+GC)GD = h_2 + \bar{h}_1^{(2)}h_1 + \bar{h}_2^{(2)}h_0, \tag{C.22}$$

the next Markov parameter is

$$h_2 = CAB = \bar{h}_2^{(1)} - \bar{h}_1^{(2)} h_1 - \bar{h}_2^{(2)} h_0. \tag{C.23}$$

Note that the sum of subscript(s) of each individual term both sides is identical. By induction, the general relationship between the actual system Markov parameters and the observer Markov parameters is

$$h_0 = \bar{h}_0, \tag{C.24a}$$

$$h_k = \bar{h}_k^{(1)} - \sum_{i=1}^{k} \bar{h}_i^{(2)} h_{k-i}, \quad \text{for } k \geq 1. \tag{C.24b}$$

Appendix D
System Realization Theory for Linear Time-Varying Systems

The methods developed in this appendix arise from a perspective of generalizing the classical Ho-Kalman approach with OKID/ERA to the case of time-varying systems. It is shown that the generalization thus made enables the identification of time varying plant models that are in arbitrary coordinate systems at each time step. Furthermore, the coordinate systems at successive time steps are compatible with one another and makes the model sequences realized, useful in state propagation. The first two section review the basics of time-varying linear system identification while the third section discusses the important notion of time-varying coordinate frames. The fourth section introduces the time-varying version of ERA, TVERA, and the computational methods of generalized Markov parameters using the input output map are subsequently discussed in the fifth section. Numerical examples demonstrating the theoretical developments conclude the chapter.

D.1 Introduction on Linear Discrete-Time Time-Varying State-Space Models

A linear discrete-time varying system is given by

$$\boldsymbol{x}_{k+1} = A_k \boldsymbol{x}_k + B_k \boldsymbol{u}_k \tag{D.1a}$$

$$\boldsymbol{y}_k = C_k \boldsymbol{x}_k + D_k \boldsymbol{u}_k \tag{D.1b}$$

together with an initial state vector $\boldsymbol{x}_0$, where $\boldsymbol{x}_k \in \mathbb{R}^n$, $\boldsymbol{u}_k \in \mathbb{R}^r$ and $\boldsymbol{y}_k \in \mathbb{R}^m$ are the state, control input and output vectors respectively, $k \geq 0$. Similarly to the time invariant case, the time-varying (non constant) matrices A_k, B_k, C_k and D_k with appropriate dimensions represent the internal operation of the linear system, and are used to determine the system's response to any input. The solution of the difference equation, given in Eqs. (D.1) in the time varying case, is now given by

$$\boldsymbol{x}_k = \Phi_{k,0} \boldsymbol{x}_0 + \sum_{i=0}^{k-1} \Phi_{k,i+1} B_i \boldsymbol{u}_i, \tag{D.2a}$$

$$\boldsymbol{y}_k = C_k \Phi_{k,0} \boldsymbol{x}_0 + \sum_{i=0}^{k-1} C_k \Phi_{k,i+1} B_i \boldsymbol{u}_i + D_k \boldsymbol{u}_k, \tag{D.2b}$$

where the state transition matrix is defined in terms of its components by

$$\Phi_{k,k_0} = \begin{cases} A_{k-1} A_{k-2} \dots A_{k_0} & \text{for } k > k_0, \\ I & \text{for } k = k_0, \\ \text{undefined} & \text{for } k < k_0. \end{cases} \tag{D.3}$$

This state transition matrix is associated with the homogeneous part of Eq. (D.2) ($\boldsymbol{u}_k = \boldsymbol{0}$, initial condition response). By defining the generalized Markov parameters (or pulse response matrix) as

$$h_{k,i} = \begin{cases} C_k \Phi(k, i+1) B_i & \text{for } i < k-1, \\ C_k B_{k-1} & \text{for } i = k-1, \\ 0 & \text{for } i > k-1, \end{cases} \tag{D.4}$$

the input-output relationship in terms of this two index coefficients is written as

$$\boldsymbol{y}_k = C_k \Phi_{k,0} \boldsymbol{x}_0 + \sum_{i=0}^{k-1} h_{k,i} \boldsymbol{u}_i + D_k \boldsymbol{u}_k. \tag{D.5}$$

For time-invariant systems, $h_{k,i}$ depends on $k - i$ only.

D.2 Controllability and Observability

The two central notions of controllability and observability for linear time-varying systems do not evolve from the time-invariant case. The two matrices are time-varying and defined as

$$\begin{aligned} \boldsymbol{R}_k^{(q)} &= \begin{bmatrix} IB_k & \Phi_{k+1,k} B_{k-1} & \Phi_{k+1,k-1} B_{k-2} & \cdots & \Phi_{k+1,k-q+2} B_{k-q+1} \end{bmatrix} \\ &= \begin{bmatrix} B_k & A_k B_{k-1} & A_k A_{k-1} B_{k-2} & \cdots & A_k \dots A_{k-q+2} B_{k-q+1} \end{bmatrix}, \end{aligned} \tag{D.6}$$

and

$$\boldsymbol{O}_k^{(p)} = \begin{bmatrix} C_k I \\ C_{k+1} \Phi_{k+1,k} \\ C_{k+2} \Phi_{k+2,k} \\ \vdots \\ C_{k+p-1} \Phi_{k+p-1,k} \end{bmatrix} = \begin{bmatrix} C_k \\ C_{k+1} A_k \\ C_{k+2} A_{k+1} A_k \\ \vdots \\ C_{k+p-1} A_{k+p-2} \dots A_k \end{bmatrix}. \tag{D.7}$$

At time k, a system of order n is controllable if and only if the $n \times qr$ block controllability matrix $\boldsymbol{R}_k^{(q)}$ has rank n. Similarly, a system of order n is observable if and only if the $pm \times n$ block observability

matrix $\boldsymbol{O}_k^{(p)}$ has rank n. The reasoning behind these affirmation is the same as in the time-invariant case.

D.3 Time-Varying Coordinate Systems and Transformations

Time-varying coordinate systems and their corresponding transformations is a fundamental notion in understanding the identification process of linear time-varying systems. The transition operators involved during the development play an important role in the qualitative and quantitative analysis of linear time-varying ordinary difference equations. While previous sections already introduced similarity transforms for linear time-invariant systems, coordinate transformations for linear time-varying systems are much more challenging. This section purposely takes a step back to some basic definitions and explains in more details the fundamentals of topological spaces and homeomorphisms in the context of dynamical system theory.

Let's consider two functions $\boldsymbol{f}$ and $\boldsymbol{g}$ that represent the dynamics of a continuous time-invariant system in two different coordinate systems where $\boldsymbol{f}$ performs on the topological space E, $\boldsymbol{g}$ performs on the topological space F:

$$\forall\ \boldsymbol{x} \in E,\ \ \dot{\boldsymbol{x}} = \boldsymbol{f}(\boldsymbol{x}, t) \tag{D.8a}$$

$$\forall\ \boldsymbol{y} \in F,\ \ \dot{\boldsymbol{y}} = \boldsymbol{g}(\boldsymbol{y}, t) \tag{D.8b}$$

Since $\boldsymbol{f}$ and $\boldsymbol{g}$ represent the same dynamics but in two different coordinate systems E and F, there must exist a homeomorphism $\boldsymbol{h}$ between E and F such that

$$\forall\ \boldsymbol{x} \in E,\quad \boldsymbol{h} \circ \boldsymbol{f}(\boldsymbol{x}) = \boldsymbol{g} \circ \boldsymbol{h}(\boldsymbol{x}), \tag{D.9}$$

or equivalently,

$$\forall\ \boldsymbol{y} \in F,\quad \boldsymbol{f} \circ \boldsymbol{h}^{-1}(\boldsymbol{y}) = \boldsymbol{h}^{-1} \circ \boldsymbol{g}(\boldsymbol{y}). \tag{D.10}$$

Figure D.1 below illustrates the relationship between the operators. For continuous linear dynamics, replacing $\boldsymbol{f}$ and $\boldsymbol{g}$ by the operators A and $\tilde{A}$ respectively, Eqs. Eq. (D.8a) and Eq. (D.8b) become

$$\forall\ \boldsymbol{x} \in E,\ \ \dot{\boldsymbol{x}} = A\boldsymbol{x} \tag{D.11a}$$

$$\forall\ \boldsymbol{y} \in F,\ \ \dot{\boldsymbol{y}} = \tilde{A}\boldsymbol{y}. \tag{D.11b}$$

If the induced homeomorphism $\boldsymbol{h}$ is linear (hence replaced by the square invertible matrix T), it yields the following relation

$$\forall\ \boldsymbol{x} \in E,\quad TA\boldsymbol{x} = \tilde{A}T\boldsymbol{x} \Rightarrow A = T^{-1}\tilde{A}T, \tag{D.12}$$

with $\boldsymbol{y} = T\boldsymbol{x}$. For linear discrete-time invariant systems, a linear topological transformation is also a similarity transformation and this discussion completes what has already been presented in section. Similar to the time invariant case, time-varying discrete-time systems are also invariant to coordinate

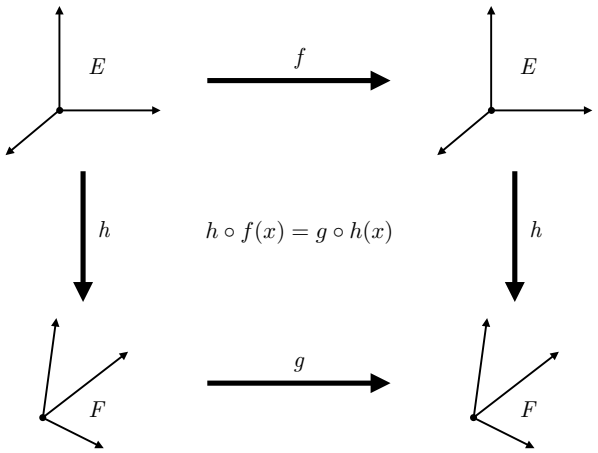


Figure D.1: An homeomorphism $\boldsymbol{h}$ between two topological spaces E and F.

(similarity) transformations; however, the isomorphic transformation between successive frames needs particular attention. Compared to invariant systems with fixed frames E and F, time-varying systems act on time-varying frames. As presented in Figure D.2, the two refrence frames E and F are now time varying and the homeomorphism between two corresponding frames are also time-varying. The

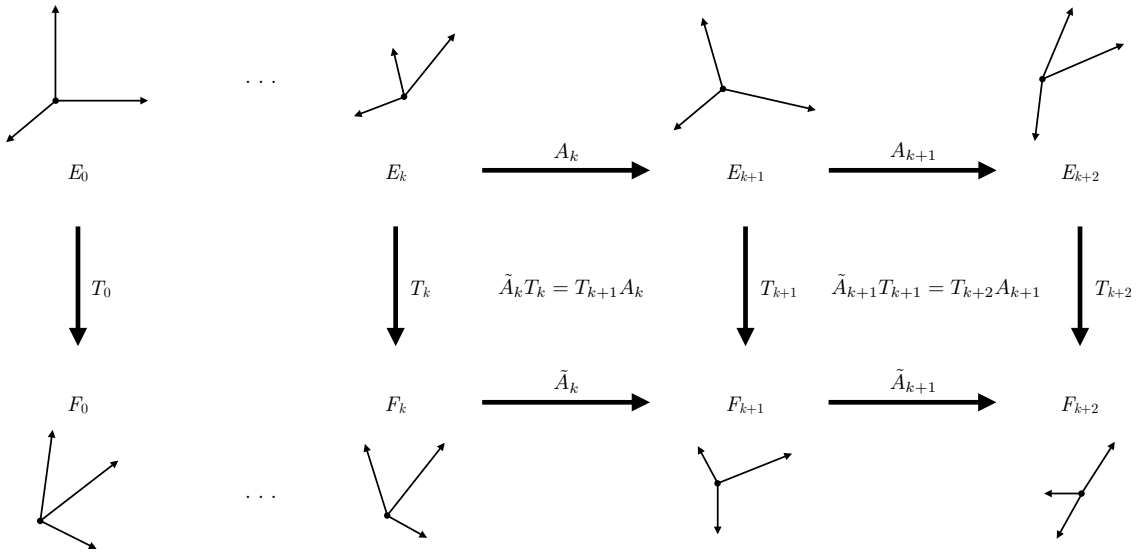


Figure D.2: Similarity transform between consecutive time-varying frames.

system representation $\{A_k, B_k, C_k, D_k\}$ is said to be topologically equivalent to the representation $\left\{\tilde{A}_k, \tilde{B}_k, \tilde{C}_k, \tilde{D}_k\right\}$ if there exists a sequence of invertible, square matrices (and not necessarily related to each other) T_k such that

$$\tilde{A}_k = T_{k+1} A_k T_k^{-1}, \tag{D.13a}$$

$$\tilde{B}_k = T_{k+1} B_k, \tag{D.13b}$$

$$\tilde{C}_k = C_k T_k^{-1}, \tag{D.13c}$$

$$\tilde{D}_k = D_k. \tag{D.13d}$$

The question, then, arises: how to calculate or estimate these sequence of matrices T_k?
First, the relationships for the controllability and observability matrices between topologically equivalent representations are given by

$$\begin{aligned}\tilde{\boldsymbol{R}}_k^{(q)} &= \left[\tilde{B}_k \quad \tilde{A}_k\tilde{B}_{k-1} \quad \cdots \quad \tilde{A}_k \ldots \tilde{A}_{k-q+2}\tilde{B}_{k-q+1}\right], \\ &= T_k^{-1}\left[B_k \quad A_kB_{k-1} \quad \cdots \quad A_k \ldots A_{k-q+2}B_{k-q+1}\right], \\ &= T_k^{-1}\boldsymbol{R}_k^{(q)},\end{aligned} \tag{D.14}$$

and

$$\tilde{\boldsymbol{O}}_k^{(p)} = \begin{bmatrix} \tilde{C}_k \\ \tilde{C}_{k+1}\tilde{A}_k \\ \vdots \\ \tilde{C}_{k+p-1}\tilde{A}_{k+p-2}\ldots\tilde{A}_k \end{bmatrix} = \begin{bmatrix} C_k \\ C_{k+1}A_k \\ \vdots \\ C_{k+p-1}A_{k+p-2}\ldots A_k \end{bmatrix} T_{k+1} = \boldsymbol{O}_k^{(p)}T_{k+1}. \tag{D.15}$$

Inverting the corresponding controllability and observability matrices provide the transformation matrices at proper time.

D.4 Time-Varying Eigensystem Realization Algorithm

D.4.1 Generalized Hankel Matrix

Similarly to the time-invariant case, Hankel matrices will play a major role during the identification process. The classical Hankel matrix becomes a generalized Hankel matrix dependent of time, populated using the generalized Markov parameters:

$$\boldsymbol{H}_k^{(p,q)} = \begin{bmatrix} h_{k,k-1} & h_{k,k-2} & \cdots & h_{k,k-q} \\ h_{k+1,k-1} & h_{k+1,k-2} & \cdots & h_{k+1,k-q} \\ \vdots & \vdots & \ddots & \vdots \\ h_{k+p-1,k-1} & h_{k+p-1,k-2} & \cdots & h_{k+p-1,k-q} \end{bmatrix} = \boldsymbol{O}_k^{(p)}\boldsymbol{R}_{k-1}^{(q)}. \tag{D.16}$$

with the parameters p and q chosen such that the generalized Hankel matrix retains the rank n, the true state dimension. Indeed, if $pm \geq n$ and $qr \geq n$, matrices $\boldsymbol{R}_k^{(q)}$ and $\boldsymbol{O}_k^{(p)}$ are of rank maximum n. If the system is controllable and observable, the block matrices $\boldsymbol{R}_k^{(q)}$ and $\boldsymbol{O}_k^{(p)}$ are of rank n and so is $\boldsymbol{H}_k^{(p,q)}$. Again, identifying the number of dominant singular values of the Hankel matrix will provide an indication about the unknown order of the reduced model to be identified. Differing ranks are possible for this generalized time-varying Hankel matrix $\boldsymbol{H}_k^{(p,q)}$ at every time step for the variable state dimension problem. However, it is assumed that the state dimension does not change with the time index and it is not difficult to see that this assumption can be relaxed, given some adjustments. We retain the

assumption owing to our focus on mechanical systems, in which the connection between physical degrees of freedom and the number of state variables allows us to hold the dimensionality of the state space fixed throughout the time interval of interest. Note that the definition of the generalized Hankel matrix above is only valid for $k \geq q$. In practical experiments, inputs cannot be applied at negative time index and generalized Markov parameters have no meaningful sense for negative indexes. In the next section, we provide the detailed calculation of a discrete time-varying realization sequence based on the singular value decomposition of $\boldsymbol{H}_k^{(p,q)}$.

D.4.2 Calculation of Discrete-Time Varying Realizations

As for the ERA in the time-invariant case, using the singular value decomposition of $\boldsymbol{H}_k^{(p,q)}$, we can write

$$\boldsymbol{H}_k^{(p,q)} = \boldsymbol{U}_k \boldsymbol{\Sigma}_k \boldsymbol{V}_k^{\mathsf{T}} = \begin{bmatrix} \boldsymbol{U}_k^{(n)} & \boldsymbol{U}_k^{(0)} \end{bmatrix} \begin{bmatrix} \boldsymbol{\Sigma}_k^{(n)} & \boldsymbol{0} \\ \boldsymbol{0} & \boldsymbol{\Sigma}_k^{(0)} \end{bmatrix} \begin{bmatrix} \boldsymbol{V}_k^{(n)\mathsf{T}} \\ \boldsymbol{V}_k^{(0)\mathsf{T}} \end{bmatrix} \tag{D.17a}$$

$$= \boldsymbol{U}_k^{(n)} \boldsymbol{\Sigma}_k^{(n)} \boldsymbol{V}_k^{(n)\mathsf{T}} + \underbrace{\boldsymbol{U}_k^{(0)} \boldsymbol{\Sigma}_k^{(0)} \boldsymbol{V}_k^{(0)\mathsf{T}}}_{\simeq \boldsymbol{0}} \tag{D.17b}$$

$$\simeq \boldsymbol{U}_k^{(n)} \boldsymbol{\Sigma}_k^{(n)} \boldsymbol{V}_k^{(n)\mathsf{T}} \tag{D.17c}$$

at a given time step k. In terms of the corresponding controllability and observability matrices,

$$\boldsymbol{H}_k^{(p,q)} = \boldsymbol{U}_k^{(n)} \boldsymbol{\Sigma}_k^{(n)} \boldsymbol{V}_k^{(n)\mathsf{T}} = \boldsymbol{O}_k^{(p)} \boldsymbol{R}_{k-1}^{(q)} \Rightarrow \begin{cases} \boldsymbol{O}_k^{(p)} &= \boldsymbol{U}_k^{(n)} \boldsymbol{\Sigma}_k^{(n)\,1/2} \\ \boldsymbol{R}_{k-1}^{(q)} &= \boldsymbol{\Sigma}_k^{(n)\,1/2} \boldsymbol{V}_k^{(n)\mathsf{T}} \end{cases} . \tag{D.18}$$

The same procedure at time step $k+1$ will lead to

$$\boldsymbol{H}_{k+1}^{(p,q)} = \boldsymbol{U}_{k+1}^{(n)} \boldsymbol{\Sigma}_{k+1}^{(n)} \boldsymbol{V}_{k+1}^{(n)\,\mathsf{T}} = \boldsymbol{O}_{k+1}^{(p)} \boldsymbol{R}_k^{(q)} \Rightarrow \begin{cases} \boldsymbol{O}_{k+1}^{(p)} &= \boldsymbol{U}_{k+1}^{(n)} \boldsymbol{\Sigma}_{k+1}^{(n)\,1/2} \\ \boldsymbol{R}_k^{(q)} &= \boldsymbol{\Sigma}_{k+1}^{(n)\,1/2} \boldsymbol{V}_{k+1}^{(n)\,\mathsf{T}} \end{cases} . \tag{D.19}$$

Considering the block shifted controllability matrix

$$\boldsymbol{R}_k^{(q)\leftarrow} = \begin{bmatrix} A_k B_{k-1} & A_k A_{k-1} B_{k-2} & \cdots & A_k \dots A_{k-q+1} B_{k-q} \end{bmatrix} \tag{D.20a}$$

$$= A_k \boldsymbol{R}_{k-1}^{(q)}, \tag{D.20b}$$

and block shifted observability matrix

$$\boldsymbol{O}_k^{(p)\uparrow} = \begin{bmatrix} C_{k+1} A_k \\ C_{k+2} A_{k+1} A_k \\ \vdots \\ C_{k+p-1} A_{k+p-2} \dots A_k \end{bmatrix} = \boldsymbol{O}_{k+1}^{(p)} A_k, \tag{D.21}$$

we get

$$\hat{A}_k = \boldsymbol{O}_{k+1}^{(p)\ \dagger} \boldsymbol{O}_k^{(p)\uparrow} = \boldsymbol{R}_k^{(q)\leftarrow} \boldsymbol{R}_{k-1}^{(q)\ \dagger} \tag{D.22}$$

as an estimate for the identified time-varying discrete system transition matrix. Moreover, the first r columns of $\boldsymbol{R}_k^{(q)}$ form an estimate for the identified control influence matrix,

$$\hat{B}_k = \boldsymbol{R}_k^{(q)} \boldsymbol{E}_r. \tag{D.23}$$

Similarly, the first m rows of $\boldsymbol{O}_k^{(p)}$ give an estimate for the identified output influence matrix is

$$\hat{C}_k = \boldsymbol{E}_m^{\intercal} \boldsymbol{O}_k^{(p)}. \tag{D.24}$$

D.4.3 Calculation of Discrete-Time Varying Realizations for the First Few Time Steps

As explained before, the definition of the generalized Hankel matrix is only valid for $k \geq q$. The methodology detailed in the previous section can only be employed once a full rank Hankel matrix can be populated. This section presents a method for computing the first few time step models using an additional set of experimental data, the free response experiments. The output data of N free response experiments (also known as the zero input response) are given by

$$\tilde{\boldsymbol{H}}_k^{(p,N)} = \begin{bmatrix} \boldsymbol{y}_k^{\#1} & \boldsymbol{y}_k^{\#2} & \cdots & \boldsymbol{y}_k^{\#N} \\ \boldsymbol{y}_{k+1}^{\#1} & \boldsymbol{y}_{k+1}^{\#2} & \cdots & \boldsymbol{y}_{k+1}^{\#N} \\ \vdots & \vdots & \ddots & \vdots \\ \boldsymbol{y}_{k+p-1}^{\#1} & \boldsymbol{y}_{k+p-1}^{\#2} & \cdots & \boldsymbol{y}_{k+p-1}^{\#N} \end{bmatrix} = \boldsymbol{O}_k^{(p)} \boldsymbol{X}_k^{(N)}, \tag{D.25}$$

where $\boldsymbol{O}_k^{(p)}$ is the observability matrix

$$\boldsymbol{O}_k^{(p)} = \begin{bmatrix} C_k \\ C_{k+1}A_k \\ C_{k+2}A_{k+1}A_k \\ \vdots \\ C_{k+p-1}A_{k+p-2}\ldots A_k \end{bmatrix}, \tag{D.26}$$

and $\boldsymbol{X}_k^{(N)}$ is a state variable ensemble at time k:

$$\boldsymbol{X}_k^{(N)} = \begin{bmatrix} \Phi_{k,0}\boldsymbol{x}_0^{\#1} & \Phi_{k,0}\boldsymbol{x}_0^{\#2} & \cdots & \Phi_{k,0}\boldsymbol{x}_0^{\#N} \end{bmatrix} \in \mathbb{R}^{n \times N}. \tag{D.27}$$

As for the procedure when $k \geq q$, using the singular value decomposition of $\tilde{\boldsymbol{H}}_k^{(p,N)}$, we can write

$$\tilde{\boldsymbol{H}}_k^{(p,N)} = \boldsymbol{U}_k \boldsymbol{\Sigma}_k \boldsymbol{V}_k^{\mathsf{T}} = \begin{bmatrix} \boldsymbol{U}_k^{(n)} & \boldsymbol{U}_k^{(0)} \end{bmatrix} \begin{bmatrix} \boldsymbol{\Sigma}_k^{(n)} & \boldsymbol{0} \\ \boldsymbol{0} & \boldsymbol{\Sigma}_k^{(0)} \end{bmatrix} \begin{bmatrix} \boldsymbol{V}_k^{(n)\mathsf{T}} \\ \boldsymbol{V}_k^{(0)\mathsf{T}} \end{bmatrix} \tag{D.28a}$$

$$= \boldsymbol{U}_k^{(n)} \boldsymbol{\Sigma}_k^{(n)} \boldsymbol{V}_k^{(n)\mathsf{T}} + \underbrace{\boldsymbol{U}_k^{(0)} \boldsymbol{\Sigma}_k^{(0)} \boldsymbol{V}_k^{(0)\mathsf{T}}}_{\simeq \boldsymbol{0}} \tag{D.28b}$$

$$\simeq \boldsymbol{U}_k^{(n)} \boldsymbol{\Sigma}_k^{(n)} \boldsymbol{V}_k^{(n)\mathsf{T}} \tag{D.28c}$$

at a given time step k. In terms of the corresponding observability and state variable ensemble matrices,

$$\tilde{\boldsymbol{H}}_k^{(p,N)} = \boldsymbol{U}_k^{(n)} \boldsymbol{\Sigma}_k^{(n)} \boldsymbol{V}_k^{(n)\mathsf{T}} = \boldsymbol{O}_k^{(p)} \boldsymbol{X}_k^{(N)} \Rightarrow \begin{cases} \boldsymbol{O}_k^{(p)} = \boldsymbol{U}_k^{(n)} \boldsymbol{\Sigma}_k^{(n)\,1/2} \\ \boldsymbol{X}_k^{(N)} = \boldsymbol{\Sigma}_k^{(n)\,1/2} \boldsymbol{V}_k^{(n)\mathsf{T}} \end{cases} . \tag{D.29}$$

The same procedure at time step $k+1$ will lead to

$$\tilde{\boldsymbol{H}}_{k+1}^{(p,N)} = \boldsymbol{U}_{k+1}^{(n)} \boldsymbol{\Sigma}_{k+1}^{(n)} \boldsymbol{V}_{k+1}^{(n)\,\mathsf{T}} = \boldsymbol{O}_{k+1}^{(p)} \boldsymbol{X}_{k+1}^{(N)} \Rightarrow \begin{cases} \boldsymbol{O}_{k+1}^{(p)} = \boldsymbol{U}_{k+1}^{(n)} \boldsymbol{\Sigma}_{k+1}^{(n)\,1/2} \\ \boldsymbol{X}_{k+1}^{(N)} = \boldsymbol{\Sigma}_{k+1}^{(n)\,1/2} \boldsymbol{V}_{k+1}^{(n)\,\mathsf{T}} \end{cases} . \tag{D.30}$$

Note that the state variable ensemble matrix $\boldsymbol{X}_{k+1}^{(N)}$ at time $k+1$ is related to the state variable ensemble matrix $\boldsymbol{X}_k^{(N)}$ at time k by

$$\boldsymbol{X}_{k+1}^{(N)} = A_k \boldsymbol{X}_k^{(N)} \tag{D.31}$$

which leads to the estimate

$$\hat{A}_k = \boldsymbol{X}_{k+1}^{(N)} \boldsymbol{X}_k^{(N)\,-1} \tag{D.32}$$

for the time-varying state matrix at times $k = 0, 1, \ldots, q-1$. The calculation of the corresponding $\hat{C}_k$ is accomplished by setting

$$\hat{C}_k = \boldsymbol{E}_m^{\mathsf{T}} \boldsymbol{O}_k^{(p)}. \tag{D.33}$$

Finally, the estimate for $\hat{B}_k$ is calculated by forming the partial Hankel matrices for the first few time steps ($k = 0, 1, \ldots, q-1$):

$$\boldsymbol{H}_{k+1}^{(p,1)} = \begin{bmatrix} h_{k+1,k} \\ h_{k+2,k} \\ \vdots \\ h_{k+p,k} \end{bmatrix} = \begin{bmatrix} C_{k+1} B_k \\ C_{k+2} A_{k+1} B_k \\ \vdots \\ C_{k+p} A_{k+p-1} \ldots A_{k+1} B_k \end{bmatrix} = \boldsymbol{O}_{k+1}^{(p)} B_k, \tag{D.34}$$

leading to

$$\hat{B}_k = \boldsymbol{O}_{k+1}^{(p)\,\dagger} \boldsymbol{H}_{k+1}^{(p,1)}. \tag{D.35}$$

However, the model matrices determined Eq. (D.32), Eq. (D.33) and Eq. (D.35) are of little use in practice. This is because of the fact that the first few models developed in this manner ($k < q$) are in a totally different coordinate system, derived from the free response singular value decomposition. Models for $k \geq q$ are in the coordinate system derived from the forced response singular value decomposition. Hence, one cannot use the models thus developed in state propagation because they have a jump discontinuity at the time step $k = q$ in their coordinate systems. The first option to alleviate this issue is to apply the coordinate transformation theory developed in the previous section. The second option is to estimate the state at time $k = q$. This approach is explained in the next section.

D.4.4 Estimation of Initial Conditions from Identified Time-Varying Model

Writing the input and output from a general k^{th} time step, for p more time steps, one obtain a set of equations that can be written in a matrix form as

$$\bar{\boldsymbol{y}} = \boldsymbol{O}_k^{(p)} \boldsymbol{x}_k + \boldsymbol{\Delta}_k \bar{\boldsymbol{u}} \tag{D.36}$$

with

$$\bar{\boldsymbol{y}} = \begin{bmatrix} \boldsymbol{y}_k \\ \boldsymbol{y}_{k+1} \\ \vdots \\ \boldsymbol{y}_{k+p-1} \end{bmatrix}, \qquad \bar{\boldsymbol{u}} = \begin{bmatrix} \boldsymbol{u}_k \\ \boldsymbol{u}_{k+1} \\ \vdots \\ \boldsymbol{u}_{k+p-1} \end{bmatrix}, \tag{D.37}$$

and

$$\boldsymbol{\Delta}_k = \begin{bmatrix} D_k & & & \\ C_{k+1}B_k & D_{k+1} & & \\ \vdots & \vdots & \ddots & \\ C_{k+p-1}A_{k+p-2}\dots B_k & C_{k+p-1}A_{k+p-2}\dots B_{k+1} & \cdots & D_{k+p-1} \end{bmatrix}. \tag{D.38}$$

Eq. (D.36) can be solved using the least-squares solution:

$$\boldsymbol{x}_k = \boldsymbol{O}_k^{(p)\,\dagger} \left[\bar{\boldsymbol{y}} - \boldsymbol{\Delta}_k \bar{\boldsymbol{u}} \right]. \tag{D.39}$$

Appendix E
Time-Varying Observer Kalman Identification Algorithm (TVOKID)

E.1 Introduction

In stark contrast to the time invariant (shift invariant) systems, the generalized Markov parameters determine the pulse response characteristics of the true plant in a much more general fashion. Note that the number of independent degrees of freedom to describe the input–output relationship increases tremendously for the case of time-varying systems, as the response of the system ($h_{k,i}$) not only depends upon the time difference from the applied input ($\boldsymbol{u}(i)$) but also on the time instant i at which the said input is applied. Therefore, a practical method is needed to calculate the generalized Markov parameters without resorting to a high dimensional calculation. This calculation becomes further compounded in systems in which the stability of the origin cannot be ascertained, because the number of potentially significant generalized Markov parameters grows rapidly. In other words, in the case of the problems with unstable origins, the output at every time step in the time-varying case depends on the linear combinations of the pulse response functions and all the inputs applied until that instant. Therefore, the number of unknowns increase by $m \times r$ for each time step in the model sequence and, consequently, the analyst is required to perform more experiments if a refined discrete-time model is sought.

E.2 Classical Formulation

The input-output relationship in terms of thegeneralized Markov parameters is

$$\boldsymbol{y}_k = C_k \Phi_{k,0} \boldsymbol{x}_0 + \sum_{i=0}^{k-1} h_{k,i} \boldsymbol{u}_i + D_k \boldsymbol{u}_k. \tag{E.1}$$

In a matrix form, Eq. (E.1) is written as

$$\boldsymbol{y}_k = C_k \Phi_{k,0} \boldsymbol{x}(0) + \boldsymbol{Y} \tilde{\boldsymbol{u}} \tag{E.2}$$

with

$$\boldsymbol{Y} = \begin{bmatrix} D_k & C_k B_{k-1} & C_k A_{k-1} B_{k-2} & \cdots & C_k A_{k-1} \ldots A_1 B_0 \end{bmatrix}, \tag{E.3a}$$

$$\bar{\boldsymbol{u}} = \begin{bmatrix} \boldsymbol{u}_k \\ \boldsymbol{u}_{k-1} \\ \boldsymbol{u}_{k-2} \\ \vdots \\ \boldsymbol{u}_0 \end{bmatrix}. \tag{E.3b}$$

As was pointed out, such a relationship between the input and output leads to a problem that increases by $m \times r$ parameters for every time step considered. Thus, it becomes difficult to compute the increasing number of unknown parameters. In the special case of systems for which the open loop is asymptotically stable, this is not a problem. However, frequently, one tries to use identification in problems that do not have a stable origin for control and estimation purposes and may be required to compute time-varying model sequences with higher resolution. Again, rather than identifying the generalized system Markov parameters, one can use an asymptotically stable observer to remedy this problem of unbounded growth in the number of experiments. In addition, the algorithm presented in this section gives an estimate on the minimum number of experiments needed to perform identification and/or recovery of all the Markov parameters of interest until the specific time instant considered: a central result is to make the number of repeated experiments independent of the desired resolution of the model.

E.3 State-Space Observer Model

Following the work from time-invariant systems, consider the use of a time-varying output–feedback-style gain sequence in the difference model Eq. (D.1a), producing

$$\begin{aligned} \boldsymbol{x}_{k+1} &= A_k \boldsymbol{x}_k + B_k \boldsymbol{u}_k + G_k \boldsymbol{y}_k - G_k \boldsymbol{y}_k && \text{(E.4a)} \\ &= (A_k + G_k C_k)\, \boldsymbol{x}_k + (B_k + G_k D_k)\, \boldsymbol{u}_k - G_k \boldsymbol{y}_k && \text{(E.4b)} \\ &= \bar{A}_k \boldsymbol{x}_k + \bar{B}_k \boldsymbol{v}_k && \text{(E.4c)} \end{aligned}$$

where

$$\bar{A}_k = A_k + G_k C_k, \tag{E.5a}$$

$$\bar{B}_k = \begin{bmatrix} B_k + G_k D_k & -G_k \end{bmatrix}, \tag{E.5b}$$

$$\boldsymbol{v}_k = \begin{bmatrix} \boldsymbol{u}_k \\ \boldsymbol{y}_k \end{bmatrix}, \tag{E.5c}$$

and G_k is an arbitrary matrix chosen to make the matrix $\bar{A}_k$ as stable as desired. The general I/O relationship is

$$\boldsymbol{y}_k = C_k \bar{A}_{k-1} \dots \bar{A}_0 \boldsymbol{x}_0 + \sum_{i=1}^{k} \bar{h}_{k,k-i} \boldsymbol{v}_{k-i} + D_k \boldsymbol{u}_k. \tag{E.6}$$

and the generalized Markov parameters of this observer system are referred as *generalized observer Markov parameters*, constituted of two block components similar to the linear time-invariant case, shown in the partitions to be

$$\bar{h}_{k,i} = C_k \bar{A}_{k-1} \dots \bar{A}_{i+1} \bar{B}_i \tag{E.7a}$$

$$= \begin{bmatrix} C_k \bar{A}_{k-1} \dots \bar{A}_{i+1} (B_i + G_i D_i) & -C_k \bar{A}_{k-1} \dots \bar{A}_{i+1} G_i \end{bmatrix} \tag{E.7b}$$

$$= \begin{bmatrix} \bar{h}^{(1)}_{k,i} & -\bar{h}^{(2)}_{k,i} \end{bmatrix}. \tag{E.7c}$$

The closed loop, thus constructed, is now forced to have an asymptotically stable origin. The first step involved in achieving this goal of closed-loop asymptotic stability is to choose a number of time steps l_k (variable each time, in general) sufficiently large, so that the output of the plant at time $k + l_k$ strictly depends only on the $l_k + 1$ previously augmented control inputs $\{\boldsymbol{v}_{k+i-1}\}_{i=1\dots l_k}$ and $\boldsymbol{u}_{k+l_k}$ and independent of the state at every time k. Therefore, one can write

$$\boldsymbol{y}_{k+l_k} = C_{k+l_k} \bar{A}_{k+l_k-1} \dots \bar{A}_k \boldsymbol{x}_k + \sum_{i=1}^{l_k} \bar{h}_{k+l_k,k+i-1} \boldsymbol{v}_{k+i-1} + D_{k+l_k} \boldsymbol{u}_{k+l_k} \tag{E.8a}$$

$$\simeq \sum_{i=1}^{l_k} \bar{h}_{k+l_k,k+i-1} \boldsymbol{v}_{k+i-1} + D_{k+l_k} \boldsymbol{u}_{k+l_k} \tag{E.8b}$$

where

$$C_{k+l_k} \bar{A}_{k+l_k-1} \dots \bar{A}_k \boldsymbol{x}_k \simeq \boldsymbol{0}, \tag{E.9}$$

with exact equality in the absence of measurement noise. Note that the order l_k of the model can change with time. In using this I/O relationship instead of the exact relationship given in Eq. Eq. (E.1), damping is introduced into the closed loop. For simplicity and ease in implementation and understanding, the variable order l_k remains fixed and minimum (time-varying deadbeat) at each time step and set to $l_{\min}$, where $l_{\min}$ is the smallest positive integer such that $l_{\min} > mn$.

The output at time k ($k \geq l_{\min}$) is written as

$$\boldsymbol{y}_k = \tilde{\bar{\boldsymbol{Y}}}_k \bar{\boldsymbol{v}}_k \tag{E.10}$$

where

$$\tilde{\bar{\boldsymbol{Y}}}_k = \begin{bmatrix} D_k & \bar{h}_{k,k-1} & \bar{h}_{k,k-2} & \cdots & \bar{h}_{k,k-l_{\min}} \end{bmatrix} \tag{E.11a}$$

$$\bar{\boldsymbol{v}}_k = \begin{bmatrix} \boldsymbol{u}_k \\ \boldsymbol{v}_{k-1} \\ \boldsymbol{v}_{k-2} \\ \vdots \\ \boldsymbol{v}_{k-l_{\min}} \end{bmatrix} \tag{E.11b}$$

This represents a set of m equations for $m \times ((r+m)l_{\min} + r)$ unknowns. In this case, the number of unknowns remains constant which makes the number of repeated experiments required to compute these parameters now constant: it is observed that a minimum of

$$N_{\min} = (r+m)l_{\min} + r \tag{E.12}$$

experiments are necessary to determine the observer Markov parameters uniquely. From the developments of the subsequent sections, this is the minimum number of repeated experiments one should perform in order to realize the time-varying system models desired from the TVERA. Eq. Eq. (E.10), with N repeated experiments yields:

$$\bar{\boldsymbol{y}}_k = \widetilde{\bar{\boldsymbol{Y}}}_k \bar{\boldsymbol{V}}_k, \tag{E.13}$$

where

$$\bar{\boldsymbol{y}}_k = \begin{bmatrix} \boldsymbol{y}_k & \boldsymbol{y}_1 & \boldsymbol{y}_2 & \cdots & \boldsymbol{y}_{l-1} \end{bmatrix}, \tag{E.14a}$$

$$\widetilde{\bar{\boldsymbol{Y}}}_k = \begin{bmatrix} D_k & \bar{h}_{k,k-1} & \bar{h}_{k,k-2} & \cdots & \bar{h}_{k,k-l_{\min}} \end{bmatrix}, \tag{E.14b}$$

$$\bar{\boldsymbol{V}}_k = \begin{bmatrix} \boldsymbol{u}_k^{\#1} & \boldsymbol{u}_k^{\#2} & \cdots & \boldsymbol{u}_k^{\#N} \\ \boldsymbol{v}_{k-1}^{\#1} & \boldsymbol{v}_{k-1}^{\#2} & \cdots & \boldsymbol{v}_{k-1}^{\#N} \\ \boldsymbol{v}_{k-2}^{\#1} & \boldsymbol{v}_{k-2}^{\#2} & \cdots & \boldsymbol{v}_{k-2}^{\#N} \\ \vdots & \vdots & \ddots & \vdots \\ \boldsymbol{v}_{k-l_{\min}}^{\#1} & \boldsymbol{v}_{k-l_{\min}}^{\#2} & \cdots & \boldsymbol{v}_{k-l_{\min}}^{\#N} \end{bmatrix}. \tag{E.14c}$$

Therefore, the least-squares solution for the generalized observer Markov parameters is given for each time step as

$$\widetilde{\bar{\boldsymbol{Y}}}_k = \bar{\boldsymbol{y}}_k \bar{\boldsymbol{V}}_k^{\dagger}. \tag{E.15}$$

E.4 Computation of Markov Parameters from Generalized Markov Parameters

Considering the definition of the generalized observer Markov parameters, one can write

$$\bar{h}_{k,k-1} = C_k \bar{B}_{k-1} \tag{E.16a}$$

$$= C_k \begin{bmatrix} B_{k-1} + G_{k-1}D_{k-1} & -G_{k-1} \end{bmatrix} \tag{E.16b}$$

$$= \begin{bmatrix} \bar{h}^{(1)}_{k,k-1} & -\bar{h}^{(2)}_{k,k-1} \end{bmatrix}. \tag{E.16c}$$

where the superscripts are used to. The following manipulation relates the generalized Markov parameter $h_{k,k-1}$ to the components of the generalized observer Markov parameter $\bar{h}_{k,k-1}$:

$$h_{k,k-1} = C_k B_{k-1} = \bar{h}^{(1)}_{k,k-1} - \bar{h}^{(2)}_{k,k-1} D_{k-1}. \tag{E.17}$$

A similar expression for Markov parameters with two time-steps between them yields

$$\bar{h}^{(1)}_{k,k-2} - \bar{h}^{(2)}_{k,k-2} D_{k-2} = C_k \bar{A}_{k-1} \bar{B}_{k-2} - C_k \bar{A}_{k-1} G_{k-2} D_{k-2} \tag{E.18a}$$

$$= C_k \bar{A}_{k-1} \left(B_{k-2} + G_{k-2} D_{k-2} \right) - C_k \bar{A}_{k-1} G_{k-2} D_{k-2} \tag{E.18b}$$

$$= C_k \bar{A}_{k-1} B_{k-2} \tag{E.18c}$$

$$= C_k \left(A_{k-1} + G_{k-1} C_{k-1} \right) B_{k-2} \tag{E.18d}$$

$$= C_k A_{k-1} B_{k-2} + \bar{h}^{(2)}_{k,k-1} C_{k-1} B_{k-2} \tag{E.18e}$$

$$= h_{k,k-2} + \bar{h}^{(2)}_{k,k-1} h_{k-1,k-2}. \tag{E.18f}$$

This manipulation leads to an expression for the generalized system Markov parameter $h_{k,k-2}$ to be calculated from generalized observer Markov parameters at time step k and the generalized system Markov parameters at previous time steps. This recursive relationship holds in general and enables the calculation of the generalized system Markov parameters from the generalized observer Markov parameters. Figure E.1 displays an overview of the TVOKID algorithm.

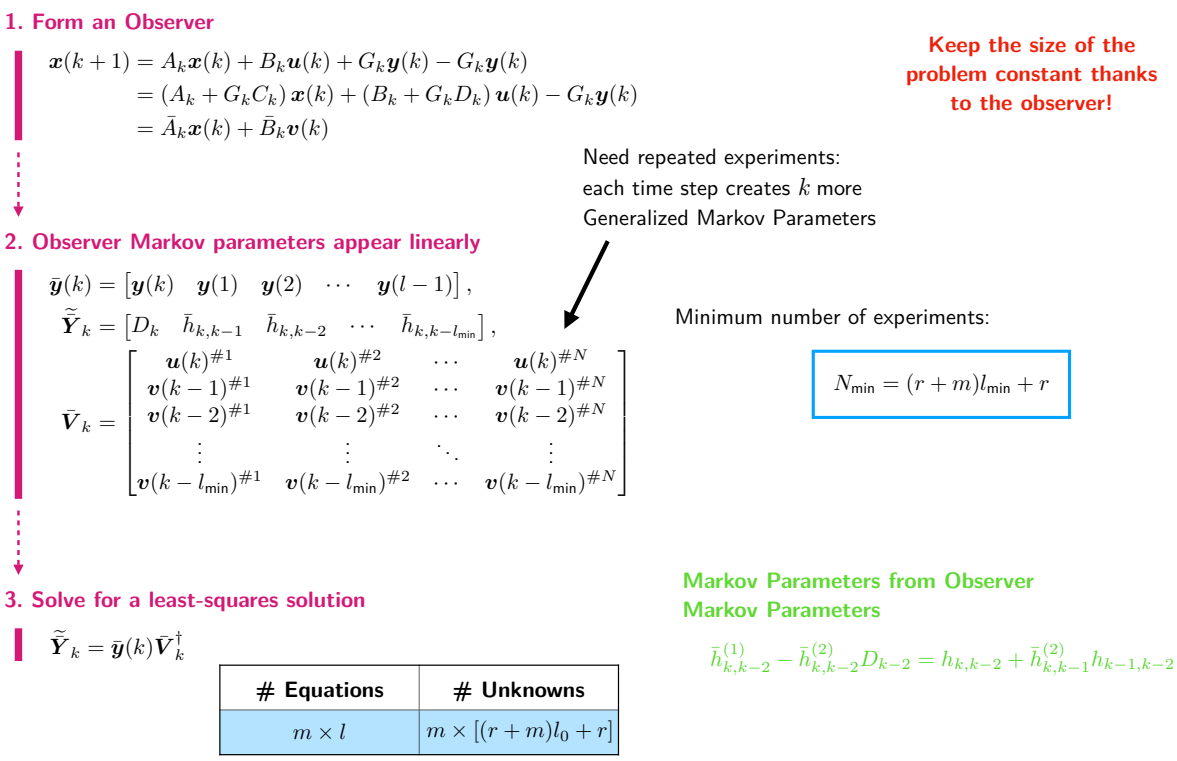


Figure E.1: Overview of the TVOKID algorithm

Bibliography

[1] Juang, J.-N. (1994) *Applied System Identification*, Prentice Hall, Englewoods Cliffs, NJ, United States.

[2] Nelles, O. (2000) *Nonlinear System Identification: From Classical Approaches to Neural Networks and Fuzzy Models*, Springer Science, Berlin, Germany.

[3] Gilbert, E. G. (1963) "Controllability and Observability in Multivariable Control Systems," *Journal of the Society for Industrial and Applied Mathematics, Series A Control*, **1**(2), pp. 128–151.

[4] Kálmán, R. E. (1963) "Mathematical Description of Linear Dynamical Systems," *Journal of the Society for Industrial and Applied Mathematics, Series A Control*, **1**(2), pp. 152–192.

[5] Ho, B. L. and R. E. Kálmán (1966) "Effective construction of linear state-variable models from input/output functions," *Regelungstechnik*, **14**(12), pp. 545–548.

[6] Alvin, K. F. and K. C. Park (1994) "Second Order Structural Identification Procedure via State-Space Based System Identification," *AIAA Journal*, **32**(2), pp. 397–406.

[7] Anderson, B. D. O. and R. E. Skelton (1988) "The Generation of all q-Markov Covers," *IEEE Transactions on Circuits and Systems*, **46**, pp. 351–356.

[8] Juang, J.-N., J. E. Cooper, and J. R. Wright (1988) "An Eigensystem Realization Algorithm Using Data Correlation (ERA/DC) for Modal Parameter Identification," *Control Theory and Advanced Technology*, **4**(1), pp. 5–14.

[9] Liu, K., R. E. Skelton, and J. P. Sharkey (1994) "Modeling Hubble Space Telescope flight data by Q-Markov cover identification," *Journal of Guidance, Control, and Dynamics*, **17**(2), pp. 250–256.

[10] Skelton, R. E. and G. Shi (1996) "Iterative identification and control using a weighted q-Markov cover with measurement noise," *Signal Processing*, **52**(2), pp. 217–234.

[11] Juang, J.-N. (2009) "Generalized Bilinear System Identification," *The Journal of the Astronautical Sciences*, **57**(1 and 2), pp. 261–273.

[12] ——— (2005) "Continuous-Time Bilinear System Identification," *Nonlinear Dynamics*, **39**, pp. 79–94.

[13] Juang, J.-N. and R. S. Pappa (1985) "An Eigensystem Realization Algorithm (ERA) for Modal Parameter Identification and Model Reduction," *Journal of Guidance, Control, and Dynamics*, **8**(5), pp. 620–627.

[14] ——— (1986) "Effects of noise on modal parameters identified by the Eigensystem Realization Algorithm," *Journal of Guidance, Control, and Dynamics*, **9**(3), pp. 294–303.

[15] Juang, J.-N., M. Phan, L. G. Horta, and R. W. Longman (1993) "Identification of Observer/Kalman Filter Markov Parameters: Theory and Experiments," *Journal of Guidance, Control, and Dynamics*, **16**(2), pp. 320–329.

[16] Juang, J.-N. and R. W. Longman (1999) *Optimized System Identification*, *Technical report 1999-209711*, National Aeronautics and Space Administration.

[17] Juang, J.-N., L. G. Horta, W. K. Belvin, J. Sharkey, and F. H. Bauer (1993) "An application of the Observer/Kalman Filter Identification (OKID) technique to Hubble flight data," .

[18] Phan, M., L. G. Horta, J.-N. Juang, and R. W. Longman (1993) "Linear System Identification via an Asymptotically Stable Observer," *Journal of Optimization Theory and Applications*, **79**(1), pp. 59–86.

[19] Chen, C. W., G. Lee, and J.-N. Juang (1992) "Several Recursive Techniques for Observer/Kalman Filter System Identification From Data," 92-4386, Hilton Head Island, SC, U.S.A.

[20] Schmid, P. J. (2010) "Dynamic mode decomposition of numerical and experimental data," *Journal of Fluid Mechanics*, **656**, pp. 5–28.

[21] Tu, J. H., C. W. Rowley, D. M. Luchtenburg, S. L. Brunton, and J. N. Kutz (2014) "On Dynamic Mode Decomposition: Theory and Applications," *Journal of Computational Dynamics*, **1**(2), pp. 391–421.

[22] Proctor, J. L. and P. A. Eckhoff (2015) "Discovering dynamic patterns from infectious disease data using dynamic mode decomposition," *International Health*, **7**, pp. 139–45.

[23] Kutz, J. N., S. L. Brunton, B. W. Brunton, and J. L. Proctor (2016) *Dynamic Mode Decomposition: Data-Driven Modeling of Complex Systems*, SIAM.

[24] Shokoohi, S. and L. M. Silverman (1987) "Identification and Model Reduction of Time-varying Discrete-time Systems," *Automatica*, **23**(4), pp. 509–521.

[25] Dewilde, P. and A. J. Van Der Veen (1998) *Time Varying Systems and Computations*, Kluwer Academic Publisher.

[26] Verhaegen, M. (1991) *Identification of Time-Varying State Space Models from Input-Output Data*, Workshop on Advanced Algorithms and their Realization, Bonas.

[27] Verhaegen, M. and X. Yu (1995) "A Class of Subspace Model Identification Algorithms to Identify Periodically and Arbitrarily Time Varying Systems," *Automatica*, **31**(2), pp. 201–216.

[28] Majji, M., J.-N. Juang, and J. L. Junkins (2010) "Time-Varying Eigensystem Realization Algorithm," *Journal of Guidance, Control, and Dynamics*, **33**(1), pp. 13–28.

[29] ——— (2010) "Observer/Kalman-Filter Time-Varying System Identification," *Journal of Guidance, Control, and Dynamics*, **33**(3), pp. 887–900.

[30] Rugh, W. J. (1981) *Nonlinear System Theory: The Volterra/Wiener Approach*, The Johns Hopkins University Press, Baltimore, MD, United States.

[31] Kvaternik, R. G. and W. A. Silva (2008) *A Computational Procedure for Identifying Bilinear Representations of Nonlinear Systems Using Volterra Kernels*, *Technical report 2008-215320*, National Aeronautics and Space Administration, Langley Research Center, Hampton VA.

[32] Singla, P. and J. L. Junkins (2008) *Multi-Resolution Methods for Modeling and Control of Dynamical Systems*, Chapman and Hall CRC Applied Mathematics and Nonlinear Science, London, United Kingdom.

[33] Crutchfield, J. P. and B. S. McNamara (1987) "Equations of Motion from a Data Series," *Complex Systems*, **1**, pp. 417–452.

[34] Narendra, K. S. and K. Parthasarathy (1990) "Identification and Control of Dynamical Systems using Neural Networks," *IEEE Transactions on Neural Networks*, **1**(1), pp. 4–27.

[35] Haykin, S. (1998) *Neural Networks: A Comprehensive Foundation*, Prentice Hall.

[36] ——— (2009) *Neural Networks and Learning Machines*, Prentice Hall.

[37] Daniels, B. C. and I. Nemenman (2015) "Efficient Inference of Parsimonious Phenomenological Models of Cellular Dynamics Using S-Systems and Alternating Regression," *PLoS ONE*, **10**(3), p. e0119821.

[38] ——— (2015) "Automated adaptive inference of phenomenological dynamical models," *Nature Communications*, **6**(1), p. 8133.

[39] Proctor, J. L., S. L. Brunton, B. W. Brunton, and J. N. Kutz (2014) "Exploiting sparsity and equation-free architectures in complex systems," *The European Physical Journal Special Topics*, **223**, pp. 2665–2684.

[40] Brunton, S. L., J. L. Proctor, and J. N. Kutz (2016) "Discovering governing equations from data: Sparse identification of nonlinear dynamical systems," *Proceedings of the National Academy of Sciences*, **113**(15), pp. 3932–3937.

[41] AlMomani, A. A. R., J. Sun, and E. Bollt (2020) "How Entropic Regression Beats the Outliers Problem in Nonlinear System Identification," *Chaos: An Interdisciplinary Journal of Nonlinear Science*, **30**(1), p. 013107.

[42] Zhang, S. and G. Lin (2019) "Robust subsampling-based sparse Bayesian inference to tackle four challenges (large noise, outliers, data integration, and extrapolation) in the discovery of physical laws from data," *arXiv:1907.07788v2*, pp. 80309–0526.

[43] ——— (2021) "SubTSBR to tackle high noise and outliers for data-driven discovery of differential equations," *Journal of Computational Physics*, **428**, p. 109962.

[44] Pantazis, Y. and I. Tsamardinos (2019) "A unified approach for sparse dynamical system inference from temporal measurements," *Bioinformatics*, **35**(18), pp. 3387–3396.

[45] Schaeffer, H. and S. G. McCalla (2017) "Sparse model selection via integral terms," *Physical Review E*, **96**(2), p. 023302.

[46] Holmes, P. and J. Guckenheimer (1983) *Nonlinear oscillations, dynamical systems, and bifurcations of vector fields*, vol. 42 of *Applied Mathematical Sciences*, Springer-Verlag, Berlin, Heidelberg.

[47] Koopman, B. O. (1931) "Hamiltonian Systems and Transformation in Hilbert Space," *Proceedings of the National Academy of Sciences*, **17**(5), pp. 315–318.

[48] Koopman, B. O. and J.-V. Neumann (1932) "Dynamical Systems of Continuous Spectra," *Proceedings of the National Academy of Sciences*, **18**(3), pp. 255–263.

[49] Jin, C., P. Singla, and T. Singh (2010) "A Multi-Resolution Approach with Sparseness Property for Input-Output Approximation," 2010-8156, Toronto, Ontario, Canada.

[50] of the 2010 American Control Conference, P. (ed.) (2010) *A multi-resolution approach for tumor motion modeling*, IEEE, Manhattan, New York, United States.

[51] Schmidt, M. D., R. R. Vallabhajosyula, J. W. Jenkins, J. E. Hood, A. S. Soni, J. P. Wikswo, and H. Lipson (2011) "Automated refinement and inference of analytical models for metabolic networks," *Physical biology*, **8**(5), p. 055011.

[52] Guého, D., P. Singla, and R. G. Melton (2020) "Data-driven sparse approximation for the identification of nonlinear dynamical systems: applications in astrodynamics," in *Spaceflight Mechanics 2020*, Advances in the Astronautical Sciences, Univelt Inc., Escondido, CA.

[53] Guého, D., P. Singla, M. Majji, and R. G. Melton (2022) "A Filtered Integral Formulation of the Sparse Model Identification Problem," *Journal of Guidance, Control, and Dynamics*, **45**(2), pp. 232–247.

[54] Mezić, I. and A. Banaszuk (2004) "Comparison of systems with complex behavior," *Physica D: Nonlinear Phenomena*, **197**, pp. 101–133.

[55] Mezić, I. (2005) "Spectral Properties of Dynamical Systems, Model Reduction and Decompositions," *Nonlinear Dynamics*, **41**, pp. 309–325.

[56] Korda, M. and I. Mezic (2018) "Linear predictors for nonlinear dynamical systems: Koopman operator meets model predictive control," *Automatica*, **93**, pp. 149–160.

[57] Brunton, S. L., B. W. Brunton, J. L. Proctor, and J. N. Kutz (2016) "Koopman Invariant subspaces and Finite Linear Representations of Nonlinear Dynamical Systems for Control," *PLoS ONE*, **11**(1), p. e0150171.

[58] Williams, M. O., I. G. Kevrekidis, and C. W. Rowley (2015) "A Data–Driven Approximation of the Koopman Operator: Extending Dynamic Mode Decomposition," *Journal of Nonlinear Science*, **25**, pp. 1307–1346.

[59] Brunton, S. L., B. W. Brunton, J. L. Proctor, E. Kaiser, and J. N. Kutz (2017) "Chaos as an intermittently forced linear system," *Nature Communications*, **8**(19).

[60] Takeishi, N., Y. Kawahara, and T. Yairi (2017) "Learning Koopman Invariant Subspaces for Dynamic Mode Decomposition," *Advances in Neural Information Processing Systems*, pp. 1130–1140.

[61] Zhang, H., C. W. Rowley, E. A. Deem, and L. N. Cattafesta (2019) "Online Dynamic Mode Decomposition for Time-Varying Systems," *SIAM Journal on Applied Dynamical Systems*, **18**(3), pp. 1586–1609.

[62] Carleman, T. (1932) "Application de la théorie des équations intégrales linéaires aux systèmes d'équations différentielles non linéaires," *Acta Mathematica*, **59**(1), pp. 63–87.

[63] ——— (1933) "Sur la théorie de l'équation intégrodifférentielle de Boltzmann," *Acta Mathematica*, **60**(1), pp. 91–146.

[64] Banks, S. P. (1992) "Infinite-dimensional Carleman linearization, the Lie series and optimal control of non-linear partial differential equations," *International Journal of Systems Science*, **23**(5), pp. 663–675.

[65] Kowalski, K. and W.-H. Steeb (1991) *Nonlinear dynamical systems and Carleman linearization*, World Scientific.

[66] Steeb, W.-H. and F. Wilhelm (1980) "Non-Linear Autonomous Systems of Differential Equations and Carleman Linearization Procedure," *Journal of Mathematical Analysis and Applications*, **77**(2), pp. 601–611.

[67] Svoronos, S., D. Papageorgiou, and C. Tsiligiannis (1994) "Discretization of Nonlinear Control Systems via the Carleman Linearization," *Chemical Engineering Science*, **49**(19), pp. 3263–3267.

[68] Majji, M., J. L. Junkins, and J. D. Turner (2008) "A High Order Method for Estimation of Dynamic Systems," *The Journal of the Astronautical Sciences*, **56**(3), pp. 401–440.

[69] Younes, A. B. (2019) "Exact Computation of Higher-Order State Transition Tensors for Perturbed Orbital Motion," *Journal of Guidance, Control, and Dynamics*, **42**(6), pp. 1365–1371.

[70] Govindarajan, N., R. Mohr, S. Chandrasekaran, and I. Mezić (2018), "On the approximation of Koopman spectra of measure-preserving flows," `1806.10296`.

[71] Adurthi, N., P. Singla, and M. Majji (2017) "Sparse approximation-based collocation scheme for nonlinear optimal feedback control design," *Journal of Guidance, Control, and Dynamics*, **40**(2), pp. 248–264.

[72] Boyd, S. and L. Vandenberghe (2004) *Convex Optimization*, Cambridge University Press.

[73] Guého, D., P. Singla, and R. G. Melton (2018) "Learning capabilities of neural networks and Keplerian dynamics," in *AAS/AIAA Astrodynamics Specialist Conference, 2018*, Advances in the Astronautical Sciences, Univelt Inc., Escondido, CA, pp. 2293–2310.

[74] ——— (2019) "Investigation of different neural network architectures for dynamic system identification: Applications to orbital mechanics," in *Spaceflight Mechanics 2019*, Advances in the Astronautical Sciences, Univelt Inc., Escondido, CA, pp. 1789–1803.

[75] Rowley, C. W., I. Mezic, S. Bagheri, P. Schlatter, and D. Henningson (2009) "Spectral analysis of nonlinear flows," *Journal of Fluid Mechanics*, **641**, pp. 115–127.

[76] Brunton, S. L., M. Budišić, E. Kaiser, and J. N. Kutz (2021) "Modern Koopman Theory For Dynamical Systems," .

[77] Ko, J., T. W. Strganac, J. L. Junkins, M. R. Akella, and A. J. Kurdila (2002) "Structured Model Reference Adaptive Control for a Wing Section with Structural Nonlinearity," *Journal of Vibration and Control*, **8**, pp. 553–573.

[78] O'Neil, T. and T. W. Strganac (1998) "Aeroelastic Response of a Rigid Wing Supported by Nonlinear Springs," *Journal of Aircraft*, **35**(4), pp. 616–622.

[79] Block, J. J. and T. W. Strganac (1998) "Applied Active Control for a Nonlinear Aeroelastic Structure," *Journal of Guidance, Control, and Dynamics*, **21**(6), pp. 838–845.

[80] Gilliatt, H. C., T. W. Strganac, and A. J. Kurdila (1998) "On the Presence of Internal Resonances in Aeroelastic Systems," in *39th AIAA Structures, Structural Dynamics, and Materials Conference*, AIAA Paper 98-1955, Long Beach, CA, pp. 2045–2055.

[81] Fung, Y. C. (1955) *An introduction to the theory of aeroelasticity*, John Wiley and Sons, New York.

[82] Ko, J., A. J. Kurdila, and T. W. Strganac (1997) "Nonlinear Control of a Prototypical Wing Section with Torsional Nonlinearity," *Journal of Guidance, Control, and Dynamics*, **20**(6), pp. 1181–1189.

[83] Theodorsen, T. and I. E. Garric (1940) *Mechanism of flutter: A theoretical and experimental investigation of the flutter problem, Tech. Rep. 685*, NACA Report.

[84] Ko, J., T. W. Strganac, and A. J. Kurdila (1998) "Stability and Control of a Structurally Nonlinear Aeroelastic System," *Journal of Guidance, Control, and Dynamics*, **21**(5), pp. 718–725.

[85] Jain, S. and G. Haller (2021) "How to Compute Invariant Manifolds and their Reduced Dynamics in High-Dimensional Finite-Element Models," *Nonlinear Dynamics*.

[86] Jain, S., P. Tiso, and G. Haller (2018) "Exact nonlinear model reduction for a von Kármán beam: Slow-fast decomposition and spectral submanifolds," *Journal of Sound and Vibration*, **423**, pp. 195–211.

[87] McNamara, J. J. and P. P. Friedmann (2011) "Aeroelastic and Aerothermoelastic Analysis in Hypersonic Flow: Past, Present, and Future," *AIAA Journal*, **49**(6), pp. 1089–1122.

[88] Bowcutt, K. G. (2018) "Physics Drivers of Hypersonic Vehicle Design," in *AIAA 2018–5373, 22nd AIAA International Space Planes and Hypersonics Systems and Technologies Conference*, Orlando, FL, pp. 1–22.

[89] Huang, D. (2019) *Development of a Hypersonic Aerothermoelastic Framework and Its Application to Flutter and Aerothermoelastic Scaling of Skin Panels*, Ph.D. thesis, University of Michigan, Ann Arbor.

[90] Lamorte, N., P. P. Friedmann, B. Glaz, A. J. Culler, A. R. Crowell, and J. J. McNamara (2014) "Uncertainty Propagation in Hypersonic Aerothermoelastic Analysis," *Journal of Aircraft*, **51**(1), pp. 192–203.

[91] Smarslok, B. P. (2015) *Quantifying Confidence in Model Predictions for Hypersonic Aircraft Structures*, *Tech. rep.*, AIR FORCE RESEARCH LAB WRIGHT-PATTERSON AFB OH AEROSPACE SYSTEMS DIRECTORATE.

[92] Culler, A. J. and J. J. McNamara (2011) "Impact of Fluid-Thermal-Structural Coupling on Response Prediction of Hypersonic Skin Panels," *AIAA Journal*, **49**(11), pp. 2393–2406.

[93] Falkiewicz, N. J. and C. E. Cesnik (2017) "Enhanced Modal Solutions for Structural Dynamics in Aerothermoelastic Analysis," *Journal of Aircraft*, **54**(3), pp. 870–889.

[94] Huang, D. and P. P. Friedmann (2020) "An Aerothermoelastic Analysis Framework with Reduced-Order Modeling Applied to Composite Panels in Hypersonic Flows," *Journal of Fluids and Structures*.

[95] Forrester, A., A. Sobester, and A. Keane (2008) *Engineering Design Via Surrogate Modelling: A Practical Guide*, John Wiley & Sons.

[96] Guého, D., P. Singla, and M. Majji (2021) "Time-Varying Koopman Operator Theory for Nonlinear Systems Prediction," IEEE Conference on Decision and Control.

[97] Dowell, E. H. (1974) *Aeroelasticity of Plates and Shells*, Noordhoff International Publishing, Leyden.

[98] Jain, S., J. Marconi, and P. Tiso (2020), "YetAnotherFECode v1.1.1," . URL `https://doi.org/10.5281/zenodo.4011281`

[99] Adurthi, N., P. Singla, and T. Singh (2012) "The Conjugate Unscented Transform - An approach to evaluate multi-dimensional expectation integrals," Montreal, QC, Canada.

[100] Vicario, F. (2014) *OKID as a general approach to linear and bilinear system identification*, Ph.D. thesis, Columbia University.

[101] Lee, C.-H. and J.-N. Juang (2012) "Nonlinear System Identification — A Continuous-Time Bilinear State Space Approach," *The Journal of the Astronautical Sciences*, **59**(1 and 2), pp. 409–431.

[102] Juang, J.-N. and C.-H. Lee (2012) "Continuous-time bilinear system identification using single experiment with multiple pulses," *Nonlinear Dynamics*, **69**, pp. 1009–1021.

[103] Majji, M., J.-N. Juang, and J. L. Junkins (2009) "Continuous Time Bilinear System Identification Using Repeated Experiments," .

[104] Junkins, J. L. and Y. Kim (1993) *Introduction to Dynamics and Control of Flexible Structures*, AIAA Education Series, AIAA.

[105] Dym, C. and I. Shames (1973) *Solid Mechanics: A Variational Approach*, McGraw-Hill Book Company.

[106] Petrolito, J. (2000) "Approximate Solutions of Differential Equations Using Galerkin's Method and Weighted Residuals," *Journal of Mechanical Engineering Education*, **28**(1), pp. 14–26.

[107] Recht, B., M. Fazel, and P. A. Parrilo (2008) "Guaranteed Minimum-Rank Solutions of Linear Matrix Equations via Nuclear Norm Minimization," *SIAM Review*, **52**(3), pp. 471–501.

Vita

Damien Guého

Damien Guého was born in Argenteuil, France, on October 13th, 1995. After preparatory classes in Paris and two years of engineering studies in Ecole Centrale de Lyon, France, he joined the Control and Analysis of Stochastic Systems (CASS) Lab at Penn State University to begin graduate work in 2017. He received his masters thesis in May 2019, which focused on the learning capabilities of neural networks with applications in astrodynamics. In parallel, he received his engineering degree from Ecole Centrale de Lyon in November 2019. Under the supervision of Dr. Puneet Singla and Dr. Robert Melton, Damien focused his research work on a wide range of topics in data-driven analysis of dynamical systems, with particular interests for high-dimensional and complex dynamical systems, data-driven system identification, reduced-order modeling, stochastic analysis, uncertainty quantification and data-driven control.