UNIVERSITY OF CALIFORNIA, MERCED

MODELING, PREDICTION AND CONTROL OF ELECTRO-MECHANICAL SYSTEMS USING ARTIFICIAL INTELLIGENCE

by

Ghazaale Leylaz

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

in

Mechanical Engineering

Committee in charge:
Professor Jian-Qiao Sun, Chair
Professor YangQuan Chen
Professor Reza Ehsani
Professor Roummel Marcia

The proposal thesis of Ghazaale Leylaz is approved:

Jian-Qiao Sun, Chair	Date
YangQuan Chen	Date
Reza Ehsani	Date
Roummel Marcia	Date

University of California, Merced

To the magnificence of God, to my parents, and my beloved husband, Amirhossein. Without their support and love, none of this would be possible. I am truly blessed.

ACKNOWLEDGEMENTS

First of all, I would like to express heartfelt gratitude to my advisor Professor Jian-Qiao Sun for his guidance, patience, motivation, encouragement, and immense knowledge not only on my research but also on my life and mental growth. I would like to thank all of the committee members, Professor YangQuan Chen, Professor Reza Ehsani, and Professor Roummel Marcia for their invaluable recommendations. Also, I really feel grateful to work with my current and former colleagues: Dr. Frank (Shangjie) Ma, David Laredo, Dr. Ala'aldin Alafaghani, Anni Zhao, Dr. Cheng Chen, and Amir Sharafi. I would like to appreciate the support and immense love of my husband Dr. Amirhossein Norouzi and my parents, Fereshteh Rahmati and Abbas Leylaz. Without their support, I would not be where I am today.

CURRICULUM VITAE

Education

M.S. in Mechanical Engineering, University of California at Merced (CA, USA), 2018.

B.S. in Mechanical Engineering, Amirkabir University of Technology (Tehran, Iran), 2011.

B.S. in Physics, Amirkabir University of Technology (Tehran, Iran), 2009.

Honors

Dual Degree Talented Student (2007), Amirkabir University of Technology (Tehran, Iran).

Mechanical Engineering Bobcat Fellowship and Travel Award (Summer 2019, Spring 2019, Summer 2020, Spring 2021), University of California at Merced.

Fellowship in Graduate Research and Teaching (2016, 2017, 2018, 2019, 2020 and 2021), University of California at Merced.

Publications

Leylaz G, Ma SF, Sun JQ (2021) "An optimal model identification algorithm of nonlinear dynamical dystems with the algebraic method ". *Journal of Vibration and Acoustics* **143**(2), 021002.

Leylaz G, Wang S, Sun JQ (2021) "Identification of nonlinear dynamical systems with time delay ". *International Journal of Dynamics and Control*, 1-12.

Ma SF, Leylaz G, Sun JQ (2021) "Data-driven robust tracking control of underactuated mechanical systems using identified flat output and active disturbance rejection control ". *International Journal of Control*, 1-17.

Ma SF, Leylaz G, Sun JQ (2021) "Identification of differentially flat output of underactuated dynamic systems ". *International Journal of Control*, 1-17.

Laredo D, Ma SF, Leylaz G, Schütze O, Sun JQ (2021) "Automatic model selection for fully connected neural networks ". *International Journal of Dynamics and Control* **8**(4), 1063-1079.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

LIST OF SYMBOLS

J_l link moment of inertia ($kg.m^2$)
J_{eq} equivalent moment of inertia ($kg.m^2$)
B_l Flexible link damping
B_{eq} Equivalent viscous damping
k_s link stiffness ($\frac{N.m}{rad}$)
k_g High gear total gear ratio
k_t Motor current torque constant ($\frac{N.m}{A}$)
k_m Motor back-emf constant ($\frac{v}{rad/s}$)
R_m Motor armature resistance (Ω)
η_g Gearbox efficiency
η_m Motor efficiency
m Mass
k_i Stiffness
c_i Damping coefficients
k_p Proportional controller gain
τ Time delay
M Torque
V_m The voltage input of DC servomotor
α The deflection angle of link
θ The rotation angle of link or servo motor
λ Sparse regulator
T_k Time instances of the K fold
σ Standard deviation
x_d Desired trajectory
$e_f(t)$ Tracking error at time t
$s(t)$ Sliding surface at time t
V Value function
r Reward value
$\phi(x)$ Activation function
W Neural network's weights
J Cost function
γ_k Adaptive learning rate

Subscripts

$\widehat{}$ Estimated values

* Optimal values

$\tilde{}$ Error estimation with respect to the optimal value

$\dot{}$ First-order derivative with respect to time

$\ddot{}$ Second-order derivative with respect to time

$^{(n)}$ n-order derivative with respect to time

ABSTRACT

Data-driven techniques are growing at an unprecedented pace due to the recent super-fast computational tools and resources and the availability of big data by sensors. In the field of dynamics and control, many researchers are investigating algorithms to learn from data to model systems, estimate physical parameters, and design controllers, especially for complex dynamical systems. However, many of these researches are still limited to simulations due to the unavoidable noise in practical cases and the limitations in data acquisition. Some techniques lack any physical meaning and it makes it hard to analyze the effect of parameters in the system's performance. Low performance to predict the system response for unseen data is another issue that researchers are dealing with. Therefore, it is of great potential to develop intelligent and robust data-driven algorithms to model and estimate parameters of the system, where the reliable model can be used for the model_based control design.

We propose a nonparametric system identification technique to discover the governing equation of nonlinear dynamic systems with a focus on practical aspects. The algorithm builds on Brunton's work in 2016 and combines the sparse regression with algebraic calculus to estimate the required derivatives of the measurements. This reduces the required derivative data for system identification. Furthermore, we make use of the concepts of K-fold cross validation from machine learning and information criteria for model selection. This allows the system identification with fewer measurements than the typically required data for the sparse regression. The result is an optimal model for the underlining system of the data with a minimum number of terms. The proposed system identification method is applicable for multiple-input–multiple-output systems. Two examples are presented to demonstrate the proposed method. The first one makes use of the simulated data of a nonlinear oscillator to show the effectiveness and accuracy of the proposed method. The second example is a nonlinear rotary flexible beam. Experimental responses of the beam are used to identify the underlining model. The Coulomb friction in the servo motor together with other nonlinear terms of the system variables are found to be important components of the model. These are, otherwise, not available in the theoretical linear model of the system.

We also extend the sparse optimization algorithm to nonlinear systems with time delay. We further integrate the bootstrapping resampling technique with the sparse regression to obtain the statistical properties of estimation. We use Taylor

expansion to parameterize time delay. The proposed algorithm in this paper is computationally efficient and robust to noise. A nonlinear Duffing oscillator is simulated to demonstrate the efficiency and accuracy of the proposed technique. An experimental example of a nonlinear rotary flexible joint is presented to further validate the proposed method.

Finally, we examine the efficiency of our identified model to design model_based controllers. First, we propose a robust flat output-based sliding mode control for trajectory tracking and to deal with under-actuated degree of freedoms. Moreover, we investigate the optimal control design. Optimal control design needs the solution of Hamilton-Jacobi-Bellman equation, where the nonlinearities in the model make the solution challenging or even infeasible. We propose an efficient algorithm to estimate a neural network solution to gain the feedback control law. We examine the efficiency of our algorithm through several popular examples in the optimal control community and more importantly our identified nonlinear model of rotary flexible manipulator link.

Chapter 1

INTRODUCTION

Recent technologies have enthusiasm about data-driven techniques and solutions. In the recent decade, thanks to the explosive growth of available data and computing resources, data-driven techniques and machine learning approaches have yielded transformative results across diverse scientific disciplines of robotics, natural language processing, economics, etc. In the field of mechanical systems, the high-tech systems tends to add complexity such that the classical methods and techniques are not able to deal with them. That's why data-driven modeling and control design techniques are growing at the fast pace. The present thesis focuses on developing algorithms to find data-driven solutions for optimal modeling or system identification, time delays estimations, and optimal controller design. We first give a brief introduction about our works in this chapter and then in the next chapters we detail the problem statements, algorithms, results and conclusions. In this chapter, we first give a literature review in the field of system identification and mention our contributions, especially in practical aspects. Then, we describe the works are done to detect the time delays in the systems and count the highlights of our work in this field. Finally, after a brief definition of data-driven control systems, we focus on model-based controller design where the system's model is obtained by data-driven methods. In particular, we detail in the literature done in the field of optimal control and reinforcement learning-based controllers for nonlinear and high-dimensional systems.

1.1 Optimal Data-Driven Modeling

Modern engineering systems can be highly complex with time-varying nonlinearities and uncertainties. It is a challenge to develop accurate mathematical models of such systems, which are often needed for designing precision controls and for optimal operations. In the age of high-tech and super-computing, it has become increasingly feasible to use a large number of sensors to measure the input and output of complex systems over a long time. With the availability of big data of input and output, we should be able to identify the underlining mathematical model of the system and determine the model parameters without much knowledge about the system. Chapter 2 presents a study of this kind. The study reported in this work

falls in the general area of system identification, which is a well established technical field.

Data are abundant whereas models often remain elusive, as in climate science, neuroscience, ecology, finance, epidemiology, *etc.* In biology, data-driven modeling of fish trajectories in group or individual stands for complex stochastic systems [4,5]. Porfiri's group models an individual zebrafish trajectory through stochastic equations and Markov chain. The calibration of the model parameters through experimental data by the maximum likelihood and weighted least squares method allows studies of how the fish behaves in front of constrains like walls and when exposed to the material such as caffeine [6]. In transportation, machine learning-based approaches aim at accurate trajectory prediction of aircraft to manage air traffic and control environment [7] and and of vehicles for safe and efficient autonomous driving [8].

Various system identification techniques have been developed over the years and are well documented in the survey papers [9, 10]. During the past decade, nonlinear system identification has attracted a great deal of attraction. A comprehensive survey of nonlinear system identification techniques from 2006 to 2016 can be found in [11]. Using time-varying linear system model to describe the responses of nonlinear dynamic systems is frequently studied [12–14]. Factors including the looseness of structural joints, amplitude dependent material properties and boundary conditions with variable stiffness constraints make the behavior of engineering structures highly nonlinear and complex [15]. Other intrinsic nonlinear phenomena such as hysteresis, friction, inelasticity, harmonics, jumps or modal interactions call for nonlinear system model [11].

The system identification is often done in frequency-domain. An efficient data-driven technique to identify a differentially flat output of under-actuated dynamic systems is developed in [16]. A modified H_2 algorithm for frequency response function estimation is proposed in [17]. The modification of the algorithm introduces additional correlated output to improve the computational effectiveness. A generalized approach to apply the method of "Reverse Path" to continuous mechanical systems with multiple nonlinearities is introduced in [18]. The methods for linear systems are extended to nonlinear systems with the help of associated linear equations (ALEs) and associated frequency response functions (AFRFS) [19]. Some technical issues and limitations are addressed in this reference in order to apply the method to MDOF systems. Another technique in frequency domain interprets nonlinearities as feedback forces acting on the underlying linear system [20]. The system identification can also be done in time domain or in time-frequency mixed domain [21–23]. The method of nonlinear subspace identification is formulated in time and frequency domain for a strongly nonlinear satellite structure [24].

A general framework is still not available for nonlinear system identification problems. The well-known model of Nonlinear Auto-Regressive Moving Average

with eXogenous Input (NARMAX) is considered for the detection of damage in engineering structures [25]. The work in [26] studies an amplitude and frequency modulation method (AFMM) for extracting characteristics of nonlinear systems and intermittent transient responses by processing stationary/transient responses using the empirical mode decomposition, Hilbert-Huang transform (HHT), and nonlinear dynamic characteristics derived from perturbation analysis. The experimental extraction of nonlinear normal modes (NNMs) under broadband forcing is investigated in [21], where a state-space model is first established, and then is converted to the system in the modal space.

Nonparametric or black-box methods directly work with input-output data and are very popular. Some common examples include subspace identification, kernel methods, statistical learning, data mining, artificial neural networks (ANNs), auxiliary model identification, multi-innovation identification and hierarchical identification [27–32]. The black-box model does not need the prior knowledge of the system and sometimes has no clear physical meaning [15]. Raissi and colleagues have developed a deep learning based algorithm and illustrated the method on complex dynamic systems of Lorenz, fluid flow behind a cylinder, Hopf bifurcation and the Glycoltic oscillator [30]. A long short-term memory (LSTM) recurrent neural networks (RNNs) has been recently introduced to predict high-dimensional chaotic systems [33]. A reservoir-based computing algorithm gives impressive predictions for chaotic systems [34]. It should be noted that the high-dimensional neural network models are lack of physical meaning. As a tool for regression, neural networks also don't predict well out of the range of the training data set [15, 29]. The Bayesian framework is currently attracting a great attention from the research community [35–37]. This method tries to find the best fitting model among the proposed models by studying the occurrence probabilities of the proposed functions [38].

When the dynamic model of the system is represented by differential equations, we often have to create the time series of derivatives of the measured data. The algebraic method and operational calculus are excellent tools for computing derivatives of the measured signals by means of integration, thus to avoid amplifying the measurement noise in the data [39, 40]. The algebraic framework in [41] has been studied for identification of first-order linear systems. The algebraic identification has been used to estimate parameters of a servo motor [42], highly damped flexible structures [43], and mass-spring-damper systems [44]. In this chapter, we apply the algebraic method to identify nonlinear dynamical systems. We use extensive simulations and experimental examples to show that the algebraic method is effective for nonlinear dynamical systems and is robust to measurement noise.

Polynomials have been a popular functional choice for modeling nonlinear systems. From the point of view of fitting the data, whether we use polynomials, artificial neural networks or any other functions, as long as a sufficient number of functions are used, the resulting model will fit the data to any degree of accuracy.

In this case, the major concern is overfitting when the obtained model offers little flexibility to describe the dynamic behavior of the system that may not have been contained in the data used for the system modeling. For a given set of measurements, an important question to ask is: What is the simplest model that can fit the measured data sufficiently accurate and still provide enough flexibility to make reliable predictions with the unseen data? This chapter investigates the answer to the question.

Brunton and colleagues proposed a method called the sparse identification of nonlinear dynamics (SINDy) in 2016 for creating a sparse representation of the unknown nonlinear function of the system [45]. The method has attracted a great attention from the community. The method assumes that only a few important terms out of a library of functions are needed to describe the dynamics of the system. It combines machine learning and sparsity-promoting techniques to find the sparse representation in the space of possible functions to model nonlinear dynamical systems. The connection of the SINDy method to the Akaike information criteria (AIC) for model selection has been studied in [46]. The promising results for difficult system identification problems such as hybrid dynamical systems, chaotic Lorenz system and Burger's partial differential equation have been obtained [46, 47]. The SINDy method applied to the model predictive control delivers better performance, requires significantly less data, and is more computationally efficient and robust to noise than the neural networks model [48]. Studies of the SINDy method have reported the usage of high volume of time series data from different initial conditions for model selection and for cross-validation [46]. While this is feasible for simulation studies, it can be expensive for applications with real time data. Robustness to noise and requirement for measurements of derivatives are concerns with the SINDy approach [29]. The total variation regularized derivatives are commonly used to deal with these concerns [49,50]. An integral form of equations of motion in combination with sparse regression is proposed in [51]. An application to model identification of nonlinear mechanical systems is reported in [52]. In Chapter 2, we extend the SINDy approach by combining it with the algebraic method and a K-fold cross validation approach to deal with the issue of measurement noise, initial conditions and derivatives.

We focus on the model development and parameter identification of mechanical systems satisfying Newton's second law of motion. That is, the acceleration of the system is partly due to a restoring force, which is a nonlinear function of the displacement and velocity. There is a well-known method called the method of restoring force surface, first proposed by Masri and Caughey in 1979 [53] for identifying the nonlinear restoring force function. The method continues to receive lots of attention from researchers [11]. The original method of restoring force surface is non-parametric and calls for extensive data to construct the restoring functions in

the state space. More recent studies involve piecewise linear functions and polynomials for fitting the data to create the restoring force function. In our algorithm in Chapter 2, we use polynomials to represent the restoring force functions.

The main contribution of our work is a comprehensive approach to develop an optimal sparse representation of the restoring force function. The procedure includes the adoption of the algebraic method for dealing with the dependence on the initial conditions of the measurements and measurement noise, and a K-fold cross validation approach for training and validating the model. The result is a sparse polynomial of the minimum order that accurately describes the underlining nonlinear dynamic system of the data.

1.2 Time-delayed System Identification

Time delay exists in many engineering, physics, chemistry, biology and economics systems. In control systems, time delay due to sensor and actuator dynamics, signal transmission, and digital computations is an important factor that influences the stability and control performance. To make matter worse, time delay is often unknown. Time delay estimation in a control system is a challenging problem. It is even more challenging when the system dynamics is nonlinear and unknown. Chapter 3 presents a nonparametric identification technique to identify nonlinear dynamic systems and estimate time delay introduced by the feedback control.

There have been many studies of time delay identification of control systems. Richard presented an overview of time delay estimation methods in [54]. The time delay estimation techniques based on pulsed inputs have been developed in [55, 56]. The Padé approximation [57], the modified least square and recursive methods [58–62] , instrumental variable identification [63], neural networks [64], algebraic estimation [65, 66], adaptive techniques [67, 68], and non-commutative rings [69] are just a few popular methods for time delay estimation.

The methods for time delay estimation can be in frequency or time domain [70, 71]. In Chapter 3, our focus is on the approaches in time domain. Since time delay usually appears in the system implicitly, the methods for conventional parameter estimation of dynamic systems cannot be directly applied to estimate time delay. Time delay τ usually appears in the exponential term $e^{-\tau s}$ in the transfer function of the system. The expansion techniques can parameterize it, including the classical Padé approximation, the Laguerre Fourier series, the Kautz series, the second-order Padé shift and the diagonal Padé shift. The main concern with the rational approximation is the truncation error and stability complications. Even though higher order expansions can reduce the truncation error, the system can become unstable even when the system is linear with a constant time delay [54]. In Chapter 3, we employ the Taylor expansion. Xu demonstrates that the low order Taylor expansion gives promising estimation of small time delay [72].

The work in [73] presents a nonlinear least square-based algorithm in which the instrumental variable method estimates the parameters of the transfer function of the system while an adaptive gradient-based iteration finds the optimal time delay from the filtered irregularly sampled data. The main problem with this algorithm is that the proposed cost function may have several local minima. Therefore, it highly depends on the initial guess for the parameters and especially the time delay. To deal with this issue, the authors use a low-pass filter to widen the convergence region around the global minimum. Another nonlinear recursive optimization algorithm is proposed in [74], which combines the linear method of Levenberg-Marquardt to compute the plant parameters with a modified Gauss-Newton algorithm to estimate time delays. A low-pass filter and a binary transformation are applied to the data corrupted by the white noise to create the regressor matrix for minimizing a quadratic cost function of the estimation error. The identification is online and is demonstrated on a MISO plant with multiple time delays. Similarly, the algorithm in [62] employs the Gauss-Newton method to estimate time delay when the simplified refined instrumental variable ($SRIVC$) method is used to find the plant parameters.

The Taylor expansion is used to parameterize the system with explicit time delay in [68]. An adaptive law for the parameter estimation is proposed such that the estimation error is converged. A recursive formula is introduced in [59] to improve the accuracy and convergency rate of another recursive algorithm for online estimation in [75]. The strategy in [61] aims at fractional time delay identification for discrete-time systems. It separates the influence of the system structure and the time delay by discretizing the system. With the help of the Kalman filter, the parameters are estimated recursively. The algorithm in [64] first parameterizes the system by a polynomial function and trains a neural network to estimate the parameters and time delay. A Schweizer and Wolff's σ measure denoted as σ_{SW} from the copula theory is introduced to study the relationship of input-output signals for SISO systems [76]. It is found that the measure reaches its maximum when the time delay is removed from the data. This property offers an approach to estimate time delay without the need of estimation of other parameters.

Most existing methods for time delay estimation rely on the knowledge of the system model. In fact, precise model of the system is needed. In Chapter 3, we propose the application of SINDy technique in [45] for the time delay estimation. Indeed, we extend the application of our algorithm in Chapter 2 to identify nonlinear dynamical systems with time delay.

In practical cases, the data is often contaminated with noise. Fliess and Ramirez proposed a robust and fast algebraic identification technique to estimate time invariant linear systems without time delay [77]. Inspired by the algebraic operation, Belkoura in [65] and [66] first investigated the identifiability conditions for a general class of systems described by convolution equations. Then, an algebraic

formulation is introduced for online estimation of time delay and parameters of structured and arbitrary input-outputs.

In Chapter 3, we extend the SINDy approach by combining it with the algebraic signal processing method to deal with the issue of measurement noise, initial conditions and derivatives. The algebraic operation generates useful signals for system identification while filtering out the noise. We apply the Taylor expansion to make the time delay appear as a parameter of the model to identify. A nonlinear extended state estimator is adopted for derivative estimation. As a result, we arrive at a robust sparse regression combined with cross validation and bootstrapping techniques for nonparametric system identification. The simulation and experimental results illustrate that the proposed algorithm can overcome the following limitations of the existing techniques:

- **Multiple local minima:** Achieving the global minimum is the main challenging for modified least square methods with recursive approach. The sparse regression is a convex optimization problem with a global minimum [15].
- **Nonparametric nonlinear identification:** The only assumption of the proposed algorithm about the structure of the system is that the system is sparse in the space of base functions. The proposed algorithm relies on the data to make selection and is not limited to linear and SISO systems.
- **Noise resistant:** The sparse regression is already robust. The algebraic operation offers additional filtering of the noise. Furthermore, the proposed algorithm is equipped with bootstrapping to study the statistics of estimation such as mean and standard deviations.
- **Unstructured entries:** Many classical strategies are designed for pulsed entries. The proposed approach analyzes the input and output data without any frame and structure assumption for entries.
- **Initial conditions:** Another general assumption in time delay estimation is that the initial conditions are zero while in most of practical applications, it is not true. The algebraic operation and operational calculus make the proposed algorithm independent of the initial conditions.

1.3 Data-Driven Control Design for Trajectory Tracking

Data-driven and machine learning-based techniques have made a revolution in the field of control systems. Data-driven control systems are a broad family of control systems where the identification of the model and/or the control design are totally based on the experimental data collected from the plant. Control design often begins with a model of the system. Notable exceptions include mode-free

adaptive control strategies and many uses of PID control. For mechanical systems of moderate dimension, it may be possible to write down a model based on the Newtonian, Lagrangian, or Hamiltonian formalism and linearize the dynamics about a fixed point or periodic orbit. However, for modern systems of interest, typically there are no simple models suitable for control design.

Real-world systems are usually nonlinear and the control objective is not readily achieved via linear techniques. It is still difficult to find a simple yet reliable nonlinear model for a physical system, that includes only those dynamics of the system that are of interest for the control specifications. From the recent studies, we can mention the researches on differentially flat output model of systems for control design [16, 78]. In Chapter 4, we investigate how our system identification algorithm delivers optimal models regarding control design for trajectory tracking or stabilization problems. First, we propose robust sliding model control and then we study how to solve nonlinear optimal control for our system identification approach. As a case study, we focus on control design for our identified rotary flexible link in Chapter 2.

RFL is widely used in industrial automation and there has been many studies on the control design for the past few decades. From those, we can mention the data-driven reinforcement learning-based controller in [79], backstepping design scheme in [80], fractional order control in [81], active fuzzy logic and neural networks-based control in [82], modified PID controller in [83], and adaptive fuzzy output feedback control approach in [84]. An optimal sliding mode control (SMC) is also proposed in [85]. SMC is known as a robust controller but depends on the full knowledge of the system model. Therefore, to validate the efficiency of our identified model for control design, we suggest SMC. RFL is an under-actuated system in which the number of actuators is less than the number of degree-of-freedom to be controlled. Therefore; we involve the concept of flatness in [86, 87] in our control design and define a flat output-based SMC. Flatness, in its more popular conception, is a property that readily trivializes the exact linearization problem in a nonlinear system.

Optimal control has emerged as a powerful modern controller that finds the feedback controller law by optimizing the value function as the performance index of the controller's operation and energy usage over time. In fact, in the field of control, the line between optimal control and Reinforcement Learning (RL) can be quite blurry, as optimal control involves the essential elements underlying the theory and algorithms of modern RL. Optimal control is mainly a mathematical optimization problem, where the learning and data-driven algorithms are playing a crucial rule. Finding a reliable predictive model of the system dynamic or model-based techniques can make the learning algorithms more effective and data-efficient. We investigate the efficiency of our data driven modeling algorithm in Chapter 2 to deliver simple but complete models to design optimal control.

Identified models usually contain nonlinearities. Computing optimal feedback controls for nonlinear systems generally requires solving the Hamilton-Jacobi-Bellman (HJB) equation to find the value function, which are notoriously difficult especially when the state dimension is large [88]. Moreover, the non-linearities in the identified model causes an optimization problem with a non-convex cost function landscape with multiple local minima. Therefore, there is an extensive literature on the methods of finding approximate solutions for HJB equations. Some key approach includes patchy dynamic programming [89], series expansions [90, 91], level set methods [92], semi-Lagrangian methods [93, 94], methods of characteristics and Hopf formula-based algorithms [95–97], and polynomial approximation [98].

Using neural networks (NNs) as a basis for solving HJB equations is not by itself a new idea, and deep learning approaches have led to promising results; see for instance [99–104]. To the best of our knowledge, state-of-the-art NN-based techniques generally rely on either minimizing the residual of the PDE and (artificial) boundary conditions at randomly sampled collocation points [99, 100, 103, 104]; or, due to computational limitations, approximating the control and/or HJB solution and its gradient in a small neighborhood of a nominal trajectory [101, 102]. In [105, 106], a specialized NN architecture is proposed to solve some classes of Hamilton-Jacobi equations, but this method has yet to be generalized to state-dependent HJB equations arising in optimal control. Deep learning techniques have also been proposed for solving high-dimensional stochastic optimal control problems [107,108].

To solve HJB equation for nonlinear dynamics, we propose a NNs-based estimation algorithm to solve the continuous-time infinite horizon optimal control problem. In fact, we model the solutions to HJB equation or the same value function with NNs trained on data generated with discretizing the state space. We selected the basis functions in NNs as the quadratic vector in the state components. To train NNs, we suggest ADAM algorithm in [109] which is well-known algorithm for deep networks where the number of parameters is huge. Machine learning is complementary, as it constitutes a growing set of techniques that may be broadly described as performing nonlinear optimization in a high-dimensional space from data. The candidate optimal feedback control is then easily computed by the value function in real-time. Comparing our results with the the reinforcement learning algorithm, online actor-critic approach proposed by Frank Lewis in [99], our algorithm gains more accuracy. More importantly, the algorithm in [99] requires a persistence of excitation condition to guarantee the convergence, while this condition is hard to generate. Our approach is free from this condition, although it depends on the sufficient data points used for training NNs.

The main contribution of our work in Chapter 4 is the validation of the system identification in Chapter 2 to design model-based controller. Furthermore, our flat out-put based sliding mode controller design successfully perform the tracking controller for the under-actuated system. Moreover, our NNs-based estimation to solve

HJB equation can be potentially advanced to solve higher-dimensional problems.

1.4 Thesis Organization

This dissertation consists of 4 chapters. Chapter 1 presents the introduction and dissertation organization; this is while Chapter 2 focuses on the algorithm development for optimal model identification of nonlinear dynamical systems with the algebraic method. Chapter 3 provides details about the identification of nonlinear dynamical systems with time delay. In Chapter 4, we focus on trajectory tracking control design for our data-driven model of the systems, where the designer is dealing with the non-linearities, high-dimensionality, and under actuated degree of freedom.

Chapter 2

DATA-DRIVEN MODELING

In the present chapter, we present the work in [52]. It proposes a procedure to develop mathematical models of nonlinear dynamical systems and to identify their parameters at the same time. We use polynomials as an example to model nonlinear systems and have developed a criterion for identifying an optimal model in the sense that the order of the polynomial to best model the system from a given data set is minimum. We use a nonlinear Duffing oscillator to motivate the development of the proposed algorithm. Experimental response time histories of a nonlinear flexible beam with friction are used as an example to validate the proposed algorithm.

The rest of the chapter is organized as follows. In Section 2.1, we present the problem statement and the method for identifying the optimal model from the measurement. In particular, we discuss the algebraic method in Section 2.1.2, and the sparse representation in Section 2.1.3. The proposed method is summarized in the form of an algorithm. We then present two examples to demonstrate the method. Section 2.2 makes use of the simulated data of a nonlinear oscillator to show the accuracy and effectiveness of the proposed algorithm of. Section 2.3 presents the model of a nonlinear flexible beam identified from the experimental data. Finally, Section 3.5 concludes the paper.

2.1 The Proposed Method

2.1.1 Polynomial Model of Restoring Force

As an example, we shall use a single-degree-of-freedom system to illustrate the research concept. Assume that the system with an inertial element of mass m satisfies Newton's second law such that the acceleration $\ddot{x}$ is governed by

$$m\ddot{x} + g(x, \dot{x}) = f(t) \tag{2.1}$$

where $f(t)$ the excitation and $g(x, \dot{x})$ represents the internal and nonlinear restoring force of the system. In general, we can express $g(x, \dot{x})$ in terms of a series expansion of regular or orthogonal polynomials. Because the range of the variables $(x, \dot{x})$ may not be bounded and is unknown, it is natural to consider the regular polynomials with time-invariant coefficients given by,

$$g(x, \dot{x}) = \sum_{i=0}^{N} \sum_{j=0}^{M} c_{ij} x^i \dot{x}^j \tag{2.2}$$

where c_{ij} are the unknown coefficients of the polynomial. The coefficients c_{ij} are usually related to the physical properties of the dynamic system. For the system without any prior knowledge, the orders N and M of the polynomial are unknown. This study develops an algorithm to identify the polynomial defined in Equation (2.2) from the input and output data such that it has a minimum number of terms and with the minimum orders N and M.

2.1.2 Data Pre_processing with Algebraic Method

We prepare necessary signals from the measurements by using the algebraic method in order to apply the least square method to compute the parameters m and c_{ij} of the system [41]. We use a nonlinear oscillator to illustrate the steps of the method.

$$\begin{aligned} &m\ddot{x} + c_1\dot{x} + c_2\dot{x}x^2 + c_3\dot{x}^3 + k_1x + k_3x^3 = f(t) \\ &x(0) = x_0,\ \dot{x}(0) = v_0 \end{aligned} \tag{2.3}$$

where (x_0, v_0) are the initial conditions, which are generally unknown.

Remark: The input excitations must be chosen such that it can excite the system to exhibit rich dynamics in the output.

We introduce several short-hands for the nonlinear terms.

$$z_1 = \dot{x}x^2, \quad z_2 = \dot{x}^3, \quad z_3 = x^3 \tag{2.4}$$

Applying the Laplace transform to Equation (2.3), we have

$$\begin{aligned} &m\left\{s^2X(s) - sx_0 - v_0\right\} \\ &+ c_1\left\{sX(s) - x_0\right\} + k_1X(s) = G(s) \end{aligned} \tag{2.5}$$

where

$$G(s) = F(s) - c_2Z_1(s) - c_3Z_2(s) - k_3Z_3(s), \tag{2.6}$$

$X(s)$ is the Laplace transform of $x(t)$, $Z_i(s)$ is the Laplace transform of $z_i(t)$ and $F(s)$ is the Laplace transform of $f(t)$.

To eliminate the dependence on the unknown initial conditions x_0 and v_0, we take derivatives with respect to s twice in Equation (2.5) and obtain

$$\begin{aligned} &m\left\{2X(s) + 4s\frac{dX(s)}{ds} + s^2\frac{d^2X(s)}{ds^2}\right\} \\ &+ c_1\left\{2\frac{dX(s)}{ds} + s\frac{d^2X(s)}{ds^2}\right\} + k_1\frac{d^2X(s)}{ds^2} = \frac{d^2G(s)}{ds^2} \end{aligned} \tag{2.7}$$

Dividing both sides of Equation (2.7) by s^2 to avoid derivatives with respect to time, we obtain

$$\begin{aligned} & m\left\{\frac{2}{s^2}X(s)+\frac{4}{s}\frac{dX(s)}{ds}+\frac{d^2X(s)}{ds^2}\right\} \\ & +c_1\left\{\frac{2}{s^2}\frac{dX(s)}{ds}+\frac{1}{s}\frac{d^2X(s)}{ds^2}\right\} \\ & +k_1\frac{1}{s^2}\frac{d^2X(s)}{ds^2}=\frac{1}{s^2}\frac{d^2G(s)}{ds^2} \end{aligned} \tag{2.8}$$

Equation (2.8) can be rewritten in terms of the following signals

$$\begin{aligned} & mP_1(t)+c_1P_2(t)+k_1P_3(t)+ \\ & c_2P_4(t)+c_3P_5(t)+k_3P_6(t)=P_f(t) \end{aligned} \tag{2.9}$$

where

$$\begin{aligned} P_1(t) &= 2\int^{(2)}x(t)-4\int tx(t)+t^2x(t) \\ P_2(t) &= -2\int^{(2)}tx(t)+\int t^2x(t) \\ P_3(t) &= \int^{(2)}t^2x(t), \quad P_4(t)=\int^{(2)}t^2z_1(t) \\ P_5(t) &= \int^{(2)}t^2z_2(t), \quad P_6(t)=\int^{(2)}t^2z_3(t) \\ P_f(t) &= \int^{(2)}t^2f(t) \end{aligned} \tag{2.10}$$

Note that $\int^{(n)}\phi(t)$ denotes the multiple integral $\int_0^t\int_0^{\sigma_1}...\int_0^{\sigma_{n-1}}\phi(\sigma_n)d\sigma_n...d\sigma_1$. Let t_k $(k=1,2,...n_t)$ be a set of sampled times. Define an error at time t_k as

$$\begin{aligned} e(k) = & mP_1(t_k)+c_1P_2(t_k)+k_1P_3(t_k) \\ & +c_2P_4(t_k)+c_3P_5(t_k)+k_3P_6(t_k)-P_f(t_k). \end{aligned} \tag{2.11}$$

Let us introduce an $n_t\times 1$ error vector as $\mathbf{e}=[e(1),e(2),...,e(n_t)]^T$, a parameter vector as $\mathbf{c}=[m,c_1,k_1,c_2,c_3,k_3]^T$ and a force vector as $\mathbf{p}=[P_f(t_1),P_f(t_2),...,P_f(t_{n_t})]^T$. Let $\mathbf{P}(x)$ be the $n_t\times 6$ data matrix determined by the measurement of $x(t)$ such that its $(k,j)^{th}$ component is given by $P_{k,j}=P_j(t_k)$. Then, Equation (3.11) can be written in the matrix form as

$$\mathbf{e}(\mathbf{c})=\mathbf{P}(x)\mathbf{c}-\mathbf{p}. \tag{2.12}$$

To find the unknown parameters $\mathbf{c}$, we formulate a least mean square error problem as follows. For the following cost function,

$$J(\mathbf{c}) = \mathbf{e}^T\mathbf{e} = \|\mathbf{e}\|_2^2 \tag{2.13}$$

we find the parameter vector $\mathbf{c}$ such that J is minimized. The solution of the parameters $\mathbf{c}$ can be found by either the matrix inversion or by an iterative search algorithm. In general, when the numbers of unknown parameters and data points n_t are large, as is the case for multi-degree of freedom systems, it is better to use a global search algorithm to compute the parameters.

2.1.3 Sparse Representation

Assume that the system in Equation (2.3) is unknown and that only the displacement measurement $x(t)$ of the system is available and contains random noises. We consider the polynomial model for the restoring force in Equation (2.2). The questions are: What are the minimum orders N and M? Are all the terms in the polynomial needed?

One popular way is to start with high enough orders of the polynomial or a large library of functions, which may contain non-polynomial functions, and penalize the the terms with smaller coefficients through a sparse regulator by using the least absolute shrinkage and selection operator (LASSO) [110]. The recent studies of the sparse identification of nonlinear dynamics (SINDy) have developed a computationally efficient algorithm to solve the sparse regression. To find the sparse regulator, SINDy uses the "elbow" of the bias-variance tradeoff curve called the "Pareto front". In most of cases, finding the curve's elbow is ambiguous due to existence of a cluster of candidate models close to the elbow. To address this issue, an information criterion for the candidate models can be used to rank and select the right model [46]. It is important to point out that the selection of the models on the Pareto front from a large library can be computationally intensive.

In this work, we build the models starting from lower order polynomials. We divide the data into the training set and test set. In the training, we apply the sparse regression LASSO to select the models and employ the K-fold cross validation within the training set to find the sparse regulator. This process generates a model with the smallest mean squared cross validation error in the training set for a given order of the polynomial. Finally, we compute the information criterion of the trained model over the test set, and then increase the order of the polynomial. We stop increasing the order of the polynomial after the information criterion reaches a minimum value and starts to grow.

Training	Training	…..	Validation	…..	Training	Training
1	2	…..	*k*	…..	*K-1*	*K*

Figure 2.1: Illustration of the data segmentation for the K-fold cross validation.

For a given order of the polynomial, the LASSO proposes to add an L^1 regularization term to the cost function in Equation (2.13) in order to penalize the nonzero parameters that don't contribute much to the model [110].

$$\begin{aligned} J_\lambda(\mathbf{c}) &= \|\mathbf{e}\|_2^2 + \lambda \|\mathbf{c}\|_1 \\ &= \|\mathbf{P}(x)\mathbf{c} - \mathbf{p}\|_2^2 + \lambda \|\mathbf{c}\|_1 \end{aligned} \tag{2.14}$$

where $\lambda > 0$ is a preselected small positive number known as the sparse regulator and $\|\mathbf{c}\|_1 = \sum_i |c_i|$. The sparse regression algorithm in [45] is applied to find the parameter vector $\mathbf{c}$ in order to minimize J_λ in Equation (2.14). The algorithm is computationally efficient and robust to noise. It should be noted that the optimization problem in the framework of the LASSO is convex [15], which implies the existence of a unique optimal solution.

In the studies of SINDy, the sparse regulator λ is determined through cross validation [45]. The simulations from different initial conditions are usually used to generate rich data sets to train and validate the model. The training data is often much larger than the validation data. Although the rich data set leads to more accurate model with the help of the regularization term, especially for stochastic systems, it is quite expensive to generate such a data set from experiments. To resolve this issue, we apply the K-fold cross validation method by splitting the training set consisting of a long time series of the measurement $x(t)$ into K roughly equal parts or folds as Figure 2.1 illustrates. One of the K folds of the training dataset is used for cross validation of the model trained with the rest of the data in the same training dataset.

Let T_k denote the time instances of the k^{th} part of the K folds to be used for validation, and $T_{\bar{k}}$ denote its compliment, i.e. the time instances of the rest of the K folds of the data to be used for training. The model is trained on each fold of the data in $X_{\bar{k}}$ and is validated on the set X_k. For a given regulation parameter λ, the mean square cross validation error MSE_{CV} of the model on all the validation sets is defined as

$$MSE_{CV}(\lambda) = \frac{1}{n_t} \sum_{k=1}^{K} \|\mathbf{P}(x(t \in T_k))\mathbf{c}_k - \mathbf{p}(t \in T_k)\|_2^2 \tag{2.15}$$

where $\mathbf{c}_k$ denote all the model parameters trained on $T_{\bar{k}}$. MSE_{cv} is an implicit function of λ. The minimum MSE_{cv} with respect to λ represents a compromise between the accuracy and complexity of the model [45]. This leads to the optimal choice of the regulation parameter λ for the given order of the polynomial.

Next, we consider the variation of the sparse representation of the candidate model with the order of the polynomial. Here, we adopt an information criterion to rank the sparse representation of the model [46]. Popular statistical examples of information criteria include the Akaike information criterion (AIC), Bayesian information criterion (BIC), deviance information criterion (DIC) and minimum description length. This work considers the AIC for evaluating the candidate models. The AIC describes the generality and goodness of fit of the candidate models and is calculated on the test dataset that is not used in training.

Let n_c denote the number of nonzero parameters in the vector $\mathbf{c}$. The AIC index for the model is defined as

$$AIC = 2n_c - 2\ln(L(x, \mathbf{c})) \tag{2.16}$$

where $L(x, \mathbf{c}) = p(x|\mathbf{c})$ is the likelihood function of the prediction x by the model with the parameters $\mathbf{c}$. Assume that the error $\mathbf{e}$ which contains the measurement noise is normally distributed, the residual sum of squares (RSS) can be considered as the likelihood function such that the AIC for the model can be defined as [111]

$$AIC = n_t \ln(RSS/n_t) + 2n_c \tag{2.17}$$

If the least squares estimation is used, the RSS is equal to the cost function $RSS = J(\mathbf{c})$ in Equation (2.13). We summarize the previous steps in Algorithm 1.

2.2 A Numerical Example

To validate the proposed algorithm, we consider a second-order oscillatory system with nonlinear stiffness and damping.

$$m\ddot{x} + k_1 x + k_3 x^3 + c_1\dot{x} + c_2\dot{x}x^2 + c_3\dot{x}^3 = f(t) \tag{2.18}$$

where the external force $f(t)$, as an example, is given by

$$f(t) = 10\sin(t) + 40\sin(4t) + \epsilon_f \tag{2.19}$$

ϵ_f is a normally distributed Gaussian noise with zero mean and standard deviation $\sigma_{\epsilon_f} = 0.01$. In the simulations, we have set $k_1 = 6$, $k_3 = 0.1$, $c_1 = 1$, $c_2 = 0.2$ and $c_3 = 1$. A measurement noise is considered such that the system output is given by $x(t) + \epsilon_x$ where ϵ_x is another normally distributed random noise with zero mean and standard deviation $\sigma_{\epsilon_x} = 0.01$. The general restoring force function $g(x, \dot{x})$ in Equation (2.2) is used to model the system.

Algorithm 1

The proposed algorithm for nonlinear system identification with the algebraic method.

procedure SYSTEM IDENTIFICATION($x, f, k_{fold}, ratio, \lambda_{min}, \lambda_{max}$)
  for $PolyOrder \in [1, N;\ 1, M]$ **do**
    $g(x, \dot{x}) \leftarrow x, \dot{x}, PolyOrder$ ▷ Generate polynomial terms
    $P(X) \leftarrow$ library$(X, PolyOrder)$
    $\mathbf{P}(x) \leftarrow$ library(Algebraic Operation$(g(x, \dot{x}))$ ▷ Add new polynomial terms only
    $\mathbf{p} \leftarrow$ Algebraic Operation$(f(t))$
    $\mathbf{P}_{train}(x), \mathbf{p}_{train}, \mathbf{P}_{test}(x), \mathbf{p}_{test} \leftarrow ratio$ ▷ Split data for training and test
    for $\lambda \in [\lambda_{min}, \lambda_{max}]$ **do** ▷ Search for the sparse regulator
      $MSE_{cv}(\lambda) \leftarrow$ SINDy & K $-$ fold$(\mathbf{P}_{train}(x), \mathbf{p}_{train}, \lambda)$
    end for
    $\lambda_{cv,min} \leftarrow min(MSE_{cv}(\lambda))$
    Model$(PolyOrder) \leftarrow$ SINDy$(\mathbf{P}_{train}(x), \mathbf{p}_{train}, \lambda_{cv,min})$ ▷ Sparse models
  end for
  $AIC(PolyOrder) \leftarrow$ AIC(Model$, \mathbf{P}_{test}(x), \mathbf{p}_{test})$ ▷ AIC of candidate models
  Model$(\mathbf{c}) \leftarrow$ Overall $min(AIC)$ ▷ Select the model with minimum AIC
  return Model $(\mathbf{c})$
end procedure

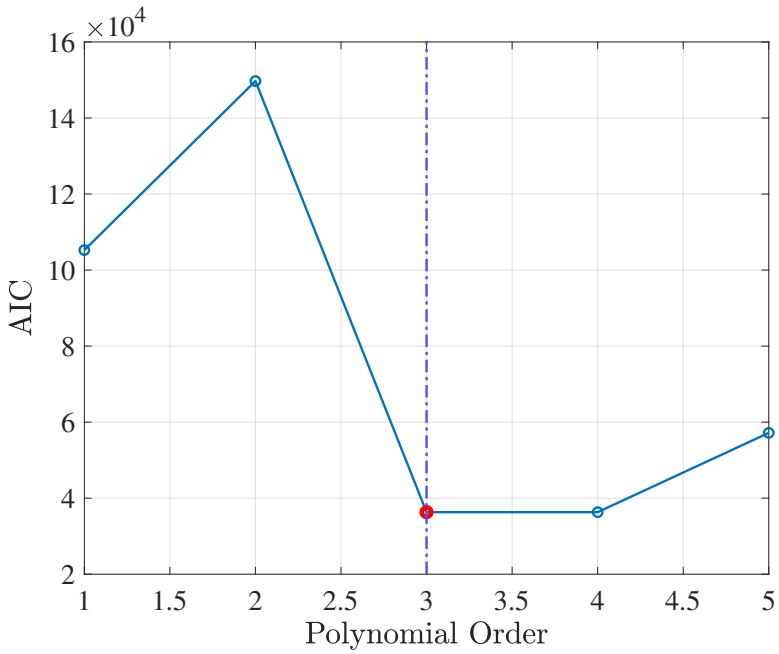

Figure 2.2: The AIC index of the model as a function of the order of polynomial for the system in Equation (2.18) when $\lambda_{\min}$ is used for all the models.

We follow the steps in Algorithm 1 to identify the best model for the data. The order of the polynomial starts from 1 and increases to 6. For simplicity, we have taken $N = M$ in the simulations. The time series generated from Equation (2.18) is equally divided into the training and test datasets. We choose $K = 5$ for the K-fold cross validation to train and select the candidate model by using the training dataset. The sparse representation of the model with a given order of the polynomial is further tested by using the test dataset. On the test dataset, the AIC index is also computed.

The results of the study are well summarized in Figures 2.2 and 2.3. Figure 2.2 shows the variation of the AIC index of the sparse model as a function of the order of the polynomial. The curve indicates that the AIC index reaches its minimum when the order is 3. This is consistent to the known order of the system.

Figure 2.3 shows the variation of the mean squared error MSE_{cv} of the K-fold cross validation for the model with the third order polynomial with the regulation parameter λ. The figure clears shows that there exists an optimal λ such that MSE_{cv} is minimum. This is true for all the orders of the polynomials we computed.

The resulting model is optimal in the sense of sparse representation of the SINDy. Table 2.1 compares the true parameters of the system with the estimated

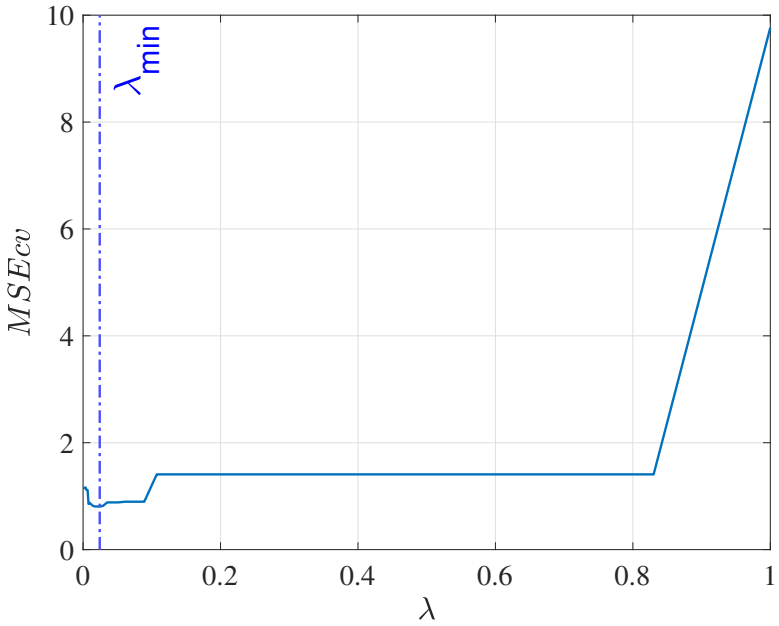

Figure 2.3: The mean squared error MSE_{cv} of the K-fold cross validation as a function of the regulation parameter λ for the system in Equation (2.18) when the model is the third order polynomial.

Table 2.1: The true and estimated coefficients of the polynomial terms in Equation (2.18).

Terms	True value	Estimated	Terms	True value	Estimated
$\ddot{x}$	2	1.9981	x	6	5.9857
x^2	0	0	x^3	0.1	0.1059
$\dot{x}$	1	0.9147	$\dot{x}^2$	0	0
$\dot{x}^3$	1	1.0064	$x\dot{x}$	0	0
$x\dot{x}^2$	0	0	$x\dot{x}^3$	0	0
$x^2\dot{x}$	0.2	0.1973	$x^2\dot{x}^2$	0	0
$x^2\dot{x}^3$	0	0	$x^3\dot{x}$	0	0
$x^3\dot{x}^2$	0	0	$x^3\dot{x}^3$	0	0

values of the parameters when the polynomial of the model is third order $N = M = 3$ when the AIC index is minimum. It is evident from the results that the proposed method for identifying the model of nonlinear dynamic systems is quite accurate and effective.

2.3 An Experimental Example

Here, we apply the proposed algorithm to identify the model of a multiple degree of freedom (MODF) system from experimental data. In particular, we consider a rotary flexible beam made by Quanser in the experimental study. Figure 2.4 depicts the setup. The flexible beam is driven by a DC servomotor with an input voltage V_m. The angle of the base holding the flexible beam is θ. The deflection angle of the beam is denoted as α. Table 2.2 lists the system parameters.

The beam has a moment of inertia J_l about the center of rotation and a mass m_l. It is rigidly fixed at one end and mounted on the servo motor. The servo motor and gear system have Coulomb friction with an unknown coefficient B_c and viscous damping coefficient B_{eq}. The beam has viscous damping due to the air with the coefficient B_l. The flexible beam has a linear stiffness K_s when the motion is small, and the stiffness becomes nonlinear when the excitation level is high.

The equations of motion for the system can be derived as [112],

$$\begin{aligned} &(J_{eq} + J_l)\ddot{\theta} + J_l\ddot{\alpha} + B_{eq}\dot{\theta} = \tau, \\ &J_l\ddot{\alpha} + J_l\ddot{\theta} + B_l\dot{\alpha} + k_s\alpha = 0 \\ &\tau = \frac{\eta_g k_g \eta_m k_t (V_m - k_g k_m \dot{\theta})}{R_m} \end{aligned} \tag{2.20}$$

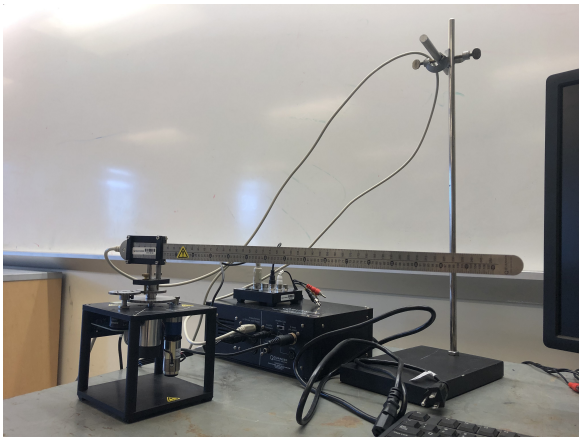

Figure 2.4: Quanser rotary flexible beam set up.

The system has two outputs θ and α under the input of voltage V_m. We assume that the torque τ is linearly dependent on voltage V_m, while the parameters of the motor are unknown. Table 2.2 shows the parameters provided by Quanser.

In terms of the control voltage, the equations of motion (2.20) can be written as

$$\begin{aligned} \frac{R_m}{\eta_g k_g \eta_m k_t}(J_{eq}\ddot{\theta} + (B_{eq} + \frac{\eta_g k_g^2 k_m \eta_m k_t}{R_m})\dot{\theta} - k_s\alpha) = V_m, \\ -\frac{R_m}{\eta_g k_g \eta_m k_t}(J_{eq}\ddot{\alpha} - (\frac{\eta_g k_g^2 k_m \eta_m k_t}{R_m} + B_{eq})\dot{\theta} + k_s\frac{J_l + J_{eq}}{J_l}\alpha) = V_m \end{aligned} \tag{2.21}$$

We assume that the restoring force of the system (2.21) can be represented by a sum of four polynomials of a single variable, i.e. θ^i, $\dot{\theta}^i$, α^j, and $\dot{\alpha}^j$, together with a friction term due to $\dot{\theta}$, without considering the cross-product terms. We should point out that the original model of the system (2.21) does not account for the effect of friction and nonlinearity, which exist in the experimental data. We assume that the restoring force of the system (2.21) can be represented by a sum of four polynomials of a single variable, i.e. θ^i, $\dot{\theta}^i$, α^j, and $\dot{\alpha}^j$, together with a friction term due to $\dot{\theta}$, without considering the cross-product terms.

Some signals contain the terms involving the first order derivative of the angle such as $\dot{\theta}^i$. We use the nonlinear state observer from the control studies to estimate the first-order derivatives. Let us take θ as an example. The second order observer with the gains β_1 and β_2 can be implemented with the following state equations.

$$\begin{aligned} \dot{z}_1 &= z_2 - \beta_1 e, \\ \dot{z}_2 &= -\beta_2 \text{fal}(e, \alpha, \delta) \end{aligned} \tag{2.22}$$

where

$$\text{fal}(e,\alpha,\delta) = \begin{cases} |e|^{\alpha}\cdot \text{sign}(e), & |e| > \delta \\ \frac{e}{\delta^{\alpha}}, & |e| \leq \delta \end{cases} \tag{2.23}$$

and the error $e = z_1 - \theta$ stands for the difference between θ and its observer estimate z_1. We have chosen $\alpha = 0.5$, $\delta = 0.05$, $\beta_1 = 100$ and $\beta_2 = 900$. According to [113, 114], the estimation error e converges to zero quickly. Consequently, by definition, z_2 is an accurate estimate of the first derivative $\dot{\theta}$. We split the data into two equal sets for training and test. We chosen $K = 10$ for the K-fold cross validation during the training. We have considered the order of the polynomials from 1 up to 4.

We again follow the steps in Algorithm 1 to identify the best model for the data. The results are summarized in Figures 2.5 and 2.6. Figure 2.5 shows the AIC indices for both the angles as a function of the order of the polynomials when the optimal regulation parameters λ_θ and λ_α are used. The figure clearly indicates that the optimal order of the polynomial for both the angles is three. This is consistent with the geometrical nonlinearity of the beam, which is typically the third order in the displacement variable.

Figure 2.6 shows the mean squared error MSE_{cv} of the K-fold cross validation for the third order polynomial model as a function of the regulation parameters λ_θ and λ_α for the system in Equation (2.21). The figure indicates the optimal values of the regulation parameters λ_θ and λ_α for both the angles α and θ.

The results in Figures 2.5 and 2.6 indicate the existence of an optimal model from the experimental data. Specifically, the following equations of motion are the optimal polynomial model of the system with the smallest AIC indices: $AIC_\theta = -1.0205 * 10^4$ and $AIC_\alpha = -8.833 * 10^3$.

$$a_{11}\ddot{\theta} + a_{12}\dot{\theta} + a_{13}\theta + a_{14}\theta^2 + a_{15}\theta^3 + a_{16}\dot{\alpha} + a_{17}\alpha + a_{18}\alpha^2 + a_{19}sign(\dot{\theta}) = V_m, \tag{2.24}$$

$$a_{21}\ddot{\theta} + a_{22}\dot{\theta} + a_{23}\theta + a_{24}\theta^2 + a_{25}\theta^3 + a_{26}\dot{\alpha} + a_{27}\alpha + a_{28}\alpha^2 + a_{29}sign(\dot{\theta}) = V_m, \tag{2.25}$$

where the terms $a_{19}sign(\dot{\theta})$ and $a_{29}sign(\dot{\theta})$ represent the Coulomb friction that exists in the experiment.

Table 2.3 compares the estimated parameters of Equations (2.24) and (2.25) with the ones provided by Quanser corresponding to the linear system of Equation (2.21). We should point out that the parameters provided by Quanser are nominal and may not represent the physical system accurately. If the nonlinear equations (2.24) and (2.25) of the identified model represents the physical system better in real time, the estimated parameters would be more reliable. One way to make this assessment is to compare the predicted responses of Equations (2.21), (2.24) and (2.25) with the experimentally measured response.

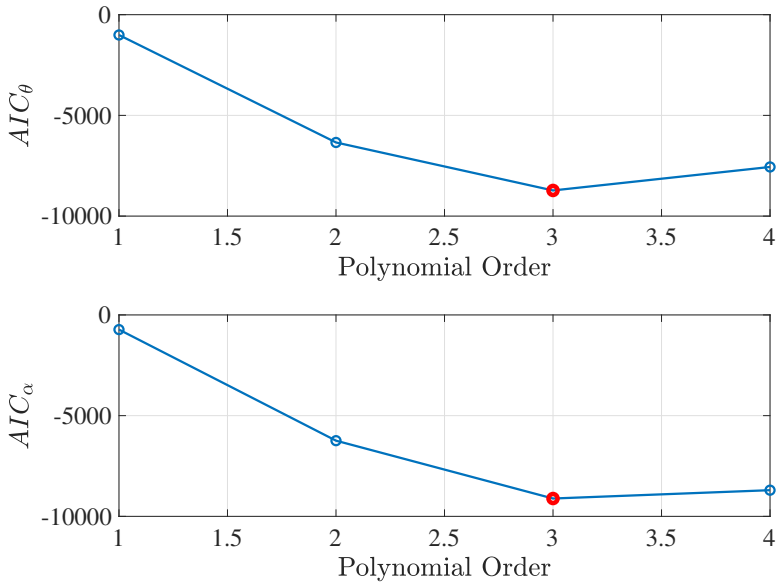

Figure 2.5: The AIC index of the model as a function of the order of polynomial for the rotatory flexible beam system in Equation (2.21) when $\lambda_{\theta,\min}$ and $\lambda_{\alpha,\min}$ are used for all the models.

Table 2.2: The parameters of the rotary flexible beam provided by Quanser [2].

Symbol	Description	Value
J_l	Cross sectional area moment of inertia	$0.0038 kg.m^2$
J_{eq}	Equivalent moment of inertia of the beam	$9.76 * 10^{-5} kg.m^2$
B_l	Damping coefficient	unknown
B_{eq}	Equivalent viscous damping	0.015 kg
k_s	Beam stiffness	$1.3 \frac{N.m}{rad}$
k_g	Gear ratio	70
k_t	Motor current torque constant	$7.68 * 10^{-3} \frac{N.m}{A}$
k_m	Motor back-emf constant	$7.68 * 10^{-3} \frac{v}{rad/s}$
R_m	Motor armature resistance	2.6Ω
η_g	Gearbox efficiency	0.9
η_m	Motor efficiency	0.69

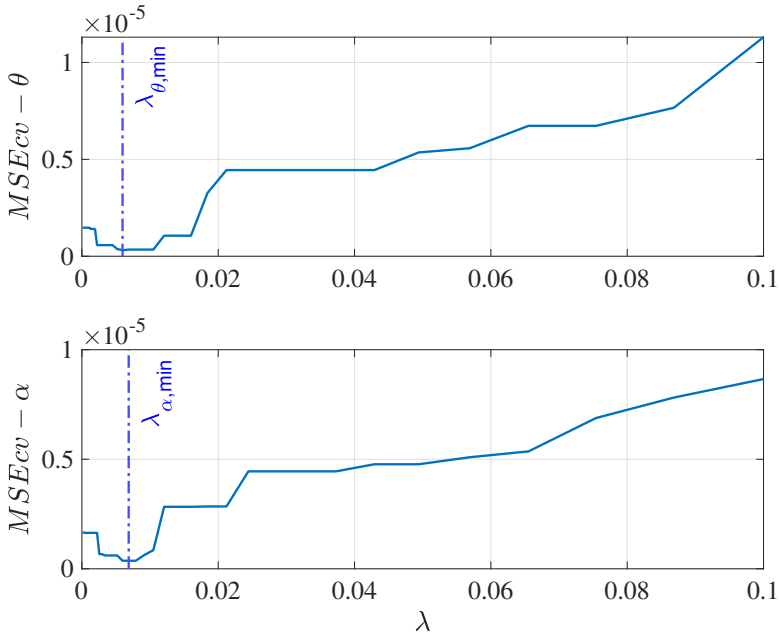

Figure 2.6: The mean squared error MSE_{cv} of the K-fold cross validation as a function of the regulation parameters λ_θ and λ_α for the rotatory flexible beam system in Equation (2.21) when the model consists of the third order polynomials.

Table 2.3: The estimated coefficients of the polynomial terms in Equations (2.24) and (2.25) for the rotary flexible listed to compare with the identified numbers. f_c is the coefficient of friction.

Eq. (2.24)	Quanser	Identified	Eq. (2.25)	Quanser	Identified
$\ddot{\theta}$	0.016231	0.0190	$\ddot{\alpha}$	-0.016231	-0.0118
θ	0	0.4151	θ	0	0.7113
θ^2	0	0.0933	θ^2	0	0.0935
θ^3	0	-1.5354	θ^3	0	-2.7442
$\dot{\theta}$	0.7323	0.6623	$\dot{\theta}$	0.7323	0.5890
$\dot{\theta}^2$	0	0	$\dot{\theta}^2$	0	0
$\dot{\theta}^3$	0	-0.0065	$\dot{\theta}^3$	0	0
α	-10.124	-6.3840	α	-15.672	-13.1712
α^2	0	5.8953	α^2	0	6.2922
α^3	0	0	α^3	0	0
$\dot{\alpha}$	0	-0.0420	$\dot{\alpha}$	0	-0.0872
$\dot{\alpha}^2$	0	0	$\dot{\alpha}^2$	0	0
$\dot{\alpha}^3$	0	0	$\dot{\alpha}^3$	0	0
f_c	0	0.1151	f_c	0	0.1474

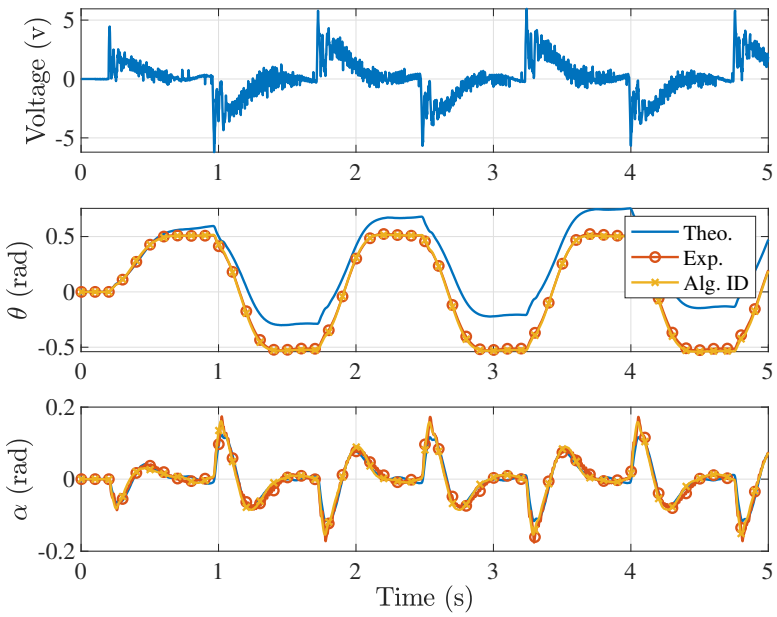

Figure 2.7: Comparison of the responses of Equations (2.21), (2.24) and (2.25) with the experimentally measured response. The response of θ follows a square trajectory with the amplitude of 30 degree and frequency 0.66 Hz. Top: Noisy Input voltage. Middle: θ angle response. Bottom: α angle response. Solid line: Equation (2.21). Solid line+circle: Experimental data. Solid line+cross: Equations (2.24) and (2.25).

Figure 2.7 compares the predicted responses of Equations (2.21), (2.24) and (2.25) with the experimentally measured response. The angle θ is driven to follow a square wave trajectory with frequency 0.66 Hz. The figure shows that there is a big difference between the response of the model of Equation (2.21) and the experimental measurements. This difference is most likely due to the fact that the model of Equation (2.21) does not include the Coulomb friction and nonlinear terms, both of which exit in the experimental data.

On the other hand, Equations (2.24) and (2.25) contain both the Coulomb friction and nonlinear terms. The Coulomb friction coefficients are estimated to be $a_{19} = 0.1151$ and $a_{29} = 0.1474$. The coefficients of nonlinear terms, e.g. a_{12} and a_{22}, are considerable. Generally, the estimated value of the coefficients of the linear terms are close to the nominal values provided by Quanser.

2.4 Conclusions

In this work, we have proposed a procedure to find the optimal polynomial model for nonlinear dynamic systems from experimental data. The model is optimal in the sense that it has a minimum order to fit the experimental data with the smallest number of terms. The model is selected by using the LASSO algorithm so that it has the smallest modeling error with respect to the order of the polynomial without overfitting and the minimum mean squared cross validation error. The algebraic method is used to create the signal matrix for estimating the parameters. Both simulated and experimental data of nonlinear dynamic systems have been used to validate the proposed method.

Chapter 3

IDENTIFICATION OF NONLINEAR DYNAMICAL SYSTEMS WITH TIME DELAY

In this chapter, we discuss our work in [115]. This work proposes an estimation technique to identify time delay and parameters of delayed nonlinear dynamical systems. We use Taylor expansion to parameterize the time delay in the motion equation and employ algebraic approach to make the algorithm resist to noise and independent from initial conditions. Moreover, the algorithm is equipped with a statistical approach to increase the accuracy of estimation. The technique is free from the limitations caused by structured entries and local minimum problems which usually happen in iterative estimation techniques. In this paper we assume the structures of the system and controller are both known.We use a nonlinear Duffing oscillator to motivate the development of the proposed algorithm. Experimental response time histories of a nonlinear flexible joint are used as an example to validate the efficiency and accuracy of the proposed algorithm. Furthermore, the experimental analysis shows the potential of the method for MIMO dynamical systems.

The rest of the paper is organized as the follows. Section 3.1 introduces the assumptions and formulates the mathematical problem of system identification. The techniques of algebraic data preprocessing and derivative estimations, sparse regression in combination with bootstrapping resampling, and cross validation are explained in Section 3.2. Section 3.3 presents an example of a simulated nonlinear mass-spring-damper system under a proportional control with time delay. The experimental validation of the proposed algorithm on the rotary flexible joint made by Quanser is presented in Section 3.4. Finally, Section 3.5 concludes the paper.

3.1 Problem Definition and Assumptions

Consider a closed-loop second order system given by,

$$m\ddot{x} + g(x, \dot{x}) = f(t) + b_1\dot{x}(t-\tau) + b_0 x(t-\tau) \tag{3.1}$$

where $m > 0$ is the mass of the system, the function $g(x, \dot{x})$ represents the nonlinear restoring or internal force of the system. The term $f(t)$ contains the reference information as well as external disturbances. The control consists of the output and

its first-order derivative with feedback gains b_0 and b_1, and a time delay τ. We make the following assumptions in this study.

Assumptions

1. Only the system response $x(t)$ is measured.
2. The system is exposed to random excitations included in $f(t)$.
3. The measurement contains Gaussian white noise with zero mean denoted as ϵ_x.
4. The system structure in terms of the function $g(x, \dot{x})$ is unknown.
5. The system is second order.

This paper is focused on nonparametric identification of closed-loop nonlinear dynamical systems with a control time delay.

3.2 The Proposed Method

For the restoring force $g(x, \dot{x})$ with an unknown structure, there are many candidate functions available to approximate it, such as polynomial, trigonometric, exponential functions or a combination of these functions. Because the polynomial is a popular base function to describe a wide range of dynamical systems, we use regular polynomials of x and $\dot{x}$ with time-invariant coefficients to explain the proposed method,

$$g(x, \dot{x}) = \sum_{i=0}^{N} \sum_{j=0}^{M} c_{ij} x^i \dot{x}^j \tag{3.2}$$

where c_{ij} are the unknown coefficients of the polynomial. Other functions can also be considered with the proposed method.

For the system without any prior knowledge, the orders N and M of the polynomial are unknown. This study develops an algorithm to identify the polynomial defined in Equation (3.2) from the response data such that it has a minimum number of terms and with the minimum orders N and M.

3.2.1 Algebraic Operation

Noise robustness has been always an inevitable part of system identification techniques as the noise is unavoidable in real data. Time delay caused by sensors and actuators is usually so small that it is difficult or even impossible to estimate correct values of time delay from noisy data. In addition to noise robustness, dynamical system identification methods require derivatives of time series of the measured response. Traditional finite difference methods to estimate derivatives can amplify

the noise. To deal with this issue, we propose to apply the algebraic operation to pre-process the data.

We use a nonlinear oscillator under a proportional control with gain k_p to illustrate the method.

$$\begin{aligned} &m\ddot{x} + c_1\dot{x} + c_2\dot{x}x^2 + c_3\dot{x}^3 + k_1x + k_3x^3 - k_px(t-\tau) = f(t) \\ &x(0) = x_0,\ \dot{x}(0) = v_0 \end{aligned} \tag{3.3}$$

where m denotes mass, c_1, c_2 and c_3 stand for damping coefficients, k_1 and k_3 are the stiffness coefficients. The system starts from initial conditions (x_0, v_0) which are generally unknown.

Remark: The input excitations must be chosen such that it can excite the system to exhibit rich dynamics in the output.

We introduce several short-hands for the nonlinear terms.

$$z_1 = \dot{x}x^2, \quad z_2 = \dot{x}^3, \quad z_3 = x^3 \tag{3.4}$$

Applying the Laplace transform to Equation (3.3), we have

$$\begin{aligned} &m\left\{s^2X(s) - sx_0 - v_0\right\} \\ &+ c_1\left\{sX(s) - x_0\right\} + k_1X(s) - k_pe^{-s\tau}X(s) = G(s) \end{aligned} \tag{3.5}$$

where

$$G(s) = F(s) - c_2Z_1(s) - c_3Z_2(s) - k_3Z_3(s), \tag{3.6}$$

$X(s)$ is the Laplace transform of $x(t)$, $Z_i(s)$ is the Laplace transform of $z_i(t)$ and $F(s)$ is the Laplace transform of $f(t)$.

Consider the Taylor expansion of the exponential term $e^{-s\tau}$

$$e^{-s\tau} = 1 - s\tau + \frac{s^2\tau^2}{2!} - \frac{s^3\tau^3}{3!} + ... \tag{3.7}$$

We should point out that keeping too many higher order terms may not be beneficial to the identification process. We only need to keep a sufficient number of terms to generate enough equations to determine the unknown parameters including time delay.

As an example, we keep the terms up to the third order and substitute the Taylor expansion in Equation (3.5).

$$\begin{aligned} &m\left\{s^2X(s) - sx_0 - v_0\right\} + c_1\left\{sX(s) - x_0\right\} \\ &+ k_1X(s) - k_p\left\{1 - s\tau + \frac{s^2\tau^2}{2!} - \frac{s^3\tau^3}{3!}\right\}X(s) = G(s) \end{aligned} \tag{3.8}$$

To eliminate the initial conditions, we differentiate Equation (3.8) with respect to s twice. To eliminate the derivative terms in time domain, we divide the resulting equation by s^3. Back to time domain, we obtain an equation of signals

$$\begin{aligned} P_f(t) = k_p\frac{\tau^3}{3!}P_1(t) + \left\{m - k_p\frac{\tau^2}{2}\right\}P_2(t) + \{c_1 + k_p\tau\}P_3(t) \\ + c_2P_4(t) + c_3P_5(t) + \{k_1 - k_p\}P_6(t) + k_3P_7(t) \end{aligned} \tag{3.9}$$

where

$$\begin{aligned} P_1(t) &= 6\int^{(2)} x(t) - 6\int tx(t) + t^2x(t) \\ P_2(t) &= 2\int^{(3)} x(t) - 4\int^{(2)} tx(t) + \int t^2x(t) \\ P_3(t) &= -2\int^{(3)} tx(t) + \int^{(2)} t^2x(t) \\ P_4(t) &= \int^{(3)} t^2z_1(t),\ P_5(t) = \int^{(3)} t^2z_2(t) \\ P_6(t) &= \int^{(3)} t^2x(t),\ P_7(t) = \int^{(3)} t^2z_3(t) \\ P_f(t) &= \int^{(3)} t^2f(t) \end{aligned} \tag{3.10}$$

Note that $\int^{(n)}\phi(t)$ denotes the multiple integral $\int_0^t\int_0^{\sigma_1}...\int_0^{\sigma_{n-1}}\phi(\sigma_n)d\sigma_n...d\sigma_1$. Let t_k $(k = 1, 2, ...n_t)$ be a set of sampled times. Define an error at time t_k as

$$\begin{aligned} e(k) = k_p\frac{\tau^3}{3!}P_1(t_k) + \left\{m - k_p\frac{\tau^2}{2}\right\}P_2(t_k) + \{c_1 + k_p\tau\}P_3(t_k) \\ + c_2P_4(t_k) + c_3P_5(t_k) + \{k_1 - k_p\}P_6(t_k) + k_3P_7(t_k) - P_f(t_k). \end{aligned} \tag{3.11}$$

Let us introduce an error vector $\mathbf{e}$, a parameter vector $\mathbf{c}$ and a force vector $\mathbf{p}$ as follows.

$$\begin{aligned} \mathbf{e} &= [e(1), e(2), ..., e(n_t)]^T, \\ \mathbf{c} &= [k_p\frac{\tau^3}{3!}, m + k_p\frac{\tau^2}{2}, c_1 - k_p\tau, c_2, c_3, k_1 + k_p, k_3]^T, \\ \mathbf{p} &= [P_f(t_1), P_f(t_2), ..., P_f(t_{n_t})]^T. \end{aligned} \tag{3.12}$$

Let $\mathbf{P}(x)$ be the $n_t \times 7$ data matrix determined by the measurement of $x(t)$ such that its $(k, j)^{th}$ component is given by $P_{k,j} = P_j(t_k)$. Then, Equation (3.11) can be written in the matrix form as

$$\mathbf{e}(\mathbf{c}) = \mathbf{P}(x)\mathbf{c} - \mathbf{p}. \tag{3.13}$$

To find the unknown parameters $\mathbf{c}$, we formulate a least mean square error problem as follows. For the following cost function,

$$J(\mathbf{c}) = \mathbf{e}^T\mathbf{e} = \|\mathbf{e}\|_2^2 \tag{3.14}$$

we find the parameter vector $\mathbf{c}$ such that J is minimized. The solution of the parameters $\mathbf{c}$ can be found by either the matrix inversion or by an iterative search algorithm. In general, when the numbers of unknown parameters and data points n_t are large, as is the case for multi-degree of freedom systems, it is better to use a global search algorithm to compute the parameters.

3.2.2 Sparse Representation

Assume that the displacement measurement $x(t)$ of the system in Equation (3.3) and external force $f(t)$ are available and contain random noises. We consider the polynomial model for the restoring force in Equation (3.2). Two questions immediately arise:

What are the minimum orders N and M?
Are all the terms in the polynomial needed?

To investigate answers for these questions, we apply the methods in statistical learning [110]. One popular way is to start with big enough orders N and M or a large library of functions which may contain non-polynomial functions, and penalize the terms with small coefficients through a sparse regulator by using the least absolute shrinkage and selection operator (LASSO) [110]. The recent studies of sparse identification of nonlinear dynamics (SINDy) have developed a computationally efficient algorithm to compute the solution of the sparse regression. LASSO regression proposes to add an L^1 regularization term to the cost function in Equation (3.14) to penalize the nonzero parameters that don't contribute to the system's dynamics

$$\begin{aligned} J_\lambda(\mathbf{c}) &= \|\mathbf{e}\|_2^2 + \lambda \|\mathbf{c}\|_1 \\ &= \|\mathbf{P}(x)\mathbf{c} - \mathbf{p}\|_2^2 + \lambda \|\mathbf{c}\|_1 \end{aligned} \tag{3.15}$$

where $\lambda > 0$ is a preselected positive number known as the sparse regulator and $\|\mathbf{c}\|_1 = \sum_i |c_i|$. The sparse regression algorithm in [45] is applied to find the parameter vector $\mathbf{c}$ in order to minimize J_λ in Equation (3.15). The algorithm is computationally efficient and robust to noise. It should be noted that the optimization problem in the framework of the LASSO is convex [15], which implies the existence of a unique optimal solution. The selection of the sparse regulator λ is critical. In Section 3.2.3, we explain how to employ cross-validation techniques to select a proper regulator value.

The LASSO penalizes the terms with small coefficients by regularization and keeps the important terms in the system model. In real world, time delay is usually small. Moreover, time delay appears in high order terms of the Taylor expansion. Therefore, the LASSO will penalize these terms, such as the coefficient of term $P_1(t)$ in Equation (3.11).

We can recover the terms involving time delay in the following way. After each sparse regression computed with the SYNDy algorithm, we keep all the terms involving time delay as well as the terms selected by the LASSO regulation. The resulting data matrix denoted by $\mathbf{P}_s(x)$ is substituted in Equation (3.15) to compute the updated coefficients $\mathbf{c}$.

3.2.3 Cross Validation and Bootstrapping

It is common to use cross-validation techniques from machine learning to determine the sparse regulation parameter λ. The value of λ in an finite interval is sampled. The cross-validation mean square error MSE_{CV} of the model over the test dataset is computed. The λ value which minimizes the cross-validation error is selected. The corresponding model is chosen as an optimal model with a proper balance of complexity and accuracy.

Let T_{cv} denote the set of time instances of the test signal to be used for cross validation. For a given regulation parameter λ, the mean square cross validation error MSE_{CV} of the model over all the validation datasets is defined as

$$MSE_{CV}(\lambda) = \frac{1}{n_t} \left\| \mathbf{P}(x(t \in T_{cv}))\mathbf{c}_\lambda - \mathbf{p}(t \in T_{cv}) \right\|_2^2 \tag{3.16}$$

where $\mathbf{c}_\lambda$ denotes the model parameters for the regulation parameter λ. MSE_{cv} is an implicit function of λ. The SINDy algorithm attempts to select λ on the Pareto front of the multi-objective optimization problem with the objectives being accuracy and complexity of the model. The elbow of the Pareto front parameterized by λ is often the choice [45].

Unfortunately, in most of cases, the elbow of the Pareto front is ambiguous due to existence of a cluster of candidate models near the elbow. The information criteria (IC) for the candidate models can help to rank and select a model with a proper trade-off between the accuracy and complexity [46]. Popular statistical examples of the information criteria include the Akaike information criterion (AIC), Bayesian information criterion (BIC), deviance information criterion (DIC) and minimum description length. The work reported in [46] makes use of a big data matrix $\mathbf{P}(x)$. The sparse regression procedure is repeated over all possible combinatorial subsets of the data matrix. The resulting models are ranked through IC scores. The model with the smallest score is selected.

However, a big data matrix $\mathbf{P}(x)$ may not always be available for real-world applications. In this work, we propose a search algorithm to determine the sparse

representation of the polynomial in Equation (3.2). We start with linear model when $N = M = 1$, and increase the polynomial order until the prediction error of the model over the test dataset reaches an acceptable low level and begins to increase. The error of the test dataset is defined as,

$$MSE_{test} = \frac{1}{n_t} \|\mathbf{P}(x(t \in T_{cv}))\hat{\mathbf{c}} - \mathbf{p}(t \in T_{cv})\|_2^2 \tag{3.17}$$

where $\hat{\mathbf{c}}$ stands for the estimated vector of coefficients. There are different ways to generate training and test datasets. When the SINDy algorithm is applied to simulation examples, we can generate rich datasets to train and validate the model by considering the system responses for different initial conditions and excitations. In this work, we assume that the dataset consists of two long time series of the system response. One is used for training and another for cross-validation. The data matrices $\mathbf{P}(x)$ are generated for different orders of the polynomial. The SINDy algorithm selects the sparse model built on the training data while the regularization parameter λ is selected with the help of cross validation on the test dataset.

Real measurements often contain noises, and can have outliers and missing data. Here, we propose to combine the sparse regression with bootstrapping in order to develop robust sparse regression. In particular, we consider K bootstrap sample vectors containing L elements of the original data points. Each vector is generated by uniform sampling of the data with replacement. For each bootstrap sample vector, the sparse regression is applied to identify the model. Finally, the parameters of the model is the average of the K estimated coefficient vectors $\hat{\mathbf{c}}_l$.

$$\tilde{\mathbf{c}} = \frac{1}{K} \sum_{l=1}^{K} \hat{\mathbf{c}}_l \tag{3.18}$$

The standard deviation of estimated coefficient vectors $\hat{\mathbf{c}}_l$ is computed to study the variation of parameter estimation.

Algorithm 2 summarizes the procedure.

Algorithm 2 Bootstrapping sparse regression algorithm for time-delay estimation.

procedure TIME-DELAY ESTIMATION($X_{train}, f_{train}, X_{test}, f_{test}, N, K, L, \lambda_{min}, \lambda_{max}$)
  for $PolyOrder \in [1, N;\ 1, M]$ **do**
    $g(x, \dot{x}) \leftarrow x, \dot{x}, PolyOrder$ ▷ Generate candidate polynomial restoring force
    $\mathbf{P}(x)_{train}, \mathbf{P}(x)_{test} \leftarrow$ library(Algebraic Operation($g(x, \dot{x})$))
    $\mathbf{p}_{train}, \mathbf{p}_{test} \leftarrow$ Algebraic Operation($f_{train}(t), f_{test}(t)$)
    for $k \in [1, K]$ **do** ▷ Generate uniformly random sampling with replacement
      $\mathbf{P}(x)_{k,train}, \mathbf{p}_{k,train} \leftarrow$ Bootsraping Sampling($\mathbf{P}(x)_{train}, \mathbf{p}_{test}, L$)
      for $\lambda \in [\lambda_{min}, \lambda_{max}]$ **do** ▷ Search to find the sparse regulator
        $MSE(\lambda) \leftarrow$ SINDy($\mathbf{P}(x)_{k,train}, \mathbf{p}_{k,train}, \lambda$)
      end for
      $\lambda_{min} \leftarrow min(MSE(\lambda))$
      Model($PolyOrder, k$) $\leftarrow$ SINDy($\mathbf{P}(x)_{k,train}, \mathbf{p}_{k,train}, \lambda_{min}$)
      if $\mathbf{c}_\tau = 0$ **then**
        Model($PolyOrder, k$) $\leftarrow$ OLS($\mathbf{P}_s(x)_{k,train}, \mathbf{p}_{k,train}$)
      end if
    end for
    Model($PolyOrder$) $\leftarrow$ Average(Model($PolyOrder, k$))
    $MSE_{test}(PolyOrder) \leftarrow MSE$(Model($PolyOrder$), $\mathbf{P}_{test}(x), \mathbf{p}_{test}$)
  end for
  Model($\mathbf{c}$) $\leftarrow$ Overall $min(MSE_{test})$ ▷ Select the model with minimum test error
  return Model ($\mathbf{c}$)
end procedure

Remarks

Some remarks on the estimation error and the stability of the system are in order. The estimation error of the coefficients $\mathbf{c}$, in particular, time delay τ, can be attributed to two sources: the truncation error of the Taylor expansion and the regression error.

While keeping more higher order terms of the Taylor expansion helps to reduce truncation error, we may run the risk of instability of the truncated model. We have found that the third order term is a good compromise for accuracy, complexity and stability.

3.3 Simulated Example

To demonstrate the proposed algorithm, we consider the second-order oscillatory system with nonlinear stiffness and damping in Equation (3.3). The following external forces are used to generate the training and test datasets.

$$f(t)_{Train} = 10\sin(4t) + 40\sin(4t^2) + \epsilon_f \tag{3.19}$$

$$f(t)_{Test} = 10\sin(2t) + 40\sin(8t) + \epsilon_f \tag{3.20}$$

For the training dataset, the system starts from the initial conditions $x_0 = 1$ and $v_0 = 0$. For the test dataset, the initial conditions are $x_0 = -1$ and $v_0 = 0.5$. The excitation contains a normally distributed random noise ϵ_f with zero mean and standard deviation $\sigma_{\epsilon_f} = 0.01$. We assume that the sensors have measurement noises such that the system output is given by $x(t) + \epsilon_x$ where ϵ_x is the normally distributed random noise with zero mean and standard deviation $\sigma_{\epsilon_x} = 0.01$.

Table 3.2 lists the system parameters used in the simulation. A proportional control with gain $k_p = 2$ is considered. We employ the third order of Taylor expansion of the time delay $x(t-\tau)$. The order of expansion is selected such that there are enough equations to solve for the unknown parameters of the system. Following Algorithm 2, we create the data matrix $\mathbf{P}(x)$ with polynomials of orders from 1 to 6 and $N = M$. For each polynomial model, 30 bootstrapping samples with the ratio of 50 percent of the total data are selected. 50 values of λ are sampled logarithmically in the range from 10^{-6} to 10^{0}.

Because the first term includes the time delay, we constrain the LASSO algorithm to keep it from being penalized and removed in the process of searching for sparse representation. The average of sparse polynomials out of all the bootstrapping samples is taken as the final result.

Figure 3.1 shows the variation of the cross validation error MSE_{cv} as a function of the order N of the polynomial. It is seen from the figure that the cross validation error has a large drop when approaching $N = 3$ and stays at the same range for $N = 4$ and starts increasing from $N = 5$. We choose $N = 3$ as the optimal order of the polynomial when the minimum validation error occurs and the order is

Table 3.1: The true and estimated parameters of the simulation example.

Terms	Parameters	Exact Value	Estimated	Standard Deviation $*10^{-3}$
$\dddot{x}$	$k_p\frac{\tau^3}{6}$	0.0026667	0.001515	0.0125
$\ddot{x}$	$m - k_p\frac{\tau^2}{2}$	0.96	0.9611	0.0137
x	$k_1 - k_p$	4	3.9830	0.1612
x^2	0	0	0	0
x^3	k_3	2	2.0041	0.0943
$\dot{x}$	$c_1 + k_p\tau$	1.4	1.3922.	0.2896
$\dot{x}^2$	0	0	0	0
$\dot{x}^3$	c_3	1	0.9973	0.0432
$x\dot{x}$	0	0	0	0
$x\dot{x}^2$	0	0	0	0
$x\dot{x}^3$	0	0	0	0
$x^2\dot{x}$	c_2	1	1.0074	0.0961
$x^2\dot{x}^2$	0	0	0	0
:	:	:	:	:
$x^3\dot{x}^3$	0	0	0	0

minimum. Table 3.1 lists the estimated coefficients of the signals in Equation (3.9). The proposed algorithm has successfully detected the sparse terms and precisely estimated the coefficients. The time delay and parameters of the original system are calculated and listed in Table 3.2 together with the known values. The accuracy of the estimated parameters is quite acceptable.

Remarks

A remark on small values of the standard deviation in Table 3.1 is in order. This fact is an indication that the randomness of the data is not significant so that the bootstrapping samples are not sufficiently different. We have checked the results several times to confirm this observation.

3.4 Experimental Example

In this section, we employ a flexible joint experimental setup, made by Quanser, to prove the efficiency of the method for experimental data from a single-input-multiple-outputs (SIMO) dynamical system. The joint is connected to the base through two springs with equal stiffness k_s and the base is fixed to the servo motor as Figure 3.2 shows. The servo motor with input voltage V_m generates a

Table 3.2: The parameters of the mass-spring-damper system for the simulation example.

Parameters	Selected Value	Estimated Value
m	1	0.9885
k_1	6	5.9830
k_3	2	2.0041
c_1	1	1.0609
c_2	1	1.0074
c_3	1	0.9973
τ	0.2	0.1656

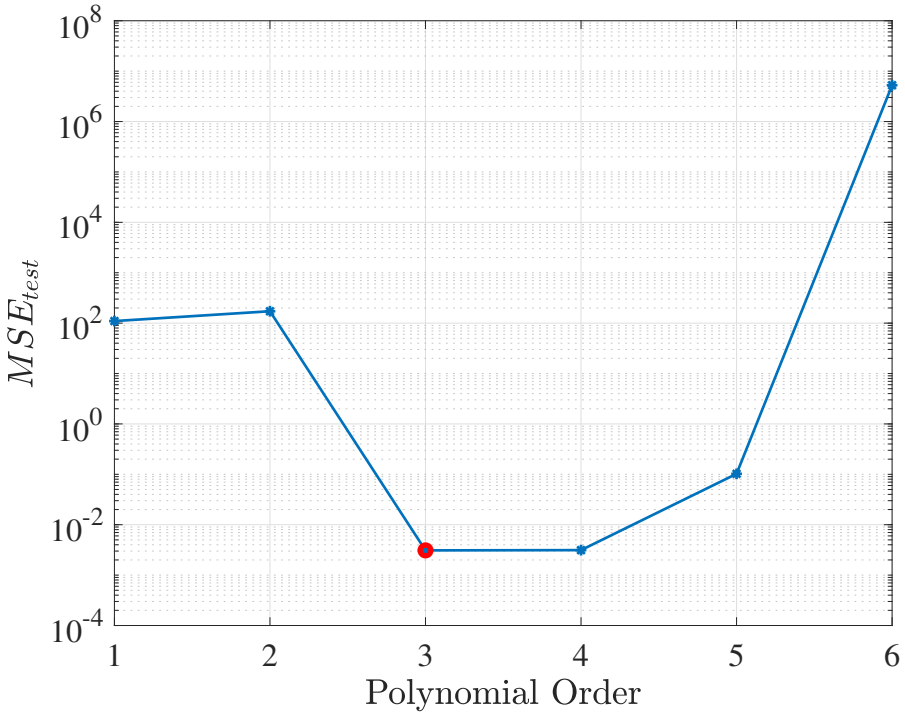

Figure 3.1: The cross validation test error for the mass-spring-damper system.

Figure 3.2: The flexible joint set up by Quanser [1].

torque M to turn the base with a rotation angle θ. The deflection angle of the joint relative to the base is α. The moment of inertia of the base is J_{eq}. The viscous damping coefficient of the base is B_{eq}. J_l stands for the moment of inertia of the joint. The equations of motion for the system can be derived as,

$$
\begin{aligned}
&(J_{eq} + J_l)\ddot{\theta} + J_l\ddot{\alpha} + B_{eq}\dot{\theta} = \tau_M, \\
&J_l\ddot{\alpha} + J_l\ddot{\theta} + B_l\dot{\alpha} + k_s\alpha = 0, \\
&\tau_M = \frac{\eta_g k_g \eta_m k_t (V_m - k_g k_m \dot{\theta})}{R_m} \equiv a \cdot V_m + b \cdot \dot{\theta}, \\
&a = \frac{\eta_g k_g \eta_m k_t}{R_m}, \quad b = -\frac{\eta_g k_g \eta_m k_t k_g k_m}{R_m},
\end{aligned}
\tag{3.21}
$$

where torque τ_M is linearly dependent on the servo motor voltage V_m and $\dot{\theta}$. We assume that the motor parameters are known. Table 3.3 lists the parameters of the motor and joint provided by Quanser. We should point out that these numbers may not be exactly the same as the physical system parameters. Therefore, we should treat these numbers as a reference.

A proportional control with gain k_p adjusts the motor voltage to make the joint follow the desired trajectory of the angle θ while the deflection angle α stays minimum. The control is given by

$$V_m = k_p * (\theta_d - \theta) \tag{3.22}$$

Table 3.3: The parameters of the rotary flexible link provided by Quanser [2].

Symbol	Description	Value
J_l	Link moment of inertia	$0.0038kg.m^2$
J_{eq}	Equivalent moment of inertia	$9.76 * 10^{-5}kg.m^2$
B_l	Flexible link torsional damping	Not available
B_{eq}	Equivalent torsional viscous damping	$0.015N.m/rad/s$
k_s	Link torsional stiffness	$1.3\frac{N.m}{rad}$
k_g	High gear total gear ratio	70
k_t	Motor current torque constant	$7.68 * 10^{-3}\frac{N.m}{A}$
k_m	Motor back-emf constant	$7.68 * 10^{-3}\frac{v}{rad/s}$
R_m	Motor armature resistance	2.6Ω
η_g	Gearbox efficiency	0.9
η_m	Motor efficiency	0.69

The control is implemented in Simulink/MATLAB. The outputs of the system are θ and α. A time delay is introduced in the feedback signal θ.

In this example, we employ the second order Taylor expansion of the time delay term.

$$\theta(t-\tau) = \theta(t) - \tau\dot{\theta}(t) + \frac{\tau^2}{2}\ddot{\theta}(t) \tag{3.23}$$

The closed-loop system can be written as,

$$(J_{eq} + ak_p\frac{\tau^2}{2})\ddot{\theta} + (B_{eq} - ak_p\tau)\dot{\theta} - B_l\dot{\alpha} - k_s\alpha = ak_p(\theta_d - \theta) + b\dot{\theta}, \tag{3.24}$$

$$J_{eq}\ddot{\alpha} - ak_p\frac{\tau^2}{2}\ddot{\theta} + (ak_p\tau - B_{eq})\dot{\theta} + \frac{B_l(J_l + J_{eq})}{J_l}\dot{\alpha} + \frac{k_s(J_l + J_{eq})}{J_l}\alpha = -ak_p(\theta_d - \theta) - b\dot{\theta}. \tag{3.25}$$

We should point out that the tracking performance of the closed-loop system will not be as good as we would like to have. This is because the proportional control alone is not adequate and the time delay further deteriorates the performance. The purpose of choosing this control with time delay is simply for generating the data in a stable manner.

To generate the training dataset, we select a desired trajectory θ_d as

$$\theta_d = 2\sin 2\pi f_1 t \tag{3.26}$$

where the frequency $f_1 = 0.66Hz$ is for the desired trajectory θ_d. To generate the test data, we choose a square-wave signal for the desired trajectory θ_d with the same amplitude and frequency as in Equation (3.26). We introduce a $0.2s$ delay in the control such that the closed-loop system is stable.

Figures 3.3 and 3.4 show the motor input voltage V_m and the sample responses of θ and α, The time series is one minute long with a sample time $0.001s$. For simplicity, we assume that for each coordinate θ and α, the restoring force can be represented by a sum of four polynomials of a single variable, i.e. θ^i, $\dot{\theta}^i$, α^j, and $\dot{\alpha}^j$, together with a friction term related to $\dot{\theta}$, without considering the cross-product terms. The second coefficient in Equation (3.25), $ak_p\frac{\tau^2}{2}$, includes the time delay. We apply the constraint to the LASSO algorithm to keep it from being penalized and removed.

Some signals in the matrix $\mathbf{P}(x)$ contain the first order derivative of the angles $\dot{\theta}$ and $\dot{\alpha}$. Recall that the real measurements contain noises. We use the nonlinear state observer from the control studies to estimate the first-order derivatives without amplifying the noise in the computation [113, 114]. The second order observer with gains β_1 and β_2 is defined by the following state equations.

$$\begin{aligned} \dot{z}_1 &= z_2 - \beta_1 e, \\ \dot{z}_2 &= -\beta_2 \text{fal}(e, \alpha, \delta) \end{aligned} \tag{3.27}$$

where

$$\text{fal}(e, \alpha, \delta) = \begin{cases} |e|^\alpha \cdot \text{sign}(e), & |e| > \delta \\ \frac{e}{\delta^\alpha}, & |e| \leq \delta \end{cases} \tag{3.28}$$

and the error $e = z_1 - \theta$. We have chosen $\alpha = 0.5$, $\delta = 0.05$, $\beta_1 = 100$ and $\beta_2 = 900$. According to [113, 114], the estimation error e converges to zero quickly. Consequently, by definition, z_2 is an accurate estimate of the first order derivative $\dot{\theta}$. Figures 3.3 and 3.4 show the plot of estimated $\dot{\theta}$ and $\dot{\alpha}$ for both training and test dataset.

To train the model, we consider 10 bootstrapping samples with the length of 50 percent of the training dataset and search the polynomial order from 1 to 7. For the sparse regulator, 100 values of λ are sampled logarithmically in the range from 10^{-10} to 10^0.

Figure 3.5 shows the variation of the cross validation error as a function of the order of polynomials. The results suggest that for both coordinates θ and α, the polynomial order $N = 5$ leads to the minimum cross validation error. This is the optimal order for the polynomials.

Table 3.4 lists the polynomial terms and the corresponding coefficients and associated standard deviation for the trained model of polynomial order $N = 5$. We should mention that the reported results have been round to 10^{-5}. The estimated value of Coulomb friction f_c is not negligible. It can play an important role in the

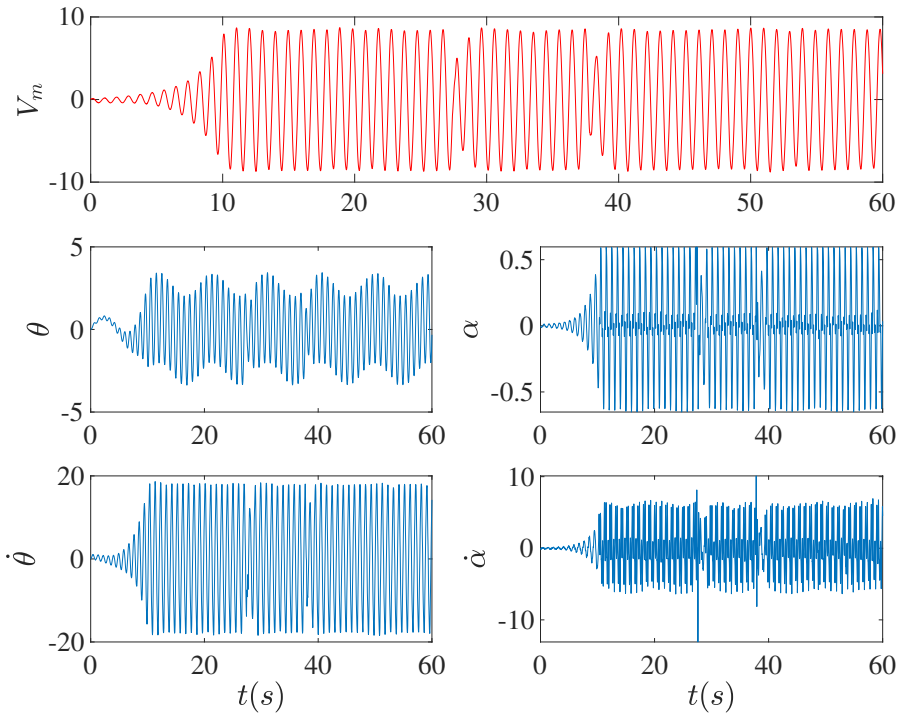

Figure 3.3: The system response to track a sinusoidal trajectory θ_d with amplitude 2 radian and frequency 0.66 Hz.

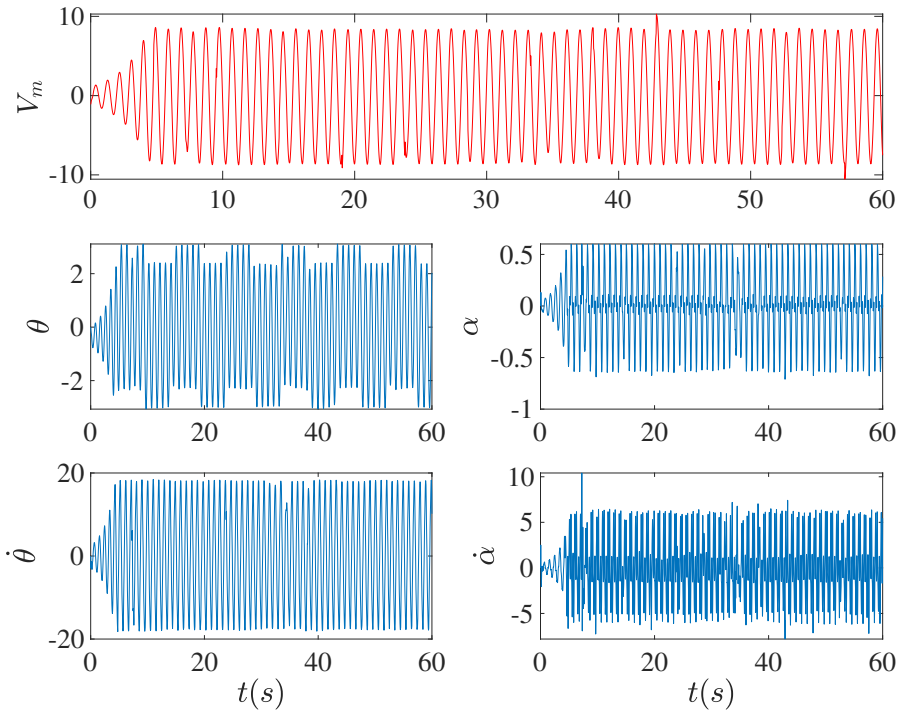

Figure 3.4: The system response to a square-wave signal θ_d with amplitude 1 radian and frequency 0.66 Hz.

system dynamics. We should point out that the linear model in Equation (3.21) does not include the friction term, although the actual device always has friction. The friction is the primary reason for the discrepancy between the predicted response by Equation (3.21) and the actual measurements.

Figures 3.3 and 3.4 indicate that the responses of the closed-loop system are not really tracking the references accurately. This is partly due to the effect of time delay and also the poor control design, as discussed earlier. The oscillatory responses of the closed-loop system are obviously responsible for the high order terms of the polynomials, particularly for the deflection angle α, as can be seen in Table 3.4. This highlights the ability of the proposed system identification algorithm to estimate the time delay and to identify the nonlinearities in the system when the linear models are no longer adequate.

The motor parameter a is known and k_p is the given control gain. Hence, from the coefficient $-ak_p\frac{\tau^2}{2}$, the time delay can be calculated and is listed in Table 3.5 in comparison with the time delay we introduced to the control. The values of the parameters in Table 3.5 fall in a wide range from 0.002 to 2. Hence, the parameters can be serval order of magnitudes apart, which makes it difficult to accurately estimate all the parameters in the presence of unwanted noises. This experimental study strongly demonstrates the robustness of the proposed algorithm to noises.

3.5 Conclusions

In summary, we have demonstrated that the proposed algorithm is effective to obtain governing equations of nonlinear dynamical systems with time delay from noisy experimental data. It can accurately estimate the time delay in the feedback control. For the first time, this study extends the sparse regression to nonlinear dynamical systems with time delay. We have equipped the sparse regression with an algebraic signal pre-processing and a nonlinear state observer. These operations are essential to compute the needed derivatives of measured time series and other signals without the need to know initial conditions, and to filter noises due to random excitations and measurements. Both simulation and experimental results have been used to validate the algorithm. The algorithm demonstrates excellent performances in identification.

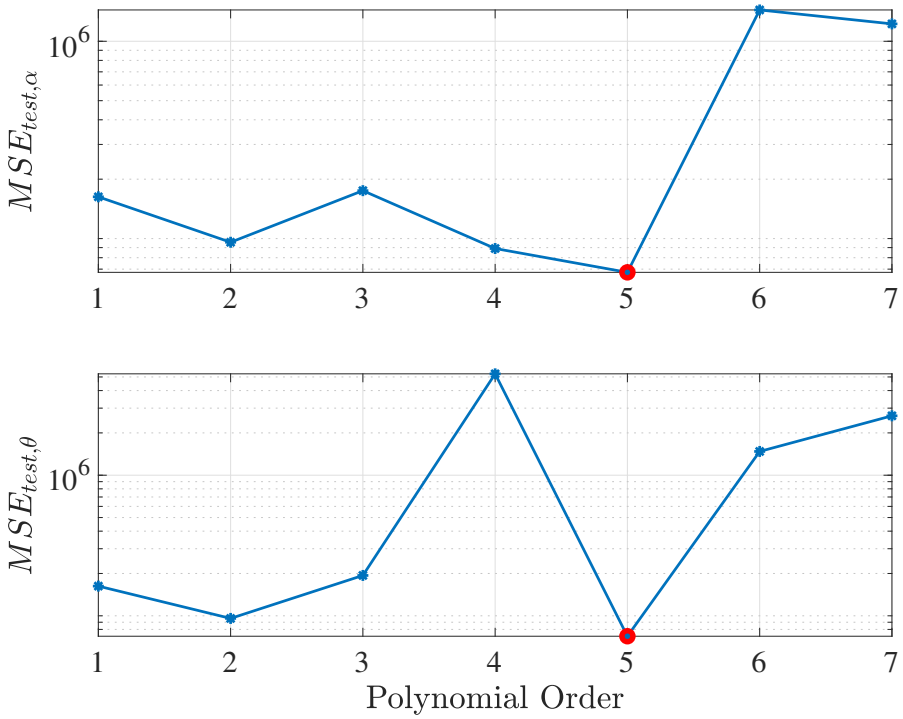

Figure 3.5: The cross-validation error of the rotary flexible joint.

Table 3.4: The θ and α terms, their coefficients and the standard deviations of the estimation for the rotary flexible joint.

Terms in θ-equation	Estimated	Standard Deviation	Terms in α-equation	Estimated	Standard Deviation
$\ddot{\theta}$	0.0061	0.	$\ddot{\theta}$	0.0012	0.0001
$\ddot{\alpha}$	–	–	$\ddot{\alpha}$	-0.0061	0
θ	0.0625	0.0026	θ	-0.0603	0.0029
θ^2	0.0003	0.0026	θ^2	0.0003	0.0018
θ^3	-0.0254	0.0009	θ^3	0.024	0.0010
θ^4	-0.0004	0.0006	θ^4	0.0003	0
θ^5	0.0019	0.0034	θ^5	-0.0019	0
$\dot{\theta}$	-0.0922	0.0002	$\dot{\theta}$	0.0920	0.0036
$\dot{\theta}^2$	0.0002	0	$\dot{\theta}^2$	-0.0002	0.0002
$\dot{\theta}^3$	0.0004	0	$\dot{\theta}^3$	-0.0004	0
$\dot{\theta}^4$	0	0	$\dot{\theta}^4$	0	0
$\dot{\theta}^5$	0	0	$\dot{\theta}^5$	0	0
α	-1.1976	0.0139	α	1.2483	0.0128
α^2	0.9040	0.0896	α^2	-0.9458	0.0991
α^3	-2.3784	0.2446	α^3	2.4237	0.2634
α^4	-0.7031	0.2916	α^4	0.6930	0.3149
α^5	8.9105	0.5177	α^5	-9.0823	0.5629
$\dot{\alpha}$	-0.0650	0.0005	$\dot{\alpha}$	0.0644	0.0008
$\dot{\alpha}^2$	-0.0215	0	$\dot{\alpha}^2$	0.0216	0.0005
$\dot{\alpha}^3$	0.0001	0	$\dot{\alpha}^3$	-0.0001	0
$\dot{\alpha}^4$	0.0004	0	$\dot{\alpha}^4$	-0.0004	0
$\dot{\alpha}^5$	0	0	$\dot{\alpha}^5$	0	0
f_c	0.0516	0.0083	f_c	-0.0477	0.0084

Table 3.5: The nominal and estimated values of the parameters of the rotary flexible link by Quanser [2].

Symbol	Nominal Values	Estimated values
$J_{eq} + ak_p\frac{\tau^2}{2}$	0.0102	0.0061
$B_{eq} - ak_p\tau$	-0.0662	-0.0922
k_s	1.3	1.1976
J_{eq}	0.00208	0.0012
B_l	Small unknown	0.0650
$ak_p\frac{\tau^2}{2}$	0.008121	0.0061
$ak_p\tau - B_{eq}$	0.0662	0.0920
$\frac{k_s(J_l+J_{eq})}{J_l}$	2.0501	1.2483
$\frac{B_l(J_l+J_{eq})}{J_l}$	Small unknown	0.064
τ	0.2	0.1734

Chapter 4

DATA-DRIVEN CONTROL OF NONLINEAR SYSTEMS FOR TRAJECTORY TRACKING

Data-driven control systems are a broad family of control systems, in which the identification of the process model and/or the design of the controller are based entirely on experimental data collected from the plant. Model-based controller designs highly depend on the model of the systems. However, it is still difficult to find a simple yet reliable model for a physical system, that includes only those dynamics of the system that are of interest for the control specifications. In this chapter, we investigate how our system identification algorithm delivers reliable and accurate models of the system to design model-based controllers.

We first start with the robust sliding model controller and add the concept of flat-output in our design for under-actuated systems. In the next design, we investigate the optimal control design procedure for the identified models. Date-driven models usually include nonlinearities to precisely describe the dynamics and physics of the systems. Furthermore, the identified models are mostly high-dimensional. Therefore, in indirect data-driven control systems, the designers are usually dealing with high-dimensional nonlinear controller designs. Optimal control design mostly focuses on solving Hamilton-Jacobi-Bellman (HJB) equation. The solution is mostly difficult or even infeasible for nonlinear and high-dimensional models. In this chapter, we also introduce an algorithm to solve HJB equation for our system identification method.

4.1 Problem Statement

The main goal of this chapter can be formulated as the control design of the identified model of rotary flexible link (RFL) in Eq. 2.24 and 2.25. The identified model can be rewritten as the state-space form in Equation (4.1),

$$\dot{x}(t) = f(x(t)) + g(x(t))u(x(t)); x(t_0) = x_0 \tag{4.1}$$

where t_0 is the initial time and x_0 is the initial state. The state $x(t)$, the drift matrix f and control matrix g are defined as,

$$\begin{aligned} x(t) &= [\theta, \alpha, \dot{\theta}, \dot{\alpha}]^T \\ f(x(t)) &= [0, 0, f_\theta(x(t)), f_\alpha(x(t))]^T \\ g &= [0, 0, B_\theta, B_\alpha]^T \ ; \ B_\theta = \frac{1}{a_{11}}, B_\alpha = \frac{1}{a_{21}} \end{aligned} \tag{4.2}$$

the control input $u(t) \in R$ is the servo-motor voltage V_m. The function $f(x)+g(x)u$ is Lipschitz continuous on a set $\Omega \subseteq R^4$ that contains the origin, and the system is stable on Ω. The identified mathematical model f_θ and f_θ include the nonlinear terms of stiffness, viscous damping and Coulomb friction. The control goal of a RFL is to keep the link tip tracking a desired trajectory while suppressing the undesired vibration. In fact, the angular location θ should follow the desired θ_d and the deflection angle α is kept zero. Therefore, the reference state is defined as,

$$x_d(t) = [\theta_d, 0, 0, 0]^T \tag{4.3}$$

the system has only one control input V_m to control two degree-of-freedom θ and α. Therefore, the system falls into the under-actuated control systems.

4.2 Flat-Output Sliding Mode Control for Identified Models

4.2.1 Design procedure

In this section, we discuss the mathematical background our proposed flat-output based sliding model controller. First, we give a brief definition of flat systems and explain how we consider the flat output of our identified nonlinear model in Eq(4.2). Then, we detail in our procedure to design the flat output-based sliding mode controller.

4.2.1.1 Flat Systems Model

Let us consider the general state-space model of nonlinear systems in Equation (4.1) and let $p = (x_e, u_e)$ denotes an operation or equilibrium point of the system. The tangent linearization of the system can be obtained as

$$\begin{aligned} \dot{x}_\delta &= (\frac{\partial f}{\partial x}|_p + \frac{\partial g}{\partial x}u(t)|_p)x_\delta + g|_p u_\delta \\ &= A_\delta x_\delta + B_\delta u_\delta \end{aligned} \tag{4.4}$$

where $x_\delta = x - x_e$, $u_\delta = u - u_e$, B_δ is the control influence vector, and A_δ is a matrix of the linear state. The tangent or Jacobian linearized system in Equation (4.4) is flat if, and only if, it is controllable, which means there exists a function y_f

of the output y_δ such that all the states x_δ are expressible in terms of y_f and its finite successive time derivatives [86, 87].

To find the flat output y_f for RFL, we consider the tangent linearization of Equation (4.2) around the equilibrium $\theta = 0$, $\dot{\theta} = 0$, $\alpha = 0$, $\dot{\alpha} = 0$, and $u = 0$. Let us ignore the friction terms such that,

$$\ddot{\theta} = a_{11}\alpha + B_\theta u \tag{4.5}$$
$$\ddot{\alpha} = a_{21}\alpha + B_\alpha u \tag{4.6}$$

then, a local differentially flat output y_f can be found analytically as

$$\begin{aligned} y_f &= C_\theta\theta + C_\alpha\alpha; \\ C_\theta &= \frac{B_\alpha}{a_{11}B_\alpha - a_{21}B_\theta} \\ C_\alpha &= -\frac{B_\theta}{a_{11}B_\alpha - a_{21}B_\theta} \end{aligned} \tag{4.7}$$

We can show that all the incremental states of the system can be expressible in terms of y_f and its time derivatives,

$$\begin{aligned} \alpha &= \ddot{y}_f \\ \dot{\alpha} &= y_f^{(3)} \\ \theta &= \frac{1}{C_\theta}(y_f - C_\alpha\alpha) \\ \dot{\theta} &= \frac{1}{C_\theta}(\dot{y}_f - C_\alpha\dot{\alpha}) = \frac{1}{C_\theta}(\dot{y}_f - C_\alpha y_f^{(3)}) \end{aligned} \tag{4.8}$$

Equation (4.8) suggests that a nonsingular state transformation is generated by the flat output y_f. The term $y^{(n)}, n = 3, 4., , ,$ denote the n order derivative of y. Using the the state in Equation (4.8) and Equation (4.5), the linearized system equations can be written in the norm form,

$$y_f^{(4)} = a_{21}\ddot{y}_f + B_\alpha u \tag{4.9}$$

to involve the nonlinearities, we consider the state $\ddot{\theta} = f_\theta + B_\theta u$ in Equation (4.2), therefore, the flat output for the nonlinear system with the states in Equation (4.8) is as,

$$y_f^{(4)} = \frac{1}{C_\alpha}\ddot{y}_f + \frac{B_\alpha}{B_\theta}f(y_f, \dot{y}_f, \ddot{y}_f) + B_\alpha u \tag{4.10}$$

with $f(.)$ representing all the nonlinearities affecting the flat output dynamics. The preferred route in sliding mode control is to overcome this quantity. This type of relation is fundamental in sliding mode creation problem for a control objective defined on the basis of the flat output. In the next section, we detail in the flat output-based sliding model control design.

4.2.1.2 Sliding-Mode Control Design

In this section, a trajectory tracking control is desired to move the system's state $x(t)$ to follow $x_d(t) = [\theta_d, 0, 0, 0]^T$. Considering the flat output y_f, one can define the flat output tracking error as $e_f = y_f - y_{f,d}$ where $y_{f,d}$ denotes the desired flat output. The relative order of the flat output system in Equation (4.10) is $n_f = 4$, therefore; a sliding surface can be readily proposed to be

$$s(e, \dot{e}, \ddot{e}, e^{(3)}) = e_f^{(3)} + \lambda_1 \ddot{e}_f + \lambda_2 \dot{e}_f + \lambda_3 e_f \tag{4.11}$$

with the control gains $\lambda_i, i = 1, 2, 3$ chosen so that the associated characteristic polynomial:

$$p(s) = s^3 + \lambda_1 s^2 + \lambda_2 s + \lambda_3 \tag{4.12}$$

is a Hurwitz polynomial. To simply the design, let us define the surface as a function of a single λ_s as

$$\begin{aligned} s(e, \dot{e}, \ddot{e}, e^{(3)}) &= (\frac{d}{dt} + \lambda_s)^{n_f - 1} e_f \\ &= e_f^{(3)} + 3\lambda_s \ddot{e}_f + 3\lambda_s^2 \dot{e}_f + \lambda_s^3 e_f \end{aligned} \tag{4.13}$$

Given the initial condition $x(t = 0)$ and therefore $y_f(t = 0)$, the problem of tracking the vector $y_{f,d}$ is now equivalent to that of remaining on the surface $s(t)$ for all $t > 0$. Indeed $s \equiv 0$ represents a linear differential equation whose unique solution is $e_f \equiv 0$. As a result, the problem of tracking vector $y_{f,d}$ can be reduced to that of keeping the scalar quantity s at zero. In fact, we can say the problem of tracking vector $y_{f,d}$ can be replaced by a first-order stabilization problem in s. Moreover, bounds on the surface s can be directly translated into bounds on the tracking error vector e_f. To guarantee the stability of the control design, it is usual to define the Lyapunov function as

$$V = \frac{1}{2} s^2 \tag{4.14}$$

such that the first time derivative is defined as

$$\dot{V} = s\dot{s} \leq -\eta_s |s| \tag{4.15}$$

where η_s is a strictly positive constant. Therefore, keeping the stability condition one can define

$$\dot{s} = -\eta_s \frac{|s|}{s} = -\eta_s sgn(s) \tag{4.16}$$

where

$$sgn(s(t)) = \begin{cases} +1 & \text{for} \quad s(t) \geq 0 \\ -1 & \text{for} \quad s(t) < 0 \end{cases} \tag{4.17}$$

making the first derivative of Equation (4.13), one obtains

$$\dot{s} = e_f^{(4)} + 3\lambda_s e_f^{(3)} + 3\lambda_s^2 \ddot{e}_f + \lambda_s^3 \dot{e}_f \tag{4.18}$$

as $e_f^{(4)} = y_f^{(4)} - y_{f,d}^{(4)}$, we replace the term $y_f^{(4)}$ by the flat output in Equation (4.10) to obtain the control law u_{SMC} as

$$u_{SMC} = \frac{1}{B_\alpha}(y_{f,d}^{(4)} - \frac{1}{C_\alpha} y_f^{(4)} - \frac{B_\alpha}{B_\theta} f_\theta - 3\lambda_s e_f^{(3)} - 3\lambda_s^2 \ddot{e}_f - \lambda_s^3 \dot{e}_f - \eta_s sgn(s)) \tag{4.19}$$

In practical cases, the term $sgn(s(t))$ in u_{SMC}causes the drawback of chattering. One approach for chattering reduction involves introducing a boundary layer around the switching surface and using a continuous control with the boundary layer. Therefore, we replace the term $sgn(s(t))$ by $sat(s(t), \phi_s)$ is defined as

$$sat(s(t), \phi_s) = \begin{cases} \frac{s(t)}{\phi_s} & \text{for} \quad |s(t)| < \phi_s \\ sgn(s(t)) & \text{for} \quad otherwise \end{cases} \tag{4.20}$$

where ϕ_s is a strictly positive constant. In the next section, we validate the feasibility and efficiency of the proposed flat output-based sliding mode controller by simulating the RFL model with the identified model.

4.2.2 Simulated Results

Using the identified nonlinear model of RFL in section 2.3, we simulate the system in MATLAB. To generate the control input u_{SMC}, we set $\eta_s = 5 \times 10^4$, $\phi_s = 500$, and $\lambda_s = 16$ such that the whole roots of Hurwitz polynomial in Equation (4.12) are located on the left half-plane of the complex plane to keep the stability. We set the desired or reference trajectory for the motor angular location θ_d as a square signal, the blue line in Figure 4.1.

The system response in Figure 4.1 shows that the controller is capable of a smooth trajectory tracking while suppressing the undesired vibration in the link. The control input is smooth and without any chattering. Moreover, the controller input satisfies the saturation limit of the servo motor $|V_m| \leq 10$. Figure 4.2 demonstrates the variation of the sliding surface during the control operation. Starting

from any initial conditions or suddenly changes in the desired trajectory, the control input makes the states trajectory slide along the surface toward the desired states when the surface value is zero or there is no trajectory error.

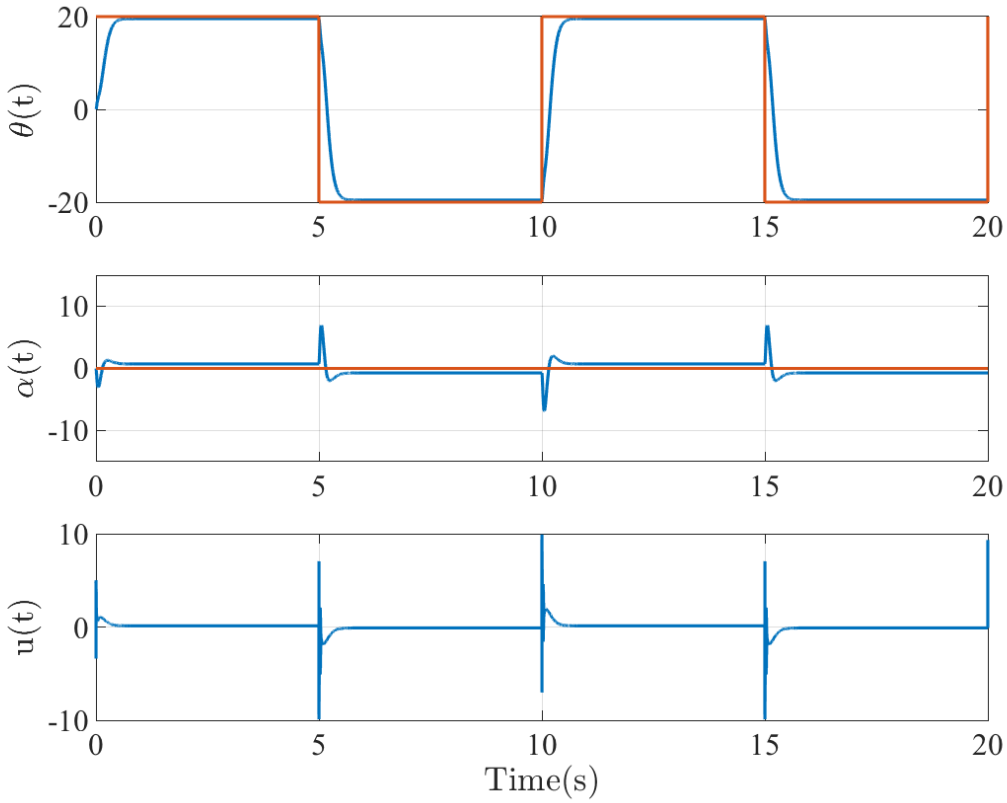

Figure 4.1: The control simulated results of the identified RFL system with a flat-output sliding model controller.

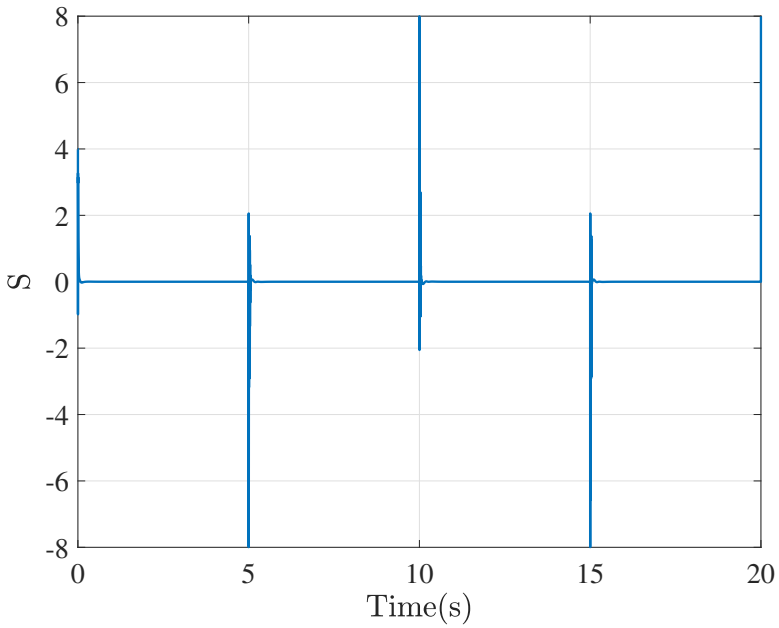

Figure 4.2: Sliding surface to track a square trajectory for rotary flexible link.

4.3 Optimal Control Design for Identified Nonlinear Rotary Flexible Link

Optimal control has emerged as one of the fundamental design philosophies of modern control systems design. Optimal control policies satisfy the specified system performance while minimizing a structured cost index which describes the balance between desired performance and available control resources. From a mathematical point of view, the solution of the optimal control problem is based on the solution of the underlying HJB equation. The solution of HJB equation is mostly challenging for nonlinear systems due to the intractability of this nonlinear differential equation for continuous-time (CT) systems, as this is the interest of this section. In this section, after giving a short review of optimal control theory, we propose a neural network solution to estimate the semi-global solution of HJB equation. Then, we validate the approach through several simulated system and finally our identified model of RFL.

4.3.1 Theoretical background

We consider the time invariant and continuous time system in Equation (4.1). Focusing on infinite horizon optimal control problem, we define the infinite horizon integral cost as,

$$V(x_0) = \int_0^\infty r(x(\tau, u(\tau))\,\mathrm{d}\tau. \tag{4.21}$$

where $r(x,u) = x^T Q x + u^T R u$ acts as the reward function. The matrix Q is positive definite and $R \in R^{m\times m}$ is a symmetric positive definite matrix. Based on the definition in [99], a control policy $\mu(x)$ is defined as admissible with respect to Equation (4.21) on domain Ω, denoted by $\mu \in \psi(\Omega)$, if $\mu(x)$ is continuous on Ω, $\mu(0) = 0$, $u(x) = \mu(x)$ stabilizes Equation (4.1) on Ω, and $V(x_0)$ is finite $\forall x_0 \in \Omega$.

For any admissible control policy $\mu \in \psi(\Omega)$, if the associated cost function

$$V^\mu(x_0) = \int_0^\infty r(x(\tau, \mu(x(\tau)))\,\mathrm{d}\tau. \tag{4.22}$$

is C^1, then an infinitesimal version of Equation (4.22) is so-called nonlinear Lyapunov equation

$$0 = r(x, \mu(x)) + (V_x^\mu)^T (f(x) + g(x)\mu(x)),\ V^\mu(0) = 0 \tag{4.23}$$

where V_x^μ denotes the partial derivative of the value function V^μ with respect to the state vector x.

The optimal control problem can now be formulated: Given the continuous-time system in Equation (4.1), the set $\mu \in \psi(\Omega)$ of admissible control policies and the infinite horizon cost functional Equation (4.21), we find an admissible control policy such that the cost index Equation (4.21) associated with the system in Equation (4.1) is minimized. Defining the Hamiltonian of the problem

$$H(x, u, V_x) = r(x(t), u(t)) + V_x^T (f(x(t)) + g(x(t))\mu(t)) \tag{4.24}$$

the optimal cost function $V^*(x)$ defined by

$$V^*(x_0) = \min_{\mu\in\psi(\Omega)} (\int_0^\infty r(x(\tau, \mu(x(\tau)))\,\mathrm{d}\tau) \tag{4.25}$$

with $x_0 = x$ is known as the **value function**, and satisfies the HJB equation

$$0 = \min_{\mu\in\psi(\Omega)} [H(x, \mu, V_x^*)]. \tag{4.26}$$

Assuming that the minimum on the right hand side of Equation (4.26) exists and is unique then the optimal control function for the given problem is

$$\mu^*(x) = -\frac{1}{2} R^{-1} g^T(x) V_x^*(x) \tag{4.27}$$

Inserting this optimal control control policy in the nonlinear Lyapunov equation we obtain the formulation of the HJB equation in terms of V_x^*

$$0 = Q(x) + V_x^{*T}(x)f(x) - \frac{1}{4}V_x^{*T}(x)g(x)R^{-1}g^T(x)V_x^*(x)$$
$$V^*(0) = 0. \tag{4.28}$$

For the linear system cases, considering a quadratic cost functional, the equivalent of this HJB equation is the well-know **Algebraic Ricacati Equation (ARE)**.

In order to find the optimal control solution for the problem one only needs to solve the HJB equation (4.28) for the value function and then substitute the solution in Equation (4.27) to obtain the optimal control. However, due to the nonlinear nature of the HJB equation finding its solution is generally difficult or impossible.

4.3.2 Algorithm to Solve Hamilton_Jacobi_Belman Equation

In this section, we aim at developing an algorithm to solve HJB equation for identified nonlinear systems. First, we estimate the value function by neural network to solve HJB equation and therefore we gain the optimal feedback control law. We propose the effective Adam optimization algorithm, the recent popular algorithm in the field of deep learning, to train our network and estimate the parameters.

4.3.2.1 Neural Network Approximation of Value Function

Therefore, it is justified to assume there exist weights W such that the value function $V(x)$ is approximated as

$$V(x) = W^T\phi(x) + \varepsilon(x) \tag{4.29}$$

Then $\phi(x)\ \ R^n \rightarrow R^N$ is called the NN activation function vector, N the number of neurons in the hidden layer, and ε accounts for the NN approximation error. As per the above, the NN activation functions $\{\phi_i(x)\ :\ i = 1, ..., N\}$ are selected so that $\{\phi_i(x)\ :\ i = 1, ..., \infty\}$ provides a complete independent basis set such that the value function $V(x)$ and its derivative

$$\frac{\partial V}{\partial x} = (\frac{\partial \phi(x)}{\partial x})^T W + \frac{\partial \varepsilon}{\partial x} = \nabla\phi^T W + \nabla\varepsilon \tag{4.30}$$

are uniformly approximated. Based on universal approximation, as the number of hidden layer neurons $N \rightarrow \infty$, the approximation error $\varepsilon \rightarrow 0$ and therefore $\nabla\varepsilon \rightarrow 0$ uniformly . Furthermore, for fixed number of neurons N, the NN approximation errors $\varepsilon(x)$, and $\nabla\varepsilon$ are bounded by constants on a compact set [116].

Considering a fixed control policy $u(t)$ and using the NN value function approximation, the nonlinear Lyapunov Equation (4.23) becomes

$$H(x,u,W) = W^T\nabla\phi(f+gu) + x^TQx + u^TRu = \varepsilon_H \tag{4.31}$$

where ε_H is approximation error and goes to zero when $N \to \infty$. The object of interest in this chapter is finding the solution of the HJB using the introduced function approximator in Section 4.3.2.1, therefore; it is interesting now to look at the effect of the approximation error on the HJB equation (4.28).

$$W^T\nabla\phi f(x) - \frac{1}{4}W^T\nabla\phi g R^{-1}g^T\nabla\phi^T W + x^TQx = \varepsilon_{HJB} \tag{4.32}$$

In the next section, we define the cost function to train the NN or estimate the parameters W based on Equation (4.32).

4.3.2.2 Cost Function to Train Neural Network

The weights or parameters of NN approximator W for the value function V in Equation (4.29) is unknown. In fact, we need to train the network to find the weights W and at each training step, we have an estimation of the weights $\hat{W}$ and consequently, the output of NN is an estimation of the value function as

$$\hat{V}(x) = \hat{W}^T\phi(x) \tag{4.33}$$

Recall that $\phi(x) : R^n \to R^N$ is the vector of activation functions, with N the number of neurons in the hidden layer. The approximate nonlinear Lyapunov equation is then

$$H(x,u,\hat{W}) = \hat{W}^T\nabla\phi(f+gu) + x^TQx + u^TRu = e_H(x) \tag{4.34}$$

defining the weight estimation error

$$\tilde{W} = \hat{W} - \hat{W} \tag{4.35}$$

Then

$$e_H(x) = -\tilde{W}^T\nabla\phi(f+gu) + \varepsilon_H \tag{4.36}$$

in fact, the total error estimation is caused by the NN approximation function and weight estimation error which is related to training NN. As our focus is on solving HJB equation, we define the residual error in HJB equation such that,

$$\hat{W}^T\nabla\phi f(x) - \frac{1}{4}\hat{W}^T\nabla\phi g R^{-1}g^T\nabla\phi^T\hat{W} + x^TQx = e(x) \tag{4.37}$$

Given any admissible control policy u, it is desired to select $\hat{W}$ to minimize the squared residual error. If we consider each state x denotes a data point and assume that $\{e_i \ : \ i = 1, ..., N_d\}$ refers the residual errors for n_d number of data

points. Then we can define the residual error vector $\mathbf{e} = [e_1, e_2, ..., e_{N_d}]$ and define the cost function J as the squared residual errors as,

$$J = \frac{1}{2}\mathbf{e}^T\mathbf{e} = \sum_{i=1}^{N_d} e_i^2 \tag{4.38}$$

training the NN by minimizing the cost function, then $\hat{W} \to W$ and $e \to \varepsilon_{HJB}$.

4.3.2.3 Data Generation to Train Neural Network

The data selection to train NN to solve HJB equation not only essentially affect the accuracy but also can mitigate the curse of dimensionality. In fact, except for systems with two or three state variables, numerically solving HJB equations for general nonlinear systems is difficult or even unfeasible.

In our algorithm, we select a roughly wide but physically feasible domain Ω for each state variable and simply discretize each dimension that results in a mesh-like grids or finite dimensional data points. Our algorithm performs fast for the nonlinear systems with two or three state variables as we report the time elapsed for the the computations. However, for the dimensions larger than three, we have to deal with the curse of dimensionality. This is also because of the fact that high-dimension systems require more activation functions ψ_i and consequently larger NN which utilize more computational sources. Moreover, high-dimensional systems require more data points on each dimension to present the state variable and give good information to train the larger NN.

We validate the accuracy of our algorithm to solve HJB equation for the nominal linear model of a rotary flexible manipulator link with four state variables. Four-dimensional systems still account for high-dimensional systems. To mitigate the curse of dimensionality, we decease the number of discretization on each dimension. The results prove the accuracy of solution, while the elapsed time is considerably long. Further developments are required as it is now an active research area in the field of optimal control. Training NN through sparse grids or the data around the desired trajectories are the proposed methods in the literature to solve HJB for higher dimensional problems.

4.3.2.4 ADAM Stochastic Gradient Descent Optimization

To obtain the exact solution for the weight vector W in HJB Equation (4.32) or train NN one needs to find the stationary points of the loss function $J(\theta)$ in Eq. (4.38), therefore; the problem is limited to an optimization problem. Our algorithm is compatible with stochastic gradient descent (SGD) optimization, which is commonly used in deep learning and has outperformed even in the non convex problems.

We use adaptive moment estimation (Adam) proposed in [117]. ADAM is a efficient method for SGD optimization that only requires first-order gradients in Eq. (4.39) with little memory requirement. The algorithm computes the adaptive learning rate for each parameter in the vector of γ_k for each iteration k.

$$\hat{W}_{k+1} = \hat{W}_k - \gamma_k \Delta_W J(W_k) \tag{4.39}$$

We introduce Adam, an algorithm for first-order gradient-based optimization of stochastic objective functions. The method is straightforward to implement and is based on adaptive estimates of lower-order moments of the gradients. The method is computationally efficient, has little memory requirements and is well suited for problems that are large in terms of data and/or parameters. The method exhibits invariance to diagonal rescaling of the gradients by adapting to the geometry of the objective function. The hyper-parameters have intuitive interpretations and typically require little tuning.

with $g_k = \Delta_W J(W_k)$, we denote the gradient vector of partial derivatives of J with respect to θ evaluated at iteration k. The algorithm updates the exponential moving average of the gradient (m_k) and the squared gradient (v_k) where the hyper parameters $\beta_1, \beta_2 \in [0, 1)$ control the exponential decay rates of these moving averages. The moving average are the estimated of the mean and the uncentered variance of the gradient.

We use a modified AdaMax, a variant of Adam based on the infinity norm. Algorithm 3 details the optimization procedure we develop for our experimental examples in the paper. We should point out that all operations on vectors are element-wise. For more mathematical details, we encourage reading the concepts in [117].

4.3.2.5 Algorithm Schematics

In this section, we summarize the algorithm to solve HJB equation for known nonlinear systems in Figure 4.3 and Algorithm Schematics 3.

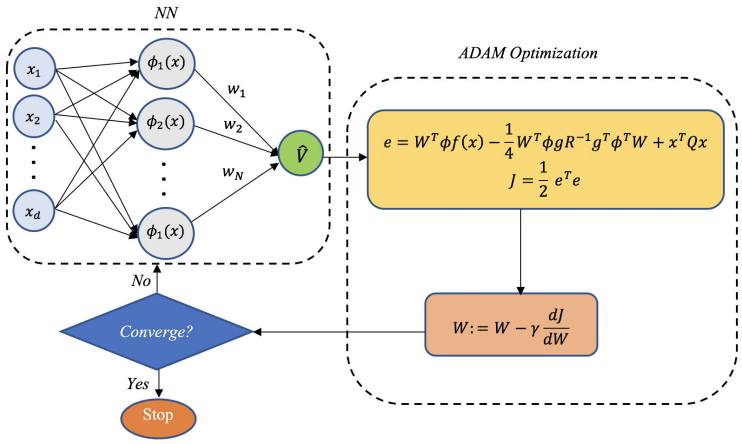

Figure 4.3: Algorithm schematics of training NNs to estimate the solution of HJB equation.

Algorithm 3 NN solution to solve HJB equation for known systems.

Input: Activation Basis function ϕ_i; Step size α
Data Collocations: N_d data point inside the domain Ω
Adam setting: Good default settings are $\beta_1 = 0.9$ and $\beta_2 = 0.999$
Initialize: Initial network parameters $\hat{W}$.
$m(0) \leftarrow 0$ ▷ Initialize first moment vector
$u(0) \leftarrow 0$ ▷ Initialize the exponentially weighted infinity norm
$k \leftarrow 0$ ▷ Initialize iteration step
while not converged **do**
    $k \leftarrow k + 1$
    $g_k \leftarrow \Delta_\theta J(\theta_{k-1})$
    $m_k \leftarrow \beta_1 m_{k-1} + (1 - \beta_1) g_k$
    $v_k \leftarrow max(\beta_2 v_{k-1}, |g_k|)$
    $\hat{W}_k \leftarrow \hat{W}_{k-1} - (\alpha/(1 - \beta_1^k)).m_k/v_k := \hat{W}_k - \gamma_k \Delta_{\hat{W}} J(W_k)$ ▷ Updating parameters
end while

4.3.3 Simulation Validations

In this section, we validate the performance of our algorithm to obtain the optimal control law for the linear and nonlinear systems proposed by Frank Lewis in [3].

4.3.3.1 Continuous-Time F16 Aircraft Plant

Consider the continuous-time F16 aircraft plant used in [118]

$$\dot{x} = \begin{bmatrix} -1.01887 & 0.90506 & -0.00215 \\ 0.82225 & -1.07741 & -0.17555 \\ 0 & 0 & -1 \end{bmatrix} x + \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix} u \tag{4.40}$$

with the state vector $x = [x_1; x_2, x_3]$. The Q and R matrices in the cost function are identity matrices of appropriate dimensions. In the linear models, the solution of the HJB equation is given by the solution of the algebraic Riccati equation (ARE) which is known as **linear-quadratic regulator (LQR)**. Since the value function is quadratic in the LQR case, we select the basis set ϕ as the quadratic vector in the state components as

$$\phi(x) = \begin{bmatrix} x_1^2 & x_1x_2 & x_1x_3 & x_2^2 & x_2x_3 & x_3^2 \end{bmatrix}^T \tag{4.41}$$

we consider the following domain

$$\begin{aligned} \Omega \ : \ -5 \leq & x_1 \leq 5 \\ -10 \leq & x_2 \leq 10 \\ -10 \leq & x_3 \leq 10 \end{aligned} \tag{4.42}$$

with the mesh discretization $10 \times 10 \times 5$. In fact, we generate totally 500 data points to train NN. We select the learning rate $\alpha = 0.001$ and randomly choose the initial values for the vector $\hat{W}$ as Fig.(4.4) shows. Fig.(4.4) shows how the NN's weights $\hat{W} = [\hat{W}_1, \hat{W}_2, ..., \hat{W}_6]$ converge to the optimal values W^* as,

$$W^* = \begin{bmatrix} 1.4245 & 2.3364 & -0.2705 & 1.4350 & -0.3002 & 0.4330 \end{bmatrix}^T \tag{4.43}$$

we gain $J = 9.3776 * 10^{-5}$ at the convergence and the computational process for 6000 iterations elapsed 39.133136 seconds being run on a 2.3 GHz Intel Core i5 processor. Using Equation (4.27) and estimated optimal weights W^*, one can simply find the optimal feedback control law in Equation (4.44). The estimated gains are the same as the LQR gains computed by the build-in function in MATLAB. This high accuracy validate the performance of the numerical solution for HJB equation.

$$
\begin{aligned}
\hat{u} &= -\frac{1}{2}R^{-1}\begin{bmatrix}0\\0\\1\end{bmatrix}^T \begin{bmatrix}2x_1 & 0 & 0\\ x_2 & x_1 & 0\\ x_3 & 0 & x_1\\ 0 & 2x_2 & 0\\ 0 & x_3 & x_2\\ 0 & 0 & 2x_3\end{bmatrix}^T \begin{bmatrix}1.4245\\ 2.3364\\ -0.2705\\ 1.4350\\ -0.3002\\ 0.4330\end{bmatrix} \\
&= -0.1352x_1 - 0.1501x_2 + 0.4329x_3 \\
&= u_{LQR}
\end{aligned} \tag{4.44}
$$

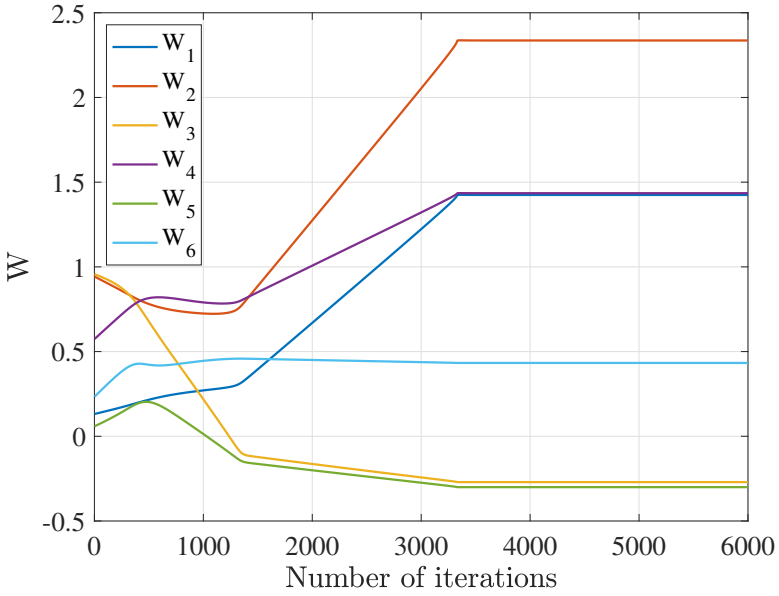

Figure 4.4: Convergence of the estimated parameters $\hat{W}$ to the optimal values W^*.

4.3.3.2 Nonlinear Systems with Known Value Function

In this section, we prove the efficiency and accuracy of our algorithm to solve HJB equation for nonlinear systems. Let us consider the nonlinear system in [3].

$$\dot{x} = f(x) + g(x)u,\ x \in R^2 \tag{4.45}$$

where

$$f(x) = \begin{bmatrix} -x_1 + x_2 \\ -0.5x_1 - 0.5x_2(1 - (cos(2x_1) + 2)^2) \end{bmatrix} \tag{4.46}$$
$$g(x) = \begin{bmatrix} 0 \\ cos(2x_1) + 2 \end{bmatrix}$$

we select $Q = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}$ and $R = 1$. Using the online reinforcement algorithm in [3], the optimal value function is

$$V^*(x) = \frac{1}{2}x_1^2 + x_2^2 \tag{4.47}$$

and therefore, the optimal control signal is

$$u^*(x) = -(cos(2x_1) + 2)x_2 \tag{4.48}$$

One selects the NN vector activation function as

$$\phi(x) = \begin{bmatrix} x_1^2 & x_1x_2 & x_2^2 \end{bmatrix} \tag{4.49}$$

we consider the following domain

$$\begin{aligned} \Omega \ : \ & -2 \leq x_1 \leq 2 \\ & -2 \leq x_2 \leq 2 \end{aligned} \tag{4.50}$$

with the mesh-grid 25×25 and therefore, totally 625 data points to train the network. Setting the learning rate $\alpha = 0.005$ and the initial weight vector $\hat{W} = [1, 1, 1]^T$, after 498 iterations we gain the cost function $J = 5.9207 * 10^{-10}$. The running time is recorded as 2.052706 seconds on a 2.3 GHz Intel Core i5 processor. Figure 4.5 shows the convergence of the estimated weights $\hat{W} = [W_1, W_2, W_3]^T$ to the optimal values W^*. Table 4.1 includes the optimal weights W^* in comparison with our estimation $\hat{W}$ and estimated weighs W_{HJB} by the online reinforcement learning approach by Frank Lewis in [3].

Table 4.1: The optimal weights W^* in comparison with our estimation $\hat{W}$ and estimated weighs W_{HJB} by the online reinforcement learning approach by Frank Lewis in [3] for the known nonlinear system in Equation (4.45).

Term	W^*	W_{HJB}	$\hat{W}$
x_1^2	0.5	0.5017	0.49998
$x_1 x_2$	0	-0.002	2.8946×10^{-5}
x_2^2	1	1.0008	0.99973

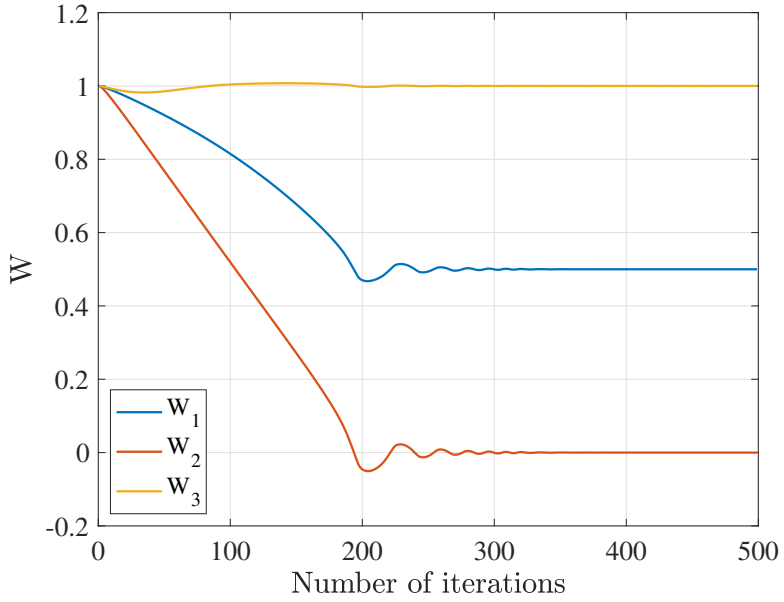

Figure 4.5: Convergence of the critic parameters W to the parameters of the optimal critic W^*.

So using the estimated optimal weights, we can simply compute the feedback control law as,

$$\hat{u}(x) = -\frac{1}{2}R^{-1}\begin{bmatrix} 0 \\ cos(2x_1)+2 \end{bmatrix}\begin{bmatrix} 2x_1 & 0 \\ x_2 & x_1 \\ 0. & 2x_2 \end{bmatrix}^T \begin{bmatrix} 0.49998 \\ 2.8946 * 10^{-5} \\ 0.99973 \end{bmatrix} \tag{4.51}$$

Figure 4.6 shows the optimal value function. The identified value function given by $\hat{V} = \hat{W}\phi(x)$ is virtually indistinguishable. In fact, Figure 4.7 shows the 3D plot of the difference between the approximated value function, by using the

proposed algorithm, and the optimal one. This error is close to zero. Good approximation of the actual value function is being evolved.

Finally Figure 4.8 shows the 3D plot of the difference between the approximated control, by using our algorithm, and the optimal one. This error is also close to zero.

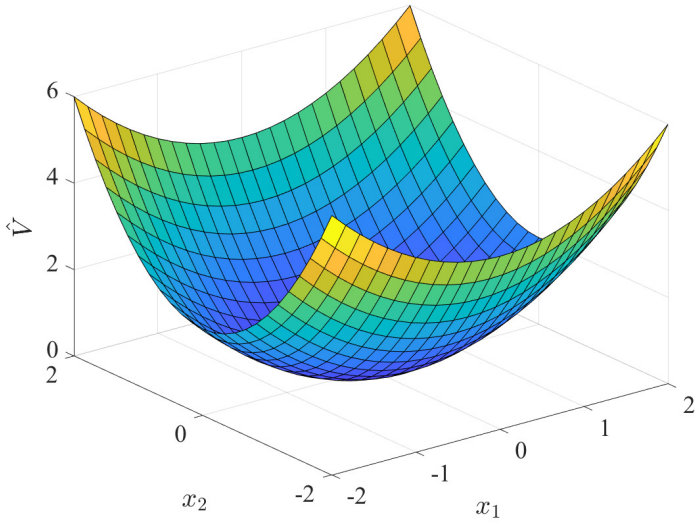

Figure 4.6: 3D plot of the estimated value function $\hat{V}$.

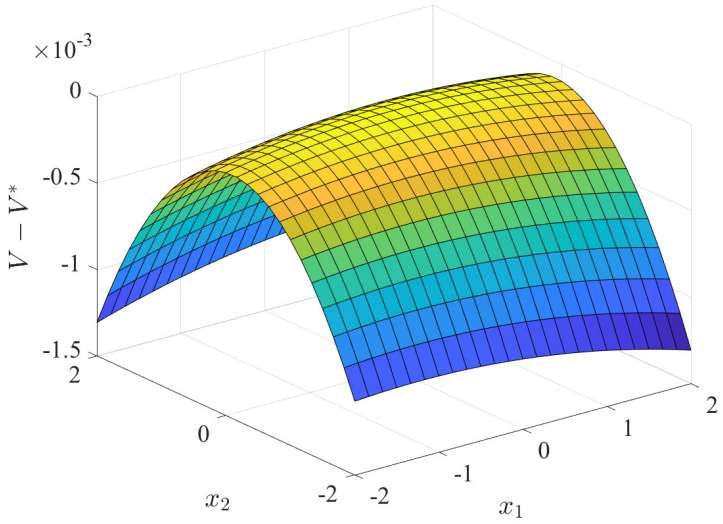

Figure 4.7: Convergence of the critic parameters W to the parameters of the optimal critic W^*.

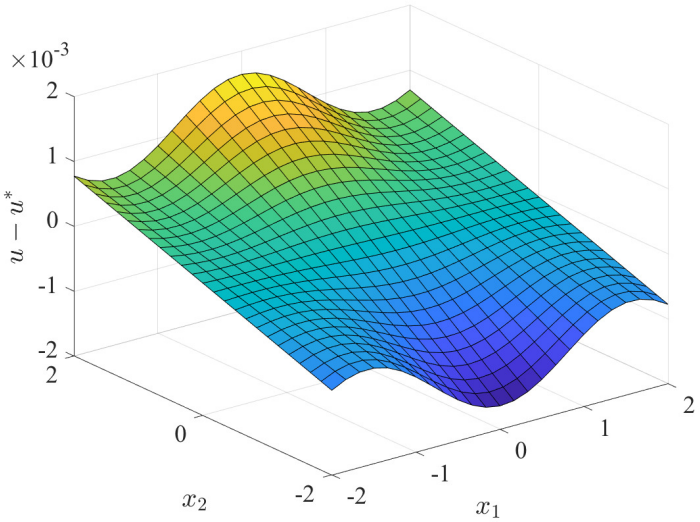

Figure 4.8: 3D plot of the approximation error for the feedback control law.

4.3.4 Optimal Feedback Control for Rotary Flexible Manipulator Link with Identified Model

In this section, we try to develop the optimal control design for rotary flexible link by using our identified model in section 2.3. The identified model includes non-linearities and accounts for high-dimensional problem in optimal control community. Rewriting the motion equation in Equation (2.21) in the form of state-space as

$$x = [x_1, x_2, x_3, x_4]^T = [\theta, \alpha, \dot{\theta}, \dot{\alpha}]^T \tag{4.52}$$

we select a quadratic value function $\hat{V}$ with the basis function vector ϕ as,

$$\phi(x) = \begin{bmatrix} x_1^2 & x_1x_2 & x_1x_3 & x_1x_4 & x_2^2 & x_2x_3 & x_2x_4 & x_3^2 & x_3x_4 & x_4^2 \end{bmatrix} \tag{4.53}$$

and we can generate data in the domain,

$$\begin{aligned} \Omega \ : \ -5\pi \leq & x_1 \leq 5\pi \\ -0.25\pi \leq & x_2 \leq 0.25\pi \\ -10 \leq & x_3 \leq 10 \\ -10 \leq & x_4 \leq 10 \end{aligned} \tag{4.54}$$

with mesh-grid $3 \times 3 \times 3 \times 3$ and therefore, totally 81 data points to train NN. To design control, we select the same Q and R matrices proposed in the device's manual in [112] as,

$$Q = \begin{bmatrix} 121 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 5 \end{bmatrix}, \ R = 1 \tag{4.55}$$

Setting the learning rate $\alpha = 0.001$ and the initial weight as the vector of ones, after 2.510^6 iterations, the network parameters $\hat{W}$ converge to the optimal W^* as Figure 4.9 shows the convergence of the network's weight $\hat{W}$ to the optimal values W^*,

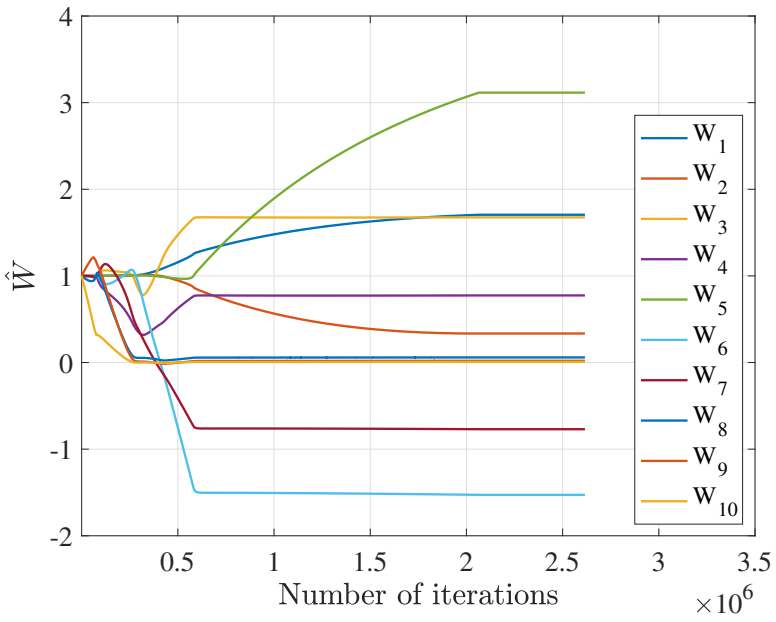

Figure 4.9: The convergence of NN's weights $\hat{W}$ to the optimal values W^* to estimate HJB solution for optimal control design.

The optimal weight W^* is recorded by

$$W^* = \begin{bmatrix} 1.7057 & 0.3348 & 1.6758 & 0.7746 & 3.1157... \\ & -1.5276 & -0.7699 & 0.0589 & 0.0198 & 0.0069 \end{bmatrix}^T \tag{4.56}$$

therefore, the optimal control for the nonlinear system can be computed in real-time as

$$\begin{aligned} V_m &= -\frac{1}{2}R^{-1} \begin{bmatrix} 0 \\ 0 \\ B_\theta \\ B_\alpha \end{bmatrix}^T \begin{bmatrix} 2x_1 & 0 & 0 & 0 \\ x_2 & x_1 & 0 & 0 \\ x_3 & 0 & x_1 & 0 \\ x_4 & 0 & 0 & x_1 \\ 0 & 2x_2 & 0 & 0 \\ 0 & x_3 & x_2 & 0 \\ 0 & x_4 & 0 & x_2 \\ 0 & 0 & 2x_3 & 0 \\ 0 & 0 & x_4 & x_3 \\ 0 & 0 & 0 & 2x_4 \end{bmatrix}^T \begin{bmatrix} 1.7057 \\ 0.3348 \\ 1.6758 \\ 0.7746 \\ 3.1157 \\ -1.5276 \\ -0.7699 \\ 0.0589 \\ 0.0198 \\ 0.0069 \end{bmatrix} \\ &= 11.344898\ x_1 - 7.640617\ x_2 + 2.264583\ x_3 - 0.060721\ x_4 \end{aligned} \tag{4.57}$$

Figure 4.10 illustrates the dynamical response of the simulated rotary flexible link with the data-deriven model to the optimal control computed in Equation (4.57) to track a square signal reference. The results prove the efficiency of the nonlinear optimal design for trajectory tracking and undesired vibration suppression.

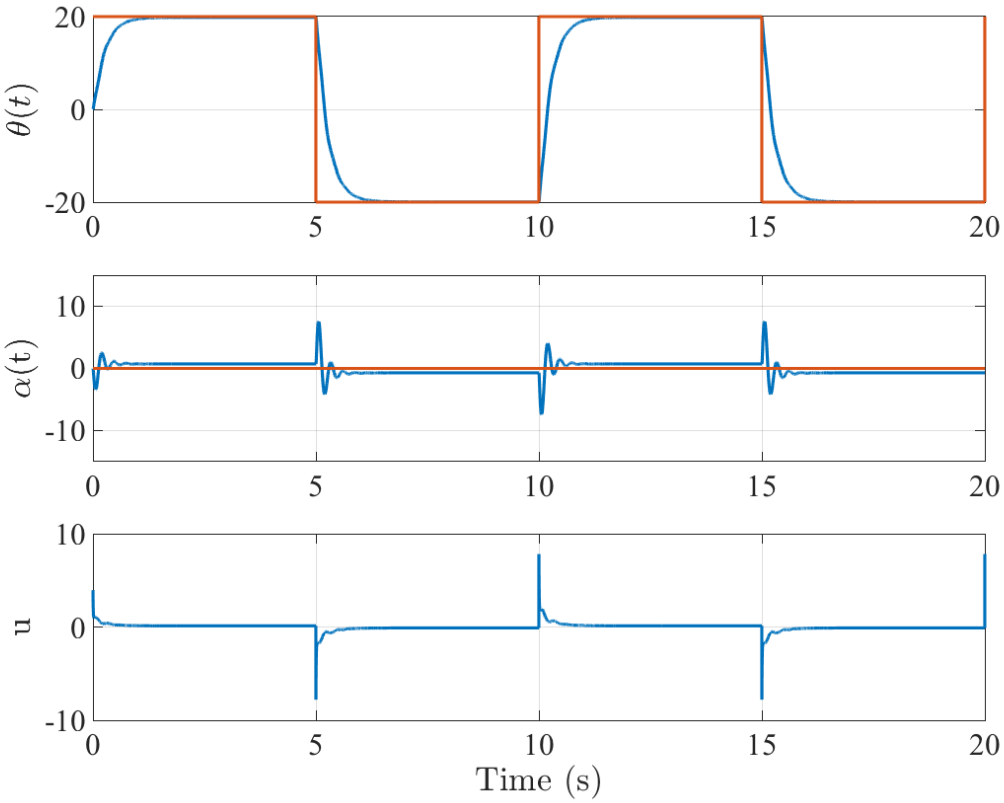

Figure 4.10: The response by RFL dynamical systems to the optimal control designed by the nonlinear identified model.

4.4 Conclusions

In this work, we develop two control design for the identified model with nonlinearities to validate the sparse system identification method. The first trajectory tracking control is designed by a sliding mode control and to overcome the under-actuation in the system we suggest a flat output to be controlled. Moreover, we prove the efficiency of our nonlinear model regarding the optimal control design. A numerical algorithm is suggested to estimate the solution of HJB equation. The control performance is validated by simulated system of RFL and popular simulated nonlinear examples in the field of optimal control.

Chapter 5

FUTURE WORKS

5.1 Conducting remarks

The present thesis mainly discusses how data-driven approach can be used to identify the optimal model of nonlinear electro-mechanical systems with and without time delay in the feedback control in time domain. The identification method involves time series data analysis and regression, sparse identification, model selection and cross validation from machine learning, the statistical concept of bootstrapping, and algebraic method dealing with high-order derivative estimation and noise removing. Comparing the common neural network training approach, the proposed method is physics-informed and involves generality for unseen data inputs. Moreover, the approach gives an accurate parameters estimation and physical properties of the system.

The result of identification can be employed by model-based control design frameworks leading to efficient data-driven controllers. The identified model gives a reliable prediction for trajectory tracking controller of nonlinear systems. Optimal control as the main principle of modern controllers is studied in this thesis. A physics-informed neural network approach is proposed to design the optimal control. The algorithm does not depend on the persistent excitation conditions which is hard to obtain it. We believe the data-driven methods and theories on system identification and control design could be further developed in a bright future.

5.2 Future Work

5.2.1 Data-driven Modeling

For the future research in the field of system identification, the proposed algorithm can be extended to multiple-input-multiple-output (MIMO) systems to model more complicated dynamic systems. Furthermore, deep learning approach can be investigated for more complex and history dependent dynamic systems. The proposed technique in this thesis is off-line while intense research regarding the online data driven estimation theories can be done. The online estimation can be beneficial for time varying systems.

In the field of time-delayed nonlinear system, the identification techniques are mostly analytical; therefore, there are great opportunities to investigate the data-driven techniques to identify systems and estimate the time delay. Complicated

case study of MIMO systems, history dependent dynamic systems, and complicated nonlinear control design can be considered. Deep learning approach can be efficiently employed for this purpose.

5.2.2 Data-driven Control Design

Optimal control solutions and reinforcement learning controllers has been taken a great attention in the recent years. For more complex systems, deep and complex neural networks can lead promising results, while the "curse of dimensionality" is still challenging. Efficient methods of data selection can save the usage of computational resources and improve the time processing. In addition to data selection, efficient optimization techniques and training approach can be deeply investigated as the future work.

BIBLIOGRAPHY

[1] Quanser Inc (2011) *Rotary Flexible Joint Workbook*. Markhan, Ontario

[2] Apkarian J, Lévis M, Gurocak H (2011) *User Manual SRV02 Rotary Servo Base Unit Set Up and Configuration*. Quanser Inc., Markham, Ontario

[3] Vamvoudakis KG, Lewis FL (2010) Online actor-critic algorithm to solve the continuous-time infinite horizon optimal control problem. *Automatica* **46**(5), 878–888

[4] Gautrais J, Ginelli F, Fournier R, Blanco S, Soria M, Hugues C, Guy T (2012) Deciphering interactions in moving animal groups. *PLOS Computational Biology*

[5] Zienkiewicz A, Barton DAW, Porfiri M, di Bernardo M (2015) Data-driven stochastic modelling of zebrafish locomotion. *Journal of Mathematical Biology* **71**(5), 1081–1105

[6] Burbano L, Daniel A, Porfiri M (2020) Data-driven modeling of zebrafish behavioral response to acute caffeine administration. *Journal of Theoretical Biology* **485**, 110,054

[7] Hernandez AM, Casado Magana EJ, Berna AG (2018) Data-driven aircraft trajectory predictions using ensemble meta-estimators. In: *Proceedings of the 37th Digital Avionics Systems Conference*, London, UK, pp 1–10

[8] Altchë F, de La Fortelle A (2017) An LSTM network for highway trajectory prediction. In: *Proceedings of the 20th International Conference on Intelligent Transportation Systems*, Yokohama, Japan, pp 353–359

[9] Wang WX, Lai YC, Grebogi C (2016) Data based identification and prediction of nonlinear and complex dynamical systems. *Physics Reports* **644**, 1–76

[10] Ljung L (2010) Perspectives on system identification. *Annual Reviews in Control* **34**(1), 1–12

[11] Noël JP, Kerschen G (2017) Nonlinear system identification in structural dynamics: 10 more years of progress. *Mechanical Systems and Signal Processing* **83**, 2–35

[12] Schoukens J, Vaes M, Pintelon R (2016) Linear system identification in a nonlinear setting: Nonparametric analysis of the nonlinear distortions and their impact on the best linear approximation. *IEEE Control Systems Magazine* **36**(3), 38–69

[13] Sracic MW, Allen MS (2014) Identifying parameters of multi-degree-of-freedom nonlinear structural dynamic systems using linear time periodic approximations. *Mechanical Systems and Signal Processing* **46**(2), 325–343

[14] Moaveni B, Asgarieh E (2012) Deterministic-stochastic subspace identification method for identification of nonlinear structures as time-varying linear systems. *Mechanical Systems and Signal Processing* **31**, 40–55

[15] Lai Z, Nagarajaiah S (2019) Sparse structural system identification method for nonlinear dynamic systems with hysteresis/inelastic behavior. *Mechanical Systems and Signal Processing* **117**, 813–842

[16] Ma SF, Leylaz G, Sun JQ (2020) Identification of differentially flat output of underactuated dynamic systems. *International Journal of Control* pp 1–12

[17] Haroon M, Adams DE (2009) A modified H2 algorithm for improved frequency response function and nonlinear parameter estimation. *Journal of Sound and Vibration* **320**(4), 822–837

[18] Magnevall M, Josefsson A, Ahlin K, Broman G (2012) Nonlinear structural identification by the "Reverse Path" spectral method. *Journal of Sound and Vibration* **331**(4), 938–946

[19] Vazquez Feijoo JA, Worden K, Montes Garcia P, Lagunez Rivera L, Juarez Rodriguez N, Pech Pérez A (2010) Analysis of MDOF nonlinear systems using associated linear equations. *Mechanical Systems and Signal Processing* **24**(8), 2824–2843

[20] Noël JP, Kerschen G (2013) Frequency-domain subspace identification for nonlinear mechanical systems. *Mechanical Systems and Signal Processing* **40**(2), 701–717

[21] Noël JP, Renson L, Grappasonni C, Kerschen G (2016) Identification of nonlinear normal modes of engineering structures under broadband forcing. *Mechanical Systems and Signal Processing* **74**, 95–110

[22] Clement S, Bellizzi S, Cochelin B, Ricciardi G (2014) Sliding window proper orthogonal decomposition: Application to linear and nonlinear modal identification. *Journal of Sound and Vibration* **333**(21), 5312–5323

[23] Worden K, Green PL (2017) A machine learning approach to nonlinear modal analysis. *Mechanical Systems and Signal Processing* **84**, 34–53

[24] Noël JP, Marchesiello S, Kerschen G (2014) Subspace-based identification of a nonlinear spacecraft in the time and frequency domains. *Mechanical Systems and Signal Processing* **43**(1), 217–236

[25] Peng ZK, Lang ZQ, Wolters C, Billings SA, Worden K (2011) Feasibility study of structural damage detection using NARMAX modelling and nonlinear output frequency response function based analysis. *Mechanical Systems and Signal Processing* **25**(3), 1045–1061

[26] Pai PF, Palazotto AN (2008) Detection and identification of nonlinearities by amplitude and frequency modulation analysis. *Mechanical Systems and Signal Processing* **22**(5), 1107–1132

[27] Brewick PT, Masri SF, Carboni B, Lacarbonara W (2016) Data-based nonlinear identification and constitutive modeling of hysteresis in NiTiNOL and Steel strands. *Journal of Engineering Mechanics* **142**(12), 04016,107

[28] Lusch B, Kutz JN, Brunton SL (2018) Deep learning for universal linear embeddings of nonlinear dynamics. *Nature Communications* **9**(1), 4950

[29] Champion K, Lusch B, Kutz JN, Brunton SL (2019) Data-driven discovery of coordinates and governing equations. *Proceedings of the National Academy of Sciences* **116**(45), 22,445

[30] Raissi M, Perdikaris P, Karniadakis G (2018) Multistep neural networks for data-driven discovery of nonlinear dynamical systems. *arXivorg* (arXiv:1801.01236)

[31] Billings SA (2013) *Nonlinear System Identification: NARMAX Methods in the Time, Frequency, and Spatio-Temporal Domains*. John Wiley & Sons, United States

[32] Fu L, Li P (2013) The research survey of system identification method. In: *Proceedings of the 5th International Conference on Intelligent Human-Machine Systems and Cybernetics*, Hangzhou, China, vol 2, pp 397–401

[33] Vlachas PR, Byeon W, Wan ZY, Sapsis TP, Koumoutsakos P (2018) Data-driven forecasting of high-dimensional chaotic systems with long short-term memory networks. *Proceedings of the Royal Society A: Mathematical, Physical and Engineering Sciences* **474**(2213)

[34] Pathak J, Hunt B, Girvan M, Lu Z, Ott E (2018) Model-free prediction of large spatiotemporally chaotic systems from data: A reservoir computing approach. *Physical Review Letters* **120**(2), 024,102

[35] Muto M, Beck JL (2008) Bayesian updating and model class selection for Hysteretic structural models using stochastic simulation. *Journal of Vibration and Control* **14**(1-2), 7–34

[36] Beck JL, Katafygiotis LS (1998) Updating models and their uncertainties. I: Bayesian statistical framework. *Journal of Engineering Mechanics* **124**(4), 455–461

[37] Green PL, Worden K (2015) Bayesian and Markov chain Monte Carlo methods for identifying nonlinear systems in the presence of uncertainty. *Proceedings of the Royal Society A: Mathematical, Physical and Engineering Sciences* **373**, 20140,405

[38] Kerschen G, Golinval JC, Hemez FM (2003) Bayesian model screening for the identification of nonlinear mechanical structures. *Journal of Vibration and Acoustics* **125**(3), 389–397

[39] Yosida K (1984) *Operational Calculus - A Theory of Hyperfunctions*. Springer Science+Business Media, New York

[40] Sira-Ramìrez H, Garcìa-Rodrìguez C, Cortés-Romero J, Luviano-Juárez A (2014) *Algebraic Identification and Estimation Methods in Feedback Control System*. John Wiley & Sons, United States

[41] Fliess M, Sira-Ramìrez H (2003) An algebraic framework for linear identification. *ESAIM: Control, Optimisation and Calculus of Variations* **9**, 151–168

[42] Garrido R, Concha A (2013) An algebraic recursive method for parameter identification of a servo model. *IEEE/ASME Transactions on Mechatronics* **18**(5), 1572–1580

[43] Trapero JR, Sira-Ramìrez H, Batlle VF (2007) A fast on-line frequency estimator of lightly damped vibrations in flexible structures. *Journal of Sound and Vibration* **307**(1), 365–378

[44] Becedas J, Mamani G, Feliu-Batlle V, Sira-Ramìrez H (2007) Algebraic identification method for mass-spring-damper system. In: *Proceedings of World Congress on Engineering and Computer Science*, San Francisco, USA

[45] Brunton SL, Proctor JL, Kutz JN (2016) Discovering governing equations from data by sparse identification of nonlinear dynamical systems. *Proceedings of the National Academy of Sciences* **113**(15), 3932

[46] Mangan N, Kutz J, Brunton S, Proctor J (2017) Model selection for dynamical systems via sparse regression and information criteria. *Proceedings of the Royal Society A: Mathematical, Physical and Engineering Science* **473**, 20170,009

[47] Kaiser E, Kutz J, Brunton S (2017) Sparse identification of nonlinear dynamics for model predictive control in the low-data limit. *Proceedings of the Royal Society A: Mathematical, Physical and Engineering Science* **474**, 20180,335

[48] Mangan N, Askham T, Brunton S, Kutz J, Proctor J (2019) Model selection for hybrid dynamical systems via sparse regression. *Proceedings of the Royal Society A: Mathematical, Physical and Engineering Sciences* **475**, 20180,534

[49] Chartrand R (2011) Numerical differentiation of noisy, nonsmooth data. *ISRN Applied Mathematics Journal* **2011**, 164,564

[50] Rudin LI, Osher S, Fatemi E (1992) Nonlinear total variation based noise removal algorithms. *Physica D* **60**, 259–268

[51] Schaeffer H, McCalla SG (2017) Sparse model selection via integral terms. *Physical Review E* **96**(2), 023,302

[52] Leylaz G, Ma SF, Sun JQ (2021) An optimal model identification algorithm of nonlinear dynamical systems with the algebraic method. *Journal of Vibration and Acoustics* **143**, 021,002–1–021,002–8

[53] Masri SF, Caughey TK (1979) A nonparametric identification technique for nonlinear dynamic problems. *Journal of Applied Mechanics* **46**(2), 433–447

[54] Richard JP (2003) Time-delay systems: an overview of some recent advances and open problems. *Automatica* **39**(10), 1667–1694

[55] Lima RBC, Barros PR (2015) Identification of time-delay systems: a state-space realization approach. *IFAC-PapersOnLine* **48**(8), 254–259

[56] Yan R, Chen F, Dong S, Liu T (2016) Gradient-based step response identification of low-order model for time delay systems. In: *Proceedings of the 2016 Chinese Control and Decision Conference (CCDC)*, Yinchuan, China, pp 5814–5819

[57] Jones RW (2004) Recursive estimation of time-delay using a high-order Pade approximant. In: *Proceedings of SICE Annual Conference*, vol 2, pp 1328–1333

[58] Gawthrop PJ, Nihtilä MT (1985) Identification of time delays using a polynomial identification method. *Systems & Control Letters* **5**(4), 267–271

[59] Ren XM, Rad AB, Chan PT, Lo WL (2005) Online identification of continuous-time systems with unknown time delay. *IEEE Transactions on Automatic Control* **50**(9), 1418–1422

[60] Bedoui S, Ltaïef M, Abderrahim K (2013) Hierarchical gradient based identification of discrete-time delay systems. In: *Proceedings of the 52nd IEEE Conference on Decision and Control*, pp 4565–4570

[61] Higo T, Kawaguchi T, Itamoto M, Adachi S (2018) Recursive identification of fractional time-delay systems using discrete-time model. *IFAC-PapersOnLine* **51**(15), 939–944

[62] Ha H, Welsh JS (2016) Parameter and delay estimation of continuous-time models utilizing multiple filtering. In: *Proceedings of the IEEE 55th Conference on Decision and Control (CDC)*, Las Vegas, NV, USA, pp 1205–1210

[63] Ha H, Welsh JS, Alamir M (2018) Useful redundancy in parameter and time delay estimation for continuous-time models. *Automatica* **95**, 455–462

[64] Yadaiah N, Deekshatulu BL, Sivakumar L, Sree Hari Rao V (2007) Neural network algorithm for parameter identification of dynamical systems involving time delays. *Applied Soft Computing* **7**(3), 1084–1091

[65] Belkoura L, Richard JP, Fliess M (2009) Parameters estimation of systems with delayed and structured entries. *Automatica* **45**(5), 1117–1125

[66] Belkoura L (2010) Identifiability and algebraic identification of time delay systems. *IFAC Proceedings* **43**(2), 1–8

[67] Jin F, Qiu T (2019) Adaptive time delay estimation based on the maximum correntropy criterion. *Digital Signal Processing* **88**, 23–32

[68] Na J, Ren X, Xia Y (2014) Adaptive parameter identification of linear SISO systems with unknown time-delay. *Systems & Control Letters* **66**, 43–50

[69] Zheng G, Barbot JP, Boutat D (2013) Identification of the delay parameter for nonlinear time-delay systems with unknown inputs. *Automatica* **49**(6), 1755–1760

[70] Beauduin T, Fujimoto H (2017) Identification of system dynamics with time delay: a two-stage frequency domain approach. *IFAC-PapersOnLine* **50**(1), 10,870–10,875

[71] Yu L, Qiu TS, Song AM (2017) A time delay estimation algorithm based on the weighted correntropy spectral density. *Circuits, Systems, and Signal Processing* **36**(3), 1115–1128

[72] Xu L (2014) A proportional differential control method for a time-delay system using the Taylor expansion approximation. *Applied Mathematics and Computation* **236**, 391–399

[73] Chen F, Garnier H, Gilson M (2015) Robust identification of continuous-time models with arbitrary time-delay from irregularly sampled data. *Journal of Process Control* **25**, 19–27

[74] Ghoul Y, Ibn Taarit K, Ksouri M (2016) Online identification of continuous-time systems with multiple unknown time delays based on optimization approaches from sampled data. In: *Proceedings of the 7th International Conference on Sciences of Electronics, Technologies of Information and Telecommunications (SETIT)*, Hammamet,Tunisia, pp 204–208

[75] Gawthrop PJ, Nihtila MT, Rad AB (1989) Recursive parameter estimation of continuous systems with unknown time delay. *Control Theory Advanced Technology* **5**, 227–248

[76] Wang X, Su J, Zhang L (2018) Time-delay estimation for SISO systems using SWσ. *ISA Transactions* **80**, 43–53

[77] Fliess M, Sira–Ramírez H (2003) An algebraic framework for linear identification. *ESAIM: Control, Optimisation and Calculus of Variations* **9**, 151–168

[78] Ma SF, Leylaz G, Sun JQ (2021) Data-driven robust tracking control of underactuated mechanical systems using identified flat output and active disturbance rejection control. *International Journal of Control* pp 1–17

[79] Xin Y, Qin ZC, Sun JQ (2020) Robust experimental study of data-driven optimal control for an underactuated rotary flexible joint. *International Journal of Control, Automation and Systems* **18**(5), 1202–1214

[80] Huang JW, Lin JS (2008) Backstepping control design of a single-link flexible robotic manipulator. *IFAC Proceedings Volumes* **41**(2), 11,775–11,780

[81] Al-Saggaf UM, Mehedi IM, Mansouri R, Bettayeb M (2017) Rotary flexible joint control by fractional order controllers. *International Journal of Control, Automation and Systems* **15**(6), 2561–2569

[82] Jnifene A, Andrews W (2005) Experimental study on active vibration control of a single-link flexible manipulator using tools of fuzzy logic and neural networks. *IEEE transactions on instrumentation and measurement* **54**(3), 1200–1208

[83] Mansour T, Konno A, Uchiyama M (2008) Modified pid control of a single-link flexible robot. *Advanced Robotics* **22**(4), 433–449

[84] Li Y, Tong S, Li T (2013) Adaptive fuzzy output feedback control for a single-link flexible robot manipulator driven dc motor via backstepping. *Nonlinear Analysis: Real World Applications* **14**(1), 483–494

[85] Qin ZC, Xiong FR, Ding Q, Hernández C, Fernandez J, Schütze O, Sun JQ (2015) Multi-objective optimal design and validation of sliding mode control. *International Design Engineering Technical Conferences and Computers and Information in Engineering Conference* **57181**, V008T13A015

[86] Fliess M, Lévine J, Martin P, Rouchon P (1995) Flatness and defect of nonlinear systems: introductory theory and examples. *International journal of control* **61**(6), 1327–1361

[87] Fliess M, Lévine J, Martin P, Rouchon P (1999) A lie-backlund approach to equivalence and flatness of nonlinear systems. *IEEE Transactions on automatic control* **44**(5), 922–937

[88] Nakamura-Zimmerer T, Gong Q, Kang W (2021) Adaptive deep learning for high-dimensional Hamilton–Jacobi–Bellman equations. *SIAM Journal on Scientific Computing* **43**(2), A1221–A1247

[89] Cacace S, Cristiani E, Falcone M, Picarelli A (2012) A patchy dynamic programming scheme for a class of hamilton–jacobi–bellman equations. *SIAM Journal on Scientific Computing* **34**(5), A2625–A2649

[90] Al'Brekht E (1961) On the optimal stabilization of nonlinear systems. *Journal of Applied Mathematics and Mechanics* **25**(5), 1254–1266

[91] Lukes DL (1969) Optimal regulation of nonlinear dynamical systems. *SIAM Journal on Control* **7**(1), 75–100

[92] Osher S, Sethian JA (1988) Fronts propagating with curvature-dependent speed: Algorithms based on hamilton-jacobi formulations. *Journal of computational physics* **79**(1), 12–49

[93] Bokanowski O, Garcke J, Griebel M, Klompmaker I (2013) An adaptive sparse grid semi-lagrangian scheme for first order hamilton-jacobi bellman equations. *Journal of Scientific Computing* **55**(3), 575–605

[94] Falcone M, Ferretti R (2013) *Semi-Lagrangian approximation schemes for linear and Hamilton—Jacobi equations*. SIAM

[95] Darbon J, Osher S (2016) Algorithms for overcoming the curse of dimensionality for certain hamilton–jacobi equations arising in control theory and elsewhere. *Research in the Mathematical Sciences* **3**(1), 1–26

[96] Chow YT, Darbon J, Osher S, Yin W (2019) Algorithm for overcoming the curse of dimensionality for state-dependent hamilton-jacobi equations. *Journal of Computational Physics* **387**, 376–409

[97] Yegorov I, Dower PM (2021) Perspectives on characteristics based curse-of-dimensionality-free numerical approaches for solving hamilton–jacobi equations. *Applied Mathematics & Optimization* **83**(1), 1–49

[98] Kalise D, Kunisch K (2018) Polynomial approximation of high-dimensional hamilton–jacobi–bellman equations and applications to feedback control of semilinear parabolic pdes. *SIAM Journal on Scientific Computing* **40**(2), A629–A652

[99] Abu-Khalaf M, Lewis FL (2005) Nearly optimal control laws for nonlinear systems with saturating actuators using a neural network HJB approach. *Automatica* **41**(5), 779–791

[100] Cheng T, Lewis FL, Abu-Khalaf M (2007) Fixed-final-time-constrained optimal control of nonlinear systems using neural network hjb approach. *IEEE Transactions on Neural Networks* **18**(6), 1725–1737

[101] Izzo D, Öztürk E, Märtens M (2019) Interplanetary transfers via deep representations of the optimal policy and/or of the value function. *Proceedings of the Genetic and Evolutionary Computation Conference Companion* pp 1971–1979

[102] Jiang F, Chou G, Chen M, Tomlin CJ (2016) Using neural networks to compute approximate and guaranteed feasible hamilton-jacobi-bellman pde solutions. *arXiv preprint arXiv:161103158*

[103] Sirignano J, Spiliopoulos K (2018) Dgm: A deep learning algorithm for solving partial differential equations. *Journal of computational physics* **375**, 1339–1364

[104] Tassa Y, Erez T (2007) Least squares solutions of the hjb equation with neural network value-function approximators. *IEEE transactions on neural networks* **18**(4), 1031–1041

[105] Darbon J, Langlois GP, Meng T (2020) Overcoming the curse of dimensionality for some hamilton–jacobi partial differential equations via neural network architectures. *Research in the Mathematical Sciences* **7**(3), 1–50

[106] Darbon J, Meng T (2021) On some neural network architectures that can represent viscosity solutions of certain high dimensional hamilton–jacobi partial differential equations. *Journal of Computational Physics* **425**, 109,907

[107] Bachouch A, Huré C, Langrené N, Pham H (2021) Deep neural networks algorithms for stochastic control problems on finite horizon: numerical applications. *Methodology and Computing in Applied Probability* pp 1–36

[108] Han J, Jentzen A, Weinan E (2018) Solving high-dimensional partial differential equations using deep learning. *Proceedings of the National Academy of Sciences* **115**(34), 8505–8510

[109] Kingma DP, Ba J (2014) Adam: A method for stochastic optimization. *arXiv:14126980*

[110] Hastie T, Tibshirani R, Friedman J (2009) *The Elements of Statistical Learning: Data Mining, Inference, and Prediction*. Springer Series in Statistics

[111] Burnham KP, Anderson DR (2002) *Model Selection and Multi-model Inference*. Springer, Berlin,Germany

[112] Apkarian J, Karam P, Lévis M (2011) *Flexible Link Experiment for MatLab/Simulink Users*. Quanser Inc., Markham, Ontario, Canada

[113] Han J (1995) A class of extended state observers for uncertain systems. *Control and Decision* **10**, 85–88

[114] Guo B, Zhao ZL (2016) *Active Disturbance Rejection Control for Nonlinear Systems: An Introduction*. John Wiley & Sons

[115] Leylaz G, Wang S, Sun JQ (2021) Identification of nonlinear dynamical systems with time delay. *International Journal of Dynamics and Control* pp 1–12

[116] Hornik K, Stinchcombe M, White H (1990) Universal approximation of an unknown mapping and its derivatives using multilayer feedforward networks. *Neural networks* **3**(5), 551–560

[117] Diederik P K, Ba J (2014) Adam: A method for stochastic optimization. *arXiv:14126980 (2014)*

[118] Stevens BL, Lewis FL, Johnson EN (2015) *Aircraft control and simulation: dynamics, controls design, and autonomous systems*. John Wiley & Sons