Summer 2014

Optimal Control of Unknown Nonlinear System From Inputoutput Data

Xin Jin
Purdue University

Graduate School ETD Form 9
(Revised 01/14)

PURDUE UNIVERSITY
GRADUATE SCHOOL
Thesis/Dissertation Acceptance

This is to certify that the thesis/dissertation prepared

By Xin Jin

Entitled
OPTIMAL CONTROL OF UNKNOWN NONLINEAR SYSTEM FROM INPUT-OUTPUT DATA

For the degree of Master of Science in Engineering

Is approved by the final examining committee:

Yung Shin

Xinyan Deng

Bin Yao

To the best of my knowledge and as understood by the student in the *Thesis/Dissertation Agreement. Publication Delay, and Certification/Disclaimer (Graduate School Form 32)*, this thesis/dissertation adheres to the provisions of Purdue University's "Policy on Integrity in Research" and the use of copyrighted material.

Approved by Major Professor(s): Yung Shin

Approved by: David Anderson 5/27/2014

Head of the Department Graduate Program Date

OPTIMAL CONTROL OF UNKNOWN NONLINEAR SYSTEM FROM INPUT-OUTPUT DATA

A Thesis

Submitted to the Faculty

of

Purdue University

by

Xin Jin

In Partial Fulfillment of the

Requirements for the Degree

of

Master of Science in Engineering

August 2014

Purdue University

West Lafayette, Indiana

ACKNOWLEDGEMENTS

I would like to give my sincere gratitude to Dr. Yung C. Shin, who has been my Master advisor and with whom I have worked in the past years. I thank him for his guidance and support to help me understand the meaning of research and learn to deal with practical problems. His continuous advice and motivation has kept me on track for my research project at all times. It has been an honor and pleasure to work with and learning from him.

I learned many basic control theories from courses and lectures of Dr. Bin Yao, whom I would like to thank. His prodigious talent, strong motivation and intense interest in control really simulate my interest in control theory. I would also like to give my genuine thanks to Dr. Peter Meckl for his kind help. He allowed me to conduct a lot of experiments on his robot and gave me a lot of suggestions for my project.

Thanks are also due to Dr. Yung Shin, Dr. Bin Yao, and Dr. Xinyan Deng for agreeing to serve on my thesis committee and for sparing their invaluable time reviewing this thesis. I would like to thank all my friends and my lab mate Phuong Ngo for providing all kinds of support when I got stuck somewhere.

TABLE OF CONTENTS

LIST OF TABLES

Table Page

LIST OF FIGURES

Figure Page

Figure Page

ABSTRACT

Jin, Xin. M.S.E., Purdue University, August 2014. Optimal Control of Unknown
Nonlinear System from Input-Output Data. Major Professor: Yung Shin. School of
Mechanical Engineering.

Optimal control designers usually require a plant model to design a controller. The problem is the controller’s performance heavily depends on the accuracy of the plant model. However, in many situations, it is very time-consuming to implement the system identification procedure and an accurate structure of a plant model is very difficult to obtain. On the other hand, neuro-fuzzy models with product inference engine, singleton fuzzifier, center average defuzzifier, and Gaussian membership functions can be easily trained by many well-established learning algorithms based on given input-output data pairs. Therefore, this kind of model is used in the current optimal controller design.

Two approaches of designing optimal controllers of unknown nonlinear systems based on neuro-fuzzy models are presented in the thesis. The first approach first utilizes neuro-fuzzy models to approximate the unknown nonlinear systems, and then the feasible-direction algorithm is used to achieve the numerical solution of the Euler-Lagrange equations of the formulated optimal control problem. This algorithm uses the steepest descent to find the search direction and then apply a one-dimensional search routine to find the best step length. Finally several nonlinear optimal control problems are simulated and the results show that the performance of the proposed approach is quite

similar to that of optimal control to the system represented by an explicit mathematical model. However, due to the limitation of the feasible-direction algorithm, this method cannot be applied to highly nonlinear and dimensional plants.

Therefore, another approach that can overcome these drawbacks is proposed. This method utilizes Takagi-Sugeno (TS) fuzzy models to design the optimal controller. TS fuzzy models are first derived from the direct linearization of the neuro-fuzzy models, which is close to the local linearization of the nonlinear dynamic systems. The operating points are chosen so that the TS fuzzy model is a good approximation of the neuro-fuzzy model. Based on the TS fuzzy model, the optimal control is implemented for a nonlinear two-link flexible robot and a rigid asymmetric spacecraft, thus providing the possibility of implementing the well-established optimal control method on unknown nonlinear dynamic systems.

CHAPTER 1. INTRODUCTION

1.1 Motivation and Literature Review

1.1.1 Difficulties in Finding an Accurate Model

Most control strategies are based on an explicit mathematical model of the system. It is well known that modeling and identification procedures for the dynamics of a given nonlinear system are most time consuming iterative endeavors that require model design, parameter identification and model validation at each step of the iteration. Moreover, it is quite reasonable to say that no matter how well one tries to describe a system in terms of a set of mathematical equations, there will always be a model mismatch. This mismatch between the model and the actual system is due to the imperfection of available system identification techniques, high nonlinearity and unknown system model form. Since there are a limited number of mathematical functions (sine, cosine, exponential, logarithm, etc), it is rather impossible to only use them to describe the actual system whose components are not fully understood at the microscopic level [1].

Traditionally, controllers have been designed from simplified models that were obtained from fundamental physical laws such as Ohm's law and Kirchhoff's voltage and current laws in electrical circuits, Faraday's law and Ampere's law in magnetic fields, Lagrange-Euler equations and Newton's formula in mechanics. To be able to apply these fundamental physical laws, many assumptions have to be made during the system identification process. The outcome of the identification process is usually an over-simplified model. Therefore, if the system for which a controller is designed is too simplified, one may cannot meet the control objective [1].

1.1.2 Neuro-Fuzzy Models

The difficulties in obtaining accurate models from fundamental physical laws motivate the use of neuro-fuzzy models to represent unknown systems. It has become an active research field because of its unique merits in representing complex nonlinear system behavior. It is essentially a multimodel approach in which individual rules (where each rule act like a “local model”) are combined to describe the global behavior of the system [2]. Primary advantages of this approach include the explicit knowledge representation in the form of if-then rules, the mechanism of reasoning in human-understandable terms, and the ability of universal approximation, which means it can approximate any nonlinear function to arbitrary accuracy as proved by the Stone-Weierstrass theorem [3]. Therefore, a neuro-fuzzy model is used in the current work to model systems instead of using physical laws.

When using a neuro-fuzzy model to approximate an unknown system, it is desired that the model can approximate the training data as closely as possible while including as few rules as possible. The tradeoff between them is a fundamental principle underlying various general theories of statistical modeling and inductive inference [4] [5]. Several research efforts have been made in the fuzzy logic community to strike a balance between reducing the fitting error and increasing the model complexity. For instance, back-propagation algorithm [6], gradient descent, least square [7], clustering [8], and orthogonal least square (OLS) algorithm [9] [10] [11] have been developed to train the neuro-fuzzy models. The most efficient and widely used method is the OLS algorithm, which is used in the thesis to train the neuro-fuzzy model. Specifically, after an initial fuzzy system is first constructed with as many fuzzy basis functions as input-output pairs, then the OLS algorithm is used to select significant fuzzy basis functions to construct a final neuro-fuzzy model [9] [10].

1.1.3 Recent Developments in Optimal Control

Optimal control theory that has played an important role in the design of modern control systems has as its objective the maximization of return from, or the minimization of the cost of, the operation of physical, social, and economic processes. It was introduced in the 1950s with use of dynamic programming (leading to Hamilton-Jacobi-Bellman partial differential equations) and the Pontryagin maximum principle (a generalization of the Euler-Lagrange equations deriving from the calculus of variations) [12] [13]. However, the optimal control of nonlinear systems is still one of the most challenging and difficult subjects in the control field. In recent years, adaptive/approximate dynamic programming algorithms [14] [15] [16] have gained much attention from researchers. It is a reinforcement learning approach based on adaptive critics to solve dynamic programming problems utilizing function approximation for the value function. It can be based on value iterations or policy iterations. In [17], a successive approximation method using generalized Hamilton-Jacobi-Bellman equation was proposed to solve the near-optimal control problem for affine nonlinear discrete time systems, which requires a small perturbation assumption and an initially stable policy. The complete dynamics of affine nonlinear systems were assumed known in the approach.

In [18], the Q-learning policy iteration method was used to solve the optimal strategies for linear discrete time without requiring known system dynamics where in the system dynamics are defined as constant matrices. However, this method is intended only for linear systems and it is not clear how to select the number of iterations required for convergence and stability.

Optimal control strategies for unknown affine nonlinear discrete time systems of the form $x(k+1) = f(x(k)) + g\big(x(k)\big)u(k)$ [19] [20] [21] [22] or continuous time linear systems [23] using offline trained neural networks have been presented. The proposed scheme does not require explicit knowledge of the system dynamics as only the learned neural network model is needed. It first uses a neural network to learn the complete plant dynamics and then offline adaptive/approximate dynamic programming is attempted to use only the learned neural network system model, resulting in a novel optimal control law. However, this scheme can only be applied to the specific type of affine nonlinear discrete time systems or continuous time linear systems. Therefore, two ways of optimal control that can be applied to the unknown nonlinear systems in a more general form of $\dot{x} = f(x,u)$ are developed in the current work.

1.1.4 Control Based on TS Fuzzy Models

The TS fuzzy model is a powerful practical engineering tool for modeling and control of complex nonlinear systems. It is shown to be a universal function approximator that can approximate any smooth nonlinear functions to any degree of accuracy [24] [25] and is less prone to the curse of dimensionality than other fuzzy models [26]. It is similar to the concept of typical piecewise linear approximation methods in nonlinear control, which is achieved by linearizing a system around a number of nominal operating points and then applying linear feedback control methods to each local linear model. However, such linear models can only guarantee the stability and performance of the system at the operating points [27]. On the other hand, the TS fuzzy model approximates an entire nonlinear dynamic system by fuzzy inferencing between affine local linear time invariant models [28]. Therefore, it provides a way of designing controllers based on local linear models and analyzing stability or performance based on the global nonlinear model [29].

The TS fuzzy model based controller is usually to design a feedback controller for each local model and the stability of the overall system is then determined by Lyapunov stability analysis [30] [31] [32]. This kind of design methods must find a common positive definite matrix to satisfy the Lyapunov equations, which can be difficult to find, especially when the number of rules required to give a good plant model is large. Besides, this method proves to be conservative in many cases. However, there has been a lot of research to relax the stability conditions by utilizing the property of the fuzzy summations at different levels [33] [34] or analyze the stability of the fuzzy controller that does not share the same premise membership functions with the TS fuzzy model [35] [36]. Moreover, some people constructed a globally sliding model fuzzy logic controller by blending all local state feedback controllers together with a sliding model controller. Therefore, they can design a globally stable fuzzy logic controller without finding a common Lyapunov function and overall robustness and tracking ability of the entire closed-loop system can also be significantly improved [37] [38]. With the development of the TS fuzzy model based controller, it begins to show its advantages over conventional nonlinear controllers.

To take advantage of TS fuzzy model based controllers, the research for establishment of a TS fuzzy model has attracted great attention. There are basically two kinds of approaches to identify a TS fuzzy model. One is to linearize the original nonlinear system in a number of operating points when the model of the system is known. The other is based on the data generated from the unknown nonlinear system [29], which is more of interest to us. The second approach was based on the idea of consecutive structure and parameter identification. Structure identification includes estimation of local points of the rules by fuzzy clustering. With fixed antecedent parameters, the TS model transforms into a linear model, where the parameters are obtained by the recursive least square method [39], back-propagation [40] or genetic algorithm [41]. The objective of this approach is to minimize the global nonlinear prediction error between the TS fuzzy model and the corresponding original nonlinear system. However, all these methods may result in local models that are significantly different from local linearization of the nonlinear systems although they can offer good global prediction performance [42]. So when the TS fuzzy model generated from data is used as a basis for a fuzzy gain-scheduled controller, these methods can hardly provide a satisfactory model for control since the local linear models are used to design local linear controllers.

The authors in [43] presented an approach to achieve accurate global nonlinear prediction and at the same time its local models that are close approximations to the local linearization of the nonlinear dynamic systems. This defines a difficult multi-objective identification problem, namely, the construction of a dynamic model that is a good approximation of both local and global dynamics of the underlying system. It has been shown that constrained and regularized identification methods may improve interpretability of constituent local models as local linearization, and the locally weighted least squares method may explicitly address the tradeoff between the local and global accuracy of TS fuzzy models. However, the practical importance of the approach was illustrated only by very simple examples. One may expect that the problems related to interpretability and identifiability will be much more pronounced when more complex higher order and multivariable examples are considered. In this work, a different identification approach to circumvent these difficulties is presented and will be used for the optimal control.

1.1.5 Control Based on Neural Network Models

Except for the TS fuzzy models, the most popular universal function approximators used in control is the neural network models. During the adaptive nature of the neural network models, most neural network controllers are adaptive neural network controllers. The first stable and efficient neural network controller designs were proposed in [46] [47]. Then the Lyapunov's stability theory was applied to the controller design and several stable adaptive neural network control approaches were developed [48] [49]. Although these approaches do not require nonlinear dynamic functions to be linearly parameterized and still can achieve a good control performance, they were applied only to a relatively simple class of nonlinear systems because the nonlinear uncertainties and interconnections must satisfy the strict matching condition. This mismatch problem was first overcome by using a modified Lyapunov function and the backstepping techniques in [50]. The developed control scheme guarantees the uniform ultimate boundedness of the closed-loop adaptive systems. Now the nonlinear adaptive neural network control has been widely used for continuous nonlinear systems [51] [52], and for nonlinear discrete time systems [53].

People began to research the adaptive neural network control in other aspects. Adaptive neural network control was developed for a class of MIMO nonlinear systems with unknown bounded disturbances in discrete-time domain [54] and continuous-time domain [55]. In [56], in view of possible time-delays in practical systems, approximation-based adaptive neural network control has been also addressed for nonlinear SISO time delay systems with constant virtual control coefficients. In [57], the nonlinear MIMO time-delay system was addressed by constructing a novel quadratic type Lyapunov functional. In [58], a robust adaptive neural network control was investigated for a general class of uncertain MIMO nonlinear systems with unknown control coefficient matrices and input nonlinearities. The variable structure control in combination with backstepping and Lyapunov synthesis was proposed for adaptive neural network control design with guaranteed stability. In [59], the switched nonlinear systems with switching jumps and uncertainties in both system models and switching signals were also addressed. Overall, during the past two decades, neural network control has attracted considerable attention because of its inherent capability for modeling and controlling highly uncertain, nonlinear and complex systems.

1.2 Research Objectives

1.2.1 Nonlinear Optimal Control with Neuro-Fuzzy Models

As in section 1.1.2, the existing optimal control schemes can only be applied to the specific type of affine nonlinear unknown discrete time systems or unknown continuous time linear systems. So this work seeks to extend them to the nonlinear systems in the form of $\dot{x} = f(x, u)$. First, a neuro-fuzzy model with product inference engine, singleton fuzzifier, center average defuzzifier, and Gaussian membership functions is trained by the OLS learning algorithm based on given input-output data pairs to model the unknown systems. Then the feasible-direction algorithm [44] [45] is used to achieve the numerical solution of the Euler-Lagrange equations of the formulated discrete time optimal control problem. This algorithm uses the steepest descent to find the search direction and then apply a one-dimensional search routine to find the best step length. It has a very high computational efficiency and very easy to implement. Finally the proposed approach is applied to several nonlinear systems to show its efficiency for control of unknown nonlinear systems. The results are quite similar to that of optimal control to the systems represented by explicit mathematical models. However, due to the limitation of the feasible-direction numerical algorithm, it cannot be applied to a too complex system or a control process with too many time steps. Therefore, a better way which also utilized the TS fuzzy model is then developed.

1.2.2 Nonlinear Optimal Control with TS Fuzzy Models

The TS fuzzy model is also used to model the complex nonlinear systems and shown to be a universal function approximator. However, unlike neuro-fuzzy model, most training algorithms can hardly provide satisfied local accuracy to be used for control. In this thesis, a novel way to derive a TS fuzzy model whose local models are close approximations to the local linearization of the nonlinear systems is presented. A fuzzy model with product inference engine, singleton fuzzifier, center average defuzzifier, and Gaussian membership functions is first used to approximate the global nonlinear systems. Then the TS fuzzy model is derived from the direct linearization of the fuzzy model at prescribed operating points. Therefore, it is close to the local linearization of the nonlinear dynamic systems. Besides, the operating points are chosen so that the TS fuzzy model is a good approximation of the neuro-fuzzy model, which means it can obtain a good approximation of the nonlinear dynamic systems. Then the optimal control is implemented on each linear affine system and the overall control action is derived from the fuzzy inferencing of each optimal control action. Finally, the proposed method is simulated for several examples and also applied to a very complex two link flexible robot in the laboratory, which demonstrates the wide practicality and effectiveness of the proposed algorithm for controlling unknown nonlinear dynamic systems.

1.3 Overview of Thesis

Chapter 2 starts introduction of neuro-fuzzy model and orthogonal least square algorithm as a background. Then optimal control using neuro-fuzzy model is discussed in details. Finally, the simulation results for several unknown nonlinear systems are shown and discussed.

Chapter 3 has a similar structure to the previous section. The TS fuzzy model is first introduced. Then the identification of TS fuzzy model and control based on it are focused. The experimental and simulation results for some practical examples are given in the last part.

Chapter 4 presents the conclusion of the current work and the recommendations for future work.

CHAPTER 2. OPTIMAL CONTROL WITH NEURO-FUZZY MODELS

2.1 Neuro-Fuzzy Models

A neuro-fuzzy model consists of four principal elements as shown in Figure 2.1 [26]: fuzzifier, fuzzy rule base, fuzzy inference engine, and defuzzifier. For the nonlinear discrete time multi-input, multi-output (MIMO) system, it can be separated into a group of multi-input, single-output (MISO) systems: $U \in R^{n+m} \to R$, where U is compact. The neuro-fuzzy model is established in state space form such that the inputs of the neuro-fuzzy system are the n states and m inputs of the system and the output of it is the each state value of the system at the next time instance.

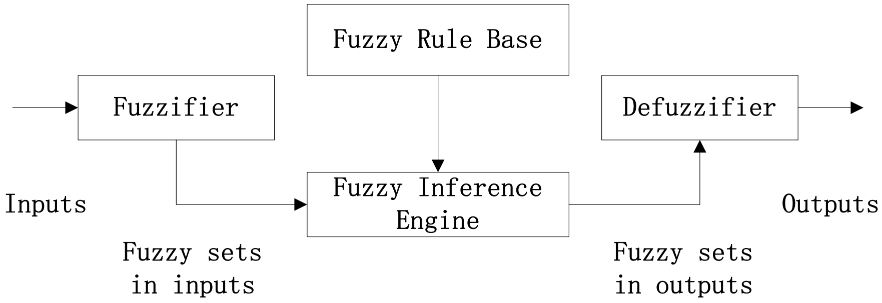

Figure 2.1. Basic Configuration of Neuro-Fuzzy Systems.

The fuzzifier performs a mapping from the observed crisp input space $U \in R^{n+m}$ to the fuzzy sets defined in U. The fuzzy rule base consists of a set of linguistic rules in the form of “IF a set of conditions are satisfied, THEN a set of consequences are inferred”. The fuzzy inference engine is the decision making logic that employs fuzzy rules from the fuzzy rule base to determine a mapping from the fuzzy sets in the input space U to the fuzzy sets in the output space R. The defuzzifier performs a mapping from the fuzzy sets in R to crisp points in R [8] [9].

MIMO neuro-fuzzy systems with singleton fuzzifier, product inference, centroid defuzzifier, and Gaussian membership function can be represented as follows, for $p = 1, \cdots, n$ [8] [9].

$$x_p(k+1) = f_p(\boldsymbol{x}(k), \boldsymbol{u}(k)) = \frac{\sum_{l=1}^{M_p} w_p^l \left(\prod_{i=1}^{n} \mu_{A_{ip}^l}(x_i(k))\right)\left(\prod_{j=1}^{m} \mu_{B_{jp}^l}(u_j(k))\right)}{\sum_{l=1}^{M_p}\left(\prod_{i=1}^{n} \mu_{A_{ip}^l}(x_i(k))\right)\left(\prod_{j=1}^{m} \mu_{B_{jp}^l}(u_j(k))\right)} \quad (1)$$

where $f_p \colon U \in R^{n+m} \to R$, $x_p(k+1)$ is the pth state of the system at time index $k+1$, w_p^l is the singleton, $\boldsymbol{x}(k) = [x_1(k), x_2(k), \cdots, x_n(k)]^T$ is the state vector of the system at time index k and $\mu_{A_{ip}^l}(x_i(k))$ is the Gaussian membership function, defined by

$$\mu_{A_{ip}^l}(x_i(k)) = \exp\left(-\frac{1}{2}\left(\frac{x_i(k) - x_{ip}^l}{\sigma_{x_i p}^l}\right)^2\right) \quad (2)$$

where x_{ip}^l and $\sigma_{x_i p}^l$ are the center and width of $x_i(k)$ respectively. Similarly $\boldsymbol{u}(k) = [u_1(k), u_2(k), \cdots, u_m(k)]^T$ is the input vector of the system at time index k and $\mu_{B_{jp}^l}(u_j(k))$ is the Gaussian membership function, defined by

$$\mu_{B_{jp}^l}(u_j(k)) = \exp\left(-\frac{1}{2}\left(\frac{u_j(k) - u_{jp}^l}{\sigma_{u_j p}^l}\right)^2\right) \quad (3)$$

where u_{jp}^l and $\sigma_{u_j p}^l$ are the center and width of $u_j(k)$.

2.2 Orthogonal Least Square Algorithm

The orthogonal least square (OLS) algorithm is a very efficient and widely used way of training a neuro-fuzzy model. The OLS algorithm is a one-pass regression procedure, and is therefore much faster than other algorithms. Also, the OLS algorithm generates a robust neuro-fuzzy model that is not sensitive to noise in its inputs [8] [9]. In this paper, the width (σ) of the neuro-fuzzy model are first fixed to cover the input state region. The resulting neuro-fuzzy model is then equivalent to a series expansion of fuzzy basis functions, which is linear in parameters [8] [9].

$$f_p(\boldsymbol{x}(k), \boldsymbol{u}(k)) = \sum_{l=1}^{M_p} w_p^l h_p^l(\boldsymbol{x}(k), \boldsymbol{u}(k)) \tag{4}$$

where

$$h_p^l(\boldsymbol{x}(k), \boldsymbol{u}(k)) = \frac{\left(\prod_{i=1}^{n} \mu_{A_{ip}^l}(x_i(k))\right)\left(\prod_{j=1}^{m} \mu_{B_{jp}^l}(u_j(k))\right)}{\sum_{l=1}^{M_p}\left(\prod_{i=1}^{n} \mu_{A_{ip}^l}(x_i(k))\right)\left(\prod_{j=1}^{m} \mu_{B_{jp}^l}(u_j(k))\right)} \tag{5}$$

However, since the normalization factor in the denominator is not known before the fuzzy basis function is selected, a pseudo-fuzzy basis functions is needed to define as follows [10]:

$$q_p^l(\boldsymbol{x}(k), \boldsymbol{u}(k)) = \left(\prod_{i=1}^{n} \mu_{A_{ip}^l}(x_i(k))\right)\left(\prod_{j=1}^{m} \mu_{B_{jp}^l}(u_j(k))\right) \tag{6}$$

Then the fuzzy basis function can be expressed in terms of pseudo-fuzzy basis functions as follows [10]:

$$h_p^l(\boldsymbol{x}(k), \boldsymbol{u}(k)) = \frac{q_p^l(\boldsymbol{x}(k), \boldsymbol{u}(k))}{\sum_{l=1}^{M_p} q_p^l(\boldsymbol{x}(k), \boldsymbol{u}(k))} \tag{7}$$

For N input-output training pairs $\left([x^t(k); u^t(k)], x_p^t(k+1)\right)$, we can get from $t = 1$ to N in the following matrix form [11]:

$$\boldsymbol{d} = \boldsymbol{H}\boldsymbol{w} + \boldsymbol{e} \tag{8}$$

where $\boldsymbol{d} = \left[x_p^1(k+1), \cdots, x_p^N(k+1)\right]^T$, $\boldsymbol{H} = \left[\boldsymbol{h}_{\boldsymbol{p}}^{\boldsymbol{1}}, \cdots, \boldsymbol{h}_{\boldsymbol{p}}^{M_p}\right]$ with

$\boldsymbol{h}_{\boldsymbol{p}}^{\boldsymbol{l}} = \left[h_p^l\big(\boldsymbol{x}^1(k); \boldsymbol{u}^1(k)\big), \cdots, h_p^l\big(\boldsymbol{x}^N(k); \boldsymbol{u}^N(k)\big)\right]^T$, $\boldsymbol{w} = [w_p^1, \cdots, w_p^{M_p}]^T$, and $\boldsymbol{e} =$

$[e^1, \cdots, e^N]^T$.

The classical Gram-Schmidt orthogonal least-squares algorithm is used to determine the significant pseudo-fuzzy basis functions which then can be normalized to fuzzy basis functions [9] and the weighting factor can be calculated as [11]:

$$\boldsymbol{w} = (\boldsymbol{H}^T\boldsymbol{H})^{-\boldsymbol{1}}\boldsymbol{H}^T\boldsymbol{d} \tag{9}$$

2.3 Optimal Control Based on Neuro-Fuzzy Models

The procedure of designing a discrete time optimal controller for nonlinear systems represented by a neuro-fuzzy model is presented in this section. In this paper, a feasible-direction algorithm is used for achieve the numerical solution of the Euler-Lagrange equations of the formulated discrete time optimal control problem [44].

The general problem considered in the solution algorithm is that of minimizing a cost function:

$$J = \theta[\boldsymbol{x}(N)] + \sum_{k=0}^{N-1} \varphi[\boldsymbol{x}(k), \boldsymbol{u}(k)] \tag{10}$$

subject to the MIMO neuro-fuzzy model trained by OLS algorithm for $p = 1, \cdots, n$:

$$x_p(k+1) = f_p(\boldsymbol{x}(k), \boldsymbol{u}(k)) = \frac{\sum_{l=1}^{M_p} w_{lp} \prod_{i=1}^{n} \exp\left(-\frac{1}{2}\left(\frac{x_i(k)-x_{ip}^l}{\sigma_{x_ip}^l}\right)^2\right) \prod_{j=1}^{m} \exp\left(-\frac{1}{2}\left(\frac{u_j(k)-u_{jp}^l}{\sigma_{u_jp}^l}\right)^2\right)}{\sum_{l=1}^{M_p} \prod_{i=1}^{n} \exp\left(-\frac{1}{2}\left(\frac{x_i(k)-x_{ip}^l}{\sigma_{x_ip}^l}\right)^2\right) \prod_{j=1}^{m} \exp\left(-\frac{1}{2}\left(\frac{u_j(k)-u_{jp}^l}{\sigma_{u_jp}^l}\right)^2\right)} \tag{11}$$

$$\boldsymbol{x}(0) = \boldsymbol{x}_0 \tag{12}$$

The augmented cost function is represented by

$$J_a = \theta[\boldsymbol{x}(N)] + \sum_{k=0}^{N-1} \varphi[\boldsymbol{x}(k), \boldsymbol{u}(k)] + \boldsymbol{\lambda}^T(k+1)\big(\boldsymbol{f}[\boldsymbol{x}(k), \boldsymbol{u}(k)] - \boldsymbol{x}(k+1)\big) \quad (13)$$

The gradient of J_a with respect to u is given by

$$\boldsymbol{g}(k) = \frac{\partial \sum_{k=0}^{N-1} \varphi[\boldsymbol{x}(k),\boldsymbol{u}(k)]}{\partial \boldsymbol{u}(k)} + \frac{\partial \boldsymbol{f}[\boldsymbol{x}(k),\boldsymbol{u}(k)]}{\partial \boldsymbol{u}(k)}^T \boldsymbol{\lambda}(k+1) \quad (14)$$

where $\frac{\partial \boldsymbol{f}[\boldsymbol{x}(k),\boldsymbol{u}(k)]}{\partial \boldsymbol{u}(k)} = \begin{pmatrix} \frac{\partial f_1[\boldsymbol{x}(k),\boldsymbol{u}(k)]}{\partial u_1(k)} & \cdots & \frac{\partial f_1[\boldsymbol{x}(k),\boldsymbol{u}(k)]}{\partial u_m(k)} \\ \vdots & \ddots & \vdots \\ \frac{\partial f_n[\boldsymbol{x}(k),\boldsymbol{u}(k)]}{\partial u_1(k)} & \cdots & \frac{\partial f_n[\boldsymbol{x}(k),\boldsymbol{u}(k)]}{\partial u_m(k)} \end{pmatrix}$ and $k = 1, 2, \dots, N-1$.

If we use an explicit mathematical model of the nonlinear system, $\frac{\partial f[x(k),u(k)]}{\partial u(k)}$ is easy to derive. However, if we only have a neuro-fuzzy model of the system, $\frac{\partial f[x(k),u(k)]}{\partial u(k)}$ should be computed as:

$$\frac{\partial f_p[\boldsymbol{x}(k),\boldsymbol{u}(k)]}{\partial u_q(k)} = \sum_{l_a=1}^{M_p} \sum_{l_b=1}^{M_p} a_p^{l_a} a_p^{l_b} q_p^{l_a} \left(w_{l_a p} - w_{l_b p}\right) \quad (15)$$

where

$$a_p^{l_a} = \prod_{i=1}^{n} \exp\left(-\frac{1}{2}\left(\frac{x_i(k) - x_{ip}^{l_a}}{\sigma_{x_i p}^{l_a}}\right)^2\right) \prod_{j=1}^{m} \exp\left(-\frac{1}{2}\left(\frac{u_j(k) - u_{jp}^{l_a}}{\sigma_{u_j p}^{l_a}}\right)^2\right) \quad (16)$$

$$a_p^{l_b} = \prod_{i=1}^{n} \exp\left(-\frac{1}{2}\left(\frac{x_i(k) - x_{ip}^{l_b}}{\sigma_{x_i p}^{l_b}}\right)^2\right) \prod_{j=1}^{m} \exp\left(-\frac{1}{2}\left(\frac{u_j(k) - u_{jp}^{l_b}}{\sigma_{u_j p}^{l_b}}\right)^2\right) \quad (17)$$

$$q_p^{l_a} = -\frac{u_q(k) - u_{qp}^{l_a}}{\left(\sigma_{u_q p}^{l_a}\right)^2} \quad (18)$$

From the gradient of J_a with respect to x, we can get

$$\boldsymbol{\lambda}(k) = \frac{\partial \sum_{k=0}^{N-1} \varphi[\boldsymbol{x}(k),\boldsymbol{u}(k)]}{\partial \boldsymbol{x}(k)} + \frac{\partial \boldsymbol{f}[\boldsymbol{x}(k),\boldsymbol{u}(k)]}{\partial \boldsymbol{x}(k)}^T \boldsymbol{\lambda}(k+1) \quad (19)$$

$$\boldsymbol{\lambda}(N) = \frac{\partial \theta[x(N)]}{\partial \boldsymbol{x}(N)} \quad (20)$$

where $\frac{\partial \boldsymbol{f}[\boldsymbol{x}(k),\boldsymbol{u}(k)]}{\partial \boldsymbol{x}(k)} = \begin{pmatrix} \frac{\partial f_1[\boldsymbol{x}(k),\boldsymbol{u}(k)]}{\partial x_1} & \cdots & \frac{\partial f_1[\boldsymbol{x}(k),\boldsymbol{u}(k)]}{\partial x_n} \\ \vdots & \ddots & \vdots \\ \frac{\partial f_n[\boldsymbol{x}(k),\boldsymbol{u}(k)]}{\partial x_1} & \cdots & \frac{\partial f_n[\boldsymbol{x}(k),\boldsymbol{u}(k)]}{\partial x_n} \end{pmatrix}$ and $k = 1, 2, \dots, N-1$.

Similar to $\frac{\partial f[x(k),u(k)]}{\partial u(k)}$, $\frac{\partial f[x(k),u(k)]}{\partial x(k)}$ for the neuro-fuzzy model should be represented as:

$$\frac{\partial f_p[\boldsymbol{x}(k),\boldsymbol{u}(k)]}{\partial x_q} = \sum_{l_a=1}^{M_p} \sum_{l_b=1}^{M_p} a_p^{l_a} a_p^{l_b} q_p^{l_a} \left(w_{l_a p} - w_{l_b p}\right) \tag{21}$$

where

$$a_p^{l_a} = \prod_{i=1}^{n} \exp\left(-\frac{1}{2}\left(\frac{x_i(k)-x_{ip}^{l_a}}{\sigma_{x_i p}^{l_a}}\right)^2\right) \prod_{j=1}^{m} \exp\left(-\frac{1}{2}\left(\frac{u_j(k)-u_{jp}^{l_a}}{\sigma_{u_j p}^{l_a}}\right)^2\right) \tag{22}$$

$$a_p^{l_b} = \prod_{i=1}^{n} \exp\left(-\frac{1}{2}\left(\frac{x_i(k)-x_{ip}^{l_b}}{\sigma_{x_i p}^{l_b}}\right)^2\right) \prod_{j=1}^{m} \exp\left(-\frac{1}{2}\left(\frac{u_j(k)-u_{jp}^{l_b}}{\sigma_{u_j p}^{l_b}}\right)^2\right) \tag{23}$$

$$q_p^{l_a} = -\frac{x_q(k)-x_{qp}^{l_a}}{\left(\sigma_{x_q p}^{l_a}\right)^2} \tag{24}$$

Then the structure of the solution algorithm to find the optimal state trajectories and control inputs can be described as follows [44]:

Step 1: Select a feasible initial control trajectory $\boldsymbol{u}^0(k)$, set the iteration index $i = 0$.

Step 2: Using $\boldsymbol{u}^i(k)$, solve (11) from the initial condition (12) to obtain $\boldsymbol{x}^i(k)$.

Step 3: Using $\boldsymbol{u}^i(k)$ and $\boldsymbol{x}^i(k)$, solve (19) from terminal condition (20) to obtain Lagrange multipliers $\boldsymbol{\lambda}^i$(k) and calculate gradients $\boldsymbol{g}^i(k)$ from (14).

Step 4: Specify a search direction: $\boldsymbol{p}^i(k) = -\boldsymbol{g}^i(k)$.

Step 5: Apply a one-dimensional search routine along $\boldsymbol{p}^i(k)$ to obtain $\boldsymbol{u}^{i+1}(k)$. The corresponding line-optimization problem minimizes the following quantity:

$$\min_{\alpha>0} J[\boldsymbol{u}^i(k)+\alpha\boldsymbol{p}^i(k)]$$

Step 6: If for a give scalar $\varepsilon > 0$, the inequalities

$$\left[\boldsymbol{g}^i(k),\boldsymbol{g}^i(k)\right] < \varepsilon$$

hold, stop. Otherwise, set $i = i + 1$ and go to step 2.

In step 4, several methods such as conjugate gradient methods or quasi-Newton methods can also be used for the specification of the search direction $p^i(k)$ [60]. All these methods use a search direction that satisfies $\left[p^i(k), g^i(k)\right] < 0$, which guarantees that the derivative $\frac{\partial J}{\partial \alpha}$ is always negative for $\alpha = 0$ (except for $u^i(k)$, which is a stationary point), and therefore the objective function can be improved for some $\alpha > 0$.

In step 5, there are many different ways to search for the best step length for the line search algorithm such as Wolfe conditions, Goldstein conditions or backtracking approach [61]. For the neuro-fuzzy model, the computation procedures are rather complex. Thus, we need to use the following forward-backward method to find the best step length α in a range $[c, d]$:

(1) Given $h = 0.1$, evaluate $J(\alpha_k)$, where $\alpha_k = c$ and $k = 0$.

(2) Compare the objective function values. Set $\alpha_{k+1} = \alpha_k + h$ and evaluate $J_{k+1} = J(\alpha_{k+1})$. If $J_{k+1} < J_k$, go to forward step (3); otherwise, go to stop step (4).

(3) Forward step: Set $\alpha_k = \alpha_{k+1}, J_k = J_{k+1}$, $k = k + 1$, go to (2).

(4) Stop step: Set $c = \alpha_k$, $d = \alpha_{k+1}$, output $[c, d]$ and stop.

Then choose a smaller h and do the iteration again, until $(d - c) \leq \bar{\varepsilon}$. After that choose $\alpha = \frac{c+d}{2}$.

2.4 Simulation Results

In this section, three simulation examples that illustrate the effectiveness of the proposed method are presented. They are the carriage and nonlinear spring system [62], the rigid asymmetric spacecraft [63] and the nonlinear continuous stirred tank reactor [64].

2.4.1 Carriage with Nonlinear Spring

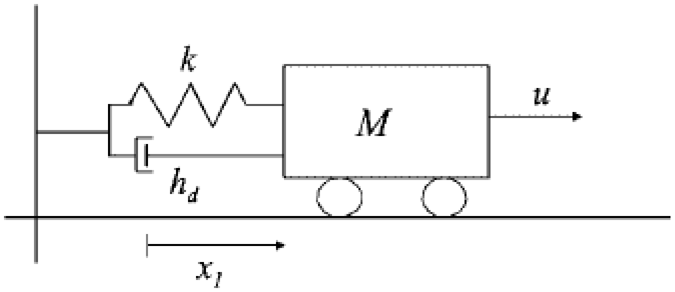

Figure 2.2. Carriage and Nonlinear Spring.

The optimal control law is applied to a cart with a mass M moving on the plane. This carriage is attached to the wall via a spring with elasticity k given by

$$k = k_0 e^{-x_1} \tag{25}$$

where x_1 is the displacement of the carriage from the equilibrium position associated with the external force u. Finally, a damper with damping factor h_d affects the system in a resistive way. The model of the system is given by the following continuous-time state-space nonlinear model [62].

$$\dot{x}_1(t) = x_2(t) \tag{26}$$

$$\dot{x}_2(t) = -\frac{k_0}{M} e^{-x_1(t)} x_1(t) - \frac{h_d}{M} x_2(t) + \frac{u(t)}{M} \tag{27}$$

where x_2 is the carriage velocity. The parameters of the system are $M = 1\ kg$, $k_0 = 0.33\ N/m$, while the damping factor is $h_d = 1.1$. An Euler approximation of system with sampling time $T_c = 0.4\ s$ is given by [62]

$$x_1(k+1) = x_1(k) + T_c x_2(k) \tag{28}$$

$$x_2(k+1) = x_2(k) - T_c \frac{k_0}{M} e^{-x_1(k)} x_1(k) - T_c \frac{h_d}{M} x_2(k) + T_c \frac{u(k)}{M} \tag{29}$$

Since the equation (28) can be easily known from the physical meanings of x_1 and x_2, only the state equation (29) needs to be approximated by the following neuro-fuzzy model:

$$x_2(k+1) = \frac{\sum_{l=1}^{M_2} w_{l2} \exp\left(-\frac{1}{2}\left(\frac{x_1(k)-x_{12}^l}{\sigma_{x_12}^l}\right)^2\right)\exp\left(-\frac{1}{2}\left(\frac{x_2(k)-x_{22}^l}{\sigma_{x_22}^l}\right)^2\right)\exp\left(-\frac{1}{2}\left(\frac{u(k)-u_{12}^l}{\sigma_{u2}^l}\right)^2\right)}{\sum_{l=1}^{M_2} \exp\left(-\frac{1}{2}\left(\frac{x_1(k)-x_{12}^l}{\sigma_{x_12}^l}\right)^2\right)\exp\left(-\frac{1}{2}\left(\frac{x_2(k)-x_{22}^l}{\sigma_{x_22}^l}\right)^2\right)\exp\left(-\frac{1}{2}\left(\frac{u(k)-u_{12}^l}{\sigma_{u2}^l}\right)^2\right)} \tag{30}$$

For $p = 2$, the widths of the states x_1 and x_2 ($\sigma_{x_1p}^l$ and $\sigma_{x_2p}^l$) were fixed to be 0.6 and the width of the input u (σ_{up}^l) is fixed to be 0.6. Then 300 input-output pairs were utilized to train the neuro-fuzzy model by the OLS algorithm to derive the centers of states and input (x_{1p}^l, x_{2p}^l, u_{1p}^l) and the weighting vector (w_{lp}). After trained, 73 rules ($M_2 = 73$) were selected to be the neuro-fuzzy model of equation (30).

To validate the neuro-fuzzy model, the system responses were simulated with the same inputs ($u(k) = -1\ N$ for $k = 0,1,\dots 9$) for the discrete time mathematical model and neuro-fuzzy model. As shown in Figure 2.3, the trained neuro-fuzzy model approximates the mathematical model well.

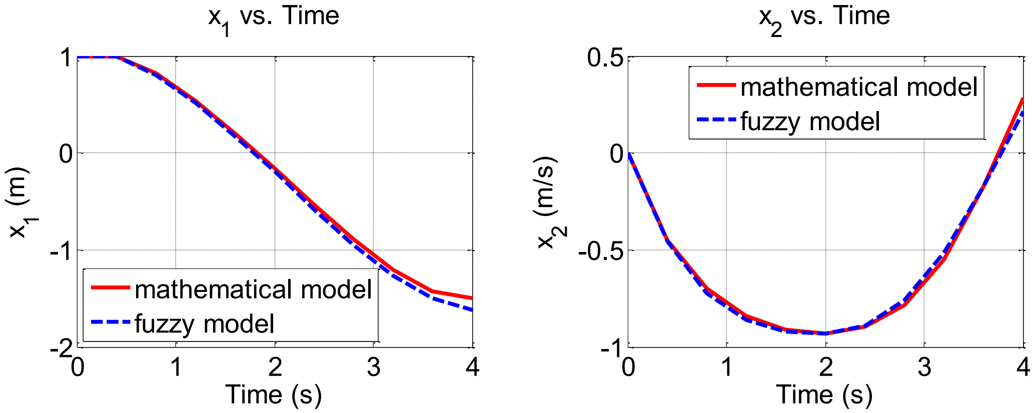

Figure 2.3. The System Responses of the Mathematical Model and Neuro-Fuzzy Model with the Same Control Inputs for the Cart with Nonlinear Spring.

The initial condition is $x_1(0) = 0\ m$ and $x_2(0) = 0\ m/s$, the command input is $r_1 = 1\ m$ and $r_2 = 0\ m/s$ and the performance index is ($N = 10$)::

$$J = \left(x_1(N) - r_1(N)\right)^2 + 0.1\left(x_2(N) - r_2(N)\right)^2$$

$$+ \sum_{k=0}^{N-1}\left(\left(x_1(k) - r_1(k)\right)^2 + 0.1\left(x_2(k) - r_2(k)\right)^2 + 0.2u(k)^2\right) \qquad (31)$$

Using the proposed algorithm, the optimal control input for the system represented by a neuro-fuzzy model was derived. The feasible-direction algorithm [24] [25] was also used to derive the optimal control inputs for the system represented by an explicit mathematical model. Then the two optimal control inputs were implemented with the mathematical model to obtain the state trajectories, as shown in Figure 2.4. From the simulation results, the optimal control results for the nonlinear system represented by an explicit mathematical model and a neuro-fuzzy model are quite similar. The performance index values for the mathematical model and neuro-fuzzy model are 3.2654 and 3.2838 respectively, which indicates the effectiveness of the proposed method.

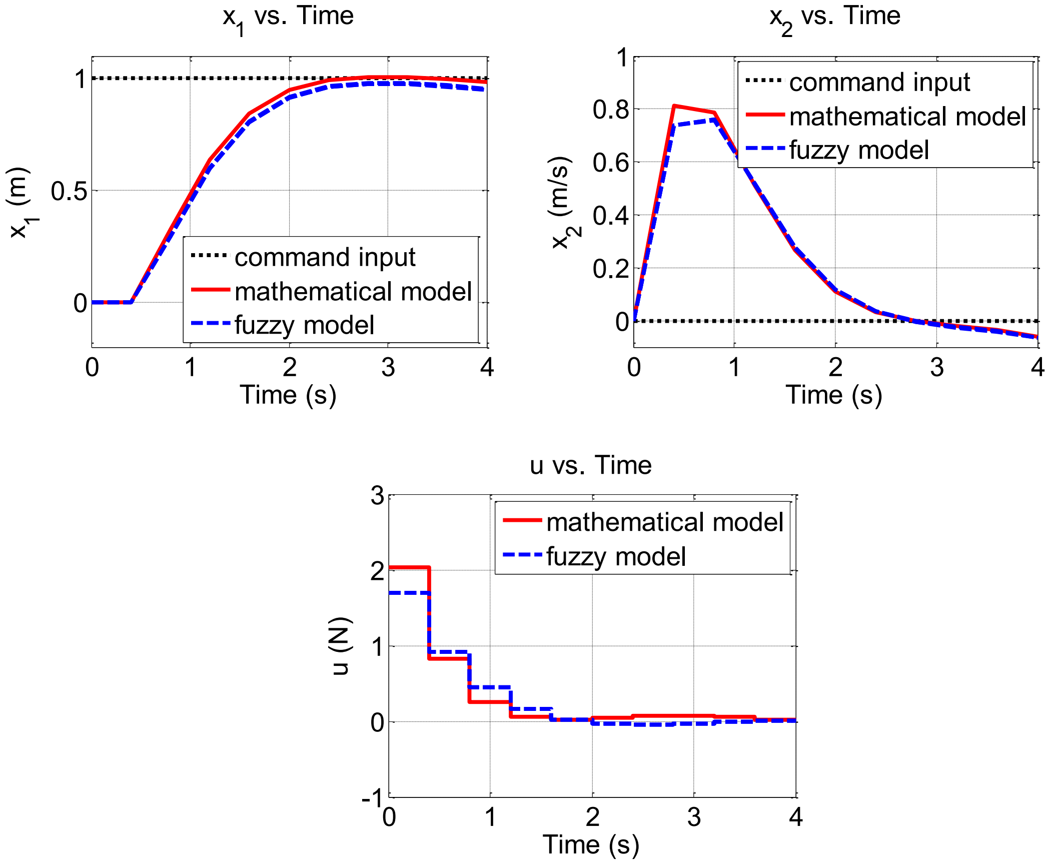

Figure 2.4. Simulation Results of Optimal Control for the Cart with Nonlinear Spring Represented by a Neuro-Fuzzy Model and an Explicit Mathematical Model.

2.4.2 Rigid Asymmetric Spacecraft

Tracking of a rigid asymmetric spacecraft is concerned with a primary attitude control task. Due to inherent nonlinearity of attitude dynamics, tracking in large and rapid maneuvers is a complex undertaking. Therefore, this tracking problem with three independent axis controls is investigated here. The Euler's equations for the angular velocities x_1, x_2, x_3 of the spacecraft are given by [63]

$$\dot{x}_1(t) = -\frac{I_3 - I_2}{I_1} x_2(t) x_3(t) + \frac{u_1(t)}{I_1} \tag{32}$$

$$\dot{x}_2(t) = -\frac{I_1 - I_3}{I_2} x_1(t) x_3(t) + \frac{u_2(t)}{I_2} \tag{33}$$

$$\dot{x}_3(t) = -\frac{I_2 - I_1}{I_3} x_1(t) x_2(t) + \frac{u_3(t)}{I_3} \tag{34}$$

where u_1, u_2, u_3 are the control torques, and $I_1 = 86.24 \text{ kg} \cdot \text{m}^2$, $I_2 = 85.07 \text{ kg} \cdot \text{m}^2$ and $I_3 = 113.59 \text{ kg} \cdot \text{m}^2$ are the spacecraft principal inertias.

An Euler approximation of the system with sampling time $T = 5$ s is given by

$$x_1(k+1) = x_1(k) + 5 \times (-0.3307 x_2(k) x_3(k) + 0.0116 u_1(k)) \tag{35}$$

$$x_2(k+1) = x_2(k) + 5 \times (+0.3215 x_1(k) x_3(k) + 0.0118 u_2(k)) \tag{36}$$

$$x_3(k+1) = x_3(k) + 5 \times (+0.0103 x_1(k) x_2(k) + 0.0088 u_3(k)) \tag{37}$$

To train the neuro-fuzzy model of the system, the widths of the states x_1, x_2 and x_3 were fixed to be 0.05, the widths of the inputs u_1, u_2 and u_3 were fixed to be 0.1. Then 2000 input-output pairs were utilized to train the neuro-fuzzy models by the OLS algorithm to derive the centers of states and inputs, and the weighting factors. For the input-output pairs, the ranges of states x_1, x_2 and x_3 were from -0.1 to 0.1, the ranges of inputs u_1, u_2 and u_3 were from -0.2 to 0.2. After trained, 500 rules were selected to be the neuro-fuzzy model of all the three equations (35), (36) and (37) respectively.

To validate the neuro-fuzzy model, the responses with constant inputs ($u_1(k) = -0.05\ Nm$, $u_2(k) = -0.05\ Nm$, $u_3(k) = -0.05\ Nm$ for $k = 0,1, \dots 14$) were simulated for the mathematical model and the neuro-fuzzy model. As shown in Figure 2.5, the neuro-fuzzy model is close to the mathematical model.

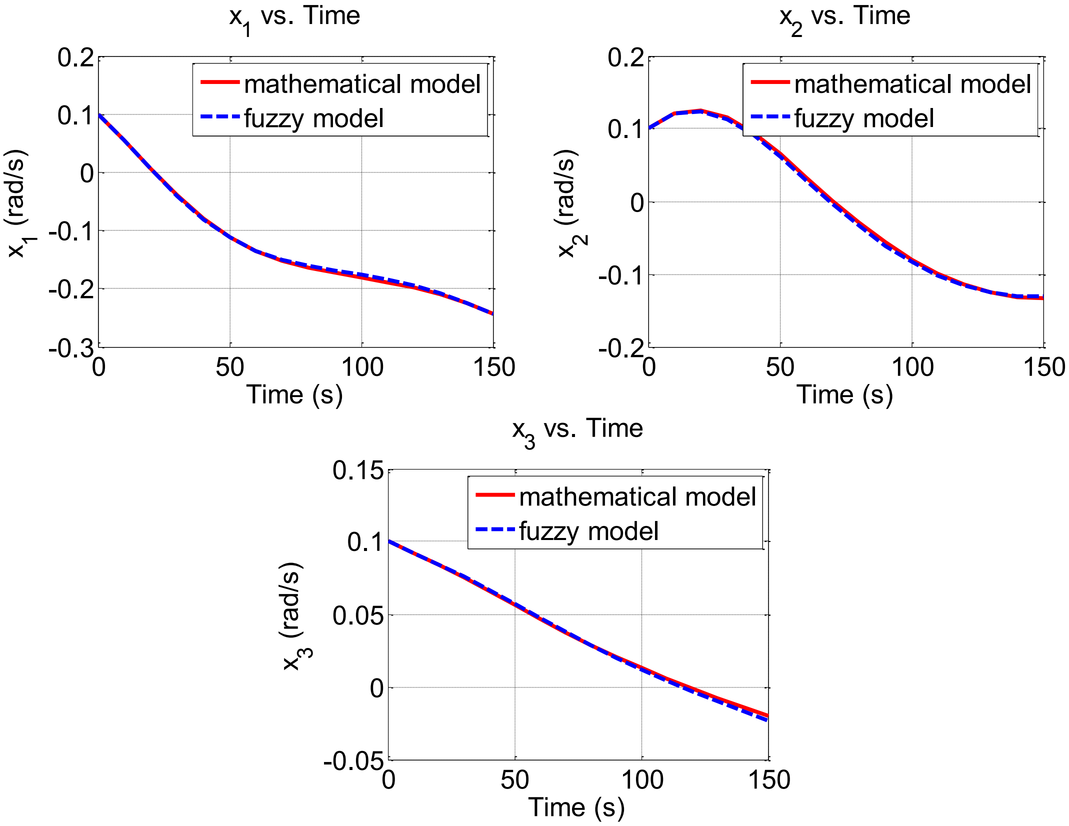

Figure 2.5. The System Responses of the Mathematical Model and Neuro-Fuzzy Model with the Same Control Inputs for Rigid Asymmetric Spacecraft.

The initial conditions are $x_1(0) = 0$ rad/s, $x_2(0) = 0$ rad/s and $x_3(0) = 0$ rad/s, the command inputs are set to $r_1 = 0.04$ rad/s, $r_2 = 0.04$ rad/s and $r_3 = 0.04$ rad/s, and the performance index is defined by $(N = 40)$:

$$J = \left(x_1(N) - r_1(N)\right)^2 + \left(x_2(N) - r_2(N)\right)^2 + \left(x_3(N) - r_3(N)\right)^2 + \sum_{k=0}^{N-1}\left(\left(x_1(k) - r_1(k)\right)^2 + \left(x_2(k) - r_2(k)\right)^2 + \left(x_3(k) - r_3(k)\right)^2 + 0.01u_1^2(k) + 0.01u_2^2(k) + 0.01u_3^2(k)\right) \quad (38)$$

Similar as in section 2.4.1, the optimal control inputs for the systems represented by a neuro-fuzzy model and an explicit mathematical model were derived and then the state trajectories were obtained, as shown in Figure. 2.6. From the simulation results, the optimal control results for the nonlinear systems represented by the two models are very close and the performance index values for the neuro-fuzzy model and mathematical model are both 0.0149. Therefore, the performance of the proposed method is very good.

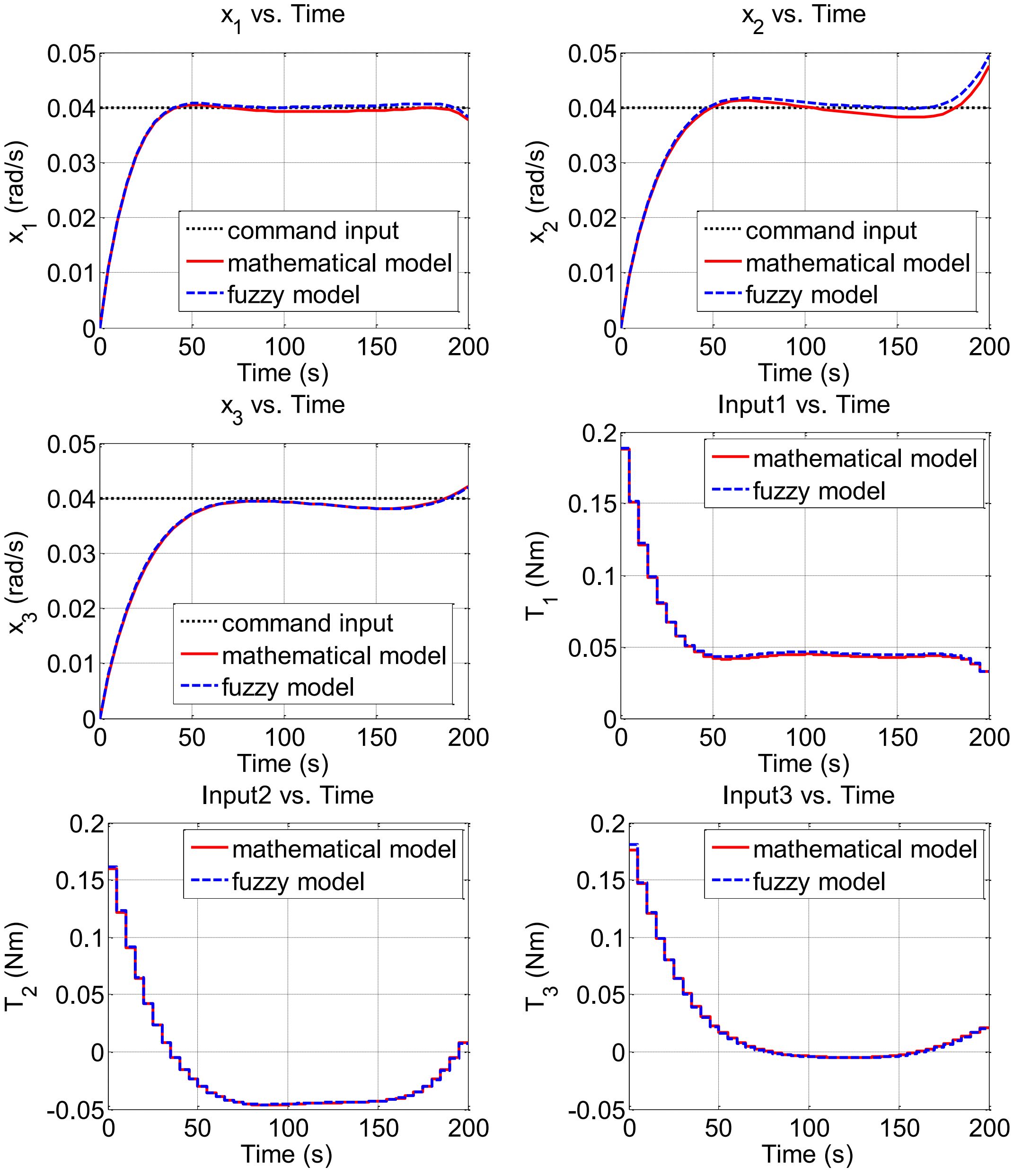

Figure 2.6. Simulation Results of Optimal Control for the Rigid Asymmetric Spacecraft Represented by a Neuro-Fuzzy Model and an Explicit Mathematical Model.

2.4.3 Continuous Stirred Tank Reactor

Consider a continuously stirred tank reactor shown in Figure 2.4.3.1. The mass and heat balance for a single reaction $A \rightleftharpoons B$ are [64]

$$\frac{dc}{dt} = \frac{1}{\theta}\left(c_f - c\right) - \hat{r} \tag{39}$$

$$\frac{dT}{dt} = \frac{1}{\theta}\left(T_f - T\right) + J\hat{r} - \alpha u(T - T_c) \tag{40}$$

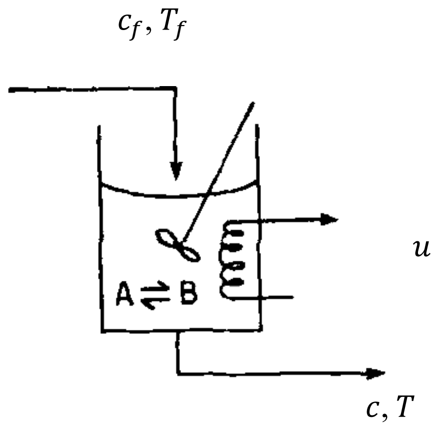

Figure 2.7. Continuously Stirred Tank Reactor. [64]

where concentration of reaction c and temperature T are two states of the system, u is the coolant flowrate control input, $\theta = 10$ min is the total start-up time of interest, $c_f = 1\ \text{mol/L}$ is the feed concentration of reaction, $T_f = 300\ K$ is the feed temperature, $J = 100$ is a chemical constant, $\hat{r}$ is the reaction rate, $\alpha = 1.95 \times 10^{-4}\ m^2$ is the dimensionless heat transfer area and $T_c = 290\ K$ is the room temperature.

If we define dimensionless quantities $y_1 = \frac{c}{c_f}$, $y_2 = \frac{T}{Jc_f}$, $y_c = \frac{T_c}{Jc_f}$, $y_f = \frac{T_f}{Jc_f}$, $r = \frac{\hat{r}}{c_f}$, one gets

$$\frac{dy_1}{dt} = \frac{1}{\theta}(1 - y_1) - r \tag{41}$$

$$\frac{dy_2}{dt} = \frac{1}{\theta}\left(y_f - y_2\right) + r - \alpha u(y_2 - y_c) \tag{42}$$

where y_1, y_2, y_c and y_f are dimensionless concentration, temperature, coolant temperature, feed temperature respectively, and the dimensionless irreversible reaction rate r is given by

$$r = k_{10} y_1 e^{-\frac{N}{y_2}} \tag{43}$$

where $k_{10} = 300$ is a pre-exponential factor of forward constant and $N = 25.2$ is a gas constant.

An Euler approximation of the system with sampling time $T = 0.4$ min is given by

$$y_1(k+1) = y_1(k) + T * \left(\frac{1}{\theta}\left(1 - y_1(k)\right) - r(k)\right) \tag{44}$$

$$y_2(k+1) = y_2(k) + T * \left(\frac{1}{\theta}\left(y_f - y_2(k)\right) + r(k) - \alpha u(k)(y_2(k) - y_c)\right) \tag{45}$$

To train the neuro-fuzzy model of the system, the widths of the state y_1 and y_2 were fixed to be 0.25 and the width of the input u was fixed to be 250. Then 500 input-output pairs were utilized to train the neuro-fuzzy models by the OLS algorithm to derive the centers of state and input, and the weighting factors. For the input-output pairs, the range of state y_1 was from 0 to 1, the range of state y_2 was from 3 to 4 and the range of input u was from 0 to 1000. After trained, 137 rules were selected to be the neuro-fuzzy models of the two equations (44) and (45) respectively. The system responses with the constant input ($u(k) = 300 \text{ L/min}$ for $k = 0,1, \dots 49$) was simulated for the mathematical model and the neuro-fuzzy model. As shown in Figure 2.8, the responses of the two models are completely same.

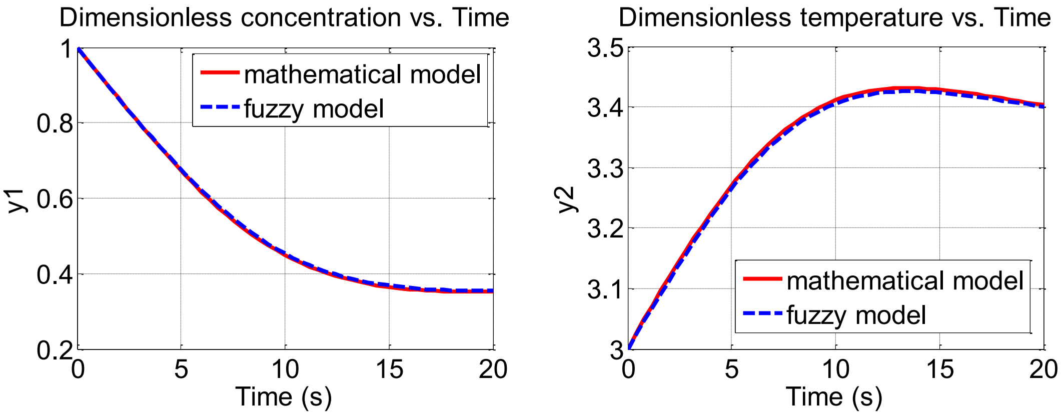

Figure 2.8. The System Responses of the Mathematical Model and Neuro-Fuzzy Model with the Same Control Inputs for Continuously Stirred Tank Reactor.

The initial conditions for the dimensionless concentration and temperature are $y_1(0) = 1$ ($c(0) = y_1(0) = 1$ mol/L) and $y_2(0) = 3$ ($T(0) = 100, y_2(0) = 300\ K$) respectively. The objective of optimal control problem is to find the coolant flowrate control $u(k) \geq 0$ such that minimize the functional

$$J = \alpha_1\big(y_1(N) - r_1(N)\big)^2 + \alpha_2\big(y_2(N) - r_2(N)\big)^2 +$$

$$\textstyle\sum_{k=0}^{N-1}\Big(\alpha_1\big(y_1(k) - r_1(k)\big)^2 + \alpha_2\big(y_2(k) - r_2(k)\big)^2 + \alpha_3\big(u(k) - u_s(k)\big)^2\Big) \quad (46)$$

where $N = 50$, $r_1 = 0.408126$, $r_2 = 3.29763$, $u_s = 370\ K$, $\alpha_1 = 100000$, $\alpha_2 = 2000$ and $\alpha_3 = 0.001$. So the desired values for concentration and temperature are $c_d = r_1 = 0.408126$ and $T_d = 100r_2 = 329.763\ K$ respectively.

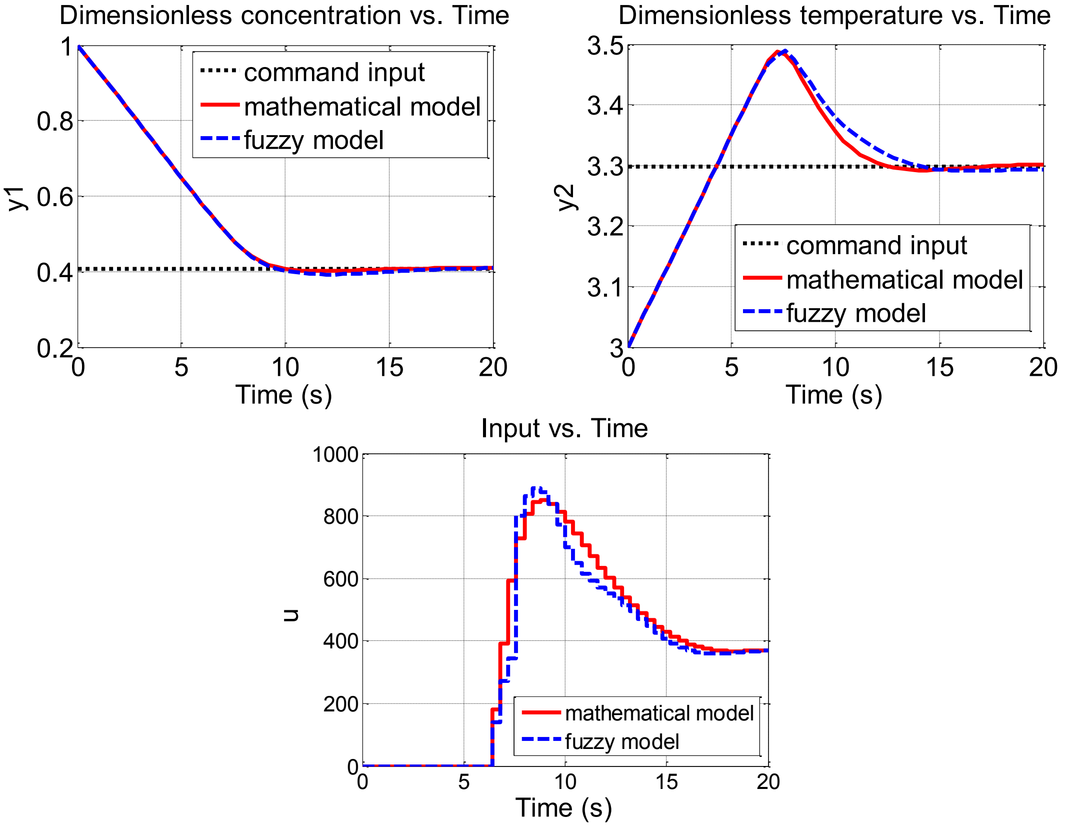

Figure 2.9. Simulation Results of Optimal Control for the Continuously Stirred Tank Reactor Represented by a Neuro-Fuzzy Model and an Explicit Mathematical Model.

Similar to the previous examples, the optimal control inputs for the systems represented by a neuro-fuzzy model and an explicit mathematical model were derived and then the state trajectories were obtained, as shown in Figure 2.9. From the simulation results, optimal control results for the nonlinear systems represented by the two models are quite close. Besides, the performance index values for the neuro-fuzzy model and mathematical model are 2.6748×10^6 and 2.6733×10^6 respectively, which are also very similar. Therefore, the performance of the proposed method is very good However, due to the limitation of the feasible-direction numerical algorithm, it cannot be applied to a too complex system or a control process with too many time steps. Therefore, a better way which also utilized the TS fuzzy model is then developed in the next Chapter.

CHAPTER 3. OPTIMAL CONTROL WITH TS FUZZY MODELS

3.1 Takagi-Sugeno Fuzzy Models

TS models can be used to represent complex MIMO systems with both fuzzy inference rules and local analytic linear dynamic models as follows: R^l: IF x_1 is $F_1^l \cdots x_n$ is F_n^l and u_1 is $F_{n+1}^l \cdots u_m$ is F_{n+m}^l, THEN $\dot{x} = A_l x(t) + B_l u(t) + d_l$, where $l \in L = \{1,2,\cdots,p\}$, R^l denotes the lth fuzzy inference rule, p is the number of inference rules, F_j^l $(j = 1,2,\cdots,n+m)$ are the fuzzy sets, $x(t) \in R^n$ is the state vector, $u(t) \in R^m$ is the input vector and (A_l, B_l, d_l) are the matrices of the lth local model [29].

By using a singleton fuzzifier, product fuzzy inference, and center-average defuzzifier, the TS fuzzy model can be rewritten as

$$\dot{x} = A(\mu)x(t) + B(\mu)u(t) + d(\mu) \tag{47}$$

where $A(\mu) = \sum_{l=1}^{p} \mu_l A_l$, $B(\mu) = \sum_{l=1}^{p} \mu_l B_l$, $d(\mu) = \sum_{l=1}^{p} \mu_l d_l$ and μ_l is the normalized membership function [29]. It is a nonlinear model in nature since the membership functions are nonlinear functions of the premise variables that contain some or all of the state variables and input variables.

3.2 Identification of Takagi-Sugeno Fuzzy Models

Because the TS fuzzy models are used to design the local linear controllers, they should achieve a good approximation of both local and global dynamics of the underlying system, or it can hardly achieve satisfied control performance. However, considering them together would become a very difficult problem because it is not straightforward to identify constituent local models of TS fuzzy models from the input-output data and the tradeoff between the local and global accuracy of TS fuzzy model should be addressed [43].

In this thesis, a novel approach that utilizes the neuro-fuzzy model to obtain a TS fuzzy model whose local models are close approximations to the local linearization of the nonlinear systems is presented. The neuro-fuzzy models can approximate any nonlinear function to arbitrary accuracy and there have been many matured training methods proposed. First, from input-output data, a MIMO neuro-fuzzy model with product inference engine, singleton fuzzifier, center average defuzzifier, and Gaussian membership functions are obtained by the orthogonal least square or least square training algorithm to approximate the global nonlinear system [43]:

$$\dot{x}_p = \frac{\sum_{l=1}^{M_p} w_{lp} \prod_{i=1}^{n} \exp\left(-\frac{1}{2}\left(\frac{x_i - x_{ip}^l}{\sigma_{x_ip}^l}\right)^2\right) \prod_{j=1}^{m} \exp\left(-\frac{1}{2}\left(\frac{u_j - u_{jp}^l}{\sigma_{u_jp}^l}\right)^2\right)}{\sum_{l=1}^{M_p} \prod_{i=1}^{n} \exp\left(-\frac{1}{2}\left(\frac{x_i - x_{ip}^l}{\sigma_{x_ip}^l}\right)^2\right) \prod_{j=1}^{m} \exp\left(-\frac{1}{2}\left(\frac{u_j - u_{jp}^l}{\sigma_{u_jp}^l}\right)^2\right)} \tag{48}$$

where $\boldsymbol{x} = (x_1, x_2 \ldots x_n)$ is the state vector, $\boldsymbol{u} = (u_1, u_2 \ldots u_m)$ is the input vector, w_{lp} is the weighting factor, x_{ip}^l and $\sigma_{x_ip}^l$ are centers and widths of the state x_p, and u_{jp}^l and $\sigma_{u_jp}^l$ are centers and widths of the input u_p and $p = 1,2 \ldots n$.

Next the TS fuzzy model is derived from the linearization of the neuro-fuzzy model. Therefore, this kind of TS fuzzy model can be interpreted as the local linearization of the nonlinear dynamic system, which is very important for the design of local linear controllers. By choosing enough operating points so that the TS fuzzy model is a good approximation of the neuro-fuzzy model, one can also obtain a good approximation of the nonlinear dynamic systems.

Linearization about one operating point $(\mathrm{x_k},\ \mathrm{u_k})$ results in

$$\dot{\boldsymbol{x}} = \mathrm{A}_k(\boldsymbol{x} - \boldsymbol{x}_k) + \mathrm{B}_k(\boldsymbol{u} - \boldsymbol{u}_k) + f_k(\boldsymbol{x}_k, \boldsymbol{u}_k) + HOT \tag{49}$$

where $\mathrm{A}_k = \left.\begin{pmatrix} \frac{\partial f_1(\boldsymbol{x},\boldsymbol{u})}{\partial x_1} & \cdots & \frac{\partial f_1(\boldsymbol{x},\boldsymbol{u})}{\partial x_n} \\ \vdots & \ddots & \vdots \\ \frac{\partial f_n(\boldsymbol{x},\boldsymbol{u})}{\partial x_1} & \cdots & \frac{\partial f_n(\boldsymbol{x},\boldsymbol{u})}{\partial x_n} \end{pmatrix}\right|_{\boldsymbol{x}=\boldsymbol{x}_k,\boldsymbol{u}=\boldsymbol{u}_k}$

$$\frac{\partial f_p(\boldsymbol{x},\boldsymbol{u})}{\partial x_q} = \sum_{l_a=1}^{M_p} \sum_{l_b=1}^{M_p} a_p^{l_a} a_p^{l_b} q_p^{l_a} \left(w_{l_a p} - w_{l_b p}\right) \tag{50}$$

where

$$a_p^{l_a} = \prod_{i=1}^{n} \exp\left(-\frac{1}{2}\left(\frac{x_i - x_{ip}^{l_a}}{\sigma_{x_i p}^{l_a}}\right)^2\right) \prod_{j=1}^{m} \exp\left(-\frac{1}{2}\left(\frac{u_j - u_{jp}^{l_a}}{\sigma_{u_j p}^{l_a}}\right)^2\right) \tag{51}$$

$$a_p^{l_b} = \prod_{i=1}^{n} \exp\left(-\frac{1}{2}\left(\frac{x_i - x_{ip}^{l_b}}{\sigma_{x_i p}^{l_b}}\right)^2\right) \prod_{j=1}^{m} \exp\left(-\frac{1}{2}\left(\frac{u_j - u_{jp}^{l_b}}{\sigma_{u_j p}^{l_b}}\right)^2\right) \tag{52}$$

$$q_p^{l_a} = -\frac{x_q - x_{qp}^{l_a}}{\left(\sigma_{x_q p}^{l_a}\right)^2} \tag{53}$$

and $\mathrm{B}_k = \left.\begin{pmatrix} \frac{\partial f_1(\boldsymbol{x},\boldsymbol{u})}{\partial u_1} & \cdots & \frac{\partial f_1(\boldsymbol{x},\boldsymbol{u})}{\partial u_m} \\ \vdots & \ddots & \vdots \\ \frac{\partial f_n(\boldsymbol{x},\boldsymbol{u})}{\partial u_1} & \cdots & \frac{\partial f_n(\boldsymbol{x},\boldsymbol{u})}{\partial u_m} \end{pmatrix}\right|_{\boldsymbol{x}=\boldsymbol{x}_k,\boldsymbol{u}=\boldsymbol{u}_k}$

$$\frac{\partial f_p[\boldsymbol{x}(k),\boldsymbol{u}(k)]}{\partial u_q(k)} = \sum_{l_a=1}^{M_p} \sum_{l_b=1}^{M_p} a_p^{l_a} a_p^{l_b} q_p^{l_a} \left(w_{l_a p} - w_{l_b p}\right) \tag{54}$$

where

$$a_p^{l_a} = \prod_{i=1}^{n} \exp\left(-\frac{1}{2}\left(\frac{x_i(k)-x_{ip}^{l_a}}{\sigma_{x_i p}^{l_a}}\right)^2\right) \prod_{j=1}^{m} \exp\left(-\frac{1}{2}\left(\frac{u_j(k)-u_{jp}^{l_a}}{\sigma_{u_j p}^{l_a}}\right)^2\right) \tag{55}$$

$$a_p^{l_b} = \prod_{i=1}^{n} \exp\left(-\frac{1}{2}\left(\frac{x_i(k)-x_{ip}^{l_b}}{\sigma_{x_i p}^{l_b}}\right)^2\right) \prod_{j=1}^{m} \exp\left(-\frac{1}{2}\left(\frac{u_j(k)-u_{jp}^{l_b}}{\sigma_{u_j p}^{l_b}}\right)^2\right) \tag{56}$$

$$q_p^{l_a} = -\frac{u_q(k)-u_{qp}^{l_a}}{\left(\sigma_{u_q p}^{l_a}\right)^2} \tag{57}$$

For simplicity, equation (49) can be rewritten as

$$\dot{\boldsymbol{x}} = \mathrm{A}_k \boldsymbol{x} + \mathrm{B}_k \boldsymbol{u} + \boldsymbol{d}_k \tag{58}$$

where

$$\boldsymbol{d}_k = f_k(\boldsymbol{x}_k, \boldsymbol{u}_k) - \mathrm{A}_k \boldsymbol{x}_k - \mathrm{B}_k \boldsymbol{u}_k \tag{59}$$

However, since the neuro-fuzzy model is trained by input-output data, the relationships of all states and inputs are not known in advance. So some unimportant states that do not influence all the other states will be modeled in the TS fuzzy model, which result in some uncontrollable and unobservable states in the system and should be eliminated.

Suppose the original system has n controllable and observable states, if we add one state that does not influence all the other states and inputs, we will get a $n+1$ states system. Because the state does not influence all the other states, so for $p = 1,2 \ldots n$, we can get

$$\dot{x}_p = f_p([x_1, x_2 \ldots x_n, x_{n+1}], \boldsymbol{u}) = \dot{x}_p' = f_p([x_1, x_2 \ldots x_n, x_{n+1} + \Delta x_{n+1}], \boldsymbol{u}) \tag{60}$$

where $\dot{x}_p$ and $\dot{x}_p'$ have the same $x_1, x_2, \dots, x_n$ but different x_{n+1}.

Therefore

$$\frac{\partial f_p(\boldsymbol{x},\boldsymbol{u})}{\partial x_{n+1}} = \frac{f_p([x_1,x_2 \dots x_n,x_{n+1}+\Delta x_{n+1}],\boldsymbol{u}) - f_p([x_1,x_2 \dots x_n,x_{n+1}],\boldsymbol{u})}{x_{n+1}+\Delta x_{n+1}-x_{n+1}} = \frac{0}{\Delta x_{n+1}} = 0 \tag{61}$$

Besides, $\frac{\partial f_{n+1}(\boldsymbol{x},\boldsymbol{u})}{\partial x_{n+1}}$ can be any value and $\frac{\partial f_{n+1}(\boldsymbol{x},\boldsymbol{u})}{\partial \boldsymbol{u}} = \boldsymbol{0}$, because the input vector $\boldsymbol{u}$ should not influence the state. So after the linearization of the system, if some states satisfy $\frac{\partial f_p(\boldsymbol{x},\boldsymbol{u})}{\partial x_q} = 0$ for all $p \neq q$ and $\frac{\partial f_q(\boldsymbol{x},\boldsymbol{u})}{\partial \boldsymbol{u}} = \boldsymbol{0}$, then we can safely eliminate them to get a completely controllable and observable system.

3.3 Optimal Control Based on TS Fuzzy Models

In order to design a global optimal controller based on the TS fuzzy model of the original nonlinear system, the parallel distributed compensation is used to derive each control rule so as to compensate each local linear model of the fuzzy system and construct a global fuzzy controller by the aggregation of the local optimal controllers with a fuzzy inference system [65].

Using the same premise as in the ith rule of the TS fuzzy model, the local controller can be obtained as follows [29]:

$$R^l\text{: IF } x_1 \text{ is } F_1^l \text{ and } \cdots x_n \text{ is } F_n^l\text{, THEN } u = u_l$$

The fuzzy controller is analytically represented by

$$u = \sum_{l=1}^{p} \mu_l u_l \tag{62}$$

where μ_l is the normalized membership function same as in (47) and u_l should be derived by the following procedure.

Assume the cost function is:

$$J_a = \frac{1}{2}\left(\boldsymbol{r}(t_f) - \boldsymbol{x}(t_f)\right)^T H\left(\boldsymbol{r}(t_f) - \boldsymbol{x}(t_f)\right) + \int_{t_0}^{t_f}((\boldsymbol{r} - \boldsymbol{x})^T Q(\boldsymbol{r} - \boldsymbol{x}) + \boldsymbol{u}^T R\boldsymbol{u})\, dt \tag{63}$$

where $\boldsymbol{r}$ is the command input, $\boldsymbol{x}$ is the state vector, $\boldsymbol{u}$ is the input vector, t_0 is the initial time, t_f is the final time, H and Q are symmetric positive semi-definite matrices and R is a symmetric positive definite matrix.

For each local optimal controller, the control action u_l can be derived by the Riccati differential equation [66]:

$$u_l = -K(\boldsymbol{r} - \boldsymbol{x}) + (\mathrm{B}_k^{\mathrm{T}}\mathrm{B}_k)^{-1}\mathrm{B}_k^{\mathrm{T}}\boldsymbol{d}_k \tag{64}$$

where K is given by

$$K = R^{-1}\mathrm{B}_k^{\mathrm{T}}P(t) \tag{65}$$

and $P(t)$ is found by solving the continuous time Riccati differential equation

$$\mathrm{A}_k^{\mathrm{T}}P(t) + P(t)\mathrm{A}_k - P(t)\mathrm{B}_k R^{-1}\mathrm{B}_k^{\mathrm{T}}P(t) + Q = -\dot{P}(t) \tag{66}$$

3.4 Simulation Examples

3.4.1 Two Link Flexible Joint Robot

Flexible robot manipulators exhibit many advantages over rigid ones: they require less material, have higher manipulation speed, lower power consumption and are safer to operate due to reduced inertia. However, control of flexible robot manipulators to maintain accurate positioning is an extremely challenging problem. Due to the flexible nature and distributed characteristics of the system, the dynamics are highly nonlinear and complex. Problems arise due to precise positioning requirement, vibration due to

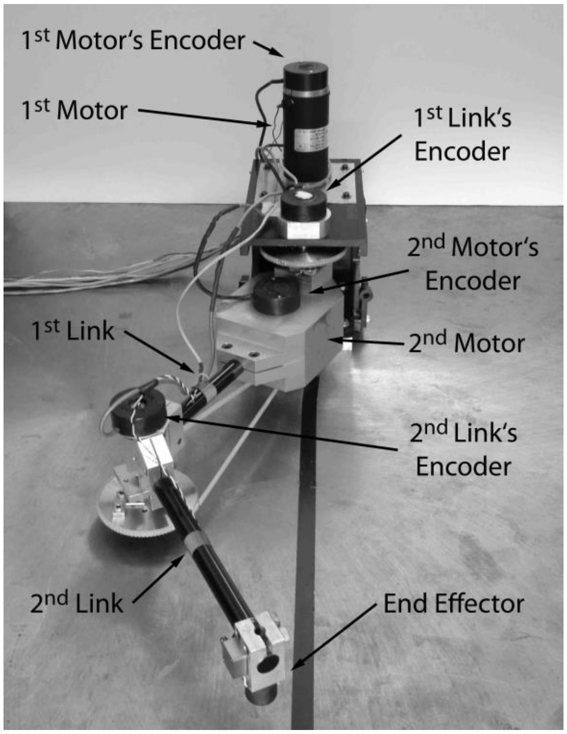

Figure 3.1. The Two-Link Flexible-Joint Robot.

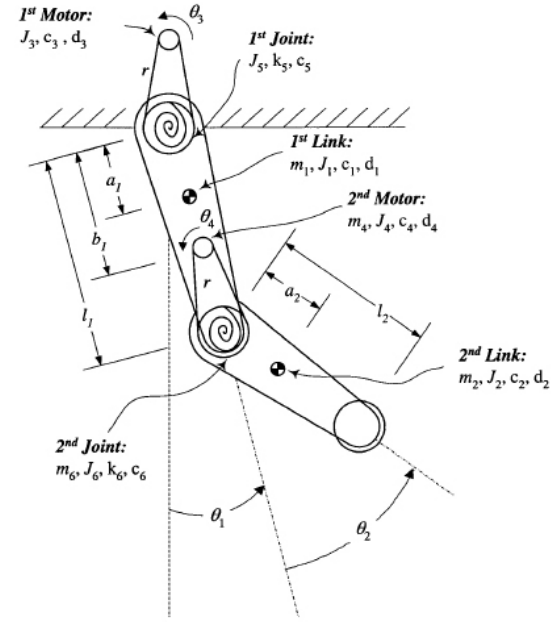

Figure 3.2. Schematic of the Robot with Physical Parameters.

system flexibility, the difficulty in obtaining accurate model of the system and non-minimum phase characteristics of the system [67]. The two-link flexible-joint robot whose model is used for simulation is schematically shown with all physical parameters in Figure 3.2.

Flexible robot manipulators exhibit many advantages over rigid ones: they require less material, have higher manipulation speed, lower power consumption and are safer to operate due to reduced inertia. However, control of flexible robot manipulators to maintain accurate positioning is an extremely challenging problem. Due to the flexible nature and distributed characteristics of the system, the dynamics are highly nonlinear and complex. Problems arise due to precise positioning requirement, vibration due to system flexibility, the difficulty in obtaining accurate model of the system and non-minimum phase characteristics of the system [67]. The two-link flexible-joint robot whose model is used for simulation is schematically shown with all physical parameters in Figure 3.2.

The reduced model is given by [68]

$$\boldsymbol{M}(\boldsymbol{\theta})\ddot{\boldsymbol{\theta}} + \boldsymbol{V}(\boldsymbol{\theta}, \dot{\boldsymbol{\theta}}) + \boldsymbol{C}\dot{\boldsymbol{\theta}} + \boldsymbol{K}\boldsymbol{\theta} = \boldsymbol{T} \tag{67}$$

where $\boldsymbol{\theta} = [\theta_1 \quad \theta_2 \quad \theta_3 \quad \theta_4]^{\boldsymbol{T}}$ is the vector of the link and motor positions. The positions of the first and second link are denoted by θ_1 and θ_2, respectively, whereas the positions of the first and second motor are denoted by θ_3 and θ_4. $\boldsymbol{M}(\boldsymbol{\theta})$ is the inertia matrix, $\boldsymbol{V}(\boldsymbol{\theta}, \dot{\boldsymbol{\theta}})$ is the vector of Coriolis and centrifugal functions, $\boldsymbol{C}$ is the viscous damping matrix, $\boldsymbol{K}$ is the matrix of the stiffness coefficients, and $\boldsymbol{T}$ is the vector of the driving torques.

The inertia matrix $\boldsymbol{M}(\boldsymbol{\theta})$ is given by

$$\boldsymbol{M(\theta)} = \begin{bmatrix} \boldsymbol{M_1}(\theta_2) & 0 \\ 0 & \boldsymbol{M_3} \end{bmatrix} \tag{68}$$

where

$$\boldsymbol{M_1}(\theta_2) = \begin{bmatrix} m_{11} & m_{12} \\ m_{21} & m_{22} \end{bmatrix} \tag{69}$$

and

$$\boldsymbol{M_3} = \begin{bmatrix} m_{33} & 0 \\ 0 & m_{44} \end{bmatrix} \tag{70}$$

The first matrix element is given by

$$m_{11} = p_1 + p_2 + 2p_3\cos(\theta_2) \tag{71}$$

with

$$p_1 = m_1a_1^2 + m_2l_1^2 + m_4b_1^2 + m_6l_1^2 + J_1 + J_4 + J_6 \tag{72}$$

$$p_2 = m_2a_2^2 + J_2 \tag{73}$$

And

$$p_3 = l_1m_2a_2 \tag{74}$$

Moreover,

$$m_{12} = m_{21} = p_2 + p_3\cos(\theta_2) \tag{75}$$

$$m_{22} = p_2 \tag{76}$$

$$m_{33} = J_3 + \frac{J_5}{r^2} \tag{77}$$

and

$$m_{44} = J_4 + \frac{J_6}{r^2} \tag{78}$$

m_i is the lumped mass, J_i is the moment of inertia of the component, l_i is the length of the link, and a_1 and a_2 denote the distance between the center of gravity of the first and second link and the first and the second joint, respectively. Furthermore, b_1 is the distance between the second motor and the first joint, and r is the gear ratio of the chain drives. The vector of the Coriolis and centrifugal functions is

$$\boldsymbol{V}(\boldsymbol{\theta},\dot{\boldsymbol{\theta}}) = \begin{bmatrix} \boldsymbol{V_L} \\ \boldsymbol{0} \end{bmatrix} = \begin{bmatrix} -p_3\left(2\dot{\theta}_1\dot{\theta}_2 + \dot{\theta}_2^2\right)\sin(\theta_2) \\ p_3\dot{\theta}_1^2\sin(\theta_2) \\ 0 \\ 0 \end{bmatrix} \tag{79}$$

and the viscous damping matrix can be written as

$$\boldsymbol{C} = \begin{bmatrix} \boldsymbol{C_L} & \boldsymbol{0} \\ \boldsymbol{0} & \boldsymbol{C_M} \end{bmatrix} \tag{80}$$

where the diagonal elements are the link damping matrix

$$\boldsymbol{C_L} = diag\{c_1, c_2\} \tag{81}$$

and the motor damping matrix

$$\boldsymbol{C_M} = diag\{c_3, c_4\} \tag{82}$$

with the viscous friction coefficients c_i. The matrix of stiffness coefficients is given by

$$\boldsymbol{K} = \begin{bmatrix} k_5 & 0 & -\frac{k_5}{r} & 0 \\ 0 & k_6 & 0 & -\frac{k_6}{r} \\ -\frac{k_5}{r} & 0 & \frac{k_5}{r^2} & 0 \\ 0 & -\frac{k_6}{r} & 0 & \frac{k_6}{r^2} \end{bmatrix} \tag{83}$$

where k_i denote the coefficients of the torsional springs, and the torque vector is

$$\boldsymbol{T} = \begin{bmatrix} \boldsymbol{0} \\ \boldsymbol{T_M} \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ T_1 \\ T_2 \end{bmatrix} \tag{84}$$

where T_1 and T_2 denoting the driving torque of the first and second motor, respectively.

With $\boldsymbol{\theta_L} = [\theta_1 \quad \theta_2]^{\boldsymbol{T}}$ and $\boldsymbol{\theta_L} = [\theta_3 \quad \theta_4]^T$, the equations of motion for the links and motors can be written separately as

Table 3.1. Estimated Values of the Robot’s Parameters. [68]

Parameter	Value	Parameter	Value
p_1	$0.1402\frac{\text{kgm}^2}{\text{rad}}$	c_4	$1.4975\times 10^{-3}\,\frac{\text{Nms}}{\text{rad}}$
$m_2a_2^2 + J_2$	$0.01962\,\frac{\text{kgm}^2}{\text{rad}}$	c_5	$0.005\frac{\text{Nms}}{\text{rad}}$
$l_1m_2a_2$	$0.02338\,\frac{\text{kgm}^2}{\text{rad}}$	c_6	$8.128\times 10^{-3}\,\frac{\text{Nms}}{\text{rad}}$
J_3	$4.1571\times 10^{-5}\,\frac{\text{kgm}^2}{\text{rad}}$	k_5	$2.848\frac{\text{Nm}}{\text{rad}}$
J_4	$7.5429\times 10^{-4}\,\frac{\text{kgm}^2}{\text{rad}}$	k_6	$2.848\frac{\text{Nm}}{\text{rad}}$
J_5	$0.025\frac{\text{kgm}^2}{\text{rad}}$	d_1	0.01987 Nm
J_6	$0.025\frac{\text{kgm}^2}{\text{rad}}$	d_2	0.0323 Nm
c_1	$0.04\frac{\text{Nms}}{\text{rad}}$	d_3	0.0053 Nm
c_2	$0.02143\frac{\text{Nms}}{\text{rad}}$	d_4	0.0271 Nm
c_3	$1.8937\times 10^{-4}\,\frac{\text{Nms}}{\text{rad}}$	r	5

$$\boldsymbol{M_1}(\theta_2)\ddot{\boldsymbol{\theta}}_{\boldsymbol{L}} + \boldsymbol{V_L}\left(\boldsymbol{\theta_L}, \dot{\boldsymbol{\theta}}_{\boldsymbol{L}}\right) + \boldsymbol{C_L}\dot{\boldsymbol{\theta}}_{\boldsymbol{L}} + \boldsymbol{K_S}\left(\boldsymbol{\theta_L} - \frac{\boldsymbol{\theta_M}}{r}\right) = \boldsymbol{T} \tag{85}$$

for the links and

$$\boldsymbol{M_3}\ddot{\boldsymbol{\theta}}_{\boldsymbol{M}} + \boldsymbol{C_M}\dot{\boldsymbol{\theta}}_{\boldsymbol{M}} + \boldsymbol{K_S}\left(\frac{\boldsymbol{\theta_M}}{r^2} - \frac{\boldsymbol{\theta_L}}{r}\right) = \boldsymbol{T_M} \tag{86}$$

for the motors. $\boldsymbol{K_S}$ is a diagonal matrix and contains the spring coefficients,

$$\boldsymbol{K_S} = diag\{k_5, k_6\} \tag{87}$$

It can be clearly seen that the links and motors are only coupled by the torsional springs in the joints.

With

$$\boldsymbol{x} = \begin{bmatrix} \boldsymbol{\theta_L} \\ \boldsymbol{\theta_M} \\ \dot{\boldsymbol{\theta}}_{\boldsymbol{L}} \\ \dot{\boldsymbol{\theta}}_{\boldsymbol{M}} \end{bmatrix} = \begin{bmatrix} \boldsymbol{x_1} \\ \boldsymbol{x_2} \\ \boldsymbol{x_3} \\ \boldsymbol{x_4} \end{bmatrix} \tag{88}$$

as state variables, equations (85) and (86) can be rewritten in the fourth order state space form [68]

$$\dot{\boldsymbol{x}}_{\boldsymbol{1}} = \boldsymbol{x_3} \tag{89}$$

$$\dot{\boldsymbol{x}}_{\boldsymbol{2}} = \boldsymbol{x_4} \tag{90}$$

$$\dot{\boldsymbol{x}}_{\boldsymbol{3}} = -\boldsymbol{M}_{\boldsymbol{1}}^{-\boldsymbol{1}}[\boldsymbol{V_L} + \boldsymbol{C_L}\boldsymbol{x_3} + \boldsymbol{D_L} + \boldsymbol{K_s}(\boldsymbol{x_1} - \boldsymbol{x_2}/r)] \tag{91}$$

$$\dot{\boldsymbol{x}}_{\boldsymbol{4}} = \boldsymbol{M}_{\boldsymbol{3}}^{-\boldsymbol{1}}[\boldsymbol{T_M} - \boldsymbol{C_M}\boldsymbol{x_4} - \boldsymbol{D_M} - \boldsymbol{K_s}(\boldsymbol{x_2}/r^2 - \boldsymbol{x_1}/r)] \tag{92}$$

The model in combination with estimated values of the robot's physical parameters is used for simulating the robot. Table 3.1 lists the used values.

To obtain the neuro-fuzzy model of the robot, the motors were excited by sine sweep torques. A combination of a 1Hz sine wave and a subsequent sine sweep signal with an initial frequency of 1Hz and a final frequency of 5Hz was used to excite the shoulder motor and the elbow motor, as shown in Figure 3.3.

To reduce noise in the velocity and acceleration signals that mainly originate from the quantization of the position signal, a low-pass Butterworth filter was applied to the position signal. After filtering, velocities and accelerations were obtained from the signal through the central finite difference method. To avoid transient effects from filtering in both directions with initializing the filter states, experiments were started with a rest

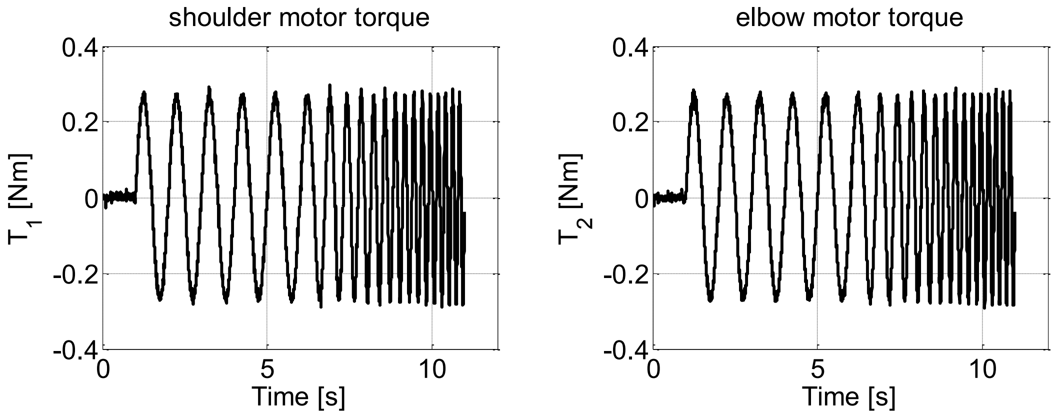

Figure 3.3. Sine Sweep Signals to Motors.

period. This period and the last 0.5 seconds of the experiment were removed from the data set for estimation. The velocity is calculated from

$$\dot{\theta}_i = \frac{\theta_i(k+1)-\theta_i(k-1)}{\Delta T} \tag{93}$$

where $\theta_i(k)$, $k = 1,2$ ... are the values of the discrete angular position measurements and $\Delta T = 0.001s$ is the sampling time. The second order central difference

$$\ddot{\theta}_i = \frac{\theta_i(k+1)-2\theta_i(k)+\theta_i(k-1)}{\Delta T^2} \tag{94}$$

is used for calculating the acceleration signal.

2000 training data chosen even from the obtained data are used to train the system and 500 testing data that are different from the training data, are used to test the accuracy of the neuro-fuzzy model and TS fuzzy model. The system is linearized at 9 or 27 operating points from the neuro-fuzzy model to get the corresponding TS fuzzy models. The 9 operating points are based on the combination of three different θ_2 values of -0.7, 0, 0.7 and one $\dot{\theta}_1$ values of 0 and three different $\dot{\theta}_2$ values of $-1, 0, 1$. For 27 operating points, except for $\dot{\theta}_1$ has three values of -1, 0 and 1, values of the other two parameters are same as that of 9 operating points.

As shown in Figure 3.4, the points in the red dashed line are testing experimental data. The points in the blue solid line and black dash-dot line are the outputs of the neuro-fuzzy model and TS fuzzy models respectively. Because the outputs of the TS fuzzy models for 9 operating points and 27 operating points are almost same, they are shown by the lines of same color in the figures. The lines in these figures overlap closely, which indicates the accuracy of the neuro-fuzzy model and TS fuzzy model.

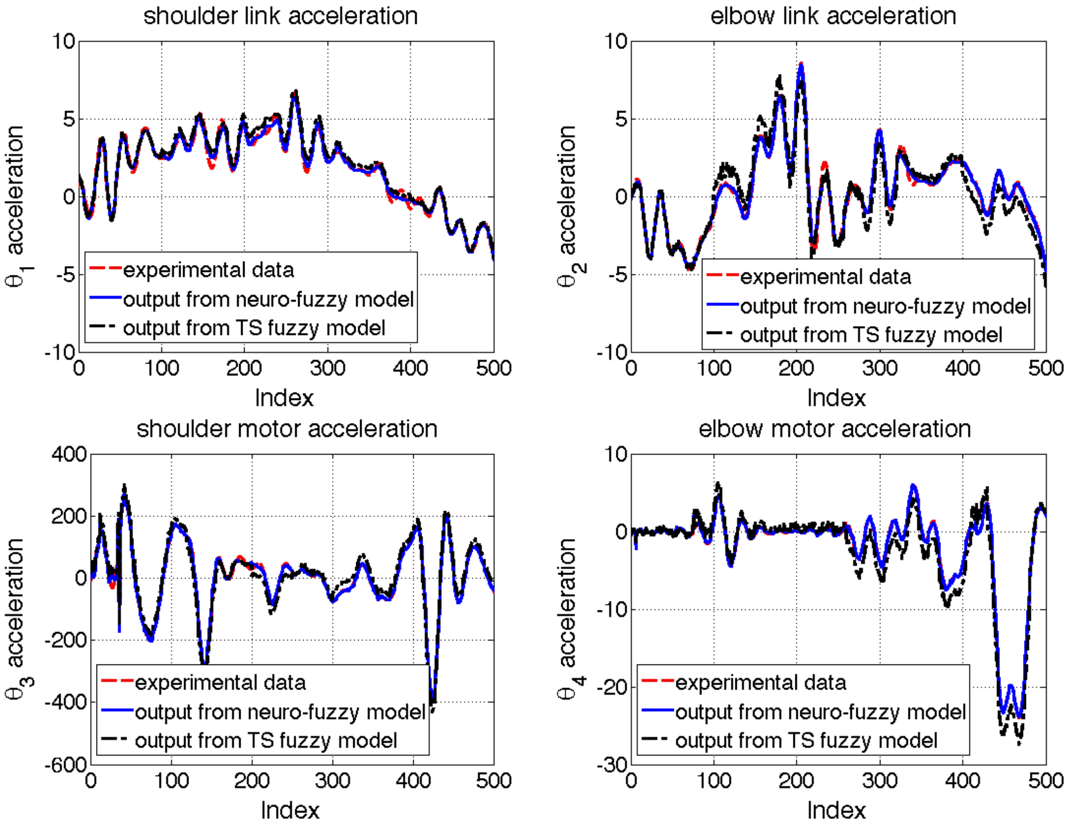

Figure 3.4. Experimental Data and Outputs from Neuro-Fuzzy Models and TS Fuzzy Models.

The initial condition is set as $\boldsymbol{x} = [0 \quad 0 \quad 0 \quad 0 \quad 0 \quad 0 \quad 0 \quad 0]^T$, and the command inputs for both θ_1 and θ_2 are given by

$$r_1 = \begin{cases} 0.4t^2 & t \le 0.4 \\ 0.064 + 0.32(t - 0.4) & 0.4 < t \le 1.875 \\ 0.6 - 0.4(t - 2.275)^2 & 1.875 < t \le 2.275 \\ 0.6 & 2.275 < t \le 4 \end{cases} \tag{95}$$

Because the gear ratio is 5, so the command inputs for both θ_3 and θ_4 are given by

$$r_2 = 5r_1 \tag{96}$$

The command inputs for both $\dot{\theta}_1$ and $\dot{\theta}_2$ are given by

$$r_3 = \begin{cases} 0.8t & t \le 0.4 \\ 0.32 & 0.4 < t \le 1.875 \\ -0.8(t - 2.275) & 1.875 < t \le 2.275 \\ 0 & 2.275 < t \le 4 \end{cases} \tag{97}$$

Similar to θ_3 and θ_4, the command inputs for both $\dot{\theta}_3$ and $\dot{\theta}_4$ are given by

$$r_4 = 5r_3 \tag{98}$$

The performance index is defined as:

$$J = \int_0^4 \left(10000\left(\boldsymbol{x_1} - \begin{bmatrix} r_1 \\ r_1 \end{bmatrix}\right)^2 + \left(\boldsymbol{x_2} - \begin{bmatrix} r_2 \\ r_2 \end{bmatrix}\right)^2 + 1000\left(\boldsymbol{x_3} - \begin{bmatrix} r_3 \\ r_3 \end{bmatrix}\right)^2 + \left(\boldsymbol{x_4} - \begin{bmatrix} r_4 \\ 4 \end{bmatrix}\right)^2 + \right.$$

$$\left. 0.1T_1^2 + 0.1T_2^2\right) dt \tag{99}$$

The optimal controllers derived from the TS fuzzy models based on 9 and 27 operating points were first simulated using the mathematical plant model in MATLAB. Figure 3.5 illustrates the response the flexible robot system with the optimal controller. The closed-loop behavior of the robot can track the designed trajectories very well. Besides, whether the TS fuzzy model has 9 operating points or 27 operating points, the results are almost the same. Therefore, 9 operating points are enough for the design of optimal controller.

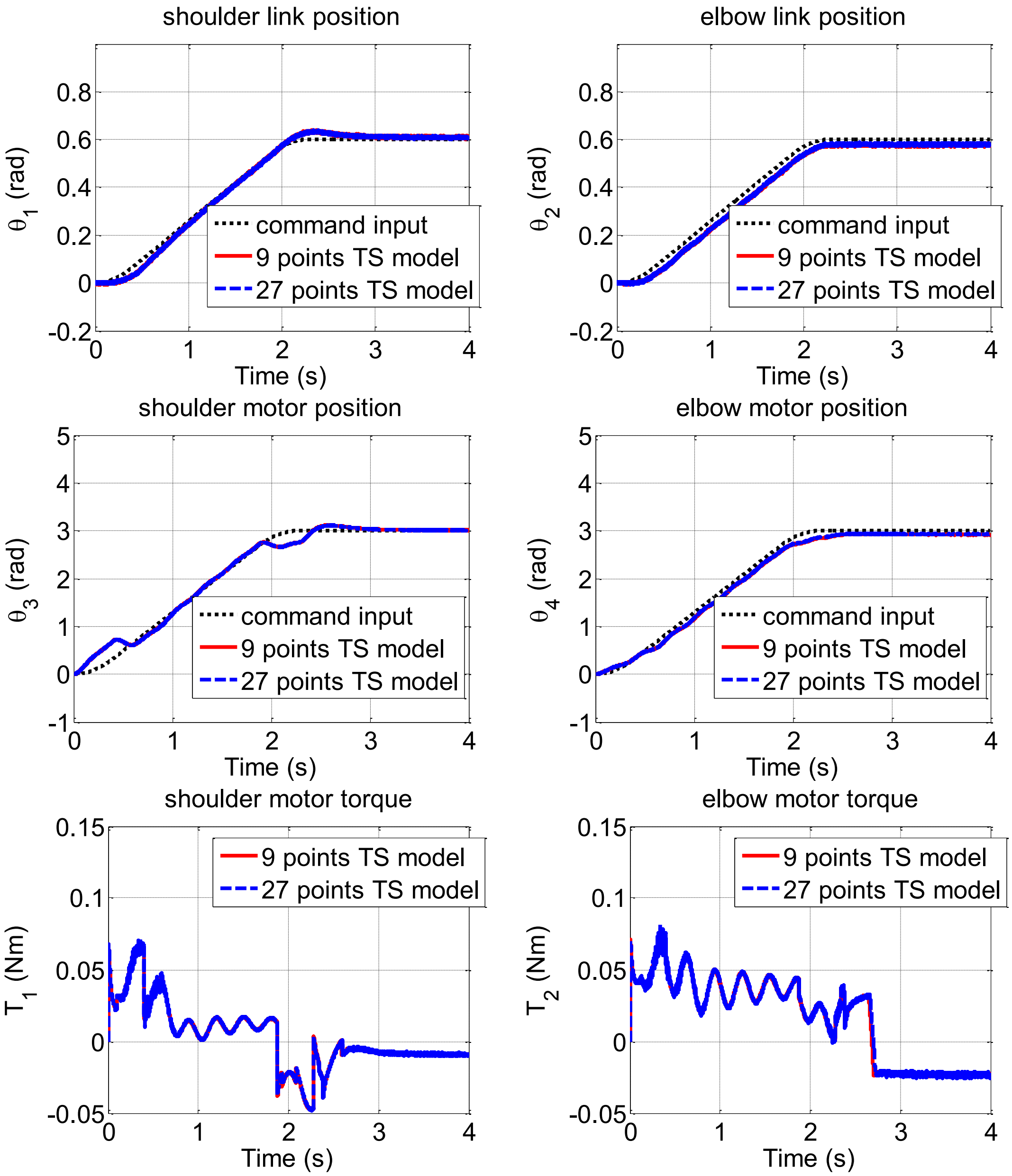

Figure 3.5. Simulation Results of the Optimal Controller for TS Models of Flexible Robot.

Then the optimal controllers derived from 9 and 27 operating points TS fuzzy model were implemented for the robot. The experimental results are shown in Figure 3.6. The results are very good except for the oscillations in the trajectories. These oscillations are caused by the flexible joints of the robot, which are very difficult to eliminate.

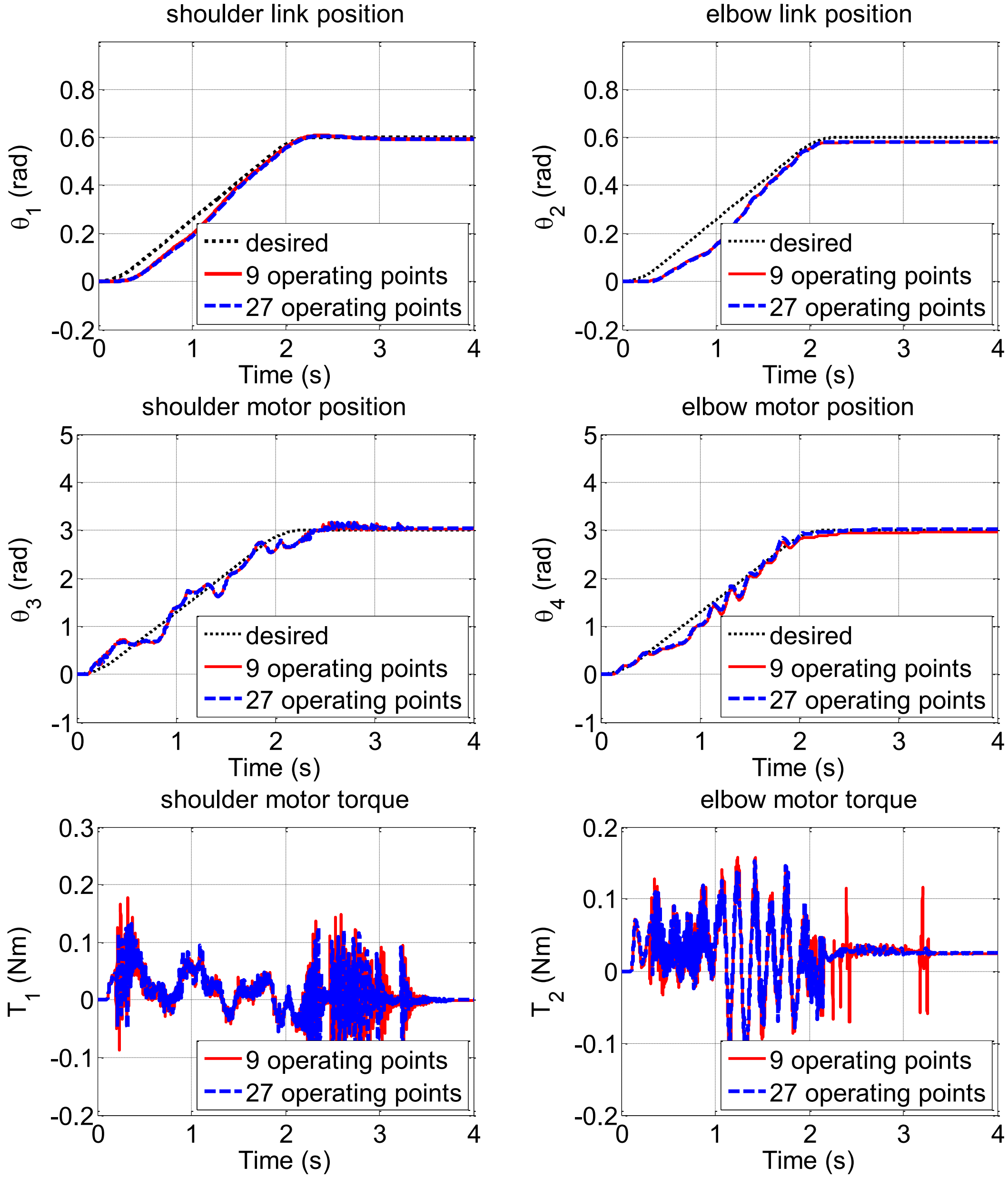

Figure 3.6. Experimental Results of the Optimal Controller for TS Models of Flexible Robot.

3.4.2 Rigid Asymmetric Spacecraft

Tracking of a rigid asymmetric spacecraft is concerned with a primary attitude control task. Due to inherent nonlinearity of attitude dynamics, tracking in large and rapid maneuvers is a complex undertaking. Therefore, this tracking problem with three independent axis controls is investigated here. The Euler's equations for the angular velocities x_1, x_2, x_3 of the spacecraft are given by [69]

$$\dot{x}_1(t) = -\frac{I_3-I_2}{I_1} x_2(t)x_3(t) + \frac{u_1(t)}{I_1} \tag{100}$$

$$\dot{x}_2(t) = -\frac{I_1-I_3}{I_2} x_1(t)x_3(t) + \frac{u_2(t)}{I_2} \tag{101}$$

$$\dot{x}_3(t) = -\frac{I_2-I_1}{I_3} x_1(t)x_2(t) + \frac{u_3(t)}{I_3} \tag{102}$$

where u_1, u_2, u_3 are the control torques, and $I_1 = 86.24$ kg · m^2, $I_2 = 85.07$ kg · m^2 and $I_3 = 113.59$ kg · m^2 are the spacecraft principal inertias.

Similar to the flexible robot, 2000 input-output data pairs were utilized to train the neuro-fuzzy models by the OLS algorithm [9]. The ranges of states x_1, x_2 and x_3 were from -0.1 to 0.1, the ranges of inputs u_1, u_2 and u_3 were from -0.2 to 0.2. The widths of the states x_1, x_2 and x_3 were fixed to be 0.05, the widths of the inputs u_1, u_2 and u_3 were fixed to be 0.1. After trained, 500 rules were selected to construct the fuzzy models corresponding to equations (101), (102) and (103) respectively.

The system is linearized at 27 operating points from the neuro-fuzzy model to get a TS fuzzy model. The 27 operating points are chosen using the combination of three different x_1 states of 0, 0.02, 0.04, three different x_2 states of 0, 0.02, 0.04, three different x_3 states of 0, 0.02, 0.04 and $u_1 = u_2 = u_3 = 0.1$.

The system responses with constant inputs $u_1 = 0.1$ Nm, $u_2 = 0.1$ Nm, $u_3 = 0.1$ Nm were simulated using the mathematical model, neuro-fuzzy model and TS fuzzy model. As shown in Figure 3.7, the system responses of the three models are almost identical.

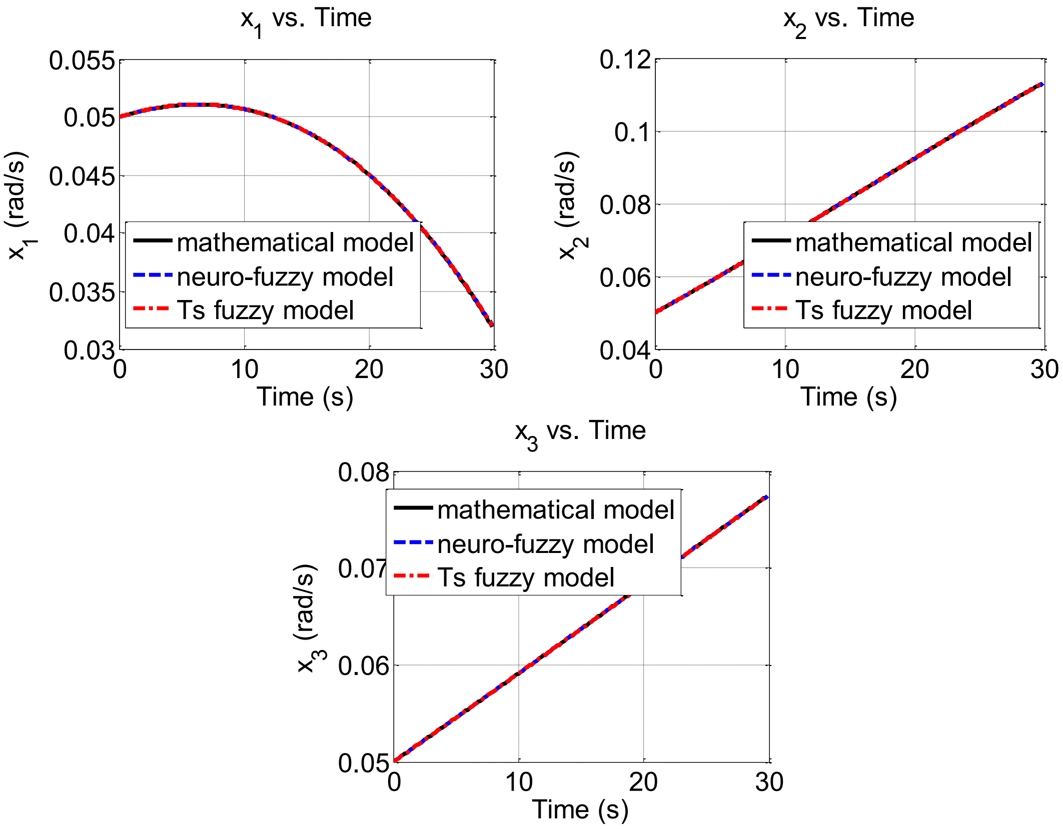

Figure 3.7. The System Responses of the Mathematical Model, Neuro-Fuzzy Model and TS Fuzzy Model with a Same Control Input.

The initial conditions are $x_1 = 0$ rad/s, $x_2 = 0$ rad/s and $x_3 = 0$ rad/s, the desired steady states are set to be $r_1 = 0.04$ rad/s, $r_2 = 0.04$ rad/s and $r_3 = 0.04$ rad/s respectively. The performance index is given by:

$$J = \int_0^{150} (100(x_1 - r_1)^2 + 100(x_2 - r_2)^2 + (x_3 - r_3)^2 + 0.01u_1^2 + 0.01u_2^2 + 0.01u_3^2)dt \quad (103)$$

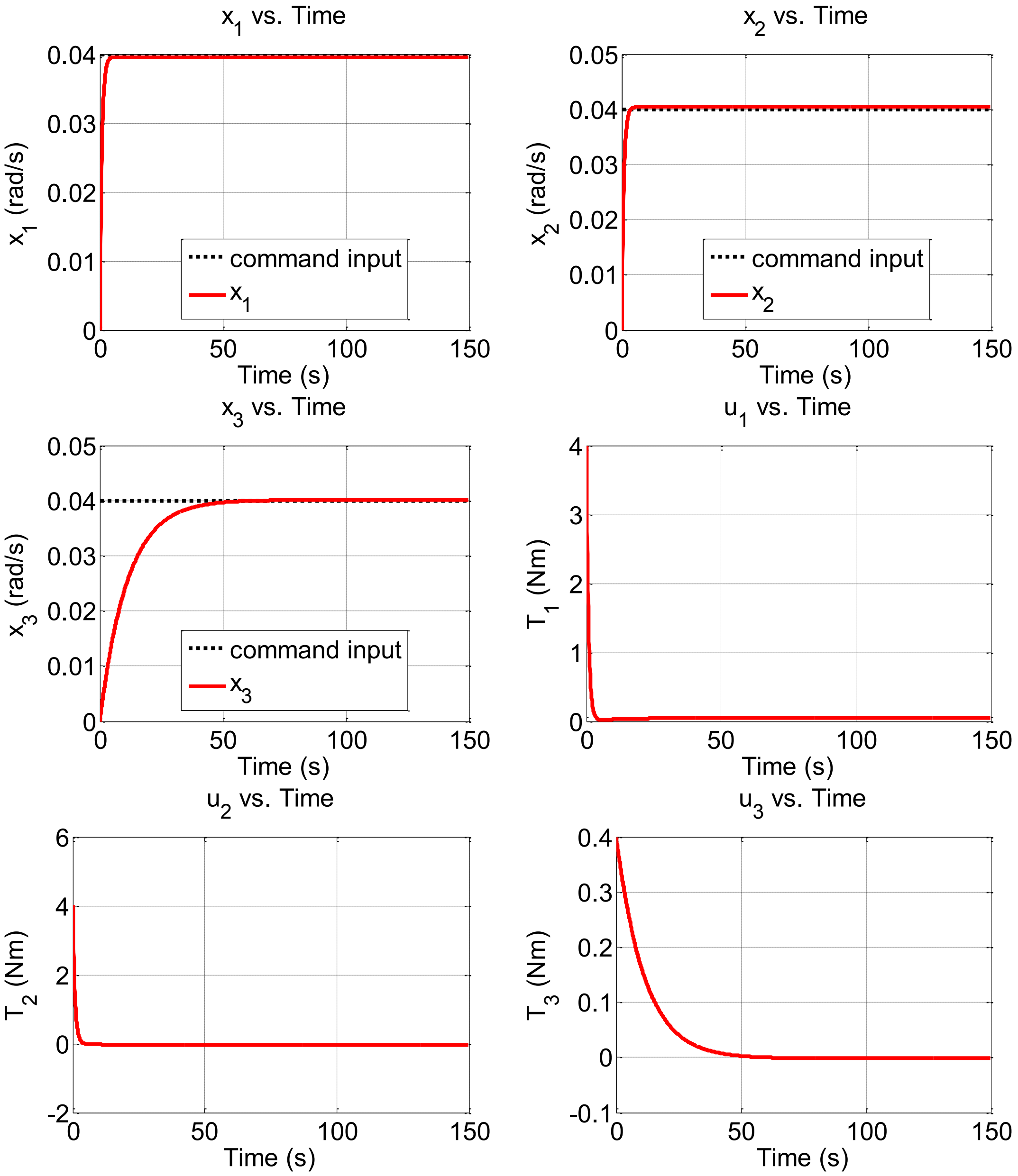

Figure 3.8. Experimental Results of the Optimal Controller for the Asymmetric Spacecraft.

The optimal controller derived from the TS fuzzy model was simulated using the mathematical plant model. The simulation results are shown in Figure 3.8. The proposed method can achieve very good simulation results for nonlinear systems. Therefore, the performance of the proposed method is very good.

CHAPTER 4. CONCLUSION

In this work, an optimal controller to the nonlinear system represented by a neuro-fuzzy model has been developed. With the product inference engine, singleton fuzzifier, center average defuzzifier and Gaussian membership functions, a fuzzy model was trained by the OLS learning algorithm which is very efficient and not sensitive to noise in its inputs, and then the optimal control problem was formulated based on the fuzzy model. The numerical solution of the problem was obtained by use of a feasible-direction algorithm. This algorithm uses the steepest descent to find the search direction and then apply a one-dimensional search routine to find the best step length. It has a very high computational efficiency and very easy to implement. The simulation results of three nonlinear optimal control examples showed that the performance of the proposed approach based on a fuzzy model was quite similar to that of optimal control to the system represented by an explicit mathematical model, thus demonstrating its efficacy for optimal control of unknown nonlinear systems. However, due to the limitation of the feasible-direction numerical algorithm, it cannot be applied to a too complex system or a control process with too many time steps. Therefore, a better way which also utilized the TS fuzzy model was then developed.

The TS fuzzy model can be also used to model the complex nonlinear systems and shown to be a universal function approximator. However, unlike neuro-fuzzy model, most training algorithms can hardly provide satisfied local accuracy to be used for control. Therefore, an explicit procedure of establishing TS fuzzy models of unknown nonlinear systems from experimental data has been presented to make the local models are close approximations to the local linearizations of the nonlinear dynamic systems The system responses of the TS fuzzy models for the same control inputs were shown to be very close to the response of the actual systems, which indicated the accuracy of the proposed identification method. Then the optimal controllers derived from TS fuzzy models were experimented to a very complex two-link flexible robot and achieved very good results, which demonstrates the practicality and effectiveness of the proposed algorithm for controlling unknown complex nonlinear dynamic system.

In future work, a better method other than the feasible-direction algorithm can be used to extend the first method to more complex systems. In addition, for the second method, a more rigorous method to eliminate the unnecessary states and how to deal with them could be studied.

LIST OF REFERENCES

LIST OF REFERENCES

[1] Withit, C., Backstepping intelligent control applited to a flexible-joint robot manipulator. Doctoral dissertation, Department of Mechanical Engineering, Purdue University, Indinan, 2006.

[2] Cornelius, L., Fuzzy theory systems techniques and applications. California: Academic Press, 1999.

[3] Kosko, B., Fuzzy systems as universal approximators, *IEEE Transactions on Computers*, 43(11), pp.1329-1333, 1994.

[4] Sugeno, M. and Kang, G. T. Structure identification of fuzzy model. *Fuzzy sets systems*, 28, pp.15-33, 1988.

[5] Sugeno, M. and Yasukawa, T. A fuzzy logic based approach to qualitative modeling. *IEEE Transactions on Fuzzy Systems* 1, pp.7-31, 1993.

[6] Wang, L. X. and Mendel, J.M., Back-propagation fuzzy system as nonlinear dynamic system identifiers, *IEEE International Conference on Fuzzy Systems*, pp.1409-1418, 1992.

[7] Wang, L. X., A Course on Fuzzy Systems. Prentice-Hall Press, USA, 1999.

[8] Chiu S. L., Fuzzy model identification based on cluster estimation, *Journal of intelligent and Fuzzy systems*, 2(3), pp.267-278, 1994.

[9] Wang, L. X. and Mendel, J. M., Fuzzy basis functions, universal approximation, and orthogonal least-squares learning, *IEEE Transactions on Neural Networks*, 3(5), pp.807-814, 1992.

[10] Cheol W. L. and Yung C. S., Construction of fuzzy systems using least-squares method and genetic algorithm, *Fuzzy Sets and Systems*, 137(3), pp.297-323, 2003.

[11] Chen, S., Cowan, C. F. N., and Grant, P. M., Orthogonal least squares learning algorithm for radial basis function networks, *IEEE Transactions on Neural Networks*, 2(2), pp.302-309, 1991.

[12] Sussmann, H. J., and Willems, J. C., 300 years of optimal control: from the brachystochrone to the maximum principle, *IEEE Control Systems*, 17(3), pp.32-44, 1997.

[13] Pontryagin, L. S., Boltyanskiy, V. G., Gramkrelidze, R. V. and Mischenko, E. F., The Mathematical Theory of Optimal Processes, Interscience, New York, 1962.

[14] Al-Tamimi, A., Lewis, F. L., and Abu-Khalaf, M., Discrete time Nonlinear HJB Solution Using Approximate Dynamic Programming: Convergence Proof, *IEEE Transactions on Systems, Man, and Cybernetics, Part B: Cybernetics*, 38(4), pp.943-949, Aug. 2008.

[15] Lewis, F.L., and Vrabie, D., Reinforcement learning and adaptive dynamic programming for feedback control, *IEEE Circuits and Systems Magazine*, 9(3), pp.32-50, 2009.

[16] Wang, F. Y., Zhang, H., and Liu, D., Adaptive Dynamic Programming: An Introduction, *IEEE Computational Intelligence Magazine*, 4(2), pp.39-47, 2009.

[17] Chen, Z., and Jagannathan, S., Generalized Hamilton–Jacobi–Bellman Formulation Based Neural Network Control of Affine Nonlinear Discrete time Systems, *IEEE Transactions on Neural Networks*, 19(1), pp.90-106, 2008.

[18] Al-Tamimi, A., Lewis, F. L., and Abu-Khalaf, M., Model-free Q-learning designs for linear discrete time zero-sum games with application to H-infinity control. *Automatica*, 43(3), pp.473–481, 2007.

[19] Dierks, T., Thumati, B. T., and Jagannathan, S., Optimal control of unknown affine nonlinear discrete time systems using offline-trained neural networks with proof of convergence. *Neural Networks*, 22(5), pp.851-860, 2009.

[20] Dierks T., and Jagannathan S., Online optimal control of nonlinear discrete time systems using approximate dynamic programming. *Journal of Control Theory and Applications*, 9(3), pp.361-369, 2011.

[21] Wei Q., Zhang H., and Dai J., Model-free multiobjective approximate dynamic programming for discrete time nonlinear systems with general performance index functions. *Neurocomputing*, 72(7), pp.1839-1848, 2009.

[22] Heydari, A.; and Balakrishnan, S. N., Finite-Horizon Control-Constrained Nonlinear Optimal Control Using Single Network Adaptive Critics, *IEEE Transactions on Neural Networks and Learning Systems*, 24(1), pp.145-157, 2013.

[23] Jiang, Y., and Jiang, Z. P., Computational adaptive optimal control for continuous-time linear systems with completely unknown dynamics, *Automatica*, 48(10), pp.2699-2704, 2012.

[24] Kotsialos, A., Papageorgiou, M., and Messmer, A., On the approximation capabilities of the homogeneous Takagi-Sugeno model, *14th International Symposium on Transportation and Traffic Theory*, pp.1067–1072, 1996.

[25] Johansen, T. A., Shorten, R., and Murray, S. R., On the interpretation and identification of dynamic Takagi–Sugeno models, *IEEE Tranasactions on. Fuzzy Systems*, 8(3), pp.297–313, 2000.

[26] Sugeno, M., On stability of fuzzy systems expressed by fuzzy rules with singleton consequents, *IEEE Tranasactions on. Fuzzy Systems*, 7(2), pp.201–224, 1999.

[27] Khalil, H. K. and Grizzle, J. W. Nonlinear systems. vol. 3. Upper Saddle River: Prentice hall, 2002.

[28] Takagi, T. and Sugeno, M., Fuzzy identification of systems and its applications to modeling and control, *IEEE Transactions on Systems, Man and Cybernetics*, 15(1), pp.116-132, 1985.

[29] Gang F., A Survey on Analysis and Design of Model-Based Fuzzy Control Systems, *IEEE Transactions on Fuzzy Systems*, 14(5), pp.676-697, 2006.

[30] Wang, H. O., Tanaka, K., and Griffin, M., Parallel distributed compensation of nonlinear systems by Takagi-Sugeno fuzzy model, *International Joint Conference of the Fourth IEEE International Conference on Fuzzy Systems and The Second International Fuzzy Engineering Symposium,* vol.2, pp.531-538, 1995.

[31] Li, J., Wang, H. O., Niemann, D., and Tanaka, K., Dynamic parallel distributed compensation for Takagi–Sugeno fuzzy systems: An LMI approach, *Information Sciences*, vol.123, no.3–4, pp.201-221, 2000.

[32] Chen, C. W. Application of fuzzy-model-based control to nonlinear structural systems with time delay: an LMI method, *Journal of Vibration and Control*, vol.16, no.11, pp: 1651-1672, 2010.

[33] Fang, C. H., Liu, Y. S., Kau, S. W., Hong, L. and Lee, C. H. A new LMI based approach to relaxed quadratic stabilization of T–S fuzzy control systems, *IEEE Transactions on Fuzzy System,* vol.14, no.3, pp.386–397, 2006.

[34] Sala, A. and Arino, C., Asymptotically necessary and sufficient conditions for stability and performance in fuzzy control: Applications of Polya's theorem, *Fuzzy Sets Systems,* vol.151, no.24, pp.2671–2686, 2007.

[35] Lam, H. K., LMI-Based Stability Analysis for Fuzzy-Model-Based Control Systems Using Artificial T–S Fuzzy Model, *IEEE Transactions on Fuzzy Systems*, vol.19, no.3, pp.505-513, 2011.

[36] Lam, H. K., and Lauber, J., Membership-function-dependent stability analysis of fuzzy model based control systems using fuzzy Lyapunov functions, *Information Sciences*, vol.232, pp.253-266, 2013.

[37] Chang, W., Park, J. B., Young, H. J., and Chen, G. Design of robust fuzzy-model-based controller with sliding mode control for SISO nonlinear systems, *Fuzzy Sets and Systems*, 125(1), pp.1-22, 2002.

[38] Chen, S. T., Stability analysis and design of Takagi–Sugeno fuzzy systems, *Information Sciences*, vol.176, no.19, pp.2817-2845, 2006.

[39] Stefan, J., Christoph, H., and Nikolaus, K., Total least squares in fuzzy system identification: an application to an industrial engine, *Engineering Applications of Artificial Intelligence*, vol.21, no.8, pp.1277–1288, 2008.

[40] Lin, F. J., Lin, C. H., and Shen, P. H., Self-constructing fuzzy neural network speed controller for permanent-magnet synchronous motor drive, *IEEE Transactions on Fuzzy Systems*, vol.9, no.5, pp.751-759, 2001.

[41] Roy, A., Briczinski, S. J., Doherty, J. F., and Mathews, J. D., Genetic algorithm based parameter estimation technique for fragmenting radar meteor head echoes, *IEEE Geoscience and Remote Sensing Letters*, vol.6, no.3, pp.363–367, 2009.

[42] Yen, J., Wang, L., and Gillespie, C. W., Improving the interpretability of TSK fuzzy models by combining global learning and local learning, *IEEE Transactions on Fuzzy Systems*, vol.6, pp.530–537, 1998.

[43] Johansen, T. A., Shorten, R., and Murray-Smith, R., On the interpretation and identification of dynamic Takagi-Sugeno fuzzy models, *IEEE Transactions on Fuzzy Systems*, vol.8, no.3, pp.297-313, 2000.

[44] Kotsialos, A., Papageorgiou, M.. and Messmer A., Optimal coordinated and integrated motorway network traffic control, *14th International Symposium on Transportation and Traffic Theory, Jerusalem, Israel*, pp.621–644, 1999.

[45] Carlson, R. C., Papamichail, I., Papageorgiou, M. and Messmer, A., Optimal motorway traffic flow control involving variable speed limits and ramp metering. *Transportation Science*, 44(2), pp.238-253, 2010.

[46] Narendra, K. S., and Parthasarathy, K. Identification and control of dynamical systems using neural networks. *IEEE Transactions on Neural Networks*, 1(6), pp.4-27, 1990.

[47] Jin, L., Nikiforuk, P. N. and Gupta, M. M.. Direct adaptive output tracking control using multilayered neural networks. *IEEE Proceedings in Control Theory and Applications*. 140(6). 1993.

[48] Yeşildirek, A. and Lewis , F. L., Feedback linearization using neural networks. *Automatica*. 31(11). pp.1659-1664, 1995.

[49] Spooner, J. T., and Kevin M. P. Stable adaptive control using fuzzy systems and neural networks. *IEEE Transactions on Fuzzy Systems*, 4(3), pp.339-359, 1996.

[50] Zhang, T., Shuzhi S. G., and Chang C. H., Adaptive neural network control for strict-feedback nonlinear systems using backstepping design. Automatica, 36(12), pp. 1835-1846. 2000.

[51] Ge, S. S., Li, G. Y., Zhang, ‚J. and Lee, T. H. Direct adaptive controlfor a class of MIMO nonlinear systems using neural networks, *IEEE Transactions on. Automatica. Control*, 49(11), pp. 2001–2006, 2004.

[52] Yang, B. J. and Anthony J. C., Adaptive control of a class of nonaffine systems using neural networks. *IEEE Transactions on Neural Networks*, 18(4), pp.1149-1159, 2007.

[53] Zhu, Q. and Guo, L. Stable adaptive neurocontrol for nonlinear discrete-time systems., *IEEE Transactions on Neural Networks*, 15(3), 653-662. 2004.

[54] Ge, S. S., Zhang, J. and Lee, T. H. Adaptive neural network control for a class of MIMO nonlinear systems with disturbances in discrete-time, *IEEE Transactions on Systems, Man, and Cybernetics, Part B: Cybernetics*, 34(4), pp.1630–1644, 2004.

[55] Ge, S. S. and Wang, C. Adaptive neural control of uncertain MIMO nonlinear systems, *IEEE Transactions on. Neural Network*, 15(3), pp.674–692, 2004.

[56] Ge, S. S., Fan, H. and Lee, T. H. Adaptive neural control of nonlinear time-delay systems with unknown virtual control coefficients. *IEEE Transactions on Systems, Man, and Cybernetics, Part B: Cybernetics,* 34(1), pp.499-516. 2004.

[57] Chen, B., Liu, X., Liu, K. and Lin, C. Novel adaptive neural control design for nonlinear MIMO time-delay systems. *Automatica*, 45(6), pp.1554-1560. 2009.

[58] Tong, S. C., Li, Y. M. and Zhang, H. G. Adaptive neural network decentralized backstepping output-feedback control for nonlinear large-scale systems with time delays. *IEEE Transactions on Neural Networks*, 22(7), pp.1073-1086. 2011.

[59] Han, T. T., Ge, S. S. and Lee, T. H. Adaptive neural control for a class of switched nonlinear systems. *Systems & Control Letters* 58(2), pp.109-118. 2009.

[60] Bertsekas, D. P., Nonlinear programming. Athena Scientific, Belmont, MA.1999.

[61] Sun, W., and Yuan, Y., Optimization theory and methods: nonlinear programming. Springer, New York. 2006.

[62] Magni, L., De Nicolao, G., Scattolini, R., and Allgöwer, F. Robust model predictive control for nonlinear discrete time systems. *International Journal of Robust and Nonlinear Control*, 13(3-4), 229-246. 2006.

[63] Jaddu, H.. Direct solution of nonlinear optimal control problems using quasilinearization and Chebyshev polynomials. *Journal of the Franklin Institute*, 339(4), 479-498. 2002.

[64] Hicks, G. A., and Ray, W. H. Approximation methods for optimal control synthesis. *The Canadian Journal of Chemical Engineering*, 49(4), 522-528. 1971.

[65] Ho, W. H., Chen, S. H., Chen, I. T., Chou, J. H., and Shu, C. C., Design of stable and quadratic-optimal static output feedback controllers for TS fuzzy-model-based control systems: an integrative computational approach, *International Journal of Innovative Computing, Information and Control*, 8(1), pp.403-418, 2012.

[66] Kirk, D. E., Optimal Control Theory, Prentice Hall. Englewood Cliffs, NJ. 1970.

[67] Mohamed, Z., and Tokhi, M.O., Command shaping techniques for vibration control of a flexible robot manipulator, Mechatronics, 14(1), pp.69-90, 2004.

[68] Scheel, A., Command shaping applied to a two-link flexible-joint robot, Master Thesis, Department of Mechanical Engineering, Purdue University, Indinan, 2012.

[69] Jaddu, H., Direct solution of nonlinear optimal control problems using quasilinearization and Chebyshev polynomials, *Journal of the Franklin Institute*, 339(4), pp.479-498.