# LEARNING AND INTELLIGENT CONTROL OF NONLINEAR SYSTEMS USING DYNAMIC NEURAL NETWORKS

YINYAN ZHANG

PhD

The Hong Kong Polytechnic University

2019

The Hong Kong Polytechnic University

Department of Computing

# Learning and Intelligent Control of Nonlinear Systems Using Dynamic Neural Networks

ZHANG Yinyan

A thesis submitted in partial fulfillment of the requirements for

the degree of Doctor of Philosophy

June 2019

# CERTIFICATE OF ORIGINALITY

I hereby declare that this thesis is my own work and that, to the best of my knowledge and belief, it reproduces no material previously published or written, nor material that has been accepted for the award of any other degree or diploma, except where due acknowledgement has been made in the text.

_____ (Signed)

ZHANG Yinyan (Name of Student)

## Abstract

Optimal control is concerned with finding a control law that drives a controlled system to a desired target in an optimal way, i.e., to minimize or maximize a predefined performance index. For nonlinear systems, the optimal control of nonlinear systems requires the solution of a partial differential equation, called the Hamilton-Jacobi-Bellman equation, for which the analytical solution is difficult or even impossible to obtain. Consequently, efforts are made on the near-optimal control, which aims at finding an approximate solution to the optimal control problem. When the system parameters are unknown or the system dynamics is unknown, the near-optimal control problem becomes more difficult. In this thesis, we are concerned with the learning and intelligent control of nonlinear systems using dynamic neural networks.

First, a unified online learning and near-optimal control framework is proposed for linear and nonlinear systems with parameter uncertainty. It is also proved that the proposed learning and near-optimal control law asymptotically converges to the optimal. The efficacy of the proposed framework and the theoretical results are validated by an application to underactuated surface vessels.

Second, a learning and near-optimal control law, which is inherently real-time, is designed to tackle the contradictory between solution accuracy and solution speed for the optimal control of a general class of nonlinear systems with fully unknown parameters. The key technique in the proposed learning and near-optimal control is to design an auxiliary system, which can be viewed as a dynamic neural network, with the aid of the sliding mode control concept to learn the dynamics of the controlled nonlinear system. Based on the sliding-mode auxiliary system and approximation of the performance index, the proposed control law guarantees asymptotic stability of the closed-system and asymptotic optimality of the performance index with time.

Third, a novel model-free learning and near-optimal control method is proposed for nonlinear systems via utilizing the Taylor expansion based problem relaxation, the uni-

versal approximation property of sigmoid neural networks, and the concept of sliding-mode control. By making approximation for the performance index, it is first relaxed to a quadratic program, and then a linear algebraic equation with unknown terms. An auxiliary system, which can be viewed as a dynamic neural network, is designed to reconstruct the input-to-output property of the control systems with unknown dynamics, so as to tackle the difficulty caused by the unknown terms, i.e., to learn the unknown dynamics.

Fourth, the learning and near-optimal distributed consensus of high-order nonlinear multi-agent systems consisting of heterogeneous agents is investigated. The consensus problem is formulated as a receding-horizon optimal control problem. For the situation with fully unknown system parameters, sliding-mode auxiliary systems, which could be viewed as dynamic neural networks and are independent for different agents, are built to reconstruct the input-output properties of agents. Based on the sliding-mode auxiliary systems, an adaptive near-optimal protocol is finally presented to control high-order nonlinear multi-agent systems with fully unknown parameters. Theoretical analysis shows that the proposed protocols can simultaneously guarantee the asymptotic optimality of the performance index and the asymptotic consensus of multi-agent systems.

Fifth, inspired by the success of the learning and near-optimal control method, we consider a special physical system, i.e., redundant manipulators. Redundancy resolution is of great importance in the control of manipulators. Among the existing results for handling this issue, the quadratic program approaches, which are capable of optimizing performance indices subject to physical constraints, are widely used. However, the existing quadratic program approaches require exactly knowing all the physical parameters of manipulators, the condition of which may not hold in some practical applications. This fact motivates us to consider the application of learning and intelligent control techniques for simultaneous parameter learning and control. We establish the first adaptive dynamic neural network with online learning for the redundancy resolution

of manipulators with unknown physical parameters so as to solve the intelligent control problem, which tackles the dilemmas in existing methods.

## List of Publications Arising From the Thesis

### Journal Papers

1. **Y. Zhang**, S. Li, and X. Liu, "Neural network-based model-free adaptive near-optimal tracking control for a class of nonlinear systems," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 29, no. 12, pp. 6227–6241, 2018.

2. **Y. Zhang**, S. Li, and L. Liao, "Near-optimal control of nonlinear dynamical systems: A brief survey," *Annual Rev. Control*, in press, doi: 10.1016/j.arcontrol. 2019.01.003.

3. **Y. Zhang**, S. Li, and X. Jiang, "Near-optimal control without solving HJB equations and its applications," *IEEE Trans. Ind. Electron.,* vol. 65, no. 9, pp. 7173–7184, 2018.

4. **Y. Zhang**, S. Li, S. Kadry, and B. Liao, "Recurrent neural network for kinematic control of redundant manipulators with periodic input disturbance and physical constraints," *IEEE Trans. Cybern.*, in press, doi: 10.1109/TCYB.2018.2859751.

5. **Y. Zhang**, S. Chen, S. Li, and Z. Zhang, "Adaptive Projection Neural Network for Kinematic Control of Redundant Manipulators With Unknown Physical Parameters," *IEEE Trans. Ind. Electron.*, vol. 65, no. 6, pp. 4909–4920, 2018.

6. **Y. Zhang** and S. Li, "Adaptive near-optimal consensus of high-order nonlinear multi-agent systems with heterogeneity," *Automatica*, vol. 85, pp. 426–432, 2017.

7. **Y. Zhang** and S. Li, "Predictive suboptimal consensus of multiagent systems with nonlinear dynamics," *IEEE Transactions on Syst., Man, Cybern., Syst.*, vol. 47, no. 7, pp. 1701–1711, 2017.

8. **Y. Zhang** and S. Li, "Time-scale expansion-based approximated optimal control for underactuated systems using projection neural networks," *IEEE Trans. Syst., Man, Cybern., Syst.*, vol. 48, no. 11, pp. 1957–1967, 2018.

9. **Y. Zhang**, S. Li, and X. Liu, "Adaptive near-optimal control of uncertain systems with

application to underactuated surface vessels," *IEEE Trans. Control Syst. Technol.*, vol. 26, no. 4, pp. 1204–1218, 2018.

**Conference Papers**

1. Y. Zhang, S. Li, X. Luo, and M. Shang, "A dynamic neural controller for adaptive optimal control of permanent magnet DC motors," *IJCNN*, Anchorage, AK, USA, 2017, pp. 839–844.

## Acknowledgements

I would like to give my deepest gratitude to my supervisor, Dr. LI Shuai for his support and guidance throughout my PhD studies. He is always willing to give me some inspiring advices on my research based on his strong background in control theory and applications. I also would like to thank my co-supervisor Dr. WANG Qixin, who is always very friendly when I need any assistance. I would like to thank Prof. ZHANG Yunong and Prof. JIN Long, who provided me with a chance to cooperate with Dr. LI when I was an undergraduate student in Sun Yat-sen University, and all the colleagues graduated from Prof. ZHANG Yunong's lab who gave me so much supports and happiness.

I would like to thank Prof. CHEN Wen-Hua with Loughborough University for his comments and discussions given to part of my works presented in this thesis. My thanks also go to Mr. CHEN Siyuan and Dr. ZHANG Zhijun for their help in conducting the experiment in Chapter 6.

I would also like to thank the colleagues in PolyU, such as Mr. Han Lei, Mr. Yang Zhonghuang, Mr. LU Zexin, and Mr. LI Zhe, for their friendship and sharing of happiness which made my life in Hong Kong much easier.

I am very grateful to my family and my girl friend, for their endless love and care. Without them, the journey of PhD study would be much more difficult.

# Table of Contents

# List of Figures

# List of Tables

# List of Abbreviations

APNN: Adaptive projection neural network

D-H: Denavit-Hartenberg

DOF: Degrees of freedom

HJB: Hamilton-Jacobi-Bellman

LNOC: Learning and near-optimal control

NPNN: Nominal projection neural network

PE: Persistent excitation

PID: Proportional-integral-derivative

PNN: Projection neural network

QP: Quadratic program

# Chapter 1

# Introduction

In this chapter, we briefly introduce the topics considered in this thesis. Then, we show the contributions and organizations of the thesis.

## 1.1 Learning and near-optimal control

Optimal control is concerned with finding a control law that drives a controlled system to a desired target in an optimal way, i.e., to minimize or maximize a predefined performance index [1]. A control law is a function with respect to the state measurements or output measurements of the control system, and defines the value of input given to the control system. Optimal control has been applied to various systems, such as power systems [2, 3], robot systems [4] and aerospace systems [5]. In classical optimal control, it is required to solve a partial differential equation called Hamilton equation, of which the analytical solution is generally intractable to obtain for nonlinear systems [1, 6–9]. For this reason, near-optimal control laws have been proposed by finding approximate solutions to the corresponding Hamilton equations. For example, in [7], a neural network Hamilton-Jacobi-Bellman approach was proposed and an off-line near-optimal state feedback controller was designed for a fully known nonlinear

system. In [10], an online approximate solution was developed, based on policy iteration, for the infinite horizon optimal control of continuous-time nonlinear systems with known dynamics.

In practical applications, there always exist modeling errors for the controlled systems [11]. To cope with modeling errors, adaptive/intelligent control techniques were developed [11–13]. However, traditional adaptive control is generally far from optimal [22]. For this reason, considerable effort has been devoted to adaptive optimal control. One of the widely used adaptive optimal control methods is approximate dynamic programming. Approximate dynamic programming (ADP) is based on iterative reinforcement learning and implemented on actor-critic structures, where two coupled learning networks, called critic and actor, are tuned online to approximate the optimal value function and optimal control solution [22–29]. Modares *et al.* [22] presented an online policy iteration algorithm for the adaptive optimal control of unknown constrained-input systems. Liu *et al.* [24] extended the work in [22] and further proposed a policy iteration adaptive dynamic programming algorithm for discrete-time nonlinear systems. By developing a novel identifier-critic-based approximate dynamic programming algorithm with a dual neural network approximation structure, Lv *et al.* [29] further proposed an online adaptive optimal control for continuous-time nonlinear systems with completely unknown dynamics. As discussed above, adaptive optimal control methods based on approximate dynamic programming require at least two learning networks, which make the control structure of these methods complicated.

As a branch of optimal control, receding horizon optimal control requires minimizing a continuous-time time-varying integral-type finite-horizon cost function, which is also termed performance index, at each time instant [32]. Control laws of receding horizon optimal control can be updated discretely in time yielding discrete-time receding horizon control [33–35]. Results about adaptive discrete-time receding horizon control have also been reported, e.g., [35, 36]. A major concern of discrete-time receding horizon control is computational burden [35]. It is worth pointing out that Chen *et al.* [6] pre-

sented an explicit control law for the continuous-time receding horizon optimal control of nonlinear systems with known dynamics, which significantly reduces the computational burden.

## 1.2 Consensus of multi-agent systems

In recent years, the consensus of multi-agent systems have been widely investigated due to the widespread applications. The consensus of integrator multi-agent systems was investigated in [146, 153, 165, 172, 173, 178]. Some protocols for linear multi-agent systems were reported in [143, 154, 155, 170, 176, 177]. The consensus of nonlinear multi-agent systems is of greater significance since physical systems are more or less nonlinear. The consensus of nonlinear multi-agent systems with identical agents was investigated in [145, 169, 179, 180]. In terms of homogeneous nonlinear multi-agent systems, only a few results have been reported [149, 181, 183]. The investigations on the consensus of nonlinear multi-agent systems with unknown parameters are of great interests [139–142, 150, 174, 183]. For example, the adaptive consensus of high-order nonlinear multi-agent systems was investigated in [141, 174]. Note that the design of adaptive consensus to tackle unknown parameters is not trivial due to the distributed nature of consensus, and it is not straightforward to extend existing adaptive control methods, e.g., [166–168], for individual agents to the consensus of multi-agents. The distributed optimal consensus problem is difficult to solve since the solution of a global optimization problem generally requires centralized, i.e. global, information [171]. This problem becomes more difficult when it comes to nonlinear multi-agent systems. In [182], the optimal consensus of a nonlinear multi-agent system with respect to local performance indices is investigated via game theory. In [151], a distributed consensus protocol was proposed for a first-order nonlinear multi-agent system with uncertain nonlinear dynamics.

## 1.3 Dynamic neural networks

Dynamic neural networks are the continuous case of recurrent neural networks used in image processing, natural language processing, etc, which at its origin aim at the implementation by using analogue circuits for parallel computations. While the mathematical description about the updating of states of recurrent neural networks in image processing is given by a difference equation, for the case of dynamic neural networks, it generally becomes a ordinary differential equation. The first dynamic neural network could be the one proposed by Hopfield and Tank for solving the traveling-salesman problem [156]. Since the pioneering work of Hopfield and Tank, many results have been reported about dynamic neural networks for solving optimization problems [157, 158], control problems [160–162], and other scientific computing problems [163, 164]. Different from the recurrent neural networks used in image processing or natural language processing, dynamic neural networks generally has a high requirement on its real-time processing ability, and is thus more suitable for the control of nonlinear systems in practice.

## 1.4 Redundancy resolution of redundant manipulators

Manipulators are said to be redundant if they have more degrees of freedom (DOF) than the required to achieve a given effector primary task [184]. Due to redundancy, for a desired end-effector trajectory, there are many alternative configurations in the joint angle space of redundant manipulators. The merit of redundancy lies in the feasibility in achieving additional objectives, such as joint physical limit avoidance [185], obstacle avoidance [186], and singularity avoidance [187]. As a result, redundant manipulators have attracted considerable research interests.

In practical applications, a critical problem referred to as the redundancy resolution is to find the joint trajectories of a redundant manipulator for a given task described in

workspace subject to certain constraints [188]. To solve this problem, various methods have been investigated based on the forward kinematics of a redundant manipulator. Due to the nonlinearity and redundancy, it is difficult to directly solve this problem at the angle level [189]. Pseudoinverse-type methods [190–193] were extensively studied in the previous decades. These methods generally formulate the solution as the sum of a minimum-norm particular solution and a homogeneous solution, requiring solving the pseudoinverse of the Jacobian matrix associated with the forward kinematics of a redundant manipulator. The limitations of pseudoinverse-type methods include the difficulty to handle joint constraints [194] and the computational intensity in performing pseudoinversion [195].

To overcome the drawbacks of pseudoinverse-type methods, quadratic program (QP) based methods have been developed and widely investigated [194–200, 212, 213]. Owing to the parallel processing capability of recurrent neural networks, QP based redundancy resolution methods often use such networks to achieve efficient computation. For example, in [198], velocity-level and acceleration-level redundancy resolution schemes were unified as a QP subject to equality and inequality/bound constraints, which is then solved via a primal-dual neural network. Hou *et al.* [199] proposed a QP based method to address the coordination of two redundant manipulators and employed a dual neural network to solve the resultant QP. Chen *et al.* [200] proposed a hybrid multi-objective scheme for redundant manipulators to simultaneously achieve the end-effector primary task, joint-physical limits avoidance, obstacle avoidance, and repetitive motion.

## 1.5 Contributions and organizations

The contributions of the thesis includes the following.

1) In Chapter 2, we proposed a state-feedback based unified online learning and

near-optimal control framework for linear and nonlinear systems with parameter uncertainty. Under this framework, auxiliary systems converging to the unknown dynamics are constructed to approximate and compensate the parameter uncertainty. With the aid of the auxiliary system, future outputs of the controlled system are predicted recursively. It is also proved that the proposed learning and near-optimal control law asymptotically converges to the optimal.

2) In Chapter 3, we proposed a output-feedback based learning and near-optimal control law, to tackle the contradictory between solution accuracy and solution speed for the near-optimal control of a general class of nonlinear systems with fully unknown parameters. The key technique in the proposed learning and near-optimal control is to design an auxiliary system with the aid of the sliding mode control concept to reconstruct the dynamics of the controlled nonlinear system. Based on the sliding-mode auxiliary system and approximation of the performance index, the proposed control law guarantees asymptotic stability of the closed-system and asymptotic optimality of the performance index with time.

3) In Chapter 4, the learning and near-optimal control method is further extended to the case with fully unknown dynamics for a class of nonlinear systems.

4) In Chapter 5, the learning and near-optimal control method is further extended to address the distributed consensus of high-order nonlinear multi-agent systems consisting of heterogeneous agents. Theoretical analysis shows that the proposed protocols can simultaneously guarantee the asymptotic optimality of the performance index and the asymptotic consensus of multi-agent systems.

5) In Chapter 6, we further extend the learning and near-optimal control method to address the redundancy resolution problem of redundant manipulators with unknown physical parameters.

Conclusions and dissections on future works are provided in Chapter 7.

# Chapter 2

# LNOC via state feedback with learning of parameters

In this chapter, a unified online learning and near-optimal control framework is proposed for linear and nonlinear systems with parameter uncertainty. [1] Under this framework, auxiliary systems converging to the unknown dynamics are constructed to approximate and compensate the parameter uncertainty. With the aid of the auxiliary system, future outputs of the controlled system are predicted recursively. By utilizing a predictive time-scale approximation technique, the nonlinear dynamic programming problem for optimal control is significantly simplified and decoupled from the parameter learning dynamics: the finite-horizon integral type objective function is simplified into a quadratic one relative to the control action and there is no need to solve time-consuming Hamilton equations. Theoretical analysis shows that closed-loop systems are asymptotically stable. It is also proved that the proposed learning and near-optimal control law asymptotically converges to the optimal. The efficacy of the proposed framework and the theoretical results are validated by an application to underactuated surface vessels.

[1] The content in this chapter has already been published. Yinyan Zhang, Shuai Li, and Xiaoping Liu, "Adaptive near-optimal control of uncertain systems with application to underactuated surface vessels," *IEEE Trans. Control Syst. Technol.*, vol. 26, no. 4, pp. 1204–1218, 2018.

## 2.1 Introduction

The tracking control of underactuated surface vehicles has attracted significant attention from both academia and industry [14–21]. For example, the global tracking control of underactuated surface vehicles with fully known parameters was addressed in [14] by Lyapunov's direct method. Under perfect knowledge on the system dynamics, in [16], a predictive tracking controller was developed based on recurrent neural networks for underactuated surface vessels. To achieve the same purpose, sliding mode controllers were developed in [17–19]. In addition, a neural network based backstepping approach was proposed in [21] for the tracking control of underactuated surface vehicles with unknown system parameters. However, to the best of the authors' knowledge, there is no existing result on the learning and near-optimal or optimal tracking control of underactuated surface vessels with theoretically guaranteed stability and optimality.

This chapter aims at proposing a unified online learning and near-optimal control (LNOC) framework for uncertain linear and nonlinear systems to solve the learning and near-optimal or optimal tracking control of underactuated surface vessels via state feedback. The parameters of the proposed LNOC laws are updated in a real-time manner and there is no need to do off-line training for the parameters of the proposed LNOC laws. The proposed framework deals with the online intelligent receding horizon optimal control and possesses a simpler structure compared with the existing adaptive optimal control methods mentioned previously. Specifically, the framework proposed in this chapter mainly consists of constructing auxiliary systems and making direct approximation of the given performance indices of the controlled systems with the aid of model information obtained from the systems. In this chapter, for better readability, we first present the results on the LNOC of linear systems. Then, we extend the results to nonlinear systems with fully unknown parameters. To reduce overshooting resulted from system transition and the big estimation error in the initial stage of control, a saturation function is introduced to guarantee control performance. The deliberate design of the

auxiliary system allows the decoupling of the control part from the estimation part. It further creates an opportunity for us to integrate optimal design into the intelligent control loop and grants great enhancements in performance optimality, and tolerance to uncertainty. Theoretical analysis shows that closed-loop systems using the proposed control laws are asymptotically stable. It is also proved that the proposed near-optimal control laws asymptotically converge to the optimal. An application of the proposed near-optimal control scheme to an underactuated surface vessel validates the efficacy of the proposed framework and the theoretical results. Note that the differences between the proposed method and ADP [22–29] lie in the following points.

1) In ADP, the near-optimal control law is iteratively calculated, and neural networks are used to approximate the solution to the Hamilton equation of the optimal control problem. Different from ADP, in the proposed approach, the performance index is relaxed to a quadratic performance index, by which an analytical near-optimal control law is derived.

2) ADP generally needs actor and critic networks to guarantee near-optimality and identifier networks to handle system uncertainty. In the proposed approach, neural networks are only used to handle the uncertainty of the controlled system.

3) ADP generally requires a stable initial policy to guarantee system stability [23], which is not required in the method proposed in this chapter.

The rest of this chapter is organized as follows. In Section 2.2, some definitions, theoretical basis and assumptions are provided. In Section 2.3, we design and propose online LNOC laws for general linear systems. In Section 2.4, we further design and propose online LNOC laws for nonlinear systems. Theoretical results are presented in Section 2.5. In Section 2.6, the proposed online LNOC is applied to an underactuacted surface vessel with parameter uncertainty. Then, in Section 2.7, we conclude this chapter with final remarks.

## 2.2 Preliminary

In this section, some useful definitions and assumptions are presented.

*Definition 1*: The nonlinear system considered in this chapter is described as follows:

$$\begin{cases} \dot{\mathbf{x}} = f(\mathbf{x}) + g(\mathbf{x})\mathbf{u}(t), \\ \mathbf{y}(t) = h(\mathbf{x}), \end{cases} \tag{2.1}$$

where $\mathbf{x}(t) \in \mathbb{R}^n$, $\mathbf{u}(t) \in \mathbb{R}^m$ (with $n > m$) and $\mathbf{y}(t) \in \mathbb{R}^m$ respectively denote the state vector, the control input vector and the output vector; functions $f : \mathbb{R}^n \to \mathbb{R}^n$, $g : \mathbb{R}^n \to \mathbb{R}^{n\times m}$ and $h : \mathbb{R}^n \to \mathbb{R}^m$ are continuously differentiable.

*Definition 2* [38]: With integer $i \geq 0$, $L_f^i h(\mathbf{x})$ denotes the $i$th Lie derivative of $h(\mathbf{x})$ with respect to $f(\mathbf{x})$. Specifically, for $i = 0$, $L_f^0 h(\mathbf{x}) = h(\mathbf{x})$; for $i = 1$,

$$L_f^1 h(\mathbf{x}) = \frac{\partial h(\mathbf{x})}{\partial \mathbf{x}} f(\mathbf{x}),$$

and, for $i > 1$, $L_f^i h(\mathbf{x})$ is defined by

$$L_f^i h(\mathbf{x}) = \frac{\partial L_f^{i-1} h(\mathbf{x})}{\partial \mathbf{x}} f(\mathbf{x}).$$

Similarly, $L_g L_f^i h(\mathbf{x})$ is defined by

$$L_g L_f^i h(\mathbf{x}) = \frac{\partial L_f^i h(\mathbf{x})}{\partial \mathbf{x}} g(\mathbf{x}).$$

*Definition 3* [38]: System (2.1) is said to have a relative degree of $\rho$ in the region of interest $\mathbb{U}$ if the following properties hold true:

- $\forall \mathbf{x} \in \mathbb{U}, \forall j \in \{1, 2, \cdots, m\}$, $L_g L_f^i h_j(\mathbf{x}) = 0$, for $0 \leq i < \rho - 1$,
- $\forall \mathbf{x} \in \mathbb{U}, \forall j \in \{1, 2, \cdots, m\}$, $L_g L_f^{\rho-1} h_j(\mathbf{x}) \neq 0$.

In this chapter, the following general assumptions are imposed on system (2.1).

1) The zero dynamics of system (2.1) are stable [39].

2) All state variables of system (2.1) are available [40, 41].

3) System (2.1) has a well-defined relative degree $\rho$ [39].

4) The output $\mathbf{y}(t)$ of system (2.1) and the desired output $\mathbf{y}_{\mathrm{d}}(t)$ are $\rho$ times continuously differentiable with respect to time $t$ [39].

## 2.3 General linear systems

In this section, we present the design procedure of control laws for general linear systems and this result will be generalized to nonlinear systems in the next section.

### 2.3.1 Problem formulation

Consider the following linear system:

$$\begin{cases} \dot{\mathbf{x}}(t) = A\mathbf{x}(t) + B\mathbf{u}(t), \\ \mathbf{y}(t) = C\mathbf{x}(t), \end{cases} \tag{2.2}$$

where $\mathbf{x} \in \mathbb{R}^n$, $\mathbf{u} \in \mathbb{R}^m$ and $\mathbf{y} \in \mathbb{R}^m$ denote the state vector, input vector and output vector, respectively. $A \in \mathbb{R}^{n\times n}$, $B \in \mathbb{R}^{n\times m}$ and $C \in \mathbb{R}^{m\times n}$ are constant parameter matrices, among which the matrix $C$ is known while matrices $A$ and $B$ are unknown. In addition, since the linear system can be viewed as a special case of nonlinear system (2.1), one can calculate the relative degree $\rho$ of the linear system by Definition 2 and Definition 3. In this chapter, we consider the situation that linear system (2.2) is controllable [42]. In addition, throughout this chapter, $\mathbf{y}_{\mathrm{d}}(t)$ is used to denote the desired output.

The finite-horizon optimal control problem about system (2.2) is formulated as

$$\begin{aligned} \text{minimize}_{\mathbf{u}(t)} \quad & J_{\mathsf{l}}(t) \\ \text{subject to} \quad & \dot{\mathbf{x}}(t) = A\mathbf{x}(t) + B\mathbf{u}(t), \\ & \mathbf{y}(t) = C\mathbf{x}(t), \end{aligned} \tag{2.3}$$

where $J_{\mathsf{l}}(t)$ denotes the performance index. In this chapter, $J_{\mathsf{l}}(t)$ is given as

$$J_{\mathsf{l}}(t) = \int_0^T (\mathbf{y}_{\mathsf{d}}(t+\tau) - \mathbf{y}(t+\tau))^{\mathsf{T}} Q (\mathbf{y}_{\mathsf{d}}(t+\tau) - \mathbf{y}(t+\tau)) \mathsf{d}\tau + \int_0^T \mathbf{u}^{\mathsf{T}}(t+\tau) R \mathbf{u}(t+\tau) \mathsf{d}\tau \tag{2.4}$$

where $Q \in \mathbb{R}^{m\times m}$ and $R \in \mathbb{R}^{m\times m}$ denote symmetric and positive-definite weight matrices, and $T > 0 \in \mathbb{R}$ denotes the predictive period.

So far, we have formulated the problem as optimal control with an integral cost function over a finite time slot and equation constraints formed by system dynamics. Note that the cost function essentially is a functional with the solution $\mathbf{u}(t)$ as a function instead of a variable. In this sense, this is a functional optimization problem, the solution of which usually is equivalent to solving a Hamilton equation.

### 2.3.2 Nominal design

We first design a nominal near-optimal control law under the assumption that all parameters are known. Let $\mathbf{w} = [1, \tau, \cdots, \tau^{\rho-1}/(\rho-1)!, \tau^{\rho}/\rho!]^{\mathsf{T}}$ and $Y_{\mathsf{d}}(t) = [\mathbf{y}_{\mathsf{d}}(t), \cdots, \mathbf{y}_{\mathsf{d}}^{[\rho-1]}, \mathbf{y}_{\mathsf{d}}^{[\rho]}(t)]$. According to Taylor expansion, given that $\tau > 0$ is small, we have

$$\begin{cases} \mathbf{y}_{\mathsf{d}}(t+\tau) \approx Y_{\mathsf{d}}(t)\mathbf{w}(\tau), \\ \mathbf{u}(t+\tau) \approx \mathbf{u}(t). \end{cases} \tag{2.5}$$

Similarly, we have the following approximation of $\mathbf{y}(t+\tau)$:

$$\mathbf{y}(t+\tau) \approx \mathbf{y}(t) + \tau\dot{\mathbf{y}}(t) + \cdots + \tau^{\rho}\frac{\mathbf{y}^{[\rho]}(t)}{\rho!}. \tag{2.6}$$

For linear system (2.2) of relative $\rho$, according to Definition 2 and Definition 3, we have $CB = 0, CAB = 0, \cdots, CA^{\rho-2}B = 0, CA^{\rho-1}B \neq 0$. It follows that

$$\begin{cases} \dot{\mathbf{y}}(t) = CA\mathbf{x}(t), \\ \qquad \vdots \\ \mathbf{y}^{[\rho-1]}(t) = CA^{\rho-1}\mathbf{x}(t), \\ \mathbf{y}^{[\rho]}(t) = CA^{\rho}\mathbf{x}(t) + CA^{\rho-1}B\mathbf{u}(t), \end{cases}$$

Let $Y_{\mathsf{l}}(t) = [\mathbf{y}(t), CA\mathbf{x}(t), \cdots, CA^{\rho}\mathbf{x}(t)]$. Equation (2.6) is then written as

$$\mathbf{y}(t+\tau) \approx Y_{\mathsf{l}}(t)\mathbf{w}(\tau) + \frac{\tau^{\rho}}{\rho!}CA^{\rho-1}B\mathbf{u}(t). \tag{2.7}$$

Substituting equations (2.5) and (2.7) into equation (2.4) yields

$$\begin{aligned} J_{\mathsf{l}}(t) \approx & \int_0^T (E_{\mathsf{l}}(t)\mathbf{w}(\tau) - \frac{\tau^{\rho}}{\rho!}CA^{\rho-1}B\mathbf{u}(t))^{\mathsf{T}}Q(E_{\mathsf{l}}(t)\mathbf{w}(\tau) - \frac{\tau^{\rho}}{\rho!}CA^{\rho-1}B\mathbf{u}(t)) \\ & + \int_0^T \mathbf{u}^{\mathsf{T}}(t)R\mathbf{u}(t)\mathsf{d}\tau \\ = & \int_0^T \mathbf{w}^{\mathsf{T}}(\tau)E_{\mathsf{l}}^{\mathsf{T}}(t)QE(t)\mathbf{w}(\tau)\mathsf{d}\tau - 2\int_0^T \frac{\tau^{\rho}}{\rho!}\mathbf{w}^{\mathsf{T}}(\tau)\mathsf{d}\tau E_{\mathsf{l}}^{\mathsf{T}}(t)CA^{\rho-1}B\mathbf{u}(t) \\ & + \int_0^T \frac{\tau^{2\rho}}{(\rho!)^2}\mathsf{d}\tau\mathbf{u}^{\mathsf{T}}(t)(CA^{\rho-1}B)^{\mathsf{T}}QCA^{\rho-1}B\mathbf{u}(t) + T\mathbf{u}^{\mathsf{T}}(t)R\mathbf{u}(t), \\ = & \hat{J}_{\mathsf{l}}(t), \end{aligned}$$

where $E_{\mathsf{l}}(t) = Y_{\mathsf{d}}(t) - Y_{\mathsf{l}}(t)$, and $\hat{J}(t)$ denotes the approximation of $J(t)$. Let

$$\mathbf{v} = \int_0^T \frac{\tau^{\rho}}{\rho!}\mathbf{w}^{\mathsf{T}}(\tau)\mathsf{d}\tau = \left[\frac{T^{\rho+1}}{(\rho+1)\rho!}, \frac{T^{\rho+2}}{(\rho+2)\rho!1!}, \cdots, \frac{T^{2\rho+1}}{(2\rho+1)(\rho!)^2}\right],$$

and

$$\kappa = \int_0^T \frac{\tau^{2\rho}}{(\rho!)^2}\mathsf{d}\tau = \frac{T^{2\rho+1}}{(2\rho+1)(\rho!)^2}.$$

Since the decision variable is $\mathbf{u}(t)$, minimizing $\hat{J}(t)$ is equivalent to minimizing

$$J_{\mathsf{e}}(t) = \mathbf{u}^{\mathsf{T}}(t)\Theta\mathbf{u}(t) + \mathbf{p}^{\mathsf{T}}(t)\mathbf{u}(t),$$

where $\Theta = TR + \kappa(CA^{\rho-1}B)^{\mathsf{T}}QCA^{\rho-1}B$ and $\mathbf{p} = (-2\mathbf{v}E_{\mathsf{l}}^{\mathsf{T}}QCA^{\rho-1}B)^{\mathsf{T}}$. Given that $\Theta$ is positive-definite, $J_{\mathsf{e}}(t)$ is convex and $\mathbf{u}(t)$ is thus obtained by solving $\partial J_{\mathsf{e}}(t)/\partial\mathbf{u} = 0$,

which gives

$$\mathbf{u}(t) = (TR + \kappa(CA^{\rho-1}B)^{\mathsf{T}}QCA^{\rho-1}B)^{-1}(QCA^{\rho-1}B)^{\mathsf{T}}(Y_{\mathsf{d}}(t) - Y_{\mathsf{l}}(t))\mathbf{v}^{\mathsf{T}}. \tag{2.8}$$

This control law requires the full knowledge of the system parameters and is thus less satisfactory in practical applications with uncertainties.

For comparison, a traditional solution for the finite-horizon optimal control problem shown in (2.3) is presented as follows [43]:

$$\mathbf{u}(t) = -R^{-1}(t)B^{\mathsf{T}}(P(t)\mathbf{x}(t) + \mathbf{s}(t)), \tag{2.9}$$

where matrix $P$ is the solution of the following differential Riccati equation:

$$-\dot{P} = P(t)A + A^{\mathsf{T}}P(t) - P(t)BR^{-1}B^{\mathsf{T}}P(t) + Q',$$

with $P(t+T) = 0$ and $Q' = C^{\mathsf{T}}QC$. Besides, $\mathbf{s}(t)$ is determined by the following differential equation [43]:

$$-\dot{\mathbf{s}} = (A - BR^{-1}B^{\mathsf{T}}P)^{\mathsf{T}}\mathbf{s}(t) + Q'C^{\mathsf{T}}(CC^{\mathsf{T}})\mathbf{x}_{\mathsf{d}}(t)$$

with $C\mathbf{x}_{\mathsf{d}}(t) = \mathbf{y}_{\mathsf{d}}(t)$ and $\mathbf{s}(t+T) = 0$. Evidently, compared with traditional solution (2.9), near-optimal control law (2.8) does not need to solve any differential equation.

### 2.3.3 Intelligent design

In this subsection, we design an online LNOC law for handling the situation when parameter matrices $A$ and $B$ are unknown.

To tackle the uncertainty of parameter matrices $A$ and $B$, a linear auxiliary system is constructed as follows:

$$\dot{\hat{\mathbf{x}}}(t) = \hat{A}\mathbf{x}(t) + \hat{B}\mathbf{u}(t) - K_{\mathsf{x}}(\hat{\mathbf{x}}(t) - \mathbf{x}(t)), \tag{2.10}$$

where $\hat{A} \in \mathbb{R}^{n\times n}$, $\hat{B} \in \mathbb{R}^{n\times m}$, $\hat{\mathbf{x}} \in \mathbb{R}^n$ is an auxiliary state vector, the diagonal matrix $K_{\mathsf{x}} = \mathsf{diag}([k_{x_1}, k_{x_2}, \cdots, k_{x_n}]) \in \mathbb{R}^{n\times n}$ is a positive-definite gain matrix, and the others

are defined as the aforementioned. It is desired that linear auxiliary system (2.10) possesses the same dynamic behavior as linear system (2.2). To achieve such an objective, we define the following evolutions for matrices $\hat{A}$ and $\hat{B}$:

$$\begin{cases} \dot{\hat{A}} = -K_{\mathsf{A}}(\hat{\mathbf{x}}(t) - \mathbf{x}(t))\mathbf{x}^{\mathsf{T}}(t), \\ \dot{\hat{B}} = -K_{\mathsf{B}}(\hat{\mathbf{x}}(t) - \mathbf{x}(t))\mathbf{u}^{\mathsf{T}}(t), \end{cases} \tag{2.11}$$

where $K_{\mathsf{A}} = \mathsf{diag}([k_{A_1}, k_{A_2}, \cdots, k_{A_n}]) \in \mathbb{R}^{n\times n}$ and $K_{\mathsf{B}} = \mathsf{diag}([k_{B_1}, k_{B_2}, \cdots, k_{B_n}]) \in \mathbb{R}^{n\times n}$ are positive-definite gain matrices. The whole linear auxiliary system is thus formulated as

$$\begin{cases} \dot{\hat{\mathbf{x}}}(t) = \hat{A}\mathbf{x}(t) + \hat{B}\mathbf{u}(t) - K_{\mathsf{x}}(\hat{\mathbf{x}}(t) - \mathbf{x}(t)), \\ \dot{\hat{A}} = -K_{\mathsf{A}}(\hat{\mathbf{x}}(t) - \mathbf{x}(t))\mathbf{x}^{\mathsf{T}}(t), \\ \dot{\hat{B}} = -K_{\mathsf{B}}(\hat{\mathbf{x}}(t) - \mathbf{x}(t))\mathbf{u}^{\mathsf{T}}(t). \end{cases} \tag{2.12}$$

Gain matrices in linear auxiliary system (2.12) are chosen in the following manner. The values of the diagonal elements in $K_{\mathsf{x}}$ are much smaller than those in $K_{\mathsf{A}}$ and $K_{\mathsf{B}}$ to guarantee that $\hat{A}$ and $\hat{B}$ take a dominant role in learning the dynamics of the controlled system. In practice, we suggest to take values of the diagonal elements in $K_{\mathsf{x}}$ at least 10 times smaller than those in $K_{\mathsf{A}}$ and $K_{\mathsf{B}}$. Under this condition, with larger values of the diagonal elements in the gain matrices, the convergence of $\hat{\mathbf{x}}(t)$ to $\mathbf{x}(t)$ is faster. Note that very large values in the gain matrices may lead to high overshooting. In this sense, they cannot be too large.

When $\hat{\mathbf{x}}(t) = \mathbf{x}(t)$, which means that the linear auxiliary system reconstructs the states and dynamics of the original linear system (2.2), linear auxiliary system (2.10) becomes

$$\dot{\mathbf{x}}(t) = \hat{A}\mathbf{x}(t) + \hat{B}\mathbf{u}(t).$$

Then, following similar steps in the previous subsection, the online LNOC law for linear system (2.2) is thus obtained as follows:

$$\mathbf{u}_{\mathsf{a}}(t) = (TR + \kappa(C\hat{A}^{\rho-1}\hat{B})^{\mathsf{T}}QC\hat{A}^{\rho-1}\hat{B})^{-1}(QC\hat{A}^{\rho-1}\hat{B})^{\mathsf{T}}(Y_{\mathsf{d}}(t) - Y_{\mathsf{l}}(t))\mathbf{v}^{\mathsf{T}}. \tag{2.13}$$

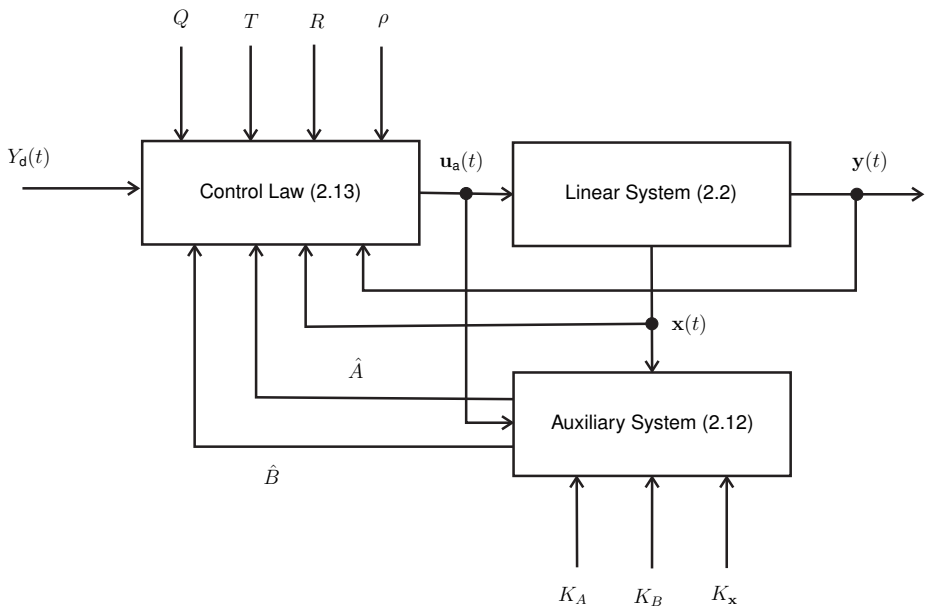



Figure 2.1: Block diagram of linear system (2.2) with parameter uncertainty synthesized by online LNOC law (2.13) and linear auxiliary system (2.12).

The block diagram of linear system (2.2) synthesized by online LNOC law (2.13) and linear auxiliary system (2.12) is shown in Fig. 2.1. As seen from this figure, by utilizing state information and input information of linear system (2.2), auxiliary system (2.12) estimates the parameter matrices of linear system (2.2). The parameter matrices generated by auxiliary system (2.12) are passed to control law (2.13), which yields LNOC action to linear system (2.2).

## 2.4 Extension to nonlinear systems

Based on the previous result on linear systems, in this section, we present the result about online LNOC of nonlinear systems.

### 2.4.1 Problem formulation

The finite-horizon optimal control problem about nonlinear system (2.1) is formulated as

$$\begin{aligned} \text{minimize}_{\mathbf{u}(t)} \quad & J_{\mathsf{n}}(t) \\ \text{subject to} \quad & \dot{\mathbf{x}} = f(\mathbf{x}) + g(\mathbf{x})\mathbf{u}(t), \\ & \mathbf{y}(t) = h(\mathbf{x}), \end{aligned} \tag{2.14}$$

where $J_{\mathsf{n}}(t)$ denotes the performance index and is given as

$$\begin{aligned} J_{\mathsf{n}}(t) = & \int_0^T (\mathbf{y}_{\mathsf{d}}(t+\tau) - \mathbf{y}(t+\tau))^{\mathsf{T}} Q (\mathbf{y}_{\mathsf{d}}(t+\tau) - \mathbf{y}(t+\tau)) \mathsf{d}\tau \\ & + \int_0^T \mathbf{u}^{\mathsf{T}}(t+\tau) R \mathbf{u}(t+\tau) \mathsf{d}\tau. \end{aligned} \tag{2.15}$$

Compared with linear system (2.2), the finite-horizon optimal control problem of nonlinear system (2.1) possesses the same performance index but different dynamic constraints. In the next subsection, the similarity between the two optimal control problems is utilized to solve the finite-horizon optimal control problem of nonlinear system (2.1).

*Remark 1:* Directly solving finite-horizon optimal control problem (2.14) requires solving a partial differential equation known as Hamilton equation, of which the analytical solution is generally difficult to obtain. Therefore, an approximation is made in the following design process, which relaxes the optimal control problem at the cost of optimality, and explicit online near-optimal control laws are obtained.

### 2.4.2 Nominal design

In this subsection, the design process of a near-optimal nominal control law for nonlinear system (2.1) is presented under the assumption that all the parameters of nonlinear system (2.1) are known.

Similar to the case of linear system (2.2), one has

$$\begin{cases} \mathbf{y}_{\mathsf{d}}(t+\tau) \approx Y_{\mathsf{d}}(t)\mathbf{w}(\tau), \\ \mathbf{u}(t+\tau) \approx \mathbf{u}(t), \\ \mathbf{y}(t+\tau) \approx \mathbf{y}(t) + \tau\dot{\mathbf{y}}(t) + \cdots + \tau^{\rho}\dfrac{\mathbf{y}^{[\rho]}(t)}{\rho!}. \end{cases}$$

For nonlinear system (2.1) with relative degree $\rho$, based on Definition 2 and Definition 3, one has

$$\begin{cases} \dot{\mathbf{y}}(t) = \dfrac{\partial h}{\partial \mathbf{x}}\dot{\mathbf{x}} = \dfrac{\partial h}{\partial \mathbf{x}} f(\mathbf{x}) = L_f h(\mathbf{x}), \\ \qquad\qquad \vdots \\ \mathbf{y}^{[\rho-1]}(t) = L_f^{\rho-1} h(\mathbf{x}), \\ \mathbf{y}^{[\rho]}(t) = L_f^{\rho} h(\mathbf{x}) + \dfrac{\partial L_f^{\rho-1} h(\mathbf{x})}{\partial \mathbf{x}} g(\mathbf{x})\mathbf{u}(t) \\ \qquad\quad = L_f^{\rho} h(\mathbf{x}) + L_g L_f^{\rho-1} h(\mathbf{x})\mathbf{u}(t). \end{cases} \tag{2.16}$$

Let $Y_{\mathsf{n}}(t) = [\mathbf{y}(t), L_f h(\mathbf{x}), \cdots, L_f^{\rho-1} h(\mathbf{x}), \cdots, L_f^{\rho} h(\mathbf{x})]$. Then, $\mathbf{y}(t+\tau)$ of nonlinear system (2.1) is approximated as

$$\mathbf{y}(t+\tau) \approx Y_{\mathsf{n}}(t)\mathbf{w}(\tau) + \frac{\tau^{\rho}}{\rho!} L_g L_f^{\rho-1} h(\mathbf{x})\mathbf{u}(t).$$

Performance index $J_{\mathsf{n}}(t)$ shown in equation (2.15) is thus approximated as

$$\begin{aligned} J_{\mathsf{n}}(t) &\approx \hat{J}_{\mathsf{n}}(t) \\ &= \int_0^T \Big(E_{\mathsf{n}}(t)\mathbf{w}(\tau) - \frac{\tau^{\rho}}{\rho!} L_g L_f^{\rho-1} h(\mathbf{x})\mathbf{u}(t)\Big)^{\mathsf{T}} Q \Big(E_{\mathsf{n}}(t)\mathbf{w}(\tau) - \frac{\tau^{\rho}}{\rho!} L_g L_f^{\rho-1} h(\mathbf{x})\mathbf{u}(t)\Big)\mathsf{d}\tau \\ &\quad + T\mathbf{u}^{\mathsf{T}}(t) R \mathbf{u}(t) \\ &= \int_0^T \mathbf{w}^{\mathsf{T}}(\tau) E_{\mathsf{n}}^{\mathsf{T}}(t) Q E_{\mathsf{n}}(t)\mathbf{w}(\tau)\mathsf{d}\tau - 2\int_0^T \frac{\tau^{\rho}}{\rho!}\mathbf{w}^{\mathsf{T}}(\tau)\mathsf{d}\tau E_{\mathsf{n}}^{\mathsf{T}}(t) Q L_g L_f^{\rho-1} h(\mathbf{x})\mathbf{u}(t) \\ &\quad + \int_0^T \frac{\tau^{2\rho}}{(\rho!)^2}\mathsf{d}\tau \mathbf{u}^{\mathsf{T}}(t)(L_g L_f^{\rho-1} h(\mathbf{x}))^{\mathsf{T}} Q L_g L_f^{\rho-1} h(\mathbf{x})\mathbf{u}(t) + T\mathbf{u}^{\mathsf{T}}(t) R \mathbf{u}(t), \end{aligned} \tag{2.17}$$

where $E_{\mathsf{n}} = Y_{\mathsf{d}}(t) - Y_{\mathsf{n}}(t)$. Similar to the control law design for linear system (2.2), since the decision variable is $\mathbf{u}(t)$, minimizing performance index $\hat{J}_{\mathsf{n}}(t)$ is equivalent to

minimizing the following quadratic performance index:

$$\Psi_{\mathsf{n}}(t) = \mathbf{u}^{\mathsf{T}}(t)\Theta_{\mathsf{n}}\mathbf{u}(t) + \mathbf{p}_{\mathsf{n}}^{\mathsf{T}}\mathbf{u}(t), \tag{2.18}$$

where $\Theta_{\mathsf{n}} = TR + \kappa(L_g L_f^{\rho-1}h(\mathbf{x}))^{\mathsf{T}} Q L_g L_f^{\rho-1}h(\mathbf{x})$ and $\mathbf{p}_{\mathsf{n}} = -2(L_g L_f^{\rho-1}h(\mathbf{x}))^{\mathsf{T}} Q^{\mathsf{T}} E_{\mathsf{n}} \mathbf{v}^{\mathsf{T}}$. Given that $\Theta_{\mathsf{n}}$ is positive-definite, performance index $\Psi_{\mathsf{n}}$ shown in (2.18) is convex and the optimal solution is thus obtained by solving $\partial\Psi_{\mathsf{n}}(t)/\partial\mathbf{u} = 0$, which gives the following nominal near-optimal control law for nonlinear system (2.1):

$$\mathbf{u}(t) = (TR + \kappa(L_g L_f^{\rho-1}h(\mathbf{x}))^{\mathsf{T}} Q L_g L_f^{\rho-1}h(\mathbf{x}))^{-1}(Q L_g L_f^{\rho-1}h(\mathbf{x}))^{\mathsf{T}}(Y_{\mathsf{d}}(t) - Y_{\mathsf{n}}(t))\mathbf{v}^{\mathsf{T}}, \tag{2.19}$$

which can only be used under the assumption that the dynamics of the controlled systems are fully known.

### 2.4.3 Intelligent design

In this subsection, we design online LNOC laws for nonlinear system (2.1) under the condition of parameter uncertainty.

Since $f(\mathbf{x})$ and $g(\mathbf{x})$ are smooth, the following parameterized network is adopted to reconstruct them:

$$\begin{cases} f(\mathbf{x}) = W_f \phi_f(\mathbf{x}), \\ g(\mathbf{x}) = W_g \phi_g(\mathbf{x}), \end{cases}$$

where $W_f \in \mathbb{R}^{n\times N_f}$ and $W_g \in \mathbb{R}^{n\times N_g}$ denote the unknown constant weight matrices; $\phi_f(\cdot) : \mathbb{R}^n \to \mathbb{R}^{N_f}$ and $\phi_g(\cdot) : \mathbb{R}^n \to \mathbb{R}^{N_g\times m}$ denote suitable basis functions; $N_f$ and $N_g$ denote the numbers of neurons. Therefore, nonlinear system (2.1) can be rewritten as the following parameterized system:

$$\begin{cases} \dot{\mathbf{x}}(t) = W_f \phi_f(\mathbf{x}) + W_g \phi_g(\mathbf{x})\mathbf{u}(t), \\ \mathbf{y}(t) = h(\mathbf{x}). \end{cases} \tag{2.20}$$

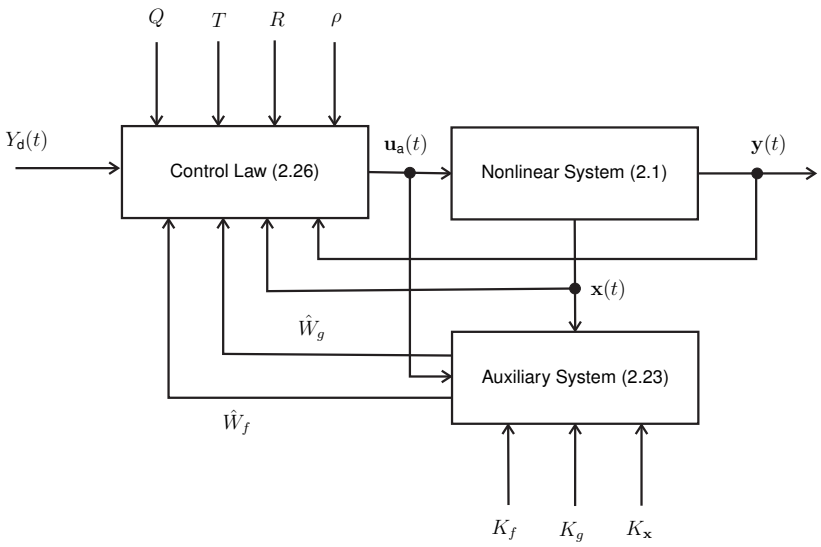



Figure 2.2: Block diagram of nonlinear system (2.1) with parameter uncertainty synthesized by online LNOC law (2.26) and auxiliary system (2.23).

In this sense, the parameter uncertainty of nonlinear system (2.1) corresponds to the uncertainty of weight matrices $W_f$ and $W_g$. Based on parameterized system (2.20), we design an LNOC law for nonlinear system (2.1) under parameter uncertainty.

Similar to the linear system case, we design the following nonlinear auxiliary system to reconstruct the states and dynamics of nonlinear system (2.20):

$$\dot{\hat{\mathbf{x}}}(t) = \hat{W}_f \phi_f(\mathbf{x}) + \hat{W}_g \phi_g(\mathbf{x}) \mathbf{u}(t) - K_{\mathbf{x}}(\hat{\mathbf{x}}(t) - \mathbf{x}(t)), \tag{2.21}$$

where parameter matrices $\hat{W}_f$ and $\hat{W}_g$ need to be obtained with $K_{\mathbf{x}}$ defined as the aforementioned. To drive $\hat{\mathbf{x}} - \mathbf{x}$ to zero, the following evolutions for matrices $\hat{W}_f$ and $\hat{W}_g$ are defined:

$$\begin{cases} \dot{\hat{W}}_f = -K_f(\hat{\mathbf{x}}(t) - \mathbf{x}(t))\phi_f^{\mathsf{T}}(\mathbf{x}), \\ \dot{\hat{W}}_g = -K_g(\hat{\mathbf{x}}(t) - \mathbf{x}(t))\mathbf{u}^{\mathsf{T}}(t)\phi_g^{\mathsf{T}}(\mathbf{x}), \end{cases} \tag{2.22}$$

where $K_f = \mathrm{diag}([k_{f_1}, k_{f_2}, \cdots, k_{f_n}]) \in \mathbb{R}^{n \times n}$ and $K_g = \mathrm{diag}([k_{g_1}, k_{g_2}, \cdots, k_{g_n}]) \in \mathbb{R}^{n \times n}$ are positive-definite gain matrices to scale the convergence. Therefore, the whole

nonlinear auxiliary system is

$$\begin{cases} \dot{\hat{\mathbf{x}}}(t) = \hat{W}_f \phi_f(\mathbf{x}) + \hat{W}_g \phi_g(\mathbf{x})\mathbf{u}(t) - K_{\mathbf{x}}(\hat{\mathbf{x}}(t) - \mathbf{x}(t)), \\ \dot{\hat{W}}_f = -K_f(\hat{\mathbf{x}}(t) - \mathbf{x}(t))\phi_f^{\mathsf{T}}(\mathbf{x}), \\ \dot{\hat{W}}_g = -K_g(\hat{\mathbf{x}}(t) - \mathbf{x}(t))\mathbf{u}^{\mathsf{T}}(t)\phi_g^{\mathsf{T}}(\mathbf{x}). \end{cases} \tag{2.23}$$

Note that the values of gain matrices $K_{\mathbf{x}}$, $K_f$, and $K_g$ in nonlinear auxiliary system (2.23) are similarly chosen as those in linear auxiliary system (2.12).

**For special case when** $\hat{\mathbf{x}}(t) = \mathbf{x}(t)$: When nonlinear auxiliary system (2.21) has reconstructed the states of nonlinear system (2.20), i.e., $\hat{\mathbf{x}}(t) = \mathbf{x}(t)$, nonlinear system (2.1) with parameter uncertainty becomes

$$\begin{cases} \dot{\mathbf{x}}(t) = \hat{W}_f \phi_f(\mathbf{x}) + \hat{W}_g \phi_g(\mathbf{x})\mathbf{u}(t), \\ \mathbf{y}(t) = h(\mathbf{x}). \end{cases} \tag{2.24}$$

We perform the controller design based on (2.24). Let $\hat{f}(\mathbf{x}) = \hat{W}_f \phi_f(\mathbf{x})$ and $\hat{g}(\mathbf{x}) = \hat{W}_g \phi_g(\mathbf{x})$. Given that the relative degree $\rho$ of nonlinear system (2.1) with parameter uncertainty is a prior knowledge, we thus have

$$\begin{cases} \dot{\mathbf{y}}(t) = \dfrac{\partial h}{\partial \mathbf{x}} \hat{f}(\mathbf{x}) = L_{\hat{f}} h(\mathbf{x}), \\ \qquad\qquad \vdots \\ \mathbf{y}^{[\rho-1]}(t) = L_{\hat{f}}^{\rho-1} h(\mathbf{x}), \\ \mathbf{y}^{[\rho]}(t) = L_{\hat{f}}^{\rho} h(\mathbf{x}) + \dfrac{\partial L_{\hat{f}}^{\rho-1} h(\mathbf{x})}{\partial \mathbf{x}} \hat{g}(\mathbf{x})\mathbf{u}(t) \\ \qquad\quad = L_{\hat{f}}^{\rho} h(\mathbf{x}) + L_{\hat{g}} L_{\hat{f}}^{\rho-1} h(\mathbf{x})\mathbf{u}(t). \end{cases} \tag{2.25}$$

Let $\hat{Y}_{\mathsf{n}}(t) = [\mathbf{y}(t), L_{\hat{f}} h(\mathbf{x}), \cdots, L_{\hat{f}}^{\rho} h(\mathbf{x})]$. Then, following similar steps in Section 2.4.2, based on system (2.24), an online LNOC law is obtained as follows for nonlinear system (2.1) with parameter uncertainty:

$$\mathbf{u}_{\mathsf{a}}(t) = (TR + \kappa (L_{\hat{g}} L_{\hat{f}}^{\rho-1} h(\mathbf{x}))^{\mathsf{T}} Q L_{\hat{g}} L_{\hat{f}}^{\rho-1} h(\mathbf{x}))^{-1} (Q L_{\hat{g}} L_{\hat{f}}^{\rho-1} h(\mathbf{x}))^{\mathsf{T}} (Y_{\mathsf{d}}(t) - \hat{Y}_{\mathsf{n}}(t)) \mathbf{v}^{\mathsf{T}}. \tag{2.26}$$

The block diagram of nonlinear system (2.1) with parameter uncertainty synthesized by LNOC law (2.26) and nonlinear auxiliary system (2.23) is shown in Fig. 2.2.

**For general situation without** $\hat{\mathbf{x}}(t) = \mathbf{x}(t)$: For the general situation, nonlinear system (2.1) with parameter uncertainty is represented as

$$\begin{cases} \dot{\mathbf{x}}(t) = \hat{W}_f \phi_f(\mathbf{x}) + \hat{W}_g \phi_g(\mathbf{x})\mathbf{u}(t) - (\tilde{W}_f \phi_f(\mathbf{x}) \\ \qquad + \tilde{W}_g \phi_g(\mathbf{x})\mathbf{u}(t)), \\ \mathbf{y}(t) = h(\mathbf{x}), \end{cases} \tag{2.27}$$

where $\tilde{W}_f = \hat{W}_f - W_f$ and $\tilde{W}_g = \hat{W}_g - W_g$. Given that the relative degree $\rho$ of nonlinear system (2.1) with parameter uncertainty is a prior knowledge, we thus have

$$\begin{cases} \dot{\mathbf{y}}(t) = L_{\hat{f}} h(\mathbf{x}) - L_{\tilde{f}} h(\mathbf{x}), \\ \qquad\qquad \vdots \\ \mathbf{y}^{[\rho-1]}(t) = L_{\hat{f}}^{\rho-1} h(\mathbf{x}) - L_{\tilde{f}}^{\rho-1} h(\mathbf{x}), \\ \mathbf{y}^{[\rho]}(t) = L_{\hat{f}}^{\rho} h(\mathbf{x}) + L_{\hat{g}} L_{\hat{f}}^{\rho-1} h(\mathbf{x})\mathbf{u}(t) - (L_{\hat{f}}^{\rho} h(\mathbf{x}) \\ \qquad + L_{\tilde{g}} L_{\tilde{f}}^{\rho-1} h(\mathbf{x})\mathbf{u}(t)), \end{cases} \tag{2.28}$$

where $L_{\tilde{f}}^{i} h(\mathbf{x}) = L_{\hat{f}}^{i} h(\mathbf{x}) - L_{f}^{i} h(\mathbf{x})$. Since $W_g$ and $W_f$ are unknown, control design cannot be done based on (2.28).

*Remark 2:* In practical applications, if a good prior knowledge about the values of all the elements of parameter matrices $W_f$ and $W_g$ of a controlled nonlinear system is available, which means that all the initial values of the elements of $\hat{W}_f$ and $\hat{W}_g$ are set to be in a small neighborhood of the corresponding values of the elements of $W_f$ and $W_g$, then controller (2.26) is a good choice since the difference between controller (2.26) and nominal controller (2.19) is small. Without a good prior knowledge, the difference may be extremely large before auxiliary system (2.23) fully reconstructs the states and dynamics of nonlinear system (2.1), which may lead to extremely large magnitude of adaptive control input $\mathbf{u}_{\mathrm{a}}(t)$. It is worth pointing out that the magnitude of the control

input that a practical system can bear is limited. To avoid too large magnitude of control input $\mathbf{u}_{\mathsf{a}}(t)$ generated by LNOC law (2.26), the following modified online LNOC law can be adopted:

$$\mathbf{u}_{\mathsf{s}}(t) = \mathsf{sat}_{\beta}(\mathbf{u}_{\mathsf{a}}(t)), \tag{2.29}$$

where $\beta = [\beta_1, \beta_2, \cdots, \beta_m]^{\mathsf{T}} \in \mathbb{R}^m$ is the bound of saturation with $\beta_j > 0$ for all $j = 1, 2, \cdots, m$. The $j$th element of $\mathsf{sat}_{\beta}(\mathbf{u}_{\mathsf{a}}(t))$ is defined as follows:

$$\mathsf{sat}_{\beta_j}(u_{\mathsf{a}j}(t)) = \begin{cases} \beta_j, & \text{if } u_{\mathsf{a}j}(t) > \beta_j, \\ u_{\mathsf{a}j}(t), & \text{if } -\beta_j \leq u_{\mathsf{a}j}(t) \leq \beta_j, \\ -\beta_j, & \text{if } u_{\mathsf{a}j}(t) < -\beta_j. \end{cases}$$

This approach can also be adopted to deal with the LNOC of linear system (2.2) to avoid very large magnitude of control inputs.

*Remark 3:* In terms of control structure complexity and computational burden, the proposed LNOC is compared with ADP [22–29] as follows.

1) For the LNOC, ADP generally needs actor and critic networks to guarantee near-optimality and identifier networks to handle system uncertainty. In the proposed approach, there is only one network associated with the auxiliary system to handle system uncertainty. In this sense, compared with ADP, the proposed LNOC has a simpler structure.

2) Corresponding to the control structure, ADP needs time to train the weights of actor, critic, and identifier networks. The values of the so-called value function and the control law are iteratively calculated based on the weights of the networks. Different from ADP, the proposed LNOC does not need the training of the weights of the networks. Besides, the proposed control law is presented in an analytical form, of which the parameters are directly obtained from the auxiliary system.

### 2.4.4 Computational complexity analysis

Since nonlinear systems are more complicated than linear systems, in this subsection, we analyze the computational complexity of the proposed LNOC method when it is applied to nonlinear systems. In terms of the linear system case, the analysis on the computational complexity can be conducted in a similar manner, and is thus omitted.

Before analyzing the computational complexity, we define a floating-point operation as one addition, subtraction, multiplication, or division of two floating-point numbers and recall the following facts [45].

1) The multiplication of a scalar and a vector of size $s_1$ requires $s_1$ floating-point operations.

2) The multiplication of a matrix and a vector, one of size $s_1 \times s_2$ and the other of size $s_2$, requires $s_1(2s_2 - 1)$ floating-point operations.

3) The multiplication of two matrices, one of size $s_1 \times s_2$ and the other of size $s_2 \times s_3$, requires $2s_1s_2s_3 - s_1s_3$ floating-point operations.

4) The addition or subtraction of two vectors of size $s_1$ requires $s_1$ floating-point operations.

5) The inversion of a square matrix of size $s_1 \times s_1$ requires $s_1^3$ floating-point operations.

The analysis on the computational complexity of the nonlinear auxiliary system (2.23) is conducted by firstly discretizing it. By using the Euler difference formula [46], the

discrete-time nonlinear auxiliary system can be obtained as follows:

$$\begin{cases} \hat{\mathbf{x}}^{k+1} = \hat{\mathbf{x}}^k + \tau(\hat{W}_f^k \phi_f(\mathbf{x}^k) + \hat{W}_g^k \phi_g(\mathbf{x}^k)\mathbf{u}^k \\ \qquad\qquad - K_{\mathbf{x}}(\hat{\mathbf{x}}^k - \mathbf{x}^k)), \\ \hat{W}_f^{k+1} = \hat{W}_f^k - \tau(K_f^k(\hat{\mathbf{x}}^k - \mathbf{x}^k)\phi_f^{\mathsf{T}}(\mathbf{x}^k)), \\ \hat{W}_g^{k+1} = \hat{W}_g^k - \tau(K_g^k(\hat{\mathbf{x}}^k - \mathbf{x}^k)\mathbf{u}^{k\mathsf{T}}\phi_g^{\mathsf{T}}(\mathbf{x}^k)), \end{cases}$$

where $\tau > 0 \in \mathbb{R}$ is the sampling period and $k = 1, 2, \cdots$ is the updating index. To utilize the proposed LNOC method, at the $k$th sampling time instant the values of $\hat{\mathbf{x}}^{k+1}$, $\hat{W}_f^{k+1}$, and $\hat{W}_g^{k+1}$ need to be calculated. Let $c_{\max}$ denotes the maximal number of floating-point operations needed for calculating the value of an element in vector $\phi_f(\mathbf{x}^k) \in \mathbb{R}^{N_f}$ or matrix $\phi_g(\mathbf{x}^k) \in \mathbb{R}^{N_g \times m}$. Based on the above facts and definitions, at the $k$th sampling time instant, computing $\hat{\mathbf{x}}^{k+1}$ requires less than $mn(1 + 2N_g) + 2n(N_f + n) + c_{\max}(n + mN_g)$ floating-point operations; computing $\hat{W}_f^{k+1}$ requires less than $mn + 2n^2 + N_f(2n + c_{\max})$ floating-point operations; computing $\hat{W}_g^{k+1}$ requires less than $mn(1 + 2N_g) + 2n^2 + nN_g + mN_g c_{\max}$ floating-point operations. In total, the nonlinear auxiliary system requires less than $mn(3 + 4N_g) + 4nN_f + 6n^2 + c_{\max}(n + 2mN_g + N_f) + nN_g$ floating-point operations at each sampling time instant.

Consider the proposed LNOC law (2.26), for which $TR \in \mathbb{R}^{m \times m}$, $\mathbf{v}^{\mathsf{T}} \in \mathbb{R}^{\rho+1}$, $Q \in \mathbb{R}^{m \times m}$, $\kappa \in \mathbb{R}$ are constant and do not need to be calculated at each sampling time instant; $Y_{\mathsf{d}}(t) \in \mathbb{R}^{m \times (\rho+1)}$ is calculated beforehand; $L_{\hat{g}} L_{\hat{f}}^{\rho-1} h(\mathbf{x}) \in \mathbb{R}^{m \times m}$ and $\hat{Y}_{\mathsf{n}}(t) \in \mathbb{R}^{m \times (\rho+1)}$ need to be updated at each sampling time instant. Let $c'_{\max}$ denotes the maximal number of floating-point operations needed for calculating the value of an element in matrices $L_{\hat{g}} L_{\hat{f}}^{\rho-1} h(\mathbf{x})$ or $\hat{Y}_{\mathsf{n}}(t)$. Therefore, based on (2.26), computing $\mathbf{u}_{\mathsf{a}}(t)$ for one time requires less than $9m^3 + (1 + 2\rho)m^2 + 2m\rho + m + c'_{\max}(m^2 + m\rho + m)$ floating-point operations.

The computational complexity of the proposed LNOC method consist of two parts: updating the auxiliary system and computing the LNOC law based on the parameter matrices generated by the auxiliary system. Therefore, the total computational complexity

of the proposed LNOC method for nonlinear system (2.1) is the sum of the two parts. It follows that the proposed LNOC method for nonlinear system (2.1) totally requires less than $mn(3+4N_g)+4nN_f+6n^2+c_{\max}(n+2mN_g+N_f)+nN_g+9m^3+(1+2\rho)m^2+2m\rho+m+c'_{\max}(m^2+m\rho+m)$ floating-point operations per updating. It is worth pointing out that the number of the required floating-point operations can be reduced by improving the implementation of the proposed control method. For example, we can introduce a variable to store the value of $\hat{\mathbf{x}}^k-\mathbf{x}^k$ so as to avoid repeatedly computing $\hat{\mathbf{x}}^k-\mathbf{x}^k$ when computing $\hat{W}_f^{k+1}$ or $\hat{W}_g^{k+1}$ at the same time instant for updating.

## 2.5 Theoretical results

In this section, considering that linear system (2.2) can be viewed as a special case of nonlinear system (2.1), we mainly present theoretical results about the performance of auxiliary system (2.23) and nonlinear system (2.1) with parameter uncertainty synthesized by online LNOC law (2.26).

### 2.5.1 Convergence of auxiliary systems

In this subsection, we present theoretical results about the convergence of auxiliary systems (2.23) and (2.12).

*Theorem 1:* The states and dynamics of nonlinear auxiliary system (2.23) converge to those of nonlinear system (2.1) with time.

*Proof:* Recall that nonlinear system (2.1) can be described by parameterized system (2.20):

$$\begin{cases}\dot{\mathbf{x}}(t)=W_f\phi_f(\mathbf{x})+W_g\phi_g(\mathbf{x})\mathbf{u}(t),\\ \mathbf{y}(t)=h(\mathbf{x}),\end{cases}$$

and nonlinear auxiliary system (2.23):

$$\begin{cases} \dot{\hat{\mathbf{x}}}(t) = \hat{W}_f \phi_f(\mathbf{x}) + \hat{W}_g \phi_g(\mathbf{x})\mathbf{u}(t) - K_{\mathbf{x}}(\hat{\mathbf{x}}(t) - \mathbf{x}(t)), \\ \dot{\hat{W}}_f = -K_f(\hat{\mathbf{x}}(t) - \mathbf{x}(t))\phi_f^{\mathsf{T}}(\mathbf{x}), \\ \dot{\hat{W}}_g = -K_g(\hat{\mathbf{x}}(t) - \mathbf{x}(t))\mathbf{u}^{\mathsf{T}}(t)\phi_g^{\mathsf{T}}(\mathbf{x}). \end{cases}$$

Let $\tilde{\mathbf{x}}(t) = \hat{\mathbf{x}}(t) - \mathbf{x}(t)$, $\tilde{W}_f = \hat{W}_f - W_f$ and $\tilde{W}_g = \hat{W}_g - W_g$. Then, one has

$$\begin{cases} \dot{\tilde{\mathbf{x}}}(t) = \tilde{W}_f \phi_f(\mathbf{x}) + \tilde{W}_g \phi_g(\mathbf{x})\mathbf{u}(t) - K_{\mathbf{x}}\tilde{\mathbf{x}}(t), \\ \dot{\tilde{W}}_f = -K_f \tilde{\mathbf{x}}(t)\phi_f^{\mathsf{T}}(\mathbf{x}), \\ \dot{\tilde{W}}_g = -K_g \tilde{\mathbf{x}}(t)\mathbf{u}^{\mathsf{T}}(t)\phi_g^{\mathsf{T}}(\mathbf{x}). \end{cases} \tag{2.30}$$

Define the following candidate Lyapunov function:

$$V_1(t) = \frac{1}{2}\tilde{\mathbf{x}}^{\mathsf{T}}(t)\tilde{\mathbf{x}}(t) + \frac{1}{2}\mathsf{tr}(\tilde{W}_f^{\mathsf{T}} K_f^{-1} \tilde{W}_f) + \frac{1}{2}\mathsf{tr}(\tilde{W}_g^{\mathsf{T}} K_g^{-1} \tilde{W}_g),$$

where $\mathsf{tr}(\cdot)$ denotes the trace of a matrix. Since $K_f$ and $K_g$ are positive-definite and diagonal, $K_f^{-1}$ and $K_g^{-1}$ are also positive-definite and diagonal. It is thus evident that $V_1(t) \geq 0$. Calculating the time derivative of $V_1(t)$ along the system dynamics yields

$$\dot{V}_1(t) = \tilde{\mathbf{x}}^{\mathsf{T}}(t)\dot{\tilde{\mathbf{x}}}(t) + \mathsf{tr}(\tilde{W}_f^{\mathsf{T}} K_f^{-1} \dot{\tilde{W}}_f) + \mathsf{tr}(\tilde{W}_g^{\mathsf{T}} K_g^{-1} \dot{\tilde{W}}_g).$$

Substituting equation (2.30) into $\dot{V}_1(t)$, one further obtains

$$\begin{aligned}
\dot{V}_1(t) &= \tilde{\mathbf{x}}^{\mathsf{T}}(t)(\tilde{W}_f\phi_f(\mathbf{x}) + \tilde{W}_g\phi_g(\mathbf{x})\mathbf{u}(t) - K_{\mathbf{x}}\tilde{\mathbf{x}}(t)) + \mathrm{tr}(\tilde{W}_f^{\mathsf{T}}K_f^{-1}(-K_f\tilde{\mathbf{x}}(t)\phi_f^{\mathsf{T}}(\mathbf{x}))) \\
&\quad + \mathrm{tr}(\tilde{W}_g^{\mathsf{T}}K_g^{-1}(-K_g\tilde{\mathbf{x}}(t)\mathbf{u}^{\mathsf{T}}(t)\phi_g^{\mathsf{T}}(\mathbf{x}))) \\
&= \tilde{\mathbf{x}}^{\mathsf{T}}(t)\tilde{W}_f\phi_f(\mathbf{x}) + \tilde{\mathbf{x}}^{\mathsf{T}}(t)\tilde{W}_g\phi_g(\mathbf{x})\mathbf{u}(t) - \tilde{\mathbf{x}}^{\mathsf{T}}(t)K_{\mathbf{x}}\tilde{\mathbf{x}}(t) - \mathrm{tr}(\tilde{W}_f^{\mathsf{T}}\tilde{\mathbf{x}}(t)\phi_f^{\mathsf{T}}(\mathbf{x})) \\
&\quad - \mathrm{tr}(\tilde{W}_g^{\mathsf{T}}\tilde{\mathbf{x}}(t)\mathbf{u}^{\mathsf{T}}(t)\phi_g^{\mathsf{T}}(\mathbf{x})) \\
&= \mathrm{tr}(\tilde{\mathbf{x}}^{\mathsf{T}}(t)\tilde{W}_f\phi_f(\mathbf{x})) + \mathrm{tr}(\tilde{\mathbf{x}}^{\mathsf{T}}(t)\tilde{W}_g\phi_g(\mathbf{x})\mathbf{u}(t)) - \tilde{\mathbf{x}}^{\mathsf{T}}(t)K_{\mathbf{x}}\tilde{\mathbf{x}}(t) - \mathrm{tr}(\tilde{W}_f^{\mathsf{T}}\tilde{\mathbf{x}}(t)\phi_f^{\mathsf{T}}(\mathbf{x})) \\
&\quad - \mathrm{tr}(\tilde{W}_g^{\mathsf{T}}\tilde{\mathbf{x}}(t)\mathbf{u}^{\mathsf{T}}(t)\phi_g^{\mathsf{T}}(\mathbf{x})) \\
&= \mathrm{tr}(\phi_f(\mathbf{x})\tilde{\mathbf{x}}^{\mathsf{T}}(t)\tilde{W}_f) + \mathrm{tr}(\phi_g(\mathbf{x})\mathbf{u}(t)\tilde{\mathbf{x}}^{\mathsf{T}}(t)\tilde{W}_g) - \tilde{\mathbf{x}}^{\mathsf{T}}(t)K_{\mathbf{x}}\tilde{\mathbf{x}}(t) - \mathrm{tr}(\tilde{W}_f^{\mathsf{T}}\tilde{\mathbf{x}}(t)\phi_f^{\mathsf{T}}(\mathbf{x})) \\
&\quad - \mathrm{tr}(\tilde{W}_g^{\mathsf{T}}\tilde{\mathbf{x}}(t)\mathbf{u}^{\mathsf{T}}(t)\phi_g^{\mathsf{T}}(\mathbf{x})) \\
&= \mathrm{tr}(\tilde{W}_f^{\mathsf{T}}\tilde{\mathbf{x}}(t)\phi_f^{\mathsf{T}}(\mathbf{x})) + \mathrm{tr}(\tilde{W}_g^{\mathsf{T}}\tilde{\mathbf{x}}(t)\mathbf{u}^{\mathsf{T}}(t)\phi_g^{\mathsf{T}}(\mathbf{x})) - \tilde{\mathbf{x}}^{\mathsf{T}}(t)K_{\mathbf{x}}\tilde{\mathbf{x}}(t) - \mathrm{tr}(\tilde{W}_f^{\mathsf{T}}\tilde{\mathbf{x}}(t)\phi_f^{\mathsf{T}}(\mathbf{x})) \\
&\quad - \mathrm{tr}(\tilde{W}_g^{\mathsf{T}}\tilde{\mathbf{x}}(t)\mathbf{u}^{\mathsf{T}}(t)\phi_g^{\mathsf{T}}(\mathbf{x})) \\
&= -\tilde{\mathbf{x}}^{\mathsf{T}}(t)K_{\mathbf{x}}\tilde{\mathbf{x}}(t).
\end{aligned}$$

Evidently, $\dot{V}_1(t) < 0$ if $\tilde{\mathbf{x}}(t) \neq 0$ and $\dot{V}_1(t) = 0$ if and only if $\tilde{\mathbf{x}}(t) = 0$. By Lyapunov theory [44], $\tilde{\mathbf{x}}(t) = 0$ is asymptotically stable. Therefore, when $t \to \infty$, one has $\hat{\mathbf{x}} = \mathbf{x}$ and auxiliary system (2.23) becomes $\dot{\mathbf{x}}(t) = \hat{W}_f\phi_f(\mathbf{x}) + \hat{W}_g\phi_g(\mathbf{x})\mathbf{u}(t)$, which means that given the same input $\mathbf{u}(t)$, the state response of auxiliary system (2.23) is the same as nonlinear system (2.1). It follows that $\lim_{t\to+\infty}(\tilde{W}_f\phi_f(\mathbf{x}) + \tilde{W}_g\phi_g(\mathbf{x})\mathbf{u}) = 0$. In other words, the states and dynamics of nonlinear auxiliary system (2.23) converge to those of nonlinear system (2.1) with time. The proof is complete. □

In Theorem 1, it is proved that a special case is achieved in an asymptotic manner, i.e., $\lim_{t\to+\infty}\hat{\mathbf{x}}(t) = \mathbf{x}(t)$. The underlying intuition is that, under the conditions specified in Theorem 1, the special case always holds in an asymptotic manner. In other words, even the special case does not hold at the beginning, it will be asymptotically achieved, making the proposed control law valid for general cases. Based on Theorem 1, we offer the following corollary about auxiliary system (2.12).

*Corollary 1:* The states and dynamics of linear auxiliary system (2.12) converge to

those of linear system (2.2) with time.

*Proof:* It can be generalized from Theorem 1 and is thus omitted. □

### 2.5.2 Stability of closed-loop systems

In terms of the stability of the closed-loop systems consisting of the controlled systems and the proposed online LNOC laws, we offer the following theoretical results.

*Lemma 1:* Invertible matrices $M$ and $N$ with the same dimension satisfy $N(MN)^{-1} = M^{-1}$.

*Proof:* Since invertible matrices $M$ and $N$ are of the same dimension, $MN$ is also invertible, i.e., $MN(MN)^{-1} = I$. It follows that $N(MN)^{-1} = M^{-1}$. The proof is complete. □

*Theorem 2:* Given that weight matrix $R = 0$, relative degree $\rho \in \{1, 2, 3, 4\}$ and matrix $L_{\hat{g}} L_{\hat{f}}^{\rho-1} h(\mathbf{x})$ is invertible, the closed-loop system consisting of nonlinear system (2.1) with parameter uncertainty and online LNOC law (2.26) is asymptotically stable in the sense of attraction and Lyapunov stability.

*Proof:* Substituting $\mathbf{u}(t) = \mathbf{u}_{\mathsf{a}}(t)$ into the last equation of (2.16) with $\mathbf{u}_{\mathsf{a}}(t)$ defined in equation (2.26) yields

$$\mathbf{y}^{[\rho]}(t) = L_f^{\rho} h(\mathbf{x}) + L_g L_f^{\rho-1} h(\mathbf{x}) \mathbf{u}_{\mathsf{a}}(t). \tag{2.31}$$

Let

$$\begin{cases} L_{\tilde{f}}^{\rho} h(\mathbf{x}) = L_{\hat{f}}^{\rho} h(\mathbf{x}) - L_f^{\rho} h(\mathbf{x}), \\ L_{\tilde{g}} L_{\tilde{f}}^{\rho-1} h(\mathbf{x}) = L_{\hat{g}} L_{\hat{f}}^{\rho-1} h(\mathbf{x}) - L_g L_f^{\rho-1} h(\mathbf{x}), \\ \eta(t) = L_{\tilde{f}}^{\rho} h(\mathbf{x}) + L_{\tilde{g}} L_{\tilde{f}}^{\rho-1} h(\mathbf{x}) \mathbf{u}_{\mathsf{a}}(t). \end{cases} \tag{2.32}$$

Equation (2.31) is thus written as

$$\mathbf{y}^{[\rho]}(t) = L_{\hat{f}}^{\rho} h(\mathbf{x}) + L_{\hat{g}} L_{\hat{f}}^{\rho-1} h(\mathbf{x}) \mathbf{u}_{\mathsf{a}}(t) - \eta(t). \tag{2.33}$$

Given that $R = 0$, LNOC law (2.26) becomes

$$\mathbf{u}(t) = (\kappa(L_{\hat{g}}L_{\hat{f}}^{\rho-1}h(\mathbf{x}))^{\mathsf{T}}QL_{\hat{g}}L_{\hat{f}}^{\rho-1}h(\mathbf{x}))^{-1}(QL_{\hat{g}}L_{\hat{f}}^{\rho-1}h(\mathbf{x}))^{\mathsf{T}}(Y_{\mathsf{d}}(t) - \hat{Y}_{\mathsf{n}}(t))\mathbf{v}^{\mathsf{T}}. \quad (2.34)$$

Substituting equation (2.34) into equation (2.33) yields

$$\mathbf{y}^{[\rho]}(t) = L_{\hat{f}}^{\rho}h(\mathbf{x}) + L_{\hat{g}}L_{\hat{f}}^{\rho-1}h(\mathbf{x})(\kappa(L_{\hat{g}}L_{\hat{f}}^{\rho-1}h(\mathbf{x}))^{\mathsf{T}}QL_{\hat{g}}L_{\hat{f}}^{\rho-1}h(\mathbf{x}))^{-1}(QL_{\hat{g}}L_{\hat{f}}^{\rho-1}h(\mathbf{x}))^{\mathsf{T}}(Y_{\mathsf{d}}(t) - \hat{Y}_{\mathsf{n}}(t))\mathbf{v}^{\mathsf{T}} - \eta(t).$$

Given that $L_{\hat{g}}L_{\hat{f}}^{\rho-1}h(\mathbf{x})$ is invertible and based on Lemma 1, one further has

$$\mathbf{y}^{[\rho]}(t) = L_{\hat{f}}^{\rho}h(\mathbf{x}) + \frac{1}{\kappa}((L_{\hat{g}}L_{\hat{f}}^{\rho-1}h(\mathbf{x}))^{\mathsf{T}}Q)^{-1}(L_{\hat{g}}L_{\hat{f}}^{\rho-1}h(\mathbf{x}))^{\mathsf{T}}Q^{\mathsf{T}}(Y_{\mathsf{d}}(t) - \hat{Y}_{\mathsf{n}}(t))\mathbf{v}^{\mathsf{T}} - \eta(t).$$

Recall that $Q$ is symmetric, i.e., $Q = Q^{\mathsf{T}}$, and positive-definite. One further obtain

$$\mathbf{y}^{[\rho]}(t) = L_{\hat{f}}^{\rho}h(\mathbf{x}) + \frac{1}{\kappa}(Y_{\mathsf{d}}(t) - \hat{Y}_{\mathsf{n}}(t))\mathbf{v}^{\mathsf{T}} - \eta(t). \quad (2.35)$$

Let $\tilde{Y}_{\mathsf{n}}(t) = \hat{Y}_{\mathsf{n}}(t) - Y_{\mathsf{n}}(t)$. Equation (2.35) is then rewritten as

$$\mathbf{y}^{[\rho]}(t) = L_{f}^{\rho}h(\mathbf{x}) + \frac{1}{\kappa}(Y_{\mathsf{d}}(t) - Y_{\mathsf{n}}(t))\mathbf{v}^{\mathsf{T}} + L_{\tilde{f}}^{\rho}h(\mathbf{x}) - \frac{1}{\kappa}\tilde{Y}_{\mathsf{n}}(t)\mathbf{v}^{\mathsf{T}} - \eta(t). \quad (2.36)$$

Let $\delta(t) = -L_{\tilde{f}}^{\rho}h(\mathbf{x}) + \tilde{Y}_{\mathsf{n}}(t)\mathbf{v}^{\mathsf{T}}/\kappa + \eta(t)$. Recalling the definition of $\eta(t)$ in (2.32), we further have $\delta(t) = L_{\tilde{g}}L_{\tilde{f}}^{\rho-1}h(\mathbf{x})\mathbf{u}_{\mathsf{a}}(t) + \tilde{Y}_{\mathsf{n}}(t)\mathbf{v}^{\mathsf{T}}/\kappa$. Then, equation (2.36) can be written as

$$\mathbf{y}^{[\rho]}(t) = L_{f}^{\rho}h(\mathbf{x}) + \frac{1}{\kappa}(Y_{\mathsf{d}}(t) - Y_{\mathsf{n}}(t))\mathbf{v}^{\mathsf{T}} - \delta(t). \quad (2.37)$$

Recall that $\hat{Y}_{\mathsf{n}}(t) = [\mathbf{y}(t), L_{\hat{f}}h(\mathbf{x}), \cdots, L_{\hat{f}}^{\rho}h(\mathbf{x})]$, $Y_{\mathsf{d}}(t) = [\mathbf{y}_{\mathsf{d}}(t), \cdots, \mathbf{y}_{\mathsf{d}}^{[\rho-1]}, \mathbf{y}_{\mathsf{d}}^{[\rho]}(t)]$,

$$\kappa = \int_0^T \frac{\tau^{2\rho}}{(\rho!)^2}\mathsf{d}\tau = \frac{T^{2\rho+1}}{(2\rho+1)(\rho!)^2},$$

and

$$\mathbf{v} = \left[\frac{T^{\rho+1}}{(\rho+1)\rho!}, \frac{T^{\rho+2}}{(\rho+2)\rho!1!}, \cdots, \frac{T^{2\rho+1}}{(2\rho+1)(\rho!)^2}\right].$$

As a result, from equation (2.37), one has

$$\mathbf{y}^{[\rho]}(t) = L_{f}^{\rho}h(\mathbf{x}) + \frac{1}{\kappa}\sum_{j=0}^{\rho-1}\frac{T^{\rho+1+j}}{(\rho+1+j)\rho!j!}(\mathbf{y}_{\mathsf{d}}^{[j]}(t) - \mathbf{y}^{[j]}(t)) + \mathbf{y}_{\mathsf{d}}^{[\rho]} - L_{f}^{\rho}h(\mathbf{x}) - \delta(t).$$

Let $\mathbf{e}(t) = \mathbf{y}_{\mathsf{d}}(t) - \mathbf{y}(t)$. It follows that the closed-loop system consisting of nonlinear system (2.1) and LNOC law (2.26) is

$$\mathbf{e}^{[\rho]}(t) = -\frac{1}{\kappa}\sum_{j=0}^{\rho-1}\frac{T^{\rho+1+j}}{(\rho+1+j)\rho!j!}\mathbf{e}^{[j]}(t) + \delta(t). \tag{2.38}$$

Note that $\delta(t)$ can be viewed as a perturbation or an input. When $\delta(t) = 0$, from equation (2.38), the unforced system is

$$\frac{1}{\kappa}\sum_{i=0}^{\rho}\frac{T^{\rho+1+i}}{(\rho+1+i)\rho!i!}\mathbf{e}^{[i]}(t) = 0. \tag{2.39}$$

When $\rho \in \{1, 2, 3, 4\}$, according to the Routh-Hurwitz criterion [42], it can be readily proved that system (2.39) is exponentially asymptotically stable. Therefore, linear system (2.38) satisfies the bounded-input bounded-output property [42].

From the proof of Theorem 1, with $\mathbf{u}(t) = \mathbf{u}_{\mathsf{a}}(t)$, $\lim_{t\to+\infty}(\tilde{W}_f\phi_f(\mathbf{x}) + \tilde{W}_g\phi_g(\mathbf{x})\mathbf{u}_{\mathsf{a}}(t)) = 0$. Given that $\phi_f(\mathbf{x})$ and $\phi_g(\mathbf{x})\mathbf{u}_{\mathsf{a}}(t)$ satisfy the persistent excitation condition [29–31], one further has $\lim_{t\to+\infty}\tilde{W}_f(t) = 0$ and $\lim_{t\to+\infty}\tilde{W}_g(t) = 0$. In light of the definitions of $L_{\hat{f}}^i h(\mathbf{x})$ and $L_{\hat{g}}L_{\hat{f}}^{\rho-1}h(\mathbf{x})$ in equation (2.25), one further has $\lim_{t\to+\infty}L_{\tilde{f}}^i h(\mathbf{x}) = 0$, $\lim_{t\to+\infty}\tilde{Y}_{\mathsf{n}}(t)\lim_{t\to+\infty}L_{\tilde{g}}L_{\tilde{f}}^{\rho-1}h(\mathbf{x})\mathbf{u}_{\mathsf{a}}(t) = 0$. Therefore,

$$\lim_{t\to+\infty}\delta(t) = \lim_{t\to+\infty}(L_{\tilde{f}}^{\rho}h(\mathbf{x}) + L_{\tilde{g}}L_{\tilde{f}}^{\rho-1}h(\mathbf{x})\mathbf{u}_{\mathsf{a}}(t) + L_{\tilde{f}}^{\rho}h(\mathbf{x}) - \tilde{Y}_{\mathsf{n}}(t)\mathbf{v}^{\mathsf{T}}/\kappa) = 0.$$

According to the bounded-input bounded-output property [42], it is further concluded that $\lim_{t\to+\infty}\|\mathbf{e}(t)\|_2 = \lim_{t\to+\infty}\|\delta(t)\|_2 = 0$, i.e., the equilibrium $\mathbf{e}(t) = 0$ is asymptotically stable. In other words, closed-looped system (2.38) consisting of nonlinear system (2.1) and online LNOC law (2.26) is asymptotically stable. The proof is thus complete. □

Now we have the following remarks about the online LNOC of nonlinear system (2.1) with parameter uncertainty.

*Remark 4:* Intuitively, the philosophy of the proposed online intelligent control scheme for nonlinear system (2.1) with parameter uncertainty lies in the exact estimation of

$W_f\phi_f(\mathbf{x})$ and $W_g\phi_g(\mathbf{x})\mathbf{u}(t)$. The exact estimation of $W_f\phi_f(\mathbf{x})$ and $W_g\phi_g(\mathbf{x})\mathbf{u}(t)$ is obtained from nonlinear auxiliary system (2.23), of which the input-to-state and input-to-output properties converge to the same as nonlinear system (2.1) with time. Note that the persistent excitation condition is generally needed to guarantee the convergence of parameters in adaptive control or parameter estimation methods [11, 30, 31].

*Remark 5:* Based on Theorem 1 and Theorem 2, it can be proved that the closed-looped system consisting of nonlinear system (2.1) and saturated online LNOC law (2.29) is also asymptotically stable. Based on Theorem 1, $\hat{W}_f$ and $\hat{W}_g$ are always bounded. Together with equation (2.26) and Theorem 2, $\mathbf{u}_{\mathsf{a}}(t)$ is bounded and continuous at each time instant $t$, when weight matrix $R = 0$, relative degree $\rho \in \{1, 2, 3, 4\}$, and matrix $L_{\hat{g}}L_{\hat{f}}^{\rho-1}h(\mathbf{x})$ is invertible. Since $\mathbf{u}_{\mathsf{a}}(t)$ is bounded and continuous at each time instant $t$, there always exist a time instant $t_{\mathsf{t}}$ and a constant saturation vector $\beta$ such that, for any $t > t_{\mathsf{t}}$, $-\beta_j \leq u_{\mathsf{a}j}(t) \leq \beta_j$ with $j = 1, 2, \cdots, m$. It follows that $\mathsf{sat}_\beta(\mathbf{u}_{\mathsf{a}}(t)) = \mathbf{u}_{\mathsf{a}}(t)$ for all $t > t_t$. Let $\mathbf{u}_{\mathsf{e}}(t) = \mathsf{sat}_\beta(\mathbf{u}_{\mathsf{a}}(t)) - \mathbf{u}_{\mathsf{a}}(t)$ and $\eta_1(t) = \eta(t) + L_gL_f^{\rho-1}h(\mathbf{x})\mathbf{u}_{\mathsf{e}}(t)$. Evidently, $\lim_{t\to+\infty}\eta_1(t) = \lim_{t\to+\infty}\eta(t)$, since $\mathbf{u}_{\mathsf{e}}(t) = 0$ for $t > t_t$. Following the steps in the proof of Theorem 2, one can thus readily proved that the closed-looped system consisting of nonlinear system (2.1) and saturated online LNOC law (2.29) is asymptotically stable.

*Remark 6:* Theorem 2 is derived under the condition that $R = 0$. In view of performance index (2.15), matrices $Q$ and $R$ are used to scale the trade-off between tracking accuracy described by $\mathbf{y}_{\mathsf{d}}(t) - \mathbf{y}(t)$ and energy consumption described by the magnitude of control effort $\mathbf{u}(t)$. Practically, $R = 0$ means that only tracking accuracy is considered. When $R \neq 0$, it means that energy consumption is also taken into account. In this situation, by the bounded-input bounded-output property [42], via the proposed method, the tracking error $\mathbf{y}_{\mathsf{d}}(t) - \mathbf{y}(t)$ can be guaranteed to be bounded.

In light of the fact that the online LNOC of linear system (2.2) is a special case of that of nonlinear system (2.1), we have the following corollary.

*Corollary 2:* Given that weight matrix $R = 0$, relative degree $\rho \in \{1, 2, 3, 4\}$, and matrix $C\hat{A}^{\rho-1}\hat{B}$ is invertible, The closed-loop system consisting of linear system (2.2) with parameter uncertainty and online LNOC law (2.13) is asymptotically stable.

*Proof:* It can be generalized from the proof of Theorem 2 and is thus omitted. □

### 2.5.3 Asymptotic optimality

In this chapter, a unified approximation of the performance indices is utilized for the optimal control of both linear system (2.2) and nonlinear system (2.1) to avoid solving Hamilton equations and to obtain analytical control laws. In this subsection, theoretical results about the optimality of the proposed control laws are presented.

*Theorem 3:* If matrix $Q$ is symmetric and positive definite, matrix $R = 0$, matrix $L_{\hat{g}}L_{\hat{f}}^{\rho-1}h(\mathbf{x})$ is invertible, and relative degree $\rho \in \{1, 2, 3, 4\}$, then performance index $J_{\mathsf{n}}(t)$ of nonlinear system (2.1) with parameter uncertainty is bounded and asymptotically converges to the optimal, i.e., online LNOC law (2.26) asymptotically converges to the optimal.

*Proof:* Based on Taylor expansion and in view of equations (2.15) and (2.17) about $J_{\mathsf{n}}(t)$ and $\hat{J}_{\mathsf{n}}(t)$, with $R = 0$, one has

$$\begin{aligned} J_{\mathsf{n}}(t) = \int_0^{\mathsf{T}} (E_{\mathsf{n}}(t)\mathbf{w}(\tau) - \frac{\tau^\rho}{\rho!}L_g L_f^{\rho-1}h(\mathbf{x}) + \frac{\tau^\rho}{\rho!}\Delta(t))^{\mathsf{T}} Q (E_{\mathsf{n}}(t)\mathbf{w}(\tau) - \frac{\tau^\rho}{\rho!}L_g L_f^{\rho-1}h(\mathbf{x}) \\ + \frac{\tau^\rho}{\rho!}\Delta(t))\mathsf{d}\tau, \end{aligned}$$

where $\Delta(t) = (\mathbf{y}_{\mathsf{d}}^{[\rho]}(t + \psi\tau) - \mathbf{y}^{[\rho]}(t + \psi\tau) - (\mathbf{y}_{\mathsf{d}}^{[\rho]}(t) - \mathbf{y}^{[\rho]}(t)))$ with $\psi \in (0, 1)$. Based on triangle inequality, we further have

$$\begin{aligned} J_{\mathsf{n}}(t) \leq & 2\hat{J}_{\mathsf{n}}(t) + 2\int_0^T \frac{\tau^{2\rho}}{(\rho!)^2}\Delta^{\mathsf{T}}(t)Q\Delta(t)\mathsf{d}\tau \\ \leq & 2\hat{J}_{\mathsf{n}}(t) + 2\frac{T^{2\rho+1}}{(2\rho+1)(\rho!)^2}\sup_{0<\psi<1}\|\Delta(t)\|_2^2\|Q\|_2, \end{aligned}$$

where $\|\cdot\|_2$ defines the two-norm of a vector or a matrix. According to Theorem 2, given that matrix $Q$ is symmetric and positive definite, matrix $R = 0$, and matrix $L_{\hat{g}}L_{\hat{f}}^{\rho-1}h(\mathbf{x})$

is invertible, the closed-loop system consisting of nonlinear system (2.1) and LNOC law (2.26) is equivalent to the following linear system:

$$\mathbf{e}^{[\rho]}(t) = -\frac{1}{\kappa}\sum_{j=0}^{\rho-1}\frac{T^{\rho+1+j}}{(\rho+1+j)\rho!j!}\mathbf{e}^{[j]}(t) + \delta(t),$$

each subsystem of which is

$$\dot{\mathbf{s}}_i(t) = W\mathbf{s}_i(t) + \nu_i(t),$$

where $\mathbf{s}_i(t) = [e_i(t), \dot{e}_i(t), \cdots, e_i^{[\rho-1]}(t)]^\mathsf{T} \in \mathbb{R}^\rho$ with $i = 1, 2, \cdots, m$, $\nu_i(t) = [0, \cdots, 0, \delta_i(t)]^\mathsf{T} \in \mathbb{R}^\rho$, and $W \in \mathbb{R}^{\rho\times\rho}$ is defined as follows:

$$W = \begin{bmatrix} 0 & 1 & 0 & \cdots & 0 \\ 0 & 0 & 1 & \cdots & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ -\kappa\frac{T^{\rho+1}}{(\rho+1)\rho!} & -\kappa\frac{T^{\rho+2}}{(\rho+2)\rho!} & -\kappa\frac{T^{\rho+3}}{(\rho+3)\rho!2!} & \cdots & -1 \end{bmatrix}.$$

It is readily checked that all the eigenvalues of $W$ are located on the left half-plane when $\rho \in \{1, 2, 3, 4\}$. It follows that linear system $\dot{\mathbf{s}}_i(t) = W\mathbf{s}_i(t)$ is exponentially stable and $\lim_{t\to+\infty}\mathbf{s}_i(t) = 0$ for all $i = 1, 2, \cdots, m$. From Theorem 2, $\lim_{t\to+\infty}\delta(t) = 0$, i.e., $\lim_{t\to+\infty}\nu_i(t) = 0$ for all $i = 1, 2, \cdots, m$. By the bounded-input bounded-output property of linear systems [42], the equilibrium $\mathbf{s}_i(t) = 0$ of linear system $\dot{\mathbf{s}}_i(t) = W\mathbf{s}_i(t) + \nu_i$ is thus asymptotically stable, i.e., $\lim_{t\to+\infty}\mathbf{s}_i(t) = 0$. Together with $\dot{\mathbf{s}}_i(t) = W\mathbf{s}_i(t) + \nu_i(t)$, it follows that

$$\lim_{t\to+\infty}\dot{\mathbf{s}}_i(t) = 0.$$

Recalling that $\mathbf{s}_i(t) = [e_i(t), \dot{e}_i(t), \cdots, e_i^{[\rho-1]}(t)]^\mathsf{T}$, one further has $\lim_{t\to+\infty}e_i^{[\rho]}(t) = 0$ for all $i = 1, 2, \cdots, m$. It follows that $\lim_{t\to+\infty}\mathbf{e}^{[\rho]}(t) = 0$, i.e., $(\mathbf{y}^{[\rho]}(t) - \mathbf{y}_\mathsf{d}^{[\rho]}(t)) = 0$. It follows that $\lim_{t\to+\infty}\sup_{0<\psi<1}\|\Delta(t)\|_2^2\|Q\|_2 = 0$. It is also obtained in the proof of Theorem 2 that $\lim_{t\to+\infty}\tilde{W}_f(t) = 0$ and $\lim_{t\to+\infty}\tilde{W}_g(t) = 0$, given that $\phi_f(\mathbf{x})$ and $\phi_g(\mathbf{x})\mathbf{u}_\mathsf{a}(t)$ satisfy the persistent excitation condition [29–31]. It follows that $\lim_{t\to+\infty}(\mathbf{u}(t) - \mathbf{u}_\mathsf{a}(t)) = 0$, where $\mathbf{u}(t)$ is defined in equation (2.19) and minimizes the convex quadratic performance index $\hat{J}_\mathsf{n}(t)$ shown in equation (2.17), for which, in the adaptive case, $\mathbf{u}(t)$ is replaced by $\mathbf{u}_\mathsf{a}(t)$. It follows that $\lim_{t\to+\infty}\hat{J}_\mathsf{n}(t) = 0$ when LNOC law (2.26) is employed.

Note that $J_{\mathsf{n}}(t) \geq 0$. Then, according to the pinching theorem [47], $\lim_{t\to\infty} J_{\mathsf{n}}(t) = 0$. The proof is complete. □

Similarly, we have the following corollary about the online LNOC of linear system (2.2).

*Corollary 3:* If matrix $Q$ is symmetric and positive definite, matrix $R = 0$, relative degree $\rho \in \{1, 2, 3, 4\}$, and matrix $C\hat{A}^{\rho-1}\hat{B}$ is invertible, then performance index $J_{\mathsf{l}}(t)$ of linear system (2.1) with parameter uncertainty is bounded and asymptotically converges to the optimal, i.e., online LNOC law (2.13) asymptotically converges to the optimal.

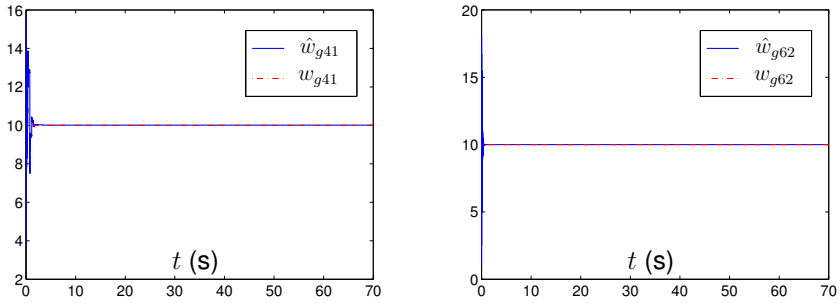
*Proof:* It can be generalized from the proof of Theorem 3 and is thus omitted. □

We offer the following remark about the situation that $L_{\hat{g}}L_{\hat{f}}^{\rho-1}h(\mathbf{x})$ or $C\hat{A}^{\rho-1}\hat{B}$ is singular, which is of great practical significance.

*Remark 7:* In control law (2.26), when the term $\kappa(L_{\hat{g}}L_{\hat{f}}^{\rho-1}h(\mathbf{x}))^{\mathsf{T}}QL_{\hat{g}}L_{\hat{f}}^{\rho-1}h(\mathbf{x})$ is singular and $R = 0$, a regulation term can be used. Specifically, one may use $\nu I_{\mathsf{l}} + \kappa(L_{\hat{g}}L_{\hat{f}}^{\rho-1}h(\mathbf{x}))^{\mathsf{T}}QL_{\hat{g}}L_{\hat{f}}^{\rho-1}h(\mathbf{x})$ to replace $\kappa(L_{\hat{g}}L_{\hat{f}}^{\rho-1}h(\mathbf{x}))^{\mathsf{T}}QL_{\hat{g}}L_{\hat{f}}^{\rho-1}h(\mathbf{x})$, where $\nu > 0 \in \mathbb{R}$ is a small positive parameter, e.g., $10^{-6}$, with $I_{\mathsf{l}}$ being an identity matrix of suitable dimension. Given that $\nu$ is small enough, the control performance is still satisfactory due to the bounded-input bounded-output property [42]. This approach also works for online LNOC law (2.13).

For the choice of the time horizon $T$, we have the following remark.

*Remark 8:* The time horizon can be chosen according to the dynamic property of the control system. When the dynamics of the system is fast, meaning that the state variables changes fast, the time horizon should be set to a small positive value, while for slow systems it can be set to a large positive value.




(a) (b)

Figure 2.3: Time history of the values of non-null elements of $\hat{W}_g$ generated by auxiliary system (2.23) synthesized by saturated online LNOC law (2.29) in comparison with the corresponding elements of $W_g$ of surface vessel system (2.40).

## 2.6 Application to uncertain underactuated surface vessel

The control of surface vessels has long been regarded a challenging problem due to the inherent underactuation of the system. To test the efficacy and generality of the proposed control scheme, in this section, the proposed online LNOC is applied to an underactuated surface vessel with large parameter uncertainty. We successively consider the case without measurement noise, and the case with measurement noises. Then, we analyze the real-time control capability of the proposed control scheme for the underactuated surface vessel.

### 2.6.1 Without measurement noises

We first consider the situation that there is no measurement noise. Consider a surface vessel operated under a failure mode, of which only two propellers work, i.e., the force in surge and the control torque in yaw. Under this realistic assumption, the kinematics and dynamics of the surface vessel are governed by the following ordinary differential

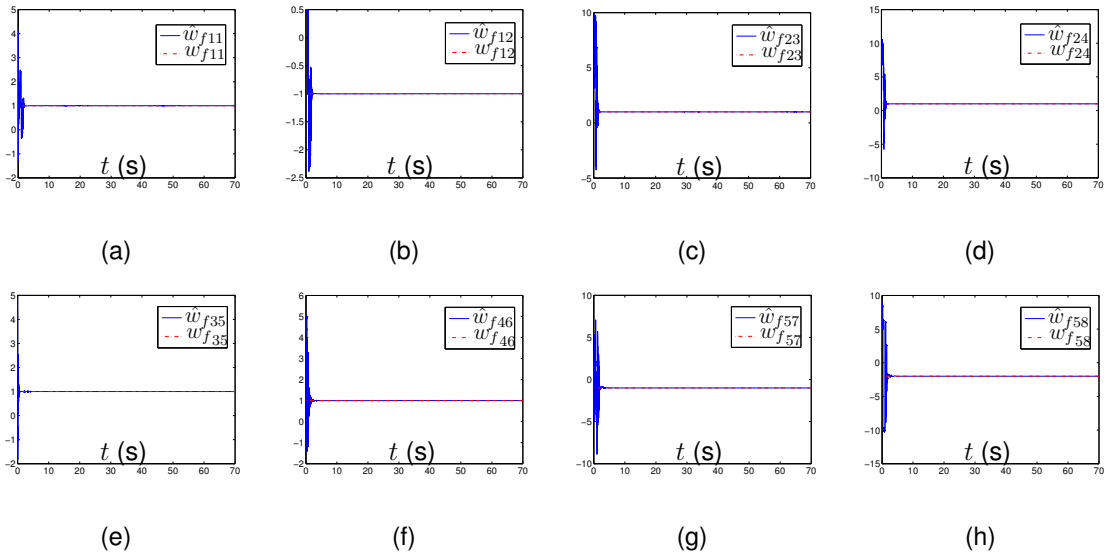
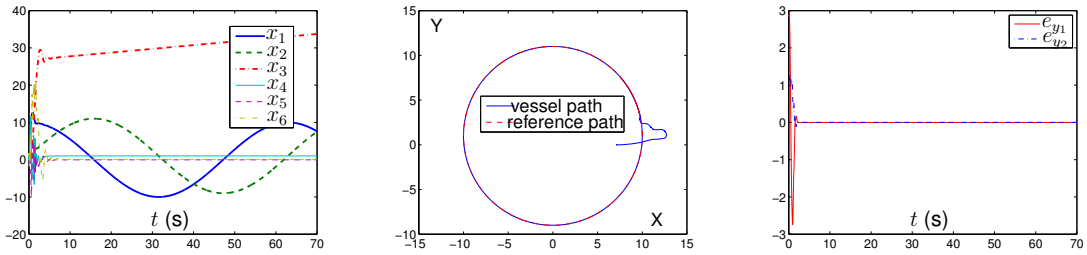



Figure 2.4: Time history of values of the non-null elements of $\hat{W}_f$ generated by auxiliary system (2.23) synthesized by saturated online LNOC law (2.29) in comparison with the corresponding elements of $W_f$ of surface vessel system (2.40).

equations [14]:

$$\begin{cases} \dot{x}_1 = x_4 \cos x_3 - x_5 \sin x_3, \\ \dot{x}_2 = x_4 \sin x_3 + x_5 \cos x_3, \\ \dot{x}_3 = x_6, \\ \dot{x}_4 = \dfrac{m_{22}}{m_{11}} x_5 x_6 - \dfrac{d_{11}}{m_{11}} x_4 + \dfrac{1}{m_{11}} u_1, \\ \dot{x}_5 = -\dfrac{m_{11}}{m_{22}} x_4 x_6 - \dfrac{d_{22}}{m_{22}} x_5, \\ \dot{x}_6 = \dfrac{m_{11} - m_{22}}{m_{33}} x_4 x_5 - \dfrac{d_{33}}{m_{33}} x_6 + \dfrac{1}{m_{33}} u_2, \end{cases} \tag{2.40}$$

where ($x_1$,$x_2$) denotes the coordinate of the surface vessel in the earth-fixed frame; $x_3$ denotes its heading angle; $x_4$, $x_5$ and $x_6$ denote the velocity in surge, sway and yaw, respectively; $u_1$ and $u_2$ denote the surge force and yaw torque, respectively. In addition, parameters $m_{11}$, $m_{22}$, $m_{33}$, $d_{11}$, $d_{22}$ and $d_{33}$ are positive constants and are given by the inertia and damping matrices of the surface vessel [14]. Evidently, the state vector is $\mathbf{x} = [x_1, x_2, x_3, x_4, x_5, x_6]^{\mathsf{T}}$; the input vector is $\mathbf{u} = [u_1, u_2]^{\mathsf{T}}$. In this application, the actual inertia and damping parameters of surface vessel system (2.40)




(a) (b) (c)

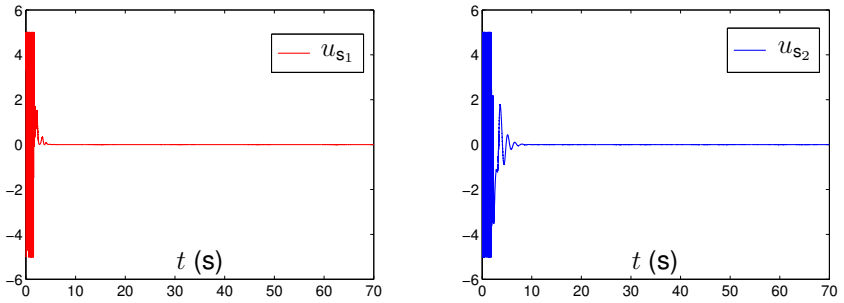
Figure 2.5: Circular path tracking performance of surface vessel system (2.40) synthesized by saturated online LNOC law (2.29). (a) Time history of state variables. (b) Vessel path plotted by $(y_1, y_2)$ data and reference path plotted by $(y_{\mathtt{d}1}, y_{\mathtt{d}2})$ data. (c) Time history of position errors defined as $e_{y_1} = y_{\mathtt{d}1} - y_1$ and $e_{y_2} = y_{\mathtt{d}2} - y_2$.

is $m_{11} = m_{22} = m_{33} = 0.1$, $d_{11} = d_{33} = 0$ and $d_{22} = 0.2$ [14].

Surface vessel system (2.40) can be rewritten in the form of parameterized system (2.20), where $\phi_f(\mathbf{x}) \in \mathbb{R}^8$ with $\phi_{f_1}(\mathbf{x}) = x_4 \cos x_3$, $\phi_{f_2}(\mathbf{x}) = x_5 \sin x_3$, $\phi_{f_3}(\mathbf{x}) = x_4 \sin x_3$, $\phi_{f_4}(\mathbf{x}) = x_5 \cos x_3$, $\phi_{f_5}(\mathbf{x}) = x_6$, $\phi_{f_6}(\mathbf{x}) = x_5 x_6$, $\phi_{f_7}(\mathbf{x}) = x_4 x_6$, and $\phi_{f_8}(\mathbf{x}) = x_5$; the non-null elements of matrix $W_f \in \mathbb{R}^{6\times 8}$ are $w_{f_{11}} = 1$, $w_{f_{12}} = -1$, $w_{f_{23}} = 1$, $w_{f_{24}} = 1$, $w_{f_{35}} = 1$, $w_{f_{46}} = m_{22}/m_{11} = 1$, $w_{f_{57}} = -m_{11}/m_{22} = -1$, $w_{f_{58}} = -d_{22}/m_{22} = -2$; $\phi_g(\mathbf{x}) = [1, 0; 0, 1]$; and the non-null elements of matrix $W_g \in \mathbb{R}^{6\times 2}$ are $w_{g_{41}} = 1/m_{11} = 10$ and $w_{g_{62}} = 1/m_{33} = 10$. It is worth pointing out that the exact values of parameter matrices $W_f$ and $W_g$ of the surface vessel system are only presented for the purpose of comparison, which are not used in the control of the surface vessel system. In other words, the proposed control scheme does not need any prior knowledge on the values of these system parameters.

If the considered output is chosen as $\mathbf{y} = [x_1, x_2]^{\mathsf{T}}$, then one readily has $L_g L_f h(\mathbf{x}) = [\cos(x_3)/m_{11}, 0; \sin(x_3)/m_{11}, 0]$. Evidently, in this case, $L_g L_f h(\mathbf{x})$ is not invertible, i.e., $L_g L_f h(\mathbf{x})$ is singular, which makes the proposed control unfeasible. To avoid the singularity problem, the considered output of the surface vessel is defined as follows:

$$\mathbf{y} = \begin{bmatrix} y_1 \\ y_2 \end{bmatrix} = \begin{bmatrix} x_1 + l\cos x_3 \\ x_2 + l \sin x_3 \end{bmatrix}, \tag{2.41}$$

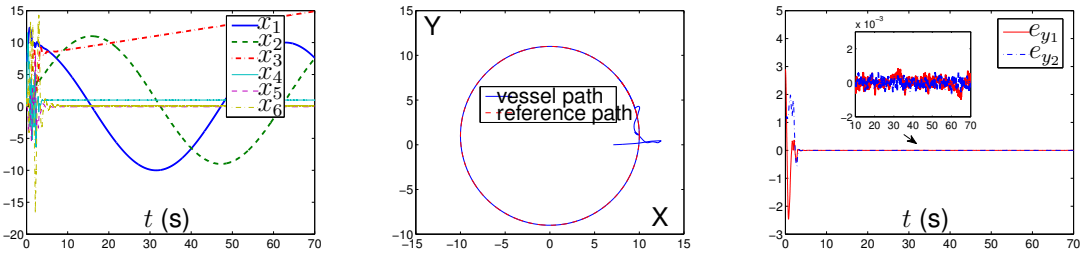



(a) (b)

Figure 2.6: Time history of input $\mathbf{u}_s(t)$ generated by saturated LNOC law (2.29). (a) Time history of $u_{s_1}(t)$. (b) Time history of $u_{s_2}(t)$.

where $l > 0 \in \mathbb{R}$ denotes the distance between the considered position ($x_1+l\cos x_3$,$x_2+l\sin x_3$) and position ($x_1$,$x_2$) of the mass center of the surface vessel. According to Definition 2 and Definition 3, the relative degree $\rho$ of the surface vessel system is 2. Besides, one readily has $L_f L_h(\mathbf{x}) = [\cos x_3/m_{11}, -l\sin x_3/m_{33}; \sin x_3/m_{11}, l\cos x_3/m_{33}]$. Since $l = 0.1 > 0$ and $m_{11} = m_{33} = 0.1 > 0$, the determinant of $L_f L_h(\mathbf{x})$ for the surface vessel system is $|L_f L_h(\mathbf{x})| = l/(m_{11}m_{33}) = 10 > 0$. It follows that $L_f L_h(\mathbf{x})$ is invertible.

Under the condition that $W_f$ and $W_g$ are uncertain, surface vessel system (2.40) is expected to track a desired circular path defined by $\mathbf{y}_d = [10\cos(0.1t), 10\sin(0.1t)+1]^\mathsf{T}$ with $R = 0$ and $Q = [1, 0; 0, 1]$ and $l = 0.1$. We choose $R = 0$ for two reasons: 1) to validate the theoretical results; 2) to achieve high accuracy. In the application, the predictive period of saturated online LNOC law (2.29) is set to $T = 0.2$ s and each element of $\beta$ is set to 5. To avoid singularity, the regulation term as shown in Remark 7 is added with $\nu = 10^{-6}$. The parameters of the corresponding auxiliary system (2.23) are set to $K_\mathbf{x} = 20 \times \text{diag}([1,1,1,1,1,1])$, $K_f = 5000 \times \text{diag}([1,1,1,1,1,1,1,1,1,1])$, and $K_g = 5000 \times \text{diag}([1,1])$. Initial state $\hat{\mathbf{x}}(0)$ of auxiliary system (2.23) is set to be the same as initial state $\mathbf{x}(0)$ of surface vessel system (2.40) with $\mathbf{x}(0) = [7,0,0,0,0,0]^\mathsf{T}$. The initial values of the non-null elements in $\hat{W}_f$ and $\hat{W}_g$ are randomly set, each of which belongs to interval $(0, 10)$. It is worth pointing out that the adaptive control in

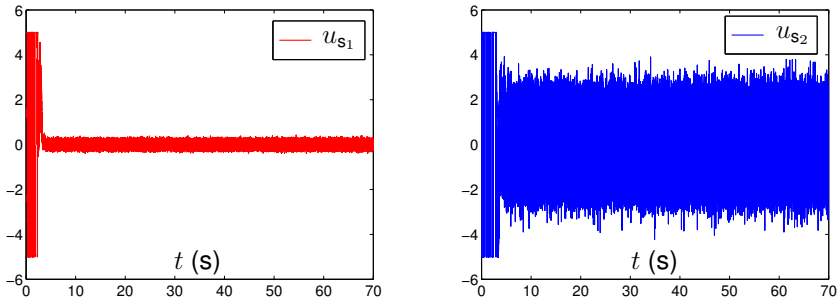



(a) (b) (c)

Figure 2.7: Circular path tracking performance of surface vessel system (2.40) synthesized by saturated online LNOC law (2.29) with independent zero-mean Gaussian measurement noises with the standard deviation being 0.01. (a) Time history of state variables. (b) Vessel path plotted by $(y_1, y_2)$ data and reference path plotted by $(y_{\mathtt{d}1}, y_{\mathtt{d}2})$ data. (c) Time history of position errors defined as $e_{y_1} = y_{\mathtt{d}1} - y_1$ and $e_{y_2} = y_{\mathtt{d}2} - y_2$.

this situation is very difficult due to the lack of a good prior knowledge of the system parameters.

With the above setups, Fig. 2.3 and Fig. 2.4 show the time history of values of the non-null elements of $\hat{W}_g$ and $\hat{W}_f$ generated by nonlinear auxiliary system (2.23) in comparison with the corresponding elements of $W_g$ and $W_f$ of the controlled surface vessel system. As seen from these figures, all the values of the non-null elements of $\hat{W}_f$ and $\hat{W}_g$ converge to those of the corresponding elements of $W_f$ and $W_g$ of the surface vessel system. The tracking performance of surface vessel system (2.40) synthesized by saturated online LNOC law (2.29) is shown in Fig. 2.5. As seen from Fig. 2.5(a), after short time, heading angle $x_3$ of the surface vessel keeps increasing and the yaw speed $x_6$ becomes constant, which means that the surface vessel keeps turning left at a fixed speed. Fig. 2.5(b) and Fig. 2.5(c) show that the surface vessel successfully tracks the reference path and position errors $e_{y_1} = y_{\mathtt{d}1} - y_1$ and $e_{y_2} = y_{\mathtt{d}2} - y_2$ converge to zero with time. Fig. 2.6 shows the time history of control input $\mathbf{u}_{\mathtt{s}}(t)$ generated by saturated LNOC law (2.29). The above results substantiate the efficacy of the proposed LNOC method under no measurement noises and verify the theoretical results.




(a) (b)

Figure 2.8: Time history of input $\mathbf{u}_s(t)$ generated by saturated LNOC law (2.29) with independent zero-mean Gaussian measurement noises with the standard deviation being 0.01. (a) Time history of $u_{s_1}(t)$. (b) Time history of $u_{s_2}(t)$.

### 2.6.2 With measurement noises

In this subsection, simulation results are presented to show the efficacy of the proposed LNOC method under measurement noises.

Consider underactuated surface vessel (2.40) again but with measurement noises taken into account, i.e., in LNOC law (2.29), the state values and output values of underactuated surface vessel (2.40) are polluted by measurement noises. Let $\zeta_s(t) \in \mathbb{R}^6$ and $\zeta_o(t) \in \mathbb{R}^2$ denote state and output measurement noises, respectively. We consider independent zero-mean Gaussian noises for each measurement with the standard deviation being 0.01. The relationship between the measured values and the actual values are $\mathbf{x}_m(t) = \mathbf{x}(t) + \zeta_s(t)$ and $\mathbf{y}_m(t) = \mathbf{y}(t) + \zeta_o(t)$, where $\mathbf{y}(t)$ is defined in (2.41). The parameters of the corresponding auxiliary system (2.23) are set as $K_{\mathbf{x}} = 5 \times \text{diag}([1,1,1,1,1,1])$, $K_f = 50 \times \text{diag}([1,1,1,1,1,1,1,1])$, and $K_g = 50 \times \text{diag}([1,1])$ to avoid potential high overshooting of the elements of $\hat{W}_f$ and $\hat{W}_g$ caused by the measurement noises. The results are shown in Fig. 2.7 and Fig. 2.8. As seen from Fig. 2.7, under the measurement noises, the tracking performance of the surface vessel system synthesized by LNOC law (2.29) is still satisfactory with the maximal steady-state error being less than 0.001. Besides, the time history of the system inputs generated by

the control law are shown in Fig. 2.8. These results substantiate the efficacy of the proposed LNOC method under measurement noises.

### 2.6.3 Capability for real-time control

In this subsection, we further analyze the real-time control capability of the proposed adaptive control method for surface vessels.

The results shown in Section 2.4.4 are employed to calculate the number of floating-point operations per sampling time instant for the proposed method. In this application, we have $m = 2$, $n = 6$, $N_f = 8$, and $N_g = 2$. In practical applications, the value of $\sin(x)$ can be calculated by $\sin(x) \approx x - x^3/6 + x^5/120 - x^7/5040 + x^9/362880 - x^{11}/39916800$, which requires 40 floating-point operations. Similarly, $\cos(x)$ can be calculated by $\cos(x) \approx 1 - x^2/2 + x^4/24 - x^6/720 + x^8/40320 - x^{10}/3628800$, which requires 34 floating-point operations. It follows that the maximal number of floating-point operations needed for calculating the value of an element in vector $\phi_f(\mathbf{x}^k)$ or matrix $\phi_g(\mathbf{x}^k)$ stated in Section VI-A is 41, i.e., $c_{\max} = 41$. Therefore, in this application, at each sampling time instant, the nonlinear auxiliary system requires less than $mn(3 + 4N_g) + 4nN_f + 6n^2 + c_{\max}(n + 2mN_g + N_f) + nN_g = 1454$ floating-point operations. Besides, the maximal number of floating-point operations needed for computing the value of an element in matrices $L_{\hat{g}} L_{\hat{f}}^{\rho-1} h(\mathbf{x})$ or $\hat{Y}_{\mathrm{n}}(t)$ is 64. Therefore, in this application, at each sampling time instant $t$, computing $\mathbf{u}_{\mathrm{a}}(t)$ requires less than $9m^3 + (1 + 2\rho)m^2 + 2m\rho + m + c'_{\max}(m^2 + m\rho + m) = 742$ floating-point operations. Therefore, practically, in the application to the underactuated surface vessel, the proposed LNOC method totally costs less than $1452 + 742 = 2194$ floating-point operations per sampling time instant. If the sample rate is selected as $100$ Hz, i.e., the sampling period is selected as $0.01$ s, then the computational burden of the proposed LNOC method for the underactuated surface vessel is less than $2.194 \times 10^5$ floating-point operations per second. It is worth pointing out that even a Pentium III 750 microprocessor

(which is very old-fashioned) has a computational capability of $3.75 \times 10^8$ floating-point operations per second [48]. Therefore, the proposed LNOC method is feasible for the real-time control of underactuated surface vessels.

## 2.7 Chapter summary

In this chapter, a unified online LNOC framework has been proposed for linear and nonlinear systems with parameter uncertainty. Based on this framework, online LNOC with a simple structure and no requirement for knowing system parameter values have been designed and proposed. Theoretical analysis has shown that closed-loop systems based on the proposed control are asymptotically stable. It has also been proved that the proposed LNOC laws asymptotically converge to the optimal. The application of the proposed near-optimal control scheme to an underactuated surface vessel system with parameter uncertainty has validated the efficacy of the framework and the theoretical results. Before ending this chapter, it is worth pointing out that the proposed framework can deal with nonlinear systems with large parameter uncertainty and enjoys model simplicity, online computation, adaptivity and near-optimality.

# Chapter 3

# LNOC via output feedback with learning of system parameters

In this chapter, a learning and near-optimal control law, which is inherently real-time, is designed to tackle the contradictory between solution accuracy and solution speed for the optimal control of a general class of nonlinear systems with fully unknown parameters.[1] The key technique in the proposed learning and near-optimal control is to design an auxiliary system with the aid of the sliding mode control concept to reconstruct the dynamics of the controlled nonlinear system. Based on the sliding-mode auxiliary system and approximation of the performance index, the proposed control law guarantees asymptotic stability of the closed-system and asymptotic optimality of the performance index with time. Two illustrative examples and an application of the proposed method to a van der Pol oscillator are presented to validate the efficacy of the proposed learning and near-optimal control. In addition, physical experiment results based on a DC motor are also presented to show the realizability, performance, and superiority of the proposed method.

[1] The content in this chapter has already been published. Yinyan Zhang, Shuai Li, and Xiangyuan Jiang, "Near-optimal control without solving HJB equations and its applications," *IEEE Trans. Ind. Electron.,* vol. 65, no. 9, pp. 7173–7184, 2018.

## 3.1 Introduction

In this chapter, an LNOC law is designed and proposed for a class of continuous-time nonlinear systems of fully-unknown parameters to simultaneously guarantee the asymptotic stability of the closed-loop system and optimality of the performance index via output feedback. The proposed LNOC is also inspired by the concept of sliding mode control [65, 66]. Sliding mode control is widely used to design controllers for systems with matched disturbances and have been widely investigated in different problems such as the satellite attitude tracking control problem [67] and the formation control problem [68]. Instead of directly using sliding mode control concept to design control laws, in this chapter, we use it to build an auxiliary system to reconstruct the dynamics of the controlled nonlinear system. This design allows partial decoupling of the control loop from the parameter learning loop and creates an opportunity to fertilize the control part with optimal design.

## 3.2 Problem formulation

Consider the following nonlinear system:

$$\begin{cases} \dot{\mathbf{x}}(t) = f(\mathbf{x}(t)) + g(\mathbf{x}(t))\mathbf{u}(t), \\ \mathbf{y}(t) = h(\mathbf{x}(t)), \end{cases} \tag{3.1}$$

where $\mathbf{x}(t) \in \mathbb{R}^n$ is the state vector, $\mathbf{y}(t) \in \mathbb{R}^m$ is the output vector, and $\mathbf{u}(t) \in \mathbb{R}^m$ is the input vector; $f(\cdot) : \mathbb{R}^n \to \mathbb{R}^n$, $g(\cdot) : \mathbb{R}^n \to \mathbb{R}^{n\times m}$, and $h(\cdot) : \mathbb{R}^n \to \mathbb{R}^m$ are smooth functions. We consider the case that the system satisfies the following conditions: 1) The zero dynamics of system (3.1) are stable [39, 58]; 2) system (3.1) has a known relative degree $\rho$ and $L_g L_f^{\rho-1} h(\mathbf{x})$ is invertible [39,58]; 3) State variables and the output of system (3.1) as well as its output derivatives up to $\rho - 1$ order are measurable or observable [58, 69, 70].

The receding-horizon optimal control problem about system (3.1) is formulated as

$$\begin{aligned} \text{minimize}_{\mathbf{u}(t)} \quad & J(t) \\ \text{subject to} \quad & \dot{\mathbf{x}}(t) = f(\mathbf{x}(t)) + g(\mathbf{x}(t))\mathbf{u}(t), \\ & \mathbf{y}(t) = h(\mathbf{x}(t)), \end{aligned} \tag{3.2}$$

where performance index $J(t)$ is defined as

$$J(t) = \int_0^T (\mathbf{y}_\mathsf{d}(t+\tau) - \mathbf{y}(t+\tau))^\mathsf{T} Q (\mathbf{y}_\mathsf{d}(t+\tau) - \mathbf{y}(t+\tau)) \mathrm{d}\tau, \tag{3.3}$$

where constant $T > 0 \in \mathbb{R}$ denotes the optimization period for each time instant $t$, $Q$ is a symmetric positive-definite weight matrix, and $\mathbf{y}_\mathsf{d}(t)$ denoting the desired output is a continuously differentiable function.

## 3.3 Control design

For nonlinear system (3.1) with relative degree $\rho$, one has

$$\mathbf{y}^{[\rho]}(t) = L_f^\rho h(\mathbf{x}(t)) + L_g L_f^{\rho-1} h(\mathbf{x}(t))\mathbf{u}(t). \tag{3.4}$$

It is assumed that (3.4) can be parameterized as follows:

$$\mathbf{y}^{[\rho]}(t) = W_1 \phi_1(\mathbf{x}(t)) + W_2 \phi_2(\mathbf{x}(t))\mathbf{u}(t) \tag{3.5}$$

where $W_1 \in \mathbb{R}^{m \times N_1}$ and $W_2 \in \mathbb{R}^{m \times N_2}$ are unknown parameter matrices; $\phi_1(\cdot) : \mathbb{R}^n \to \mathbb{R}^{N_1}$ and $\phi_2(\cdot) : \mathbb{R}^n \to \mathbb{R}^{N_2 \times m}$ are known basis functions.

The following auxiliary system is designed to reconstruct the the dynamics of system (3.1):

$$\begin{cases} \hat{\mathbf{y}}^{[\rho]}(t) = \hat{W}_1(t)\phi_1(\mathbf{x}(t)) + \hat{W}_2(t)\phi_2(\mathbf{x}(t))\mathbf{u}(t) - \dot{\mathbf{s}}(t) - \lambda \mathbf{s}(t) + \tilde{\mathbf{y}}^{[\rho]}(t), \\ \dot{\hat{W}}_1(t) = -K_1 \mathbf{s}(t) \phi_1^\mathsf{T}(\mathbf{x}(t)), \\ \dot{\hat{W}}_2(t) = -K_2 \mathbf{s}(t) \mathbf{u}^\mathsf{T}(t) \phi_2^\mathsf{T}(\mathbf{x}(t)), \end{cases} \tag{3.6}$$

where $s(t)$ is defined as

$$\mathbf{s}(t) = \sum_{j=0}^{\rho-1} \alpha_j \tilde{\mathbf{y}}^{[j]}(t). \tag{3.7}$$

Besides, $\tilde{\mathbf{y}}(t) = \hat{\mathbf{y}}(t) - \mathbf{y}(t)$ with $\hat{\mathbf{y}}(t) \in \mathbb{R}^m$ denoting the auxiliary output vector; $\hat{W}_1(t) \in \mathbb{R}^{m\times N_1}$ and $\hat{W}_2(t) \in \mathbb{R}^{m\times N_2}$ are auxiliary parameter matrices; $K_1 \in \mathbb{R}^{m\times m}$ and $K_2 \in \mathbb{R}^{m\times m}$ are diagonal positive-definite gain matrices. The auxiliary system is aided by the concept of sliding mode control with the sliding surface being $\mathbf{s}(t) = 0$, and is thus called sliding-mode auxiliary system. Note that, different from the traditional sliding mode control, in the proposed method, we only need to guarantee that $\lim_{t\to\infty} s(t) = 0$ instead of reaching it in finite time (please refer to the proof of Theorem 1 regarding $\lim_{t\to\infty} s(t) = 0$), which is enough for establishing the stability results concluded in this chapter. Let $\alpha_{\rho-1} = 1$. Then auxiliary system (3.6) becomes

$$\begin{cases} \hat{\mathbf{y}}^{[\rho]}(t) = \hat{W}_1(t)\phi_1(\mathbf{x}(t)) + \hat{W}_2(t)\phi_2(\mathbf{x}(t))\mathbf{u}(t) - \sum_{j=0}^{\rho-2} \alpha_j \tilde{\mathbf{y}}^{[j+1]}(t) - \lambda \mathbf{s}(t), \\ \dot{\hat{W}}_1(t) = -K_1 \mathbf{s}(t)\phi_1^{\mathsf{T}}(\mathbf{x}(t)), \\ \dot{\hat{W}}_2(t) = -K_2 \mathbf{s}(t)\mathbf{u}^{\mathsf{T}}(t)\phi_2^{\mathsf{T}}(\mathbf{x}(t)). \end{cases} \tag{3.8}$$

Via properly choosing parameters $\alpha_j$ for $j = 0, 1, 2, \cdots, \rho - 2$, it can be guaranteed that, on the sliding surface $\mathbf{s}(t) = 0$, $\tilde{\mathbf{y}}(t) = 0$ is exponentially stable [42].

*Theorem 1:* The dynamics of sliding-mode auxiliary system (3.8) asymptotically converges to that of nonlinear system (3.1) of fully-unknown parameters and satisfies

$$\mathbf{y}^{[\rho]}(t) = \hat{\mathbf{y}}^{[\rho]}(t) = \hat{W}_1(t)\phi_1(\mathbf{x}(t)) + \hat{W}_2(t)\phi_2(\mathbf{x}(t))\mathbf{u}(t), \tag{3.9}$$

when $t \to +\infty$.

*Proof:* Let $\tilde{W}_1(t) = \hat{W}_1(t) - W_1$ and $\tilde{W}_2(t) = \hat{W}_2(t) - W_2$. Consider the following candidate Lyapunov function:

$$V_1(t) = \frac{1}{2}\mathbf{s}^{\mathsf{T}}(t)\mathbf{s}(t) + \frac{1}{2}\mathrm{tr}(\tilde{W}_1^{\mathsf{T}}(t)K_1^{-1}\tilde{W}_1(t)) + \frac{1}{2}\mathrm{tr}(\tilde{W}_2^{\mathsf{T}}(t)K_2^{-1}\tilde{W}_2(t)).$$

Subtracting equation (3.5) from the first equation of (3.8) yields

$$\dot{\mathbf{s}}(t) = -\lambda \mathbf{s}(t) + \tilde{W}_1(t)\phi_1(\mathbf{x}(t)) + \tilde{W}_2(t)\phi_2(\mathbf{x}(t))\mathbf{u}(t). \tag{3.10}$$

Let $\alpha_{\rho-1}=1$ for $s(t)$ defined in (3.7). Based on equations (3.10) and (3.8), the following result is obtained:

$$\begin{aligned}\dot{V}_1(t) &= \mathbf{s}^{\mathsf{T}}(t)\dot{\mathbf{s}}(t) + \mathsf{tr}(\tilde{W}_1^{\mathsf{T}}(t)K_1^{-1}\dot{\tilde{W}}_1(t)) + \mathsf{tr}(\tilde{W}_2^{\mathsf{T}}(t)K_2^{-1}\dot{\tilde{W}}_2(t))\\ &= -\lambda\mathbf{s}^{\mathsf{T}}(t)\mathbf{s}(t) + \mathbf{s}^{\mathsf{T}}(t)\tilde{W}_1(t)\phi_1(\mathbf{x}(t)) + \mathbf{s}^{\mathsf{T}}(t)\tilde{W}_2(t)\phi_2(\mathbf{x}(t))\mathbf{u}(t)\\ &\quad - \mathsf{tr}(\tilde{W}_1^{\mathsf{T}}(t)\mathbf{s}(t)\phi_1^{\mathsf{T}}(\mathbf{x}(t))) - \mathsf{tr}(\tilde{W}_2^{\mathsf{T}}(t)\mathbf{s}(t)\mathbf{u}^{\mathsf{T}}(t)\phi_2^{\mathsf{T}}(\mathbf{x}(t)))\\ &= -\lambda\|\mathbf{s}(t)\|_2^2 + \mathsf{tr}(\mathbf{s}^{\mathsf{T}}(t)\tilde{W}_1(t)\phi_1(\mathbf{x}(t))) + \mathsf{tr}(\mathbf{s}^{\mathsf{T}}(t)\tilde{W}_2(t)\phi_2(\mathbf{x}(t))\mathbf{u}(t))\\ &\quad - \mathsf{tr}(\tilde{W}_1^{\mathsf{T}}(t)\mathbf{s}(t)\phi_1^{\mathsf{T}}(\mathbf{x}(t))) - \mathsf{tr}(\tilde{W}_2^{\mathsf{T}}(t)\mathbf{s}(t)\mathbf{u}^{\mathsf{T}}(t)\phi_2^{\mathsf{T}}(\mathbf{x}(t))).\end{aligned}$$

By the property of trace, we have $\mathsf{tr}(\mathbf{s}^{\mathsf{T}}(t)\tilde{W}_1(t)\phi_1(\mathbf{x}(t))) = \mathsf{tr}(\phi_1(\mathbf{x}(t))\mathbf{s}^{\mathsf{T}}(t)\tilde{W}_1(t)) = \mathsf{tr}(\tilde{W}_1^{\mathsf{T}}(t)\mathbf{s}(t)\phi_1^{\mathsf{T}}(\mathbf{x}(t)))$ and $\mathsf{tr}(\mathbf{s}^{\mathsf{T}}(t)\tilde{W}_2(t)\phi_2(\mathbf{x}(t))\mathbf{u}(t)) = \mathsf{tr}(\tilde{W}_2^{\mathsf{T}}(t)\mathbf{s}(t)\mathbf{u}^{\mathsf{T}}(t)\phi_2^{\mathsf{T}}(\mathbf{x}(t)))$. So, $\dot{V}_1(t) = -\lambda\|\mathbf{s}(t)\|_2^2$. Evidently, $\dot{V}_1(t) \leq 0$. Let set $S = \{\tilde{\mathbf{y}}(t) \in \mathbb{R}^m | \dot{V}_1(t) = 0\}$. From $\dot{V}_1(t) = 0$, one has $S = \{\tilde{\mathbf{y}}(t) \in \mathbb{R}^m | \mathbf{s}(t) = 0\}$. Recall $\mathbf{s}(t) = \sum_{j=0}^{\rho-1}\alpha_j\tilde{\mathbf{y}}^{[j]}(t)$. Given that the coefficients $\alpha_j$ for $j = 0, 1, 2, \cdots, \rho-1$ with $\alpha_{\rho-1}=1$ satisfy the Routh stability criterion [42], one has $\lim_{t\to+\infty}\tilde{\mathbf{y}}(t) = 0$. In other words, no solution can stay identically in set $S$, other than the trivial solution $\tilde{\mathbf{y}}(t) = 0$. By LaSalle's invariable set principle [44], equilibrium point $\tilde{\mathbf{y}}(t) = 0$ is asymptotic stable. It follows that, when $t \to +\infty$,

$$\mathbf{y}^{[\rho]}(t) = \hat{\mathbf{y}}^{[\rho]}(t) = \hat{W}_1(t)\phi_1(\mathbf{x}(t)) + \hat{W}_2(t)\phi_2(\mathbf{x}(t))\mathbf{u}(t).$$

In other words, the dynamics of sliding-mode auxiliary system (3.8) asymptotically converges to that of nonlinear system (3.1) of fully-unknown parameters. □

Based on Theorem 1, an LNOC law for system (3.1) can be designed based on system (3.9). Given that the outputs of system (3.1) and their time derivatives up to order $\rho-1$ are measurable. Let $\hat{E}(t) = Y_{\mathsf{d}}(t) - \hat{Y}(t)$ with $\hat{Y}(t) = [\mathbf{y}(t), \dot{\mathbf{y}}(t), \cdots, \hat{W}_1(t)\phi_1(\mathbf{x}(t))]$. Similar to the design procedure in the nominal case, a near-optimal control law for system (3.9) can be design by minimizing the following performance index:

$$\bar{J}(t) = \int_0^T \Big(\hat{E}(t)\mathbf{w}(\tau) - \frac{\tau^\rho}{\rho!}\hat{W}_2(t)\phi_2(\mathbf{x}(t))\mathbf{u}(t)\Big)^{\mathsf{T}} Q\Big(\hat{E}(t)\mathbf{w}(\tau) - \frac{\tau^\rho}{\rho!}\hat{W}_2(t)\phi_2(\mathbf{x}(t))\mathbf{u}(t)\Big)\mathsf{d}\tau. \tag{3.11}$$

Evidently,

$$\bar{J}(t) = \int_0^T \mathbf{w}^{\mathsf{T}}(\tau)\hat{E}^{\mathsf{T}}(t)Q\hat{E}(t)\mathbf{w}(\tau)\mathsf{d}\tau - 2\int_0^T \frac{\tau^{\rho}}{\rho!}\mathbf{w}^{\mathsf{T}}(\tau)\mathsf{d}\tau\hat{E}^{\mathsf{T}}(t)Q\hat{W}_2(t)\phi_2(\mathbf{x}(t))\mathbf{u}(t) + \int_0^T \frac{\tau^{2\rho}}{(\rho!)^2}\mathsf{d}\tau\mathbf{u}^{\mathsf{T}}(t)(\hat{W}_2(t)\phi_2(\mathbf{x}(t)))^{\mathsf{T}}Q\hat{W}_2(t)\phi_2(\mathbf{x}(t))\mathbf{u}(t),$$

where $E = Y_{\mathsf{d}}(t) - Y(t)$. Recall that

$$\mathbf{v} = \int_0^T \frac{\tau^{\rho}}{\rho!}\mathbf{w}^{\mathsf{T}}(\tau)\mathsf{d}\tau = \left[\frac{T^{\rho+1}}{(\rho+1)\rho!}, \frac{T^{\rho+2}}{(\rho+2)\rho!1!}, \cdots, \frac{T^{2\rho+1}}{(2\rho+1)(\rho!)^2}\right],$$

and

$$\kappa = \int_0^T \frac{\tau^{2\rho}}{(\rho!)^2}\mathsf{d}\tau = \frac{T^{2\rho+1}}{(2\rho+1)(\rho!)^2}.$$

Since the decision variable is $\mathbf{u}(t)$, minimizing performance index $\bar{J}(t)$ is equivalent to minimizing the following quadratic performance index:

$$\Psi(t) = \mathbf{u}^{\mathsf{T}}(t)\Theta\mathbf{u}(t) + \mathbf{p}^{\mathsf{T}}\mathbf{u}(t), \tag{3.12}$$

where $\Theta = \kappa(\hat{W}_2(t)\phi_2(\mathbf{x}(t)))^{\mathsf{T}}Q\hat{W}_2(t)\phi_2(\mathbf{x}(t))$ and $\mathbf{p} = -2(\hat{W}_2(t)\phi_2(\mathbf{x}(t)))^{\mathsf{T}}Q^{\mathsf{T}}E\mathbf{v}^{\mathsf{T}}$. Given that $\Theta$ is positive-definite, performance index $\Psi$ shown in (3.12) is convex and the optimal solution can thus be obtained by solving for $\mathbf{u}(t)$ from $\partial\Psi(t)/\partial\mathbf{u} = 0$. Then, given that $\hat{W}_2(t)\phi_2(\mathbf{x}(t))$ is invertible, the following control law is proposed:

$$\mathbf{u}(t) = \frac{1}{\kappa}(\hat{W}_2(t)\phi_2(\mathbf{x}(t)))^{-1}(Y_{\mathsf{d}}(t) - \hat{Y}(t))\mathbf{v}^{\mathsf{T}}, \tag{3.13}$$

which is an LNOC law for system (3.1) with fully-unknown parameters. The block diagram of nonlinear system (3.1) with fully-unknown parameters synthesized by LNOC law (3.13) and sliding-mode auxiliary system (3.8) is shown in Fig. 3.1.

*Remark 1:* For LNOC law (3.13), when $\hat{W}_2(t)\phi_2(\mathbf{x}(t))$ is singular, a regulation term can be used. Specifically, one may use $\nu I_{\mathsf{l}} + \hat{W}_2(t)\phi_2(\mathbf{x}(t))$ to replace $\hat{W}_2(t)\phi_2(\mathbf{x}(t))$, where $\nu > 0 \in \mathbb{R}$ is a small parameter, e.g., $10^{-6}$, with $I_{\mathsf{l}}$ being an identity matrix of suitable dimension. Given that $\nu$ is small enough, the control performance is still satisfactory due to the bounded-input bounded-output property [42].

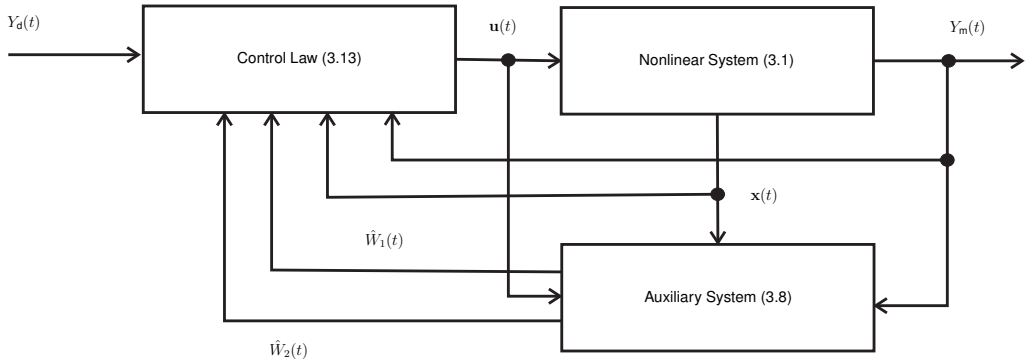



Figure 3.1: Block diagram of nonlinear system (3.1) of fully-unknown parameters synthesized by LNOC law (3.13) and sliding-mode auxiliary system (3.8) with the desired output matrix being $Y_{\mathsf{d}}(t) = [\mathbf{y}_{\mathsf{d}}(t), \dot{\mathbf{y}}_{\mathsf{d}}(t), \cdots, \mathbf{y}_{\mathsf{d}}^{[\rho]}(t)]$ and the measured output matrix being $Y_{\mathsf{m}}(t) = [\mathbf{y}(t), \dot{\mathbf{y}}(t), \cdots, \mathbf{y}^{[\rho-1]}(t)]$.

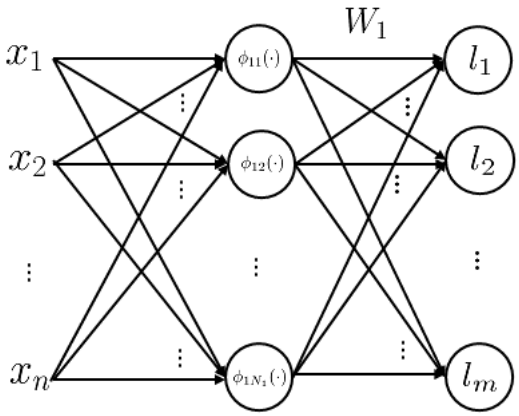



Figure 3.2: Architecture of the sub neural network associated with $W_1$ with $l_i$ denoting the $i$th element of $L_f^{\rho} h(\mathbf{x})$.

About the design methodology, we have the following remark from the perspective of neural networks.

*Remark 2:* System (3.5) can be viewed as a neural network, for which $\phi_1(\mathbf{x}(t))$ and $\phi_2(\mathbf{x}(t))$ are basis functions of two sub neural networks, for which the weight matrices $W_1$ and $W_2$ are unknown. For example, the architecture of the sub neural network associated with $W_1$ can be depicted as Fig. 3.2. The auxiliary system (3.8) provides an online training mechanism for the weight matrix $\hat{W}_1(t)$ and $\hat{W}_2(t)$ so as to make the auxiliary system dynamics converge to that of system (3.5). Then, the intelligent controller is designed based on $\hat{W}_1(t)$ and $\hat{W}_2(t)$, which is expected to drive the the system to the desired state.

*Theorem 2:* Given that relative degree $\rho \in \{1, 2, 3, 4\}$ and $\hat{W}_2(t)\phi_2(\mathbf{x}(t))$ is invertible at each time instant $t$, the closed-loop system consisting of nonlinear system (3.1) of fully-unknown parameters and LNOC law (3.13) is asymptotically stable.

*Proof:* Substituting LNOC law (3.13) into equation (3.5) yields $\mathbf{y}^{[\rho]}(t) = W_1\phi_1(\mathbf{x}(t)) + W_2\phi_2(\mathbf{x}(t))(\hat{W}_2(t)\phi_2(\mathbf{x}(t)))^{-1}(Y_\mathsf{d}(t)-\hat{Y}(t))\mathbf{v}^\mathsf{T}/\kappa$. Let $\tilde{W}_1(t) = \hat{W}_1(t)-W_1$ and $\tilde{W}_2(t) = \hat{W}_2(t) - W_2$. Then, one has

$$\begin{aligned}
\mathbf{y}^{[\rho]}(t) &= \hat{W}_1(t)\phi_1(\mathbf{x}(t)) + \frac{1}{\kappa}(Y_\mathsf{d}(t) - \hat{Y}(t))\mathbf{v}^\mathsf{T} - \delta(t) \\
&= W_1\phi_1(\mathbf{x}(t)) + \tilde{W}_1(t)\phi_1(\mathbf{x}(t)) + \frac{1}{\kappa}(Y_\mathsf{d}(t) - Y(t))\mathbf{v}^\mathsf{T} - \tilde{W}_1(t)\phi_1(\mathbf{x}(t)) - \delta(t) \\
&= W_1\phi_1(\mathbf{x}(t)) + \frac{1}{\kappa}(Y_\mathsf{d}(t) - Y(t))\mathbf{v}^\mathsf{T} - \delta(t) \\
&= \mathbf{y}^{[\rho]}(t) + \kappa\sum_{j=0}^{\rho}\frac{T^{\rho+1+j}}{(\rho+1+j)\rho!j!}\mathbf{e}^{[j]}(t) - \delta(t),
\end{aligned}$$

where $\delta(t) = \tilde{W}_1(t)\phi_1(\mathbf{x}(t)) + \tilde{W}_2(t)\phi_2(\mathbf{x}(t))\mathbf{u}(t)$ and $\mathbf{e}^{[j]} = \mathbf{y}_\mathsf{d}^{[j]}(t) - \mathbf{y}^{[j]}(t)$. It follows that the closed-loop system consisting of nonlinear system (3.1) of fully-unknown parameters and LNOC law (3.13) is

$$\kappa\sum_{j=0}^{\rho}\frac{T^{\rho+1+j}}{(\rho+1+j)\rho!j!}\mathbf{e}^{[j]}(t) = \delta(t). \tag{3.14}$$

By Routh stability criterion [42], it can be readily proved that when $\delta(t) = 0$ and $\rho \in$

$\{1,2,3,4\}$, system (3.14) is exponentially stable. Besides, according to Theorem 1, one has $\lim_{t\to+\infty}\delta(t)=\lim_{t\to+\infty}(\tilde{W}_1(t)\phi_1(\mathbf{x}(t))+\tilde{W}_2(t)\phi_2(\mathbf{x}(t))\mathbf{u}(t))=0$. According to bounded-input bounded-output stability theory [42], equilibrium point $\mathbf{e}(t)=0$ of closed-loop system (3.14) is asymptotically stable. □

*Theorem 3:* Given that relative degree $\rho\in\{1,2,3,4\}$ and $\hat{W}_2(t)\phi_2(\mathbf{x}(t))$ is invertible at each time instant $t$, performance index $J(t)$ of nonlinear system (3.1) with fully unknown parameters synthesized by LNOC law (3.13) is bounded and asymptotically converges to be optimal.

*Proof:* With equation (3.5) taken into account, based on Taylor expansion, one has

$$\begin{aligned}J(t)&=\int_0^{\mathsf{T}}\left(E(t)\mathbf{w}(\tau)-\frac{\tau^\rho}{\rho!}W_2\phi_2(\mathbf{x}(t))\mathbf{u}(t)+\Delta_1(t)\right)^{\mathsf{T}}Q\Bigg(E(t)\mathbf{w}(\tau)-\frac{\tau^\rho}{\rho!}W_2\phi_2(\mathbf{x}(t))\mathbf{u}(t)\\&\quad+\Delta_1(t)\Bigg)\mathsf{d}\tau\\&=\int_0^{\mathsf{T}}\left(\hat{E}(t)\mathbf{w}(\tau)-\frac{\tau^\rho}{\rho!}\hat{W}_2(t)\phi_2(\mathbf{x}(t))\mathbf{u}(t)+\Delta_1(t)+\Delta_2(t)\right)^{\mathsf{T}}Q\Bigg(\hat{E}(t)\mathbf{w}(\tau)\\&\quad-\frac{\tau^\rho}{\rho!}\hat{W}_2(t)\phi_2(\mathbf{x}(t))\mathbf{u}(t)+\Delta_1(t)+\Delta_2(t)\Bigg)\mathsf{d}\tau,\end{aligned}$$

where

$$\Delta_1(t)=\frac{\tau^\rho}{\rho!}(\mathbf{y}_{\mathsf{d}}^{[\rho]}(t+\varkappa\tau)-\mathbf{y}^{[\rho]}(t+\varkappa\tau)-(\mathbf{y}_{\mathsf{d}}^{[\rho]}(t)-\mathbf{y}^{[\rho]}(t)))$$

with $\varkappa\in(0,1)$ and

$$\Delta_2(t)=\tau^\rho(\tilde{W}_1(t)\phi_1(\mathbf{x}(t))+\tilde{W}_2(t)\phi_2(\mathbf{x}(t))\mathbf{u}(t))/\rho!.$$

Recall the definition of performance index $\bar{J}(t)$ in (3.11). Based on triangle inequality, one further has

$$\begin{aligned}J(t)&\le 2\bar{J}(t)+2\int_0^T\frac{\tau^{2\rho}}{(\rho!)^2}\Delta_1^{\mathsf{T}}(t)Q\Delta_1(t)\mathsf{d}\tau+2\int_0^T\frac{\tau^{2\rho}}{(\rho!)^2}\Delta_2^{\mathsf{T}}(t)Q\Delta_2(t)\mathsf{d}\tau\\&\le 2\bar{J}(t)+2\frac{T^{2\rho+1}}{(2\rho+1)(\rho!)^2}\sup_{0<\varkappa<1}\|\Delta_1(t)\|_2^2\|Q\|_2+2\frac{T^{2\rho+1}}{(2\rho+1)(\rho!)^2}\|\Delta_2(t)\|_2^2\|Q\|_2.\end{aligned}$$

According to Theorem 1,

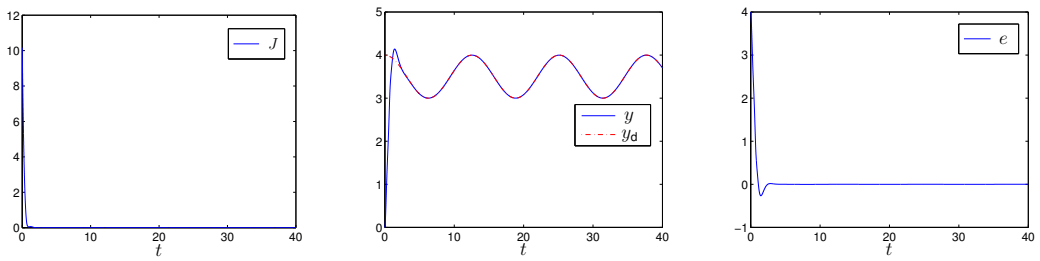
$$\lim_{t\to+\infty}(\tilde{W}_1(t)\phi_1(\mathbf{x}(t))+\tilde{W}_2(t)\phi_2(\mathbf{x}(t))\mathbf{u}(t))=0.$$

It follows that $\lim_{t\to+\infty} \|\Delta_2(t)\|_2^2\|Q\|_2 = 0$. Besides, according to Theorem 2, when $\rho \in \{1,2,3,4\}$ and $\hat{W}_2(t)\phi_2(\mathbf{x}(t))$ is invertible, nonlinear system (3.1) of fully-unknown parameters synthesized by LNOC law (3.13) satisfies $\lim_{t\to+\infty}(\mathbf{y}^{[\rho]}(t) - \mathbf{y}_{\mathsf{d}}^{[\rho]}(t)) = 0$. It follows that $\lim_{t\to+\infty} \sup_{0<\varkappa<1} \|\Delta_1(t)\|_2^2\|Q\|_2 = 0$. Note that $J(t) \geq 0$. In addition, from the design procedure of near-optimal control law (3.13), it is known that control law (3.13) is optimal in terms of convex quadratic performance index $\bar{J}(t)$ defined in equation (3.11), which guarantees $\bar{J}(t) = 0$. Then, by the pinching theorem, $\lim_{t\to\infty} J(t) = 0$. □

*Remark 3:* The performances of the proposed LNOC law (3.13) is theoretically guaranteed for nonlinear system (3.1) of fully-unknown parameters when relative degree $\rho \in \{1,2,3,4\}$. It is worth pointing out that relative degree $\rho$ of many mechanical systems is lower than 4 [39].

We also have the following remark about the comparison of the proposed method with other existing methods.

*Remark 4:* As indicated in the Introduction section, traditional optimal control approaches require solving HJB equations, which cannot be solved in a real-time manner. Through a proper approximation, the proposed one does not need to solve it. Compared with the back-stepping methods, e.g., [51], the proposed method does not require tedious choices of virtual control laws. Compared with the adaptive sliding mode control method [77], the proposed method inherently does not introduce the chattering phenomenon. Compared with the fuzzy neural network based approach [78], the computational burden of the proposed method is even more simpler owing to the lack of many Gaussian functions to be calculated.




(a) (b) (c)

Figure 3.3: Control performance of uncertain nonlinear system (3.15) synthesized by LNOC law (3.13) and sliding-mode auxiliary system (3.8) with $y_{\mathsf{d}}(t) = 0.5\cos(0.5t) + 3.5$, $T = 0.6$, and $Q = 2$. (a) Time history of performance index $J_{\mathsf{n}}(t)$ defined in equation (3.3). (b) Time histories of output $y(t)$ and desired output $y_{\mathsf{d}}(t)$. (c) Time history of tracking error $e(t) = y_{\mathsf{d}}(t) - y(t)$.

## 3.4 Illustrative examples

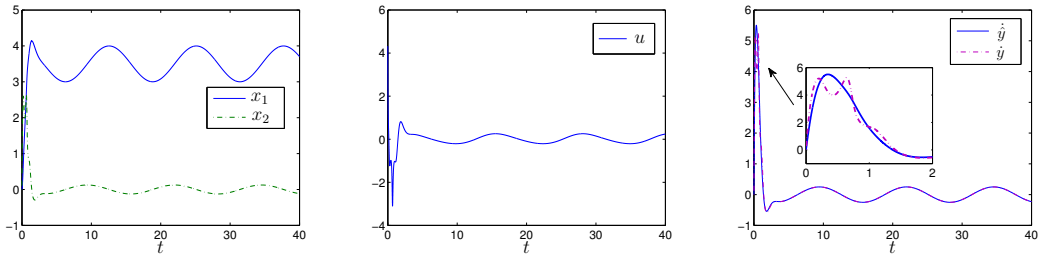
In this section, two illustrative examples are presented to show the efficacy and superiority of the proposed method and verify the theoretical results.

### 3.4.1 Example 1

Consider the following nonlinear system:

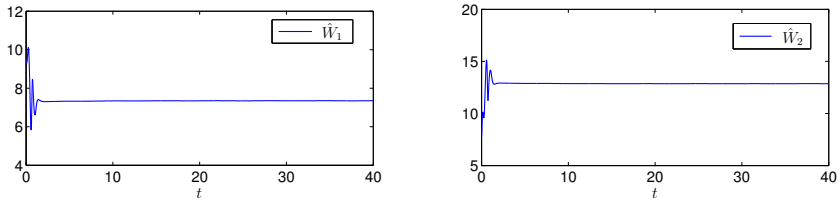
$$\begin{cases} \dot{x}_1(t) = p_1 x_2(t), \\ \dot{x}_2(t) = p_2 x_1(t) x_2(t) + p_3(\sin(x_2(t)) + 1.1)u(t), \\ y(t) = x_1(t), \end{cases} \tag{3.15}$$

The system parameters are $p_1 = 2$, $p_2 = 4$ and $p_3 = 7$. Relative degree $\rho$ of system (3.15) is 2. Note that our method does not need any prior knowledge on parameter values of the system, and the actual parameter values are only presented for readers' possible interest in validating our method. In this example, the parameters of performance index (3.3) are set as $T = 0.6$ and $Q = 2$. The desired output is given as $y_{\mathsf{d}}(t) = 0.5\cos(0.5t) + 3.5$. With $\phi_1(\mathbf{x}) = x_1 x_2$, $\phi_2(\mathbf{x}) = \sin(x_2) + 1.1$, $\lambda = 3$, $\alpha_0 = 2$,




(a) (b) (c)

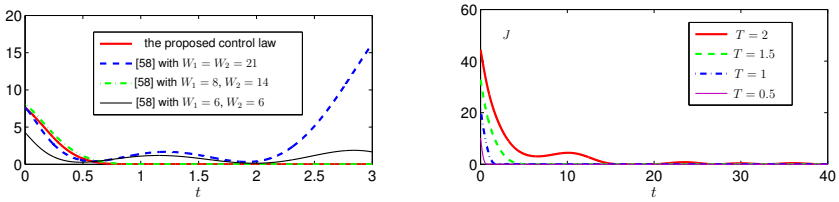
Figure 3.4: Time histories of state variables $x_1(t)$ and $x_2(t)$, control input $u(t)$, $\dot{\hat{y}}(t)$ of sliding-mode auxiliary system (3.8), and $\dot{y}(t)$ of nonlinear system (3.15) during the control process. (a) Time histories of state variables $x_1(t)$ and $x_2(t)$. (b) Time history of control input $u(t)$. (c) Time histories of $\dot{\hat{y}}(t)$ of sliding-mode auxiliary system (3.8) and $\dot{y}(t)$ of nonlinear system (3.15).




(a) (b)

Figure 3.5: Time histories of parameters $\hat{W}_1(t)$ and $\hat{W}_2(t)$ of sliding-mode auxiliary system (3.8) during the control process of nonlinear system (3.15) by LNOC law (3.13). (a) Time history of $\hat{W}_1(t)$. (b) Time history of $\hat{W}_2(t)$.

$\alpha_1 = 1$, $K_1 = 4$, $K_2 = 9$, $x_1(0) = x_2(0) = \hat{y}(0) = \dot{\hat{y}}(0) = 0$, and $\hat{W}_1(0)$ and $\hat{W}_2(0)$ randomly generated at interval $(0, 10)$, the simulation results of nonlinear system (3.15) synthesized by LNOC law (3.13) are shown in Fig. 3.3 through Fig. 3.5. From Fig. 3.3, it is observed that performance index $J_{\mathsf{n}}(t)$ rapidly converges to near zero (i.e., optimal), and the output $y(t)$ of nonlinear system (3.15) quickly tracks the desired output $y_{\mathsf{d}}(t)$ with tracking error $e(t) = y_{\mathsf{d}}(t) - y(t)$ converging to zero. This validates Theorem 2 and Theorem 3. Besides, a comparison between the time histories of $\dot{y}(t)$ of nonlinear system (3.15) and $\dot{\hat{y}}(t)$ of sliding-mode auxiliary system (3.8) during the control process is shown in Fig. 3.4(c), which validates Theorem 1. The evolutions of $\hat{W}_1(t)$ and $\hat{W}_2(t)$ are shown in Fig. 3.5. The above results validate the efficacy of the proposed LNOC




(a) (b)

Figure 3.6: Time histories of performance index $J_n$. (a) Time histories of performance index $J_n$ (3.3) of nonlinear system (3.15) synthesized by the proposed LNOC law (3.13) and by the existing nominal near-optimal control law (18) in [58], where the actual system parameter values are $W_1 = 8$ and $W_2 = 14$. (b) Time histories of $J_n(t)$ of system (3.15) synthesized by LNOC law (3.13) under different values of $T$.

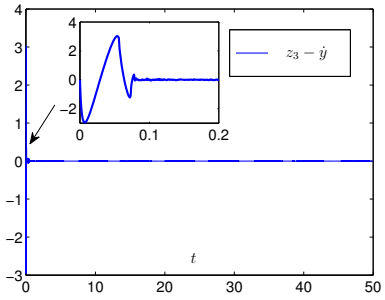



Figure 3.7: Time history of $z_3(t) - \dot{y}$ where $z_3(t)$ is the output of tracking differentiator (3.16) and $\dot{y}$ is the output derivative of system (3.15) during the control process.

and the theoretical results.

The performance of LNOC law (3.13) for nonlinear system (3.15) under parameter uncertainty is compared with that of nominal near-optimal control law (18) in [58]. The values of $\hat{W}_1(0)$ and $\hat{W}_2(0)$ for auxiliary system (3.8) are set as $\hat{W}_1(0) = \hat{W}_2(0) = 21$. The values of $W_1$ and $W_2$ in nominal near-optimal control law (18) in [58] are set as different values for comparison. The initial states of nonlinear system (3.15) are set as $x_1(0) = 3$, $x_2(0) = 5$. Meanwhile, we set $\hat{y}(0) = 3$ and $\dot{\hat{y}}(0) = 10$ for auxiliary system (3.8). Other parameter values are set the same as the previous ones. Under these setups, time histories of performance index $J_n$ (3.3) of nonlinear system (3.15) synthesized by the two control laws respectively are shown in Fig. 3.6(a). From this

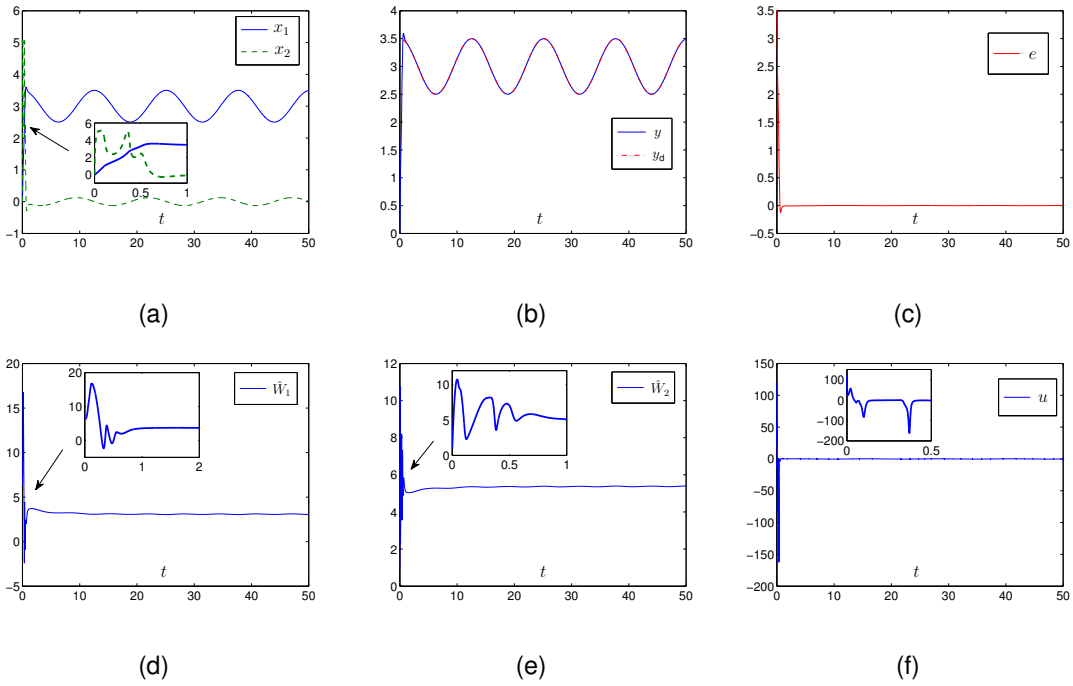



Figure 3.8: Time histories of variables during the control process of system (3.15) via LNOC law (3.13) aided with auxiliary system (3.8) and tracking differentiator (3.16). (a) Time histories of state variables $x_1(t)$ and $x_2(t)$. (b) Time histories of system output $y(t)$ and desired output $y_{\mathrm{d}}(t)$. (c) Time history of tracking error $e(t) = y_{\mathrm{d}}(t) - y(t)$. (d) Time history of parameter $\hat{W}_1(t)$. (e) Time history of parameter $\hat{W}_2(t)$. (e) Time history of input $u(t)$.

figure, it can be observed that LNOC law (3.13) is superior to the nominal near-optimal control law under parameter uncertainty of the controlled system.

In addition, with $\hat{W}_1(0) = \hat{W}_2(0) = 20$, and the other parameters being the same as those of the above simulation, time histories of $\mathrm{J_n}(t)$ (3.3) of system (3.15) synthesized by LNOC law (3.13) under different values of $T$ are shown in Fig. 3.6(b). This figure shows that under different values of $T$, performance index $\mathrm{J_n}(t)$ of system (3.15) is optimized by the proposed LNOC law (3.13). Besides, the convergence speed of the performance index is related to the value of $T > 0$. Specifically, when $T$ is smaller, the convergence is faster.

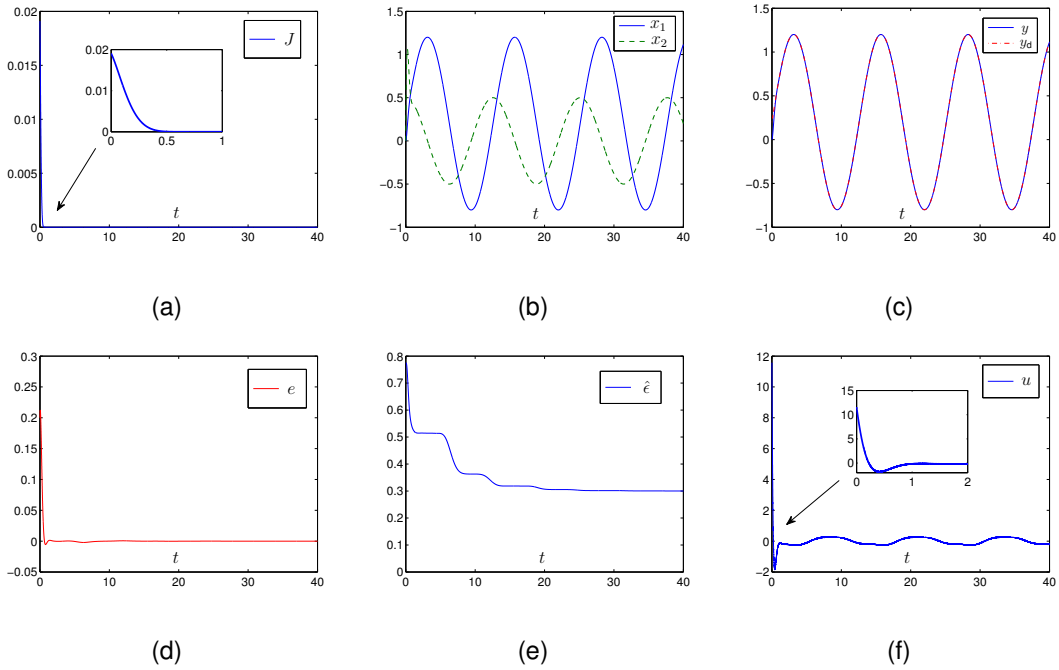



Figure 3.9: Time histories of variables during the control process of van der Pol oscillator (3.17) via LNOC law (3.13) aided with auxiliary system (3.8) and tracking differentiator (3.16). (a) Time histories of performance index $J(t)$. (b) Time histories of state variables $x_1(t)$ and $x_2(t)$. (c) Time histories of system output $y(t)$ and desired output $y_{\mathsf{d}}(t)$. (d) Time history of tracking error $e(t) = y_{\mathsf{d}}(t) - y(t)$. (e) Time history of parameter $\hat{\epsilon}$. (f) Time history of input $u(t)$.

### 3.4.2 Example 2

A major difference of the proposed method compared with most existing controllers is on the time derivatives of system outputs. When there are not available sensors to measure output derivatives of a system, tracking differentiators as in [71–73] can be used to construct the required derivatives for the output measurement. For example, to obtained the first-order derivative of a measurable function $w(t) \in \mathbb{R}$, the following first-order robust exact tracking differentiator [72] can be used:

$$\begin{cases} \dot{z}_1(t) = z_2(t) - \sqrt{C}|z_1(t) - w(t)|^{1/2}\mathsf{sign}(z_1(t) - w(t)), \\ \dot{z}_2(t) = -1.1C\mathsf{sign}(z_1(t) - w(t)), \\ z_3(t) = z_2(t) - \sqrt{C}|z_1(t) - w(t)|^{1/2}\mathsf{sign}(z_1(t) - w(t)), \end{cases} \tag{3.16}$$

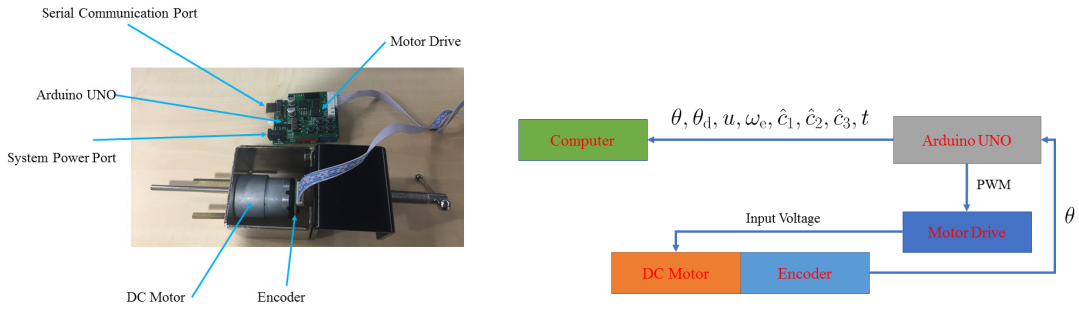
where $z_3(t)$ is the output of the tracking differentiator, which converges to $\dot{w}(t)$ in finite time; $C > 0 \in \mathbb{R}$ is the design parameters and should be large enough. Note that high-order tracking differentiators for tracking high-order derivatives can be found in [73]. To see the performance of the proposed LNOC law under the situation that the time derivatives are not directly available, simulations are also conducted based on Example 1. Specifically, the output of system (3.15) is used as the input of tracking differentiator (3.16), i.e., we let $w(t) = y(t)$. Besides, the output of tracking differentiator (3.16), i.e., $z_3$ is used in intelligent control law (3.13) to replace $\dot{y}(t)$. Without loss of generality, the initial values of the state variables of the tracking differentiator are set to zero in the simulation. Besides, according to [73], the value of parameter $C$ is set to 200. With the other setups being the same as those in Example 1, simulation results are shown in Fig. 3.7 and Fig. 3.8. These results substantiate the efficacy of the proposed intelligent control law in the case that output derivatives are not directly available.

## 3.5 Application to van der Pol oscillator

In this section, we further present the application of the proposed control method to a van der Pol oscillator [75], which is important in electronic circuits to generate respective electronic signals, such as the sine wave. The model of the van der Pol oscillator is given as follows:

$$\begin{cases} \dot{x}_1 = x_2, \\ \dot{x}_2 = -\epsilon(x_1^2 - 1)x_2 - x_1 + u, \\ y = x_1, \end{cases} \tag{3.17}$$

where $\epsilon > 0$ is a constant parameter of the system which depends on the corresponding circuit. In this application, only $y$ is directly measurable, $\epsilon = 0.3$, and the desired signal $y_{\mathsf{d}}(t) = \sin(0.5t) + 0.2$. The parameters of performance index (3.3) are set as $T = 0.3$ and $Q = 2$. The proposed method is used to address the problem without




(a) (b)

Figure 3.10: Experimental platform for the DC motor control system. (a) A picture of the DC motor control system. (b) The overall system diagram in terms of signal transmissions.

using the exact value of $\epsilon$. As in Example 2, we use tracking differentiator (3.16) to obtain the derivative of $y$. The results are shown in Fig. 3.9 for the setup that $\phi_1(\mathbf{x}) = [-(x_1^2 - 1)x_2, -x_1]^\mathsf{T}$, $\phi_2 = 1$, $W_1 = [\hat{\epsilon}, 1]^\mathsf{T}$, $W_2 = 1$, $\lambda = 3$, $\alpha_0 = 2$, $\alpha_1 = 1$, $K_1 = 7$, $x_1(0) = x_2(0) = \hat{y}(0) = \dot{\hat{y}}(0) = 0$, and $\hat{\epsilon}$ is randomly generated at interval $(0, 1)$. These results further verify the efficacy and practical significance of the proposed method.

## 3.6 Experimental validation

In this section, experimental results and comparisons are provided to show the efficacy and superiority of the proposed method.

The prototype of a motor control system shown in Fig. 3.10 is used in the experiment. The system power is provided by a 9-V 1-A AC/DC adaptor. The proposed control method is implemented in the Arduino UNO board, and the control input computed by the proposed method is converted into PWM signal which is fed into the 12-V motor drive by using a Arduino command *analogwrite(pin,value)*. The microcontroller of the Arduino UNO board used in the experiments is ATmega328P [82], which is a low-power CMOS 8-bit microcontroller and the clock speed is 16 MHz. In Arduino UNO, the *value* can be 0 to 255, which corresponds to 0 to $100\%$ PWM. The conversion rule is thus as

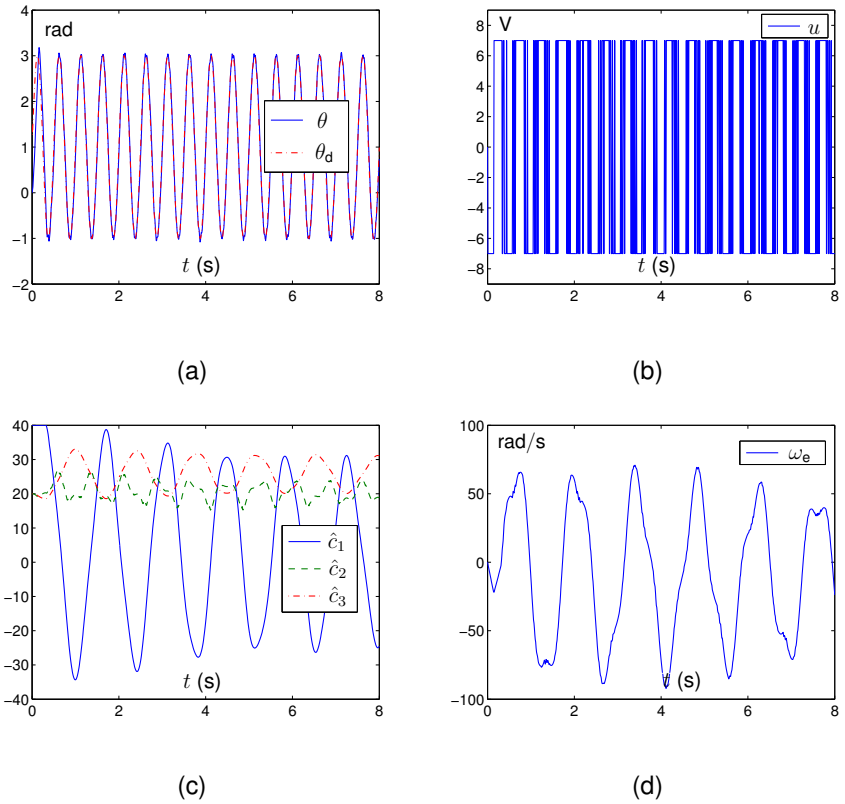



Figure 3.11: Data profiles during the experiment of the tracking control of the DC motor by the proposed LNOC method with saturation (3.19) based on the model (3.18). (a) Time history of motor angle $\theta(t)$ and desired angle $\theta_d(t)$. (b) Time histories of input $u$ generated by the proposed controller (3.13) with saturation specified in (3.19). (c) Time histories of parameters estimated by the auxiliary system. (d) Time history of $\omega$ estimated via the tracking differentiator (3.16) with the input being the joint angles measured by the encoder, which is denoted by $\omega_e$.

follows:

$$value = \frac{255u}{12},$$

where $u$ denotes the input generated by the proposed controller. The motor angle position is measured by an encoder, which generates 1040 pulses when the motor rotates for a circle. Note that, in the system configuration, the computer is only used to store the data sent by the Arduino UNO. In the experiments, all the computations are conducted by the Arduino UNO. The codes are written and complied via the open-

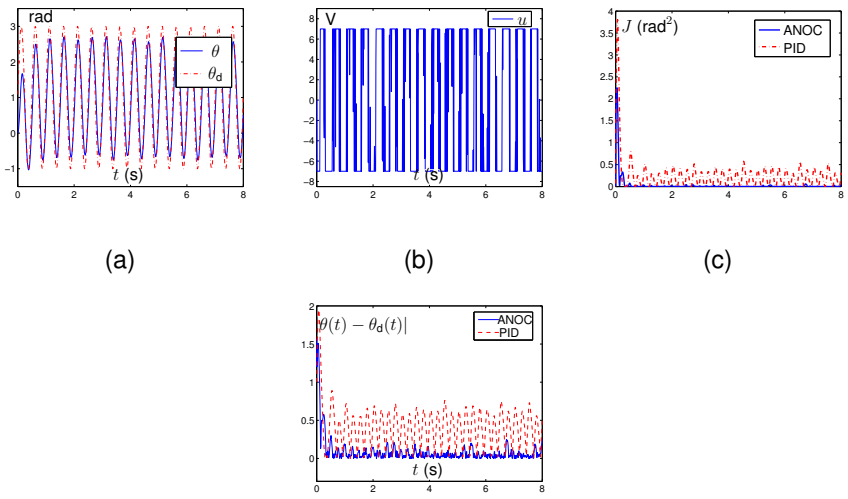



Figure 3.12: Data profiles during the experiment of the tracking control of the DC motor by the PID controller and its performance comparison with the proposed adaptive near-optimal controller (ANOC) with saturation (3.19). (a) Time history of motor angle $\theta(t)$ and desired angle $\theta_{\mathsf{d}}(t)$. (b) Time histories of input $u$ generated by the PID controller with saturation. (c) Comparison of performance index $J$ defined in (3.3) when different controllers are used. (d) Comparison of tracking error $|\theta(t) - \theta_{\mathsf{d}}(t)|$.

source Arduino software Arduino IDE. A block schematic diagram is shown Fig. 3.10 to better illustrate the proposed motor control system.

The DC motor system is essentially a nonlinear system owing to the existence of dead-zone and friction, both of which are nonlinear terms. As it is difficult to model dead-zone and friction effect accurately, we use a term $c_3$ in (3.18) to represent the sum of the nonlinear terms about the dead-zone and friction effects [83], including other unmodelled disturbances. In the experiment, the motor system dynamics is modelled as follows:

$$\begin{cases} \dot{\theta} = \omega, \\ \dot{\omega} = -c_1\omega + c_2 u + c_3, \end{cases} \tag{3.18}$$

where $\theta$ and $\omega$ denotes the motor angle and angle velocity respectively; $c_1 > 0$ and

$c_2 > 0$ are system parameters, and $c_3$ denotes the sum of disturbances and unmodelled terms, such as frictions; $u$ denotes the input voltage of the motor. The system model is based based on the Newton's law. Multiplying both sides of the second sub-equation of (3.18) with the mass yields the result obtained via the Newton's law. Note that, for the proposed method, the focus is not on identifying the parameter values of the system, but on realizing the LNOC. Evidently, the system has a well-defined relative degree $\rho = 2$. The model can be viewed as an extended version of the one in [76] by adding a disturbance term $c_3$. In the experiment, the term $\omega$, i.e., $\dot{\theta}$, is obtained via the tracking differentiator (3.16) in the manuscript with $\theta$ being the input. The obtained value of $\omega$ is denoted as $\omega_{\mathsf{e}}$. In the experiment, the motor angle $\theta$ is expected to track a desired trajectory $\theta_{\mathsf{d}} = 2\sin(4\pi t) + 1$ rad, the parameters of the performance index is set to $T = 0.001$ s and $Q = 1000$, the parameter of the tracking differentiator is set to $C = 100$, the parameters of the auxiliary system (3.6) is set to $\lambda = 0.1$, $K_1 = \mathsf{diag}([0.2, 0.4]^{\mathsf{T}})$, and $K_2 = 0.1$. As the auxiliary system is directly constructed based on $\omega_{\mathsf{e}}$ in the experiment, $\alpha_i = 0$. As the control method is implemented on the Arduino UNO, the auxiliary system and the tracking differentiator is discretized by using the Euler formula with the step-size being the sampling gap of the control system, which is set to 1 ms. For the sake of safety, the input is given as $u_{\mathsf{s}} = \mathsf{sat}(u)$ where $u$ is calculated by equation (15) in the manuscript and the saturation function $\mathsf{sat}(\cdot)$ is defined as follows:

$$\mathsf{sat}(u) = \begin{cases} 7, \text{ if } u > 7; \\ u, \text{ if } -7 < u < 7; \\ -7, \text{ if } u < -7. \end{cases} \tag{3.19}$$

The saturation function guarantees that the maximal magnitude of the input voltage to the DC motor is 7 V. During the experiment, all the necessary data are sent to a computer every 10 ms by the serial communication port of the Arduino UNO board with the baud rate being 115200. The experimental results are shown in Fig. 3.11. The maximum tracking error is $\max_{t\in[0,8]}(|\theta(t) - \theta_{\mathsf{d}}(t)|) = 1.57$ rad, which is the initial tracking error. The minimum tracking error is $\min_{t\in[0,8]}(|\theta(t) - \theta_{\mathsf{d}}(t)|) \approx 0$ rad. The

Table 3.1: Performance Measures for the Motor Control Experiment Without Artificially Added Load Disturbance

| Performance measure | Value (rad) |
|---|---|
| Maximum tracking error | 1.57* |
| Minimum tracking error | 0 |
| Average tracking error | 0.0108 |
| Standard deviation of tracking error | 0.1963 |

Note*: The maximum tracking error is the initial error.

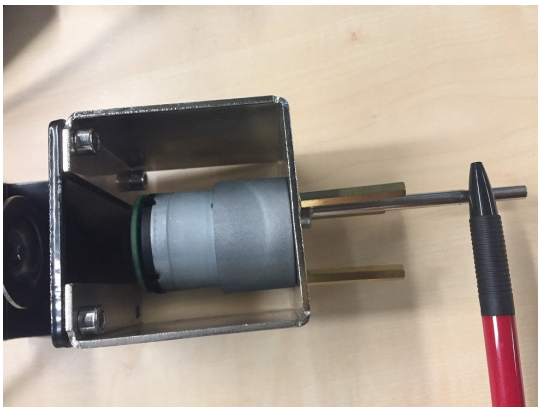


Figure 3.13: The experiment setup regarding the generation of load disturbance.

average tracking error is 0.0108 rad. The standard deviation of the tracking error is 0.1963 rad. The measures are listed in Table 3.1. The results verify the efficacy of the proposed method

Note that, in the experiments, we do not have any prior knowledge about the values of the parameters of the DC motor system, i.e., the parameters are uncertain. The proposed method does not need any prior knowledge about the values of the parameters of the control system.

For comparison, experiments are also conducted by using the following proportional-integral-derivative (PID) controller:

$$u_{\mathsf{s}}(t) = \mathsf{sat}(K_{\mathsf{p}} e(t) + K_{\mathsf{i}} \int_0^t e(\tau)\mathsf{d}\tau + K_{\mathsf{d}} \dot{e}(t)),$$

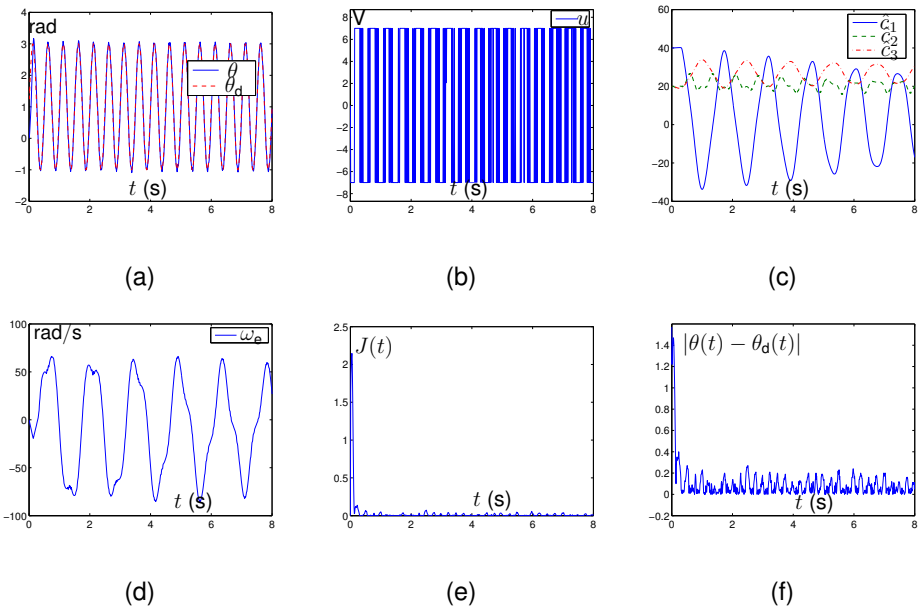



Figure 3.14: Data profiles during the experiment of the tracking control of the DC motor by the proposed LNOC method with saturation (3.19) based on the model (3.18) with artificially added load disturbance as shown in Fig. 3.13. (a) Time history of motor angle $\theta(t)$ and desired angle $\theta_{\mathsf{d}}(t)$. (b) Time histories of input $u$ generated by the proposed controller (3.13) with saturation specified in (3.19). (c) Time histories of parameters estimated by the auxiliary system. (d) Time history of $\omega$ estimated via the the tracking differentiator. (e) Time history of performance index $J(t)$. (f) Time history of tracking error $|\theta(t) - \theta_{\mathsf{d}}(t)|$.

where $e(t) = \theta_{\mathsf{d}}(t) - \theta(t)$. Under the same settings, with $K_{\mathsf{p}} = 45$, $K_{\mathsf{i}} = 0.01$, and $K_{\mathsf{d}} = 7$, which are tuned via the trial-and-error method [79–81], the experimental results are shown in Fig. 3.12. As seen from this figure, when the PID controller is used, the tracking performance becomes worse. Our reasons for the worse performance are as follows. 1) the reference signal used in the experiment is a fast time-varying signal, compared with the general regulation tasks; 2) the used DC motor has a dead-zone. As seen from the figure, the tracking performance of the PID controller becomes worse when the motor needs to change the rotation direction. Particularly, from Fig. 3.12(c), it is observed that the performance index is much better when the proposed LNOC law is used compared with the case with the PID controller.

Table 3.2: Performance Measures for the Motor Control Experiment With Artificially Added Load Disturbance

| Performance measure | Value (rad) |
|---|---|
| Maximum tracking error | 1.57* |
| Minimum tracking error | 0 |
| Average tracking error | 0.0135 |
| Standard deviation of tracking error | 0.1812 |

Note*: The maximum tracking error is the initial error.

We also conduct experiments based on the motor system for the case with load disturbance. The load disturbance is generated via using a pen to introduce more frictions to the motor shaft (as shown in Fig. 3.13). With the same parameter settings stated above, the experimental results when the proposed control method is used are shown in Fig. 3.14. As seen from the figure, the tracking performance is still satisfactory. The experimental data show that the maximum tracking error is $\max_{t\in[0,8]}(|\theta(t) - \theta_{\mathsf{d}}(t)|) = 1.57$ rad, which is the initial tracking error. The minimum tracking error is $\min_{t\in[0,8]}(|\theta(t) - \theta_{\mathsf{d}}(t)|) \approx 0$ rad. The average tracking error is 0.0135 rad. The standard deviation of the tracking error is 0.1812 rad. The measures are also listed in Table 3.2. The values of the measures are similar to the case without artificially added load disturbance.

These experimental results validate the efficacy, superiority, realizability, and online control capability of the proposed method.

Before ending this section, we offer the following remark about the performance verification of the proposed method.

*Remark 5:* Firstly, the performance of the proposed approach is theoretically guaranteed. Secondly, we use vast simulation and experimental results to verify the theoretical results and the performance of the proposed method. Meanwhile, PID control is widely used in DC motor control. The comparison with the PID controller shows the superiority

of the proposed method. As indicated in the experiment, our method does not require other physical measurement compared with the PID control. Meanwhile, our method bears online control capability as PID, which is implemented via a low-cost Arduino UNO control board.

We also have the following remark about the potential of the proposed method in industrial applications.

*Remark 6:* In terms of the DC motor control, our method can be directly applied with the same setting shown in our experiments. It can be expected that with better encoders and microcontrollers, better control performance can be achieved. The comparison of the proposed scheme with commercial DC motor controller products can be further explored.

Regarding the extension of the proposed method to discrete time, we have the following remark.

*Remark 7:* An intuitive method is to perform the Euler difference rule for the system dynamics and so for the auxiliary systems. If we would like to directly formulate the problem in the discrete-time space, meaning that both the performance index and system dynamics are in discrete time. The ideas in the proposed method could be used, i.e., using some expansion approach for the system output to have the control input term, and approximating the performance index via the expansion. This could be further investigated.

## 3.7 Chapter summary

In this chapter, an LNOC law based on a sliding-mode auxiliary system has been designed and proposed for nonlinear systems of fully-unknown parameters. Theoretical analysis has shown that the sliding-mode auxiliary system asymptotically reconstructs the dynamics of the controlled nonlinear system. Rigorous analysis has also shown

that the proposed intelligent near-optimal control law simultaneously guarantees the asymptotic stability of the closed-system and the asymptotic optimality of the performance index. Furthermore, two illustrative examples and an application to a van de Pol oscillator have validated the efficacy of the proposed LNOC law. The realizability, performance, and superiority of the proposed method has also been validated through physical experiments based on a DC motor.

# Chapter 4

# LNOC for a class of nonlinear systems with learning of system dynamics

In this chapter, a novel model-free learning and near-optimal control method is proposed for nonlinear systems via utilizing the Taylor expansion based problem relaxation, the universal approximation property of sigmoid neural networks, and the concept of sliding-mode control.[1] By making approximation for the performance index, it is first relaxed to a quadratic program, and then a linear algebraic equation with unknown terms. An auxiliary system is designed to reconstruct the input-to-output property of the control systems with unknown dynamics, so as to tackle the difficulty caused by the unknown terms. Then, by considering the property of the sliding-mode surface, an explicit LNOC law is derived from the linear algebraic equation. Theoretical analysis shows that the auxiliary system is convergent, the resultant closed-loop system is asymptotically stable, and the performance index asymptomatically converges to optimal. An illustrative example and experimental results are presented, which substantiate the efficacy of the proposed method and verify the theoretical results.

[1]The content in this chapter has already been published. Yinyan Zhang, Shuai Li, and Xiaoping Liu, "Neural network-based model-free adaptive near-optimal tracking control for a class of nonlinear systems," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 29, no. 12, pp. 6227–6241, 2018.

## 4.1 Introduction

Owing to the universal approximation capability, neural network based adaptive control methods have received more and more research interests in recent decades. For example, based on the backstepping technique, Wang *et al.* [86] proposed a neural network adaptive tracking control algorithm for uncertain nonlinear systems with unmodeled dynamics in the nonlower triangular form, which guarantees semi-global boundedness of all signals of the resultant closed-loop system. By combining backstepping and dynamic surface control techniques as well as the universal approximation capability of neural networks, Zhou *et al.* [87] proposed a novel approximation-based adaptive tracking control method for strict-feedback nonlinear systems with input saturation. The method proposed in [87] was later extended to a class of nonstrict-feedback systems [88]. Considering that backstepping based adaptive control methods may encounter the explosion of complexity and circular issues, Na *et al.* [89] proposed an adaptive control method for nonlinear pure-feedback systems without using the backstepping technique. The result in [89] indicates that, by utilizing coordinate state transformation, the state feedback control of pure-feedback systems can be transformed into the output-feedback control of canonical systems. Neural networks have also been applied to the control of robotic manipulators at the kinematics level or the dynamics level [95–97]. In [98], the adaptive neural control for the attitude and position of a flapping wing micro aerial vehicle was reported, where a disturbance observer is also adopted. In [99], a neural network based control method was proposed for a piezoelectric-actuated stick-lip device. It is observed that most of the existing adaptive control methods do not consider optimality.

In this chapter, we consider the receding-horizon optimal tracking control problem of a class of continuous-time nonlinear systems with fully unknown dynamics, which include the strict feedback nonlinear systems as considered in [105] for instance as special cases. The problem is relaxed to a quadratic program via output prediction with the

aid of Taylor expansion and the universal approximation capability of sigmoid neural networks and a linear algebraic equation with unknown terms is thus derived. Based on the sliding-mode control concept [89–91], an auxiliary system is designed to reconstruct the input-to-output property of the systems. By considering the property of the sliding-mode surface, an LNOC law is derived from the linear algebraic equation.

The rest of this chapter is organized in the following manner. In Section 4.2, the investigated problem is presented and some preliminaries are provided. In Section 4.3, the design process of the proposed model-free LNOC law is illustrated. In Section 4.4, theoretical results are provided to guarantee the performance of the proposed method. In Section 4.5, an illustrative examples is shown and discussed to substantiate the efficacy of the proposed method and to verify the theoretical results. In Section 4.6, the experimental validation for the performance of the proposed method is shown. In Section 4.7, conclusions for this chapter are provided.

## 4.2 Problem description and preliminary

In this section, the problem investigated in this chapter is presented. Besides, some helpful preliminaries are provided.

### 4.2.1 Problem description

In this chapter, the following class of nonlinear systems is considered:

$$\begin{cases} \dot{\mathbf{x}}(t) = f(\mathbf{x}(t)) + g(\mathbf{x}(t))u(t), \\ y(t) = h(\mathbf{x}(t)), \end{cases} \tag{4.1}$$

where $\mathbf{x}(t) = [x_1(t), x_2(t), \cdots, x_n(t)]^{\mathsf{T}} \in \mathbb{R}^n$ denotes the state vector, $y(t) \in \mathbb{R}$ denotes the system output, and $u(t) \in \mathbb{R}$ denotes the system input; $f(\cdot) : \mathbb{R}^n \to \mathbb{R}^n$, $g(\cdot) : \mathbb{R}^n \to \mathbb{R}^n$, and $h(\cdot) : \mathbb{R}^n \to \mathbb{R}$ denote unknown smooth functions. Note that $g(\mathbf{x}(t))$ is

referred to as the input gain. Without loss of generality, we assume that $g(\mathbf{x}(t)) \neq 0$ for any time instant $t$ and that the system output $y(t)$ has a well-defined relative degree $\rho$ [6, 39, 86, 109]. Note that examples of this class of systems with known relative degrees independent from unknown functions include many mechanical systems [90] and strict-feedback nonlinear systems [92]. Throughout this chapter, the standard Lie derivative notation and the definition of relative degree are utilized [39]. By the definition of relative degree, $L_g L_f^{\rho-1} h(\mathbf{x}(t)) \neq 0$ for all $\mathbf{x}$. Without loss of generality, in this chapter, we assumed that $L_g L_f^{\rho-1} h(\mathbf{x}(t)) \geq d_0 > 0$, where $d_0$ is a known small positive number.

In this chapter, the objective is to design a control law for system (4.1) such that the tracking error defined by $e(t) = y_{\mathsf{r}}(t) - y(t)$ between system output $y(t)$ and the reference output $y_{\mathsf{r}}(t)$ asymptotically converges to zero under the assumption that the $k$th-order time derivatives $y_{\mathsf{r}}^{[k]}(t)$, with $k = 0, 1, \cdots, \rho$, and $y_{\mathsf{r}}^{[0]}(t) = y_{\mathsf{r}}(t)$ are continuous and bounded. Considering that the integral action is capable of enhancing system tracking performance at the steady state as well as robustness against uncertainties and noise [100], the control objective is formulated as the following receding-horizon optimal tracking control problem [6, 100–103]:

$$\begin{aligned} \text{minimize}_{u(t)} \quad & J(t) \\ \text{subject to} \quad & \dot{\mathbf{x}}(t) = f(\mathbf{x}(t)) + g(\mathbf{x}(t))u(t), \\ & y(t) = h(\mathbf{x}(t)), \end{aligned} \tag{4.2}$$

where performance index $J(t)$ is defined as

$$J(t) = \int_0^T (y_{\mathsf{r}}(t+\tau) - y(t+\tau))^2 \mathsf{d}\tau, \tag{4.3}$$

where constant $T > 0 \in \mathbb{R}$ denotes the predictive period for each time instant $t$.

*Remark 1:* The main challenges for the problem considered in this chapter lie in two aspects.

1) Receding-horizon optimal tracking control problem (4.7) is a nonlinear optimization problem with an integration-type performance index, an ordinary differential

equation constraint, and an algebraic equation constraint. In this optimization problem, the decision variable is input $u(t)$, which is not explicitly included in the performance index. This problem is difficult to solve analytically, which requires the solution of nonlinear Hamilton-Jacobi-Bellman equations [6]. Up to today, there are no systematic approaches in existing literature to solve it accurately. Besides, numerically solving this complex optimization problem at each time instant is computationally intensive [6, 101].

2) Different from the cases considered in [6, 100–103], in this chapter, the system dynamics are fully unknown, i.e., $f(\mathbf{x}(t))$, $g(\mathbf{x}(t))$, and $h(\mathbf{x}(t))$ are fully unknown.

Regarding the performance index adopted in this chapter, we have the following remark.

*Remark 2:* A performance index with the control input taken into account was provided in [113], i.e., $J(t) = \nu(y_{\mathrm{r}}(t+\tau) - y(t+\tau)) + a\int_0^T (y_{\mathrm{r}}(t+\tau) - y(t+\tau))^2 \mathrm{d}\tau + b\int_0^T (u_{\mathrm{r}}(t+\tau) - u(t+\tau))^2 \mathrm{d}\tau$, where $a > 0 \in \mathbb{R}$, $b > 0 \in \mathbb{R}$, and $\nu(\cdot)$ needs to be a continuous and differentiable function with $\nu(0) = 0$ and $\nu(x) > 0, \forall x \neq 0$; $u_r(t)$ is the input function with which the output of the system $\dot{\mathbf{x}} = f(\mathbf{x}) + g(\mathbf{x})u_{\mathrm{r}}(t)$, $y(t) = h(\mathbf{x}(t))$ satisfies $y(t) = y_{\mathrm{r}}(t)$. Evidently, the physical meaning of such a performance index is not clear and it requires the knowledge of $u_{\mathrm{r}}(t)$. Actually, under system uncertainty, it is more difficult to obtain $u_{\mathrm{r}}(t)$. In practice, using the performance index (4.3) means that the control accuracy is far more important than the energy consumption.

### 4.2.2 Sigmoid neural network

Neural networks are well known for their function approximation capability. In this chapter, we use single-hidden-layer sigmoid neural networks to approximate any unknown function $F_{\mathrm{u}}(\mathbf{z}) : \mathbb{R}^n \to \mathbb{R}$ to facilitate the model-free LNOC of system (4.1). The consid-

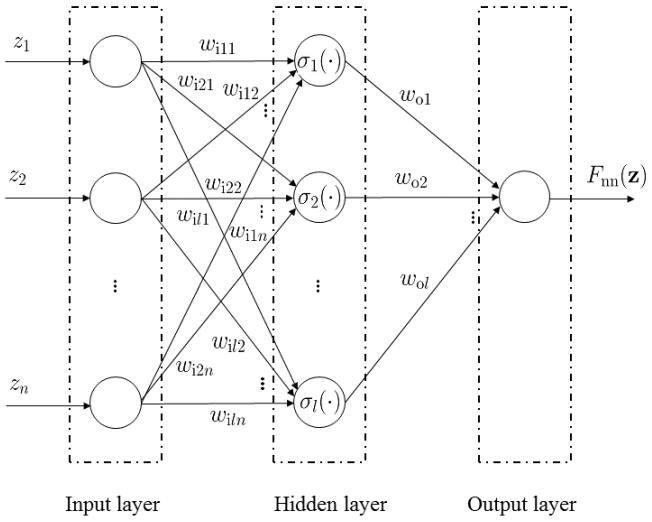



Figure 4.1: Block diagram of single-hidden-layer sigmoid neural network (4) with $l$ neurons in the hidden layer.

ered sigmoid neural network takes the following form:

$$F_{\mathsf{nn}}(\mathbf{z}) = \sum_{j=1}^{l} w_{\mathsf{o}j}\sigma_j(\mathbf{w}_{\mathsf{i}j}^{\mathsf{T}}\mathbf{z}), \tag{4.4}$$

where $\mathbf{z} = [z_1, z_2, \cdots, z_n]^{\mathsf{T}} \in \mathbb{R}^n$ denotes the input vector of the sigmoid neural network; $\mathbf{w}_{\mathsf{o}} = [w_{\mathsf{o}1}, w_{\mathsf{o}2}, \cdots, w_{\mathsf{o}n}]^{\mathsf{T}} \in \mathbb{R}^n$ is the weight vector of the output-layer; $\mathbf{w}_{\mathsf{i}j} = [w_{\mathsf{i}j1}, w_{\mathsf{i}j2}, \cdots, w_{\mathsf{i}jn}]^{\mathsf{T}} \in \mathbb{R}^l$ is the input-layer weight vector with respect to the $j$th input neuron with $j = 1, 2, \cdots, n$; $\sigma_j(\cdot)$ is the sigmoid activation function of the $j$th hidden neuron, which is defined as

$$\sigma_j(x) = \frac{1}{1 + \exp(-c_j(x - b_j))}, \tag{4.5}$$

where parameter $c_j > 0 \in \mathbb{R}$ characterizes the sharpness of the $j$th sigmoid function and $b_j \in \mathbb{R}$ is a bias term determining the center of the $j$th sigmoid function. A block diagram of the sigmoid neural network is shown in Fig. 4.1.

*Lemma 1* (Universal Approximation [94, 114, 115]): Consider sigmoid neural network (4.4) with sufficient number of neurons. If parameters $c_j$ and $b_j$ of sigmoid activation function (4.5), and $\mathbf{w}_{\mathsf{i}j}$ are randomly chosen, then for any continuous function $F : \mathbb{R}^n \rightarrow$

$\mathbb{R}$, for any $\varepsilon_0 > 0$, there exist an positive integer $l$ and an optimal weight vector $\mathbf{w}_o^*$ such that

$$\sup_{\mathbf{z}\in\mathbb{R}^n} |F(\mathbf{z}) - \sum_{j=1}^{l} w_{\mathsf{o}j}^* \sigma_j(\mathbf{w}_{\mathsf{i}j}^{\mathsf{T}}\mathbf{z})| \leq \varepsilon_0.$$

*Remark 3*: Lemma 1 is also referred to as the universal approximation property. It indicates that, if the number of neurons is sufficiently large, sigmoid neural network (4.4) can approximate any continuous function with any degree of accuracy. It follows that the difficulty caused by unknown dynamics arising in the control design for nonlinear systems can be addressed by utilizing sigmoid neural networks. In addition, Lemma 1 also indicates that when utilizing sigmoid neural networks to approximate continuous functions, constant weight vectors $\mathbf{w}_{\mathsf{i}j}$, and constant parameters $c_j$ and $b_j$ of sigmoid activation function (4.5) can be randomly chosen.

### 4.2.3 Problem reformulation

For system (4.1) with relative degree $\rho$, based on the definitions of relative degree and Lie derivatives [39], one readily has

$$y^{[\rho]}(t) = L_f^{\rho} h(\mathbf{x}(t)) + L_g L_f^{\rho-1} h(\mathbf{x}(t)) u(t).$$

By the universal approximation property (see Lemma 1) of sigmoid neural networks, ignoring the approximation error $\varepsilon_0$ [110], if the number of neurons is sufficiently large, then there exist ideal unknown weight vectors $\mathbf{w}_{\mathsf{o}}^*$ and $\mathbf{w}_{\mathsf{o}}'^*$ such that $L_f^{\rho} h(\mathbf{x}(t))$ and $L_g L_f^{\rho-1} h(\mathbf{x}(t))$ can be represented as $L_f^{\rho} h(\mathbf{x}(t)) = \sum_{j=1}^{l} w_{\mathsf{o}j}^* \sigma_j(\mathbf{w}_{\mathsf{i}j}^{\mathsf{T}}\mathbf{x}(t))$ and $L_g L_f^{\rho-1} h(\mathbf{x}(t)) = \sum_{j=1}^{l} w_{\mathsf{o}j}'^* \sigma_j(\mathbf{w}_{\mathsf{i}j}^{\mathsf{T}}\mathbf{x}(t))$, respectively. It follows that the input-to-output relationship of nonlinear system (4.1) can be reformulated as follows:

$$y^{[\rho]}(t) = \sum_{j=1}^{l} w_{\mathsf{o}j}^* \sigma_j(\mathbf{w}_{\mathsf{i}j}^{\mathsf{T}}\mathbf{x}(t)) + \sum_{j=1}^{l} w_{\mathsf{o}j}'^* \sigma_j(\mathbf{w}_{\mathsf{i}j}^{\mathsf{T}}\mathbf{x}(t)) u(t). \tag{4.6}$$

Then, problem (4.2) is reformulated as

$$\begin{aligned} &\text{minimize}_{u(t)} \qquad J(t) \\ &\text{subject to} \quad y^{[\rho]}(t) = \sum_{j=1}^{l} w_{\mathsf{o}j}^{*}\sigma_j(\mathbf{w}_{\mathsf{i}j}^{\mathsf{T}}\mathbf{x}(t)) + \sum_{j=1}^{l} w_{\mathsf{o}j}^{\prime *}\sigma_j(\mathbf{w}_{\mathsf{i}j}^{\mathsf{T}}\mathbf{x}(t))u(t), \end{aligned} \tag{4.7}$$

where $\mathbf{w}_{\mathsf{o}}^{*}$ and $\mathbf{w}_{\mathsf{o}}^{\prime *}$ are unknown.

## 4.3 Control design

In this section, the control design process is illustrated. Since the parameters of system (4.6) are unknown, we first design an auxiliary system with the aid of sliding-mode control to reconstruct the input-to-output property of the system. Then, based on the property of the sliding-mode surface, a near-optimal control law is designed and proposed. For the convenience of illustration and readability, the corresponding theoretically analysis is presented in the next section.

### 4.3.1 Problem relaxation

In this subsection, the receding-horizon optimal tracking control problem (4.7) is successively relaxed to an unconstrained quadratic program and a linear algebraic equation with unknown terms.

By Taylor expansion, the system output at time instant $t+\tau$ can be predicted via the one at time instant $t$, i.e.,

$$y(t+\tau) \approx y(t) + \tau\dot{y}(t) + \frac{\tau^2}{2!}y^{[2]}(t) + \cdots + \frac{\tau^{\rho}}{\rho!}y^{[\rho]}(t), \tag{4.8}$$

where $y^{[i]}(t)$ denotes the $i$-th order time derivative of $y(t)$ with $i = 0, 1, 2, \cdots, \rho$ and $y^{[0]}(t) = y(t)$. Substituting equation (4.6) into equation (4.8) yields

$$y(t+\tau) \approx \varrho^{\mathsf{T}}(\tau)\Upsilon(t) + \frac{\tau^{\rho}}{\rho!}\sum_{j=1}^{l} w_{\mathsf{o}j}^{\prime *}\sigma_j(\mathbf{w}_{\mathsf{i}j}^{\mathsf{T}}\mathbf{x}(t))u(t), \tag{4.9}$$

where

$$\Upsilon(t) = [y(t), \dot{y}(t), \cdots, y^{[\rho-1]}, \sum_{j=1}^{l} w_{\mathsf{o}j}^{*} \sigma_j(\mathbf{w}_{\mathsf{i}j}^{\mathsf{T}} \mathbf{x}(t))]^{\mathsf{T}}$$

and

$$\varrho(\tau) = [1, \tau, \tau^2/2, \cdots, \tau^{\rho}/\rho!]^{\mathsf{T}}.$$

Similarly, the reference output at time instant $t+\tau$ can be predicted as follows:

$$y_{\mathsf{r}}(t+\tau) \approx \varrho^{\mathsf{T}}(\tau) \Upsilon_{\mathsf{r}}(t), \tag{4.10}$$

where $\Upsilon_{\mathsf{r}}(t) = [y_{\mathsf{r}}(t), \dot{y}_{\mathsf{r}}(t), \cdots, y_{\mathsf{r}}^{[\rho]}(t)]^{\mathsf{T}}$.

Let $\check{\Upsilon}(t) = \Upsilon_{\mathsf{r}}(t) - \Upsilon(t)$. Based on equations (4.9) and (4.10), performance index (4.3) is relaxed in the following manner:

$$\begin{aligned}
J(t) &\approx \hat{J}(t) \\
&= \frac{1}{2} \int_0^T (\varrho^{\mathsf{T}}(\tau) \check{\Upsilon}(t) - \frac{\tau^{\rho}}{\rho!} \sum_{j=1}^{l} w_{\mathsf{o}j}^{\prime *} \sigma_j(\mathbf{w}_{\mathsf{i}j}^{\mathsf{T}} \mathbf{x}(t)) u(t))^2 \mathsf{d}\tau \\
&= \frac{1}{2} \int_0^T \Bigg( (\varrho^{\mathsf{T}}(\tau) \check{\Upsilon}(t))^2 - \frac{2\tau^{\rho}}{\rho!} \varrho^{\mathsf{T}}(\tau) \check{\Upsilon}(t) \sum_{j=1}^{l} w_{\mathsf{o}j}^{\prime *} \sigma_j(\mathbf{w}_{\mathsf{i}j}^{\mathsf{T}} \mathbf{x}(t)) u(t) \\
&\quad + \frac{\tau^{2\rho}}{(\rho!)^2} (\sum_{j=1}^{l} w_{\mathsf{o}j}^{\prime *} \sigma_j(\mathbf{w}_{\mathsf{i}j}^{\mathsf{T}} \mathbf{x}(t)))^2 u^2(t) \Bigg) \mathsf{d}\tau \\
&= \frac{1}{2} \int_0^T (\varrho^{\mathsf{T}}(\tau) \check{\Upsilon}(t))^2 \mathsf{d}\tau - \varrho_{\mathsf{int}} \check{\Upsilon}(t) \sum_{j=1}^{l} w_{\mathsf{o}j}^{\prime *} \sigma_j(\mathbf{w}_{\mathsf{i}j}^{\mathsf{T}} \mathbf{x}(t)) u(t) + \frac{T^{2\rho+1}}{2(2\rho+1)(\rho!)^2} (\sum_{j=1}^{l} w_{\mathsf{o}j}^{\prime *} \\
&\quad \cdot \sigma_j(\mathbf{w}_{\mathsf{i}j}^{\mathsf{T}} \mathbf{x}(t)))^2 u^2(t),
\end{aligned}$$

where constant vector $\varrho_{\mathsf{int}}$ is calculated by

$$\varrho_{\mathsf{int}} = \int_0^T \tau^{\rho} \varrho^{\mathsf{T}}(\tau)/\rho! \mathsf{d}\tau = \left[ \frac{T^{\rho+1}}{(\rho+1)\rho!}, \frac{T^{\rho+2}}{(\rho+2)\rho!1!}, \cdots, \frac{T^{2\rho+1}}{(2\rho+1)(\rho!)^2} \right]. \tag{4.11}$$

Evidently, $\hat{J}(t)$ is a quadratic performance index with system input $u(t)$ being the decision variable. In other words, the complicated nonlinear optimization problem (4.7) is relaxed to the following unconstrained quadratic program:

$$\min_{u(t)} \quad \frac{1}{2} p(t) u^2(t) - q(t) u(t) + c(t),$$

where

$$p(t) = \frac{T^{2\rho+1}(\sum_{j=1}^{l} w_{\mathsf{o}j}'^{*}\sigma_j(\mathbf{w}_{\mathsf{i}j}^{\mathsf{T}}\mathbf{x}(t)))^2}{(2\rho+1)(\rho!)^2},$$

$$q(t) = \varrho_{\mathsf{int}}\check{\Upsilon}(t)\sum_{j=1}^{l} w_{\mathsf{o}j}'^{*}\sigma_j(\mathbf{w}_{\mathsf{i}j}^{\mathsf{T}}\mathbf{x}(t)),$$

and

$$c(t) = \frac{1}{2}\int_0^T (\varrho^{\mathsf{T}}(\tau)\check{\Upsilon}(t))^2 \mathsf{d}\tau.$$

Note that $c(t)$ can be removed from the unconstrained quadratic program since $u(t)$ is not explicitly included in the expression of $c(t)$, which does not affect the solution to the program [116]. Since $p(t) \geq 0$, the quadratic program is convex. Then, the optimal input can be obtained via solving $\partial\hat{J}(t)/\partial u = 0$, i.e.,

$$-\varrho_{\mathsf{int}}\check{\Upsilon}(t) + \frac{T^{2\rho+1}\sum_{j=1}^{l} w_{\mathsf{o}j}'^{*}\sigma_j(\mathbf{w}_{\mathsf{i}j}^{\mathsf{T}}\mathbf{x}(t))u(t)}{(2\rho+1)(\rho!)^2} = 0,$$

from which the following equation is derived:

$$\begin{aligned}
&-\sum_{i=0}^{\rho-1}\frac{T^i}{(\rho+1+i)i!}(y_{\mathsf{r}}^{[i]}(t) - y^{[i]}(t)) - \frac{T^\rho}{(2\rho+1)\rho!}y_{\mathsf{r}}^{[\rho]}(t) \\
&+\frac{T^\rho}{(2\rho+1)\rho!}(\sum_{j=1}^{l} w_{\mathsf{o}j}^{*}\sigma_j(\mathbf{w}_{\mathsf{i}j}^{\mathsf{T}}\mathbf{x}(t)) + \sum_{j=1}^{l} w_{\mathsf{o}j}'^{*}\sigma_j(\mathbf{w}_{\mathsf{i}j}^{\mathsf{T}}\mathbf{x}(t))u(t)) = 0. \qquad (4.12)
\end{aligned}$$

By the above steps, receding-horizon optimal tracking control problem (4.7) for nonlinear system (4.6) is relaxed to an algebraic equation with unknown terms $\mathbf{w}_{\mathsf{o}}^{*}$ and $\mathbf{w}_{\mathsf{o}}'^{*}$.

*Remark 4:* The linear algebraic equation is a result of approximating performance index (4.3) with all the constraints in the receding-horizon optimal tracking control problem considered. Specifically, during the problem relaxation process, the algebraic equation constraint and the ordinary differential equation constraint are incorporated when approximating the system output at future time instant $t + \tau$.

### 4.3.2 Reconstruction of input-to-output dynamics

Form the previous subsection, the solution to equation (4.12) depends on the fully unknown terms $\mathbf{w}_{\mathsf{o}}^{*}$ and $\mathbf{w}_{\mathsf{o}}^{\prime *}$, which are closely related to the input-to-output property of nonlinear system (4.6). In this subsection, an auxiliary system is designed and proposed to reconstruct the input-to-output dynamics of system (4.6), which makes solving equation (4.12) feasible.

Evidently, the problem becomes finding a group of $\mathbf{w}_{\mathsf{o}j}(t)$ and $\mathbf{w}_{\mathsf{o}j}^{\prime}(t)$ to satisfy the following condition:

$$\lim_{t\to+\infty}((\sum_{j=1}^{l} w_{\mathsf{o}j}(t)\sigma_j(\mathbf{w}_{\mathsf{i}j}^{\mathsf{T}}\mathbf{x}(t)) + \sum_{j=1}^{l} w_{\mathsf{o}j}^{\prime}(t)\sigma_j(\mathbf{w}_{\mathsf{i}j}^{\mathsf{T}}\mathbf{x}(t))u(t)) - (\sum_{j=1}^{l} w_{\mathsf{o}j}^{*}\sigma_j(\mathbf{w}_{\mathsf{i}j}^{\mathsf{T}}\mathbf{x}(t)) + \sum_{j=1}^{l} w_{\mathsf{o}j}^{\prime *}\sigma_j(\mathbf{w}_{\mathsf{i}j}^{\mathsf{T}}\mathbf{x}(t))u(t))) = 0.$$

To this end, we design the following auxiliary system:

$$\begin{cases} \hat{y}^{[\rho]}(t) = \sum_{j=1}^{l} w_{\mathsf{o}j}(t)\sigma_j(\mathbf{w}_{\mathsf{i}j}^{\mathsf{T}}\mathbf{x}(t)) + \sum_{j=1}^{l} w_{\mathsf{o}j}^{\prime}(t)\sigma_j(\mathbf{w}_{\mathsf{i}j}^{\mathsf{T}}\cdot\mathbf{x}(t))u(t) - \sum_{j=0}^{\rho-2}\alpha_j\tilde{y}^{[j+1]}(t) \\ \qquad\qquad - \lambda\phi(s(t)), \\ \dot{w}_{\mathsf{o}j}(t) = -\gamma s(t)\sigma_j(\mathbf{w}_{\mathsf{i}j}^{\mathsf{T}}\mathbf{x}(t)),\ j = 1, 2, \cdots, l, \\ \dot{w}_{\mathsf{o}j}^{\prime}(t) = -\gamma s(t)\sigma_j(\mathbf{w}_{\mathsf{i}j}^{\mathsf{T}}\mathbf{x}(t))u(t) - p_j(t),\ j = 1, 2, \cdots, l, \end{cases} \tag{4.13}$$

where $s(t) = \sum_{j=0}^{\rho-1}\alpha_j\tilde{y}^{[j]}(t)$ with $\tilde{y}(t) = \hat{y}(t) - y(t)$, $\alpha_{\rho-1} = 1$, and $\alpha_j > 0$ for $j = 0, 1, \cdots, \rho-2$; $\phi(\cdot) : \mathbb{R} \to \mathbb{R}$ is a monotonically increasing odd activation function; $\lambda > 0 \in \mathbb{R}$ is a parameter used to scale the output response to its displacement compared with the output of system (4.6); $\gamma > 0 \in \mathbb{R}$ is a parameter used to scale the parameter response to the output difference between auxiliary system (4.13) and system (4.6). In addition, $p_j(t)$ is the $j$th element of $\mathbf{p}(\mathbf{w}_{\mathsf{o}}^{\prime}(t))$ and

$$\mathbf{p}(\mathbf{w}_{\mathsf{o}}^{\prime}(t)) = \begin{cases} \mathbf{w}_{\mathsf{o}}^{\prime}(t), \text{ if } \mathbf{w}_{\mathsf{o}}^{\prime}(t) \in \Omega, \\ \|\gamma s(t)\sigma_{\mathsf{vec}}(t)u(t)\|_2\left(\dfrac{(\mathbf{w}_{\mathsf{o}}^{\prime}(t))^{\mathsf{T}}\sigma_{\mathsf{vec}}(t)}{\|\sigma_{\mathsf{vec}}(t)\|_2} - \dfrac{d_0}{\|\sigma_{\mathsf{vec}}(t)\|_2}\right)\dfrac{\sigma_{\mathsf{vec}}(t)}{\|\sigma_{\mathsf{vec}}(t)\|_2}, \text{ if } \mathbf{w}_{\mathsf{o}}^{\prime}(t) \notin \Omega, \end{cases} \tag{4.14}$$

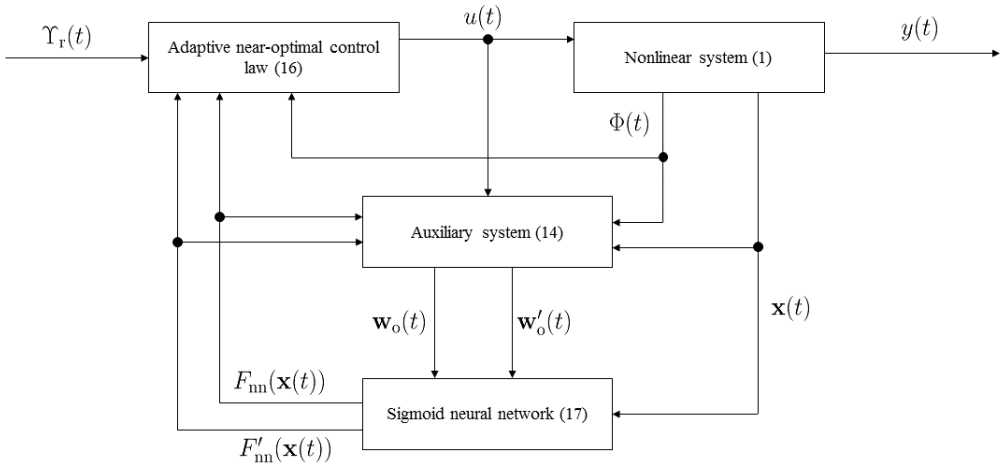



Figure 4.2: Block diagram for the implementation of the proposed LNOC for nonlinear system (4.1) with fully unknown dynamics.

where we have set $\Omega = \{\mathbf{w}'_{\rm o} \in \mathbb{R}^l | \sum_{j=1}^{l} w'_{{\rm o}j}(t)\sigma_j(\mathbf{w}_{{\rm i}j}^{\rm T}\mathbf{x}(t)) \geq d_0\}$, vector $\sigma_{\rm vec}(t) = [\sigma_1(\mathbf{w}_{\rm i1}^{\rm T}\mathbf{x}(t)), \sigma_2(\mathbf{w}_{\rm i2}^{\rm T}\mathbf{x}(t)), \cdots, \sigma_l(\mathbf{w}_{{\rm i}l}^{\rm T}\mathbf{x}(t))]^{\rm T}$, and $\|\cdot\|_2$ denotes the 2-norm of a vector. By the definition of the sigmoid function shown in equation (4.5), each element of $\sigma_{\rm vec}(t)$ is strictly bigger than zero for any time instant $t$. It follows that $\|\sigma_{\rm vec}(t)\|_2 > 0$ for any $t$, i.e., $\mathbf{p}(\mathbf{w}'_{\rm o}(t))$ is well-defined.

We offer the following remark concerning the underlying intuitions in the design of auxiliary system (4.13).

*Remark 5:* The design of auxiliary system (4.13) is inspired by the progresses of sliding-mode control [89–91, 117] and dynamical neural networks [118, 119]. In auxiliary system (4.13), the sliding-mode surface is defined by $s(t) = 0$. The parameter evolution rules in (4.13) are designed for the consideration to guarantee that $s(t)$ asymptotically converges to zero, which is theoretically analyzed latter on. On the sliding-mode surface, by properly choosing parameters $\alpha_j$ for $j = 0, 1, \cdots, \rho - 1$, such that all the roots of characteristic equation $\sum_{j=0}^{\rho-1} \alpha_j \nu^j$ are located on the left half-plane, $\tilde{y}(t) = 0$ is asymptotically stable [120]. It is worth pointing out that unlike [89–91, 117], the sliding-mode concept is utilized in this chapter to facilitate the dynamics reconstruction so as to solve the aforementioned linear algebraic equation with unknown terms, instead of

designing sliding-mode controllers or state observers. Considering that nonlinear activation functions are widely used in dynamical neural networks (e.g., [119]), which are viewed as a powerful tool to enhance convergence of dynamical systems, nonlinear activation function $\phi(\cdot)$ is incorporated in the proposed auxiliary system. In addition, the motivation to introduce the bias term $p_j(t)$ is to guarantee that, during the evolution of the parameter adaptation, $\sum_{j=1}^{l} w'_{\mathsf{o}j}(t)\sigma_j(\mathbf{w}_{\mathsf{i}j}^{\mathsf{T}}\mathbf{x}(t)) \geq d_0$, so as to avoid the so-called control singularity problem [107]. The detailed proof is latter provided in Theorem 1.

The following remark is about the selection of parameters and the activation function in the auxiliary system.

*Remark 6*: As seen from auxiliary system (4.13), together with the selection of constant parameters of sigmoid functions (which is discussed in Remark 1) and $\alpha_j$ as discussed in Remark 5, there are some other constant parameters to be selected, i.e., $\lambda$ and $\gamma$. Since $\lambda$ and $\gamma$ can be viewed as gain parameters, which have similar effects on the response of the system output or parameter adaptation as that of the proportional parameter in the traditional proportional control on system responses [121]. As a result, a larger value of $\lambda$ and $\gamma$ would lead to faster convergence, but may also lead to larger overshooting. In this sense, they should not be too large. Besides, since the main role of the auxiliary system is to capture the input-to-output dynamics of system (4.6) via the evolution of parameters $w_{\mathsf{o}j}(t)$ and $w'_{\mathsf{o}j}(t)$ (with $j = 1, 2, \cdots, l$), the value of $\gamma$ should not be smaller than that of $\lambda$. In terms of monotonically increasing odd activation function $\phi(\cdot)$, there are many alternatives, some of which can be seen from [119, 122, 123].

### 4.3.3 Control law

In this subsection, based on auxiliary system (4.13) and algebraic equation (4.12), an LNOC law is derived.

From auxiliary system (4.13), on the sliding-mode surface defined by $s(t) = 0$, it can

be derived that

$$\sum_{j=1}^{l} w_{\mathsf{o}j}(t)\sigma_j(\mathbf{w}_{\mathsf{i}j}^{\mathsf{T}}\mathbf{x}(t)) + \sum_{j=1}^{l} w'_{\mathsf{o}j}(t)\sigma_j(\mathbf{w}_{\mathsf{i}j}^{\mathsf{T}}\mathbf{x}(t))u(t) = \sum_{j=1}^{l} w^*_{\mathsf{o}j}\sigma_j(\mathbf{w}_{\mathsf{i}j}^{\mathsf{T}}\mathbf{x}(t)) + \sum_{j=1}^{l} w'^*_{\mathsf{o}j}\sigma_j(\mathbf{w}_{\mathsf{i}j}^{\mathsf{T}}\mathbf{x}(t))u(t)),$$

which is latter theoretically guaranteed by Theorem 2 shown in Section 4.4. Together with equation (4.12), one further has

$$-\sum_{i=0}^{\rho-1}\frac{T^i}{(\rho+1+i)i!}(y_{\mathsf{r}}^{[i]}(t) - y^{[i]}(t)) - \frac{T^\rho}{(2\rho+1)\rho!}y_{\mathsf{r}}^{[\rho]}(t) + \frac{T^\rho}{(2\rho+1)\rho!}\left(\sum_{j=1}^{l} w_{\mathsf{o}j}(t)\sigma_j(\mathbf{w}_{\mathsf{i}j}^{\mathsf{T}}\mathbf{x}(t)) + \sum_{j=1}^{l} w'_{\mathsf{o}j}(t)\sigma_j(\mathbf{w}_{\mathsf{i}j}^{\mathsf{T}}\mathbf{x}(t))u(t)\right) = 0. \quad (4.15)$$

Note that $\mathbf{w}_{\mathsf{o}}(t)$ and $\mathbf{w}'_{\mathsf{o}}(t)$ are generated by the designed auxiliary system and they are thus totally known. Then, by solving equation (4.15), given that $\sum_{j=1}^{l} w'_{\mathsf{o}j}(t)\sigma_j(\mathbf{w}_{\mathsf{i}j}^{\mathsf{T}}\mathbf{x}(t)) \neq 0$, an LNOC law is obtained as follows:

$$u(t) = \frac{-1}{\sum_{j=1}^{l} w'_{\mathsf{o}j}(t)\sigma_j(\mathbf{w}_{\mathsf{i}j}^{\mathsf{T}}\mathbf{x}(t))}(\sum_{i=0}^{\rho-1}\frac{T^i(2\rho+1)\rho!}{(\rho+1+i)i!T^\rho}(y_{\mathsf{r}}^{[i]}(t) - y^{[i]}(t)) - y_{\mathsf{r}}^{[\rho]}(t) - \sum_{j=1}^{l} w_{\mathsf{o}j}(t)\sigma_j(\mathbf{w}_{\mathsf{i}j}^{\mathsf{T}}\mathbf{x}(t))),$$

which can be rewritten in a more concise form:

$$u(t) = \frac{-1}{F'_{\mathsf{nn}}(\mathbf{x}(t))}(\sum_{i=0}^{\rho-1}\frac{T^i(2\rho+1)\rho!}{(\rho+1+i)i!T^\rho}(y_{\mathsf{r}}^{[i]}(t) - y^{[i]}(t)) - y_{\mathsf{r}}^{[\rho]}(t) - F_{\mathsf{nn}}(\mathbf{x}(t))),$$

with

$$\begin{cases} F'_{\mathsf{nn}}(\mathbf{x}(t)) = \sum_{j=1}^{l} w'_{\mathsf{o}j}(t)\sigma_j(\mathbf{w}_{\mathsf{i}j}^{\mathsf{T}}\mathbf{x}(t)), \\ F_{\mathsf{nn}}(\mathbf{x}(t)) = \sum_{j=1}^{l} w_{\mathsf{o}j}(t)\sigma_j(\mathbf{w}_{\mathsf{i}j}^{\mathsf{T}}\mathbf{x}(t)). \end{cases} \quad (4.16)$$

Since the basis functions and input-layer weight vectors are the same for $F_{\mathsf{nn}}(\mathbf{x}(t))$ and $F'_{\mathsf{nn}}(\mathbf{x}(t))$, they are combined into a sigmoid neural network with two outputs. Evidently, the benefit of utilizing the same basis functions lies in the reduction of neurons, which makes the structure of the sigmoid neural network simpler.

*Remark 7*: A block diagram about the implementation of the proposed LNOC method is shown in Fig. 4.2. As seen from this figure, the proposed LNOC method for nonlinear

system (4.1) with fully unknown dynamics consist of three parts, i.e., LNOC law (4.16), auxiliary system (4.13), and sigmoid neural network (4.16). Based on the output and state information of system (4.1) as well as system input $u(t)$, auxiliary system (4.13) adaptively generates better values of output-layer weight vectors $\mathbf{w}_{\mathsf{o}}(t)$ and $\mathbf{w}'_{\mathsf{o}}(t)$ for sigmoid neural network (4.16). As a result, the sigmoid neural network generates better values of $F_{\mathsf{nn}}(\mathbf{x}(t))$ and $F'_{\mathsf{nn}}(\mathbf{x}(t))$. Based on outputs $F_{\mathsf{nn}}(\mathbf{x}(t))$ and $F'_{\mathsf{nn}}(\mathbf{x}(t))$ of sigmoid neural network (4.16) and the output information $\Phi(t)$ of system (4.1), the LNOC law generates the input $u(t)$ to force output $y(t)$ of system (4.1) to track the desired output $y_r(t)$ while minimizing the performance index (4.3). By this type of feedback and information interaction, the proposed LNOC method is thus intuitively valid for system (4.1) with fully unknown dynamics.

*Remark 8:* The proposed LNOC method is a state feedback and output feedback based method. Note that most of the existing adaptive or optimal control methods are state feedback or output feedback based methods, such as the backstepping method [86, 92, 104]. In our method, together with the system states and the output, the output derivatives $y^{[i]}$ with $i = 1, 2, \cdots, \rho - 1$ need to be known. For practical systems, output derivatives may be measured by sensors [108]. Another alternative is tracking differentiators [111, 124, 125], by which the derivatives of $y(t)$ of any order can be obtained via the measurement of $y(t)$. Some existing tracking differentiators are illustrated in [111, 124, 125], some of which are finite-time convergent. The satisfactory performance of tracking differentiators has been substantiated by practical applications [112, 126].

*Remark 9:* The strict-feedback nonlinear system considered in [105] is a special case of the system considered in this chapter. In other words, compared with the backstepping approach, the proposed method allows a more general system form. Besides, the control gain is assumed to be a known constant in [105], which is considered to be a unknown function of the state variables in this chapter. Note that a wrong estimation of the control gain would severely degrade the control performance. For example, the

estimation of the control gain cannot be zero, which leads to a so-called singularity problem making the control input to be infinite. In this chapter, a projection operator is used for the estimation of the control gain so as to avoid the problem.

*Remark 10:* In practice, the input of a control system may be constrained and there are some existing results addressing the input saturation problem. Foe example, the vibration control of flexible marine riser systems with input saturation was investigated in [127]. The fuzzy tracking control problem for a class of nonlinear systems with input saturation was investigated in [128]. The attitude control of rigid spacecraft with actuator saturation was investigated in [129]. Suppose that the constraint of the control input is described by $u^- \leq u(t) \leq u^+$ with $u^-$ and $u^+$ denoting the lower bound and upper bound of the input, respectively. Let $u_{\mathsf{a}}(t)$ denotes the actual input given to the controlled system. Then, by setting

$$u_{\mathsf{a}}(t) = \begin{cases} u^+, \text{ if } u(t) > u^+ \\ u(t), \text{ if } u^- \leq u(t) \leq u^+, \\ u^-, \text{ if } u(t) < u^-, \end{cases}$$

with $u(t)$ being the input calculated based on the proposed control law, the input constraint is always not violated.

Regarding the relative degree of system (4.1) with unknown dynamics, we have the following remark.

*Remark 11:* In practice, we generally have a prior knowledge about the relative degree of a single-input single-output system. For example, a DC motor system has a relative degree of 2, which is indicated by Newton's law. For the case that such a prior knowledge is not available, we may use the controller by a trial-and-error method. Specifically, we may firstly test the performance by assuming that the relative degree is 2. If the performance is bad, we may try again by assuming that the relative degree is 3. We only need to conduct such a repetition until the performance is good. If the performance is still poor when the estimated relative degree is larger than 4, then we

may conclude that the proposed method cannot work for the system.

## 4.4 Theoretical analysis

In this section, theoretical results are provided to guarantee the performance of the proposed LNOC method, including the convergence of the proposed auxiliary system, the stability of the resultant closed-loop system, and the optimality of the performance index.

### 4.4.1 Confirmation of no singularity problems

As stated in Remark 5, a bias term $p_j(t)$ is incorporated into the design of auxiliary system (4.13) so as to avoid the singularity problem (i.e., $\sum_{j=1}^{l} w'_{\mathsf{o}j}(t)\sigma_j(\mathbf{w}_{\mathsf{i}j}^{\mathsf{T}}\mathbf{x}(t)) = 0$) that could degrade the performance of LNOC law (4.16). In this subsection, we provide a theoretical result to prove that the proposed method is singularity-free.

*Theorem 1:* If $\sum_{j=1}^{l} w'_{\mathsf{o}j}(0)\sigma_j(\mathbf{w}_{\mathsf{i}j}^{\mathsf{T}}\mathbf{x}(0)) \geq d_0 > 0$, then, for any time instant $t \geq 0$, $\mathbf{w}'_{\mathsf{o}}(t)$ of auxiliary system (4.13) satisfies $\mathbf{w}'_{\mathsf{o}}(t) \in \Omega$ with $\Omega = \{\mathbf{w}'_{\mathsf{o}} \in \mathbb{R}^l | \sum_{j=1}^{l} w'_{\mathsf{o}j}(t)\sigma_j(\mathbf{w}_{\mathsf{i}j}^{\mathsf{T}}\mathbf{x}(t)) \geq d_0\}$.

*Proof:* Define the following projection function:

$$P_{\Omega}(\mathbf{w}'_{\mathsf{o}}(t)) = \begin{cases} \mathbf{w}'_{\mathsf{o}}(t), \text{ if } \mathbf{w}'_{\mathsf{o}}(t) \in \Omega, \\ \mathbf{w}'_{\mathsf{o}}(t) - \dfrac{(\mathbf{w}'_{\mathsf{o}}(t))^{\mathsf{T}}\sigma_{\mathsf{vec}}(t) - d_0}{\sigma_{\mathsf{vec}}^{\mathsf{T}}(t)\sigma_{\mathsf{vec}}(t)}\sigma_{\mathsf{vec}}(t), \text{ if } \mathbf{w}'_{\mathsf{o}}(t) \notin \Omega, \end{cases}$$

where we have set

$$\Omega = \{\mathbf{w}'_{\mathsf{o}} \in \mathbb{R}^l | \sum_{j=1}^{l} w'_{\mathsf{o}j}(t)\sigma_j(\mathbf{w}_{\mathsf{i}j}^{\mathsf{T}}\mathbf{x}(t)) \geq d_0\}$$

and vector

$$\sigma_{\mathsf{vec}}(t) = [\sigma_1(\mathbf{w}_{\mathsf{i}1}^{\mathsf{T}}\mathbf{x}(t)), \sigma_2(\mathbf{w}_{\mathsf{i}2}^{\mathsf{T}}\mathbf{x}(t)), \cdots, \sigma_l(\mathbf{w}_{\mathsf{i}l}^{\mathsf{T}}\mathbf{x}(t))]^{\mathsf{T}}.$$

Consider the following Lyapunov function candidate:

$$V_1(t) = \|\mathbf{w}'_{\mathsf{o}}(t) - \mathsf{P}_\Omega(\mathbf{w}'_{\mathsf{o}}(t))\|_2^2/2.$$

Calculating the derivative of $V_1(t)$ yields

$$\dot{V}_1(t) = (\mathbf{w}'_{\mathsf{o}}(t) - \mathsf{P}_\Omega(\mathbf{w}'_{\mathsf{o}}(t)))^{\mathsf{T}}\dot{\mathbf{w}}'_{\mathsf{o}}(t).$$

By auxiliary system (4.13),

$$\dot{\mathbf{w}}'_{\mathsf{o}}(t) = -\gamma s(t)\sigma_{\mathsf{vec}}(t)u(t) - \frac{\|\gamma s(t)\sigma_{\mathsf{vec}}(t)u(t)\|_2}{\|\mathbf{w}'_{\mathsf{o}}(t) - P_\Omega(\mathbf{w}'_{\mathsf{o}}(t))\|_2}(\mathbf{w}'_{\mathsf{o}}(t) - P_\Omega(\mathbf{w}'_{\mathsf{o}}(t))).$$

It follows that

$$\begin{aligned}
\dot{V}_1(t) &= -(\mathbf{w}'_{\mathsf{o}}(t) - \mathsf{P}_\Omega(\mathbf{w}'_{\mathsf{o}}(t)))^{\mathsf{T}}\gamma s(t)\sigma_{\mathsf{vec}}(t)u(t) - \|\gamma s(t)\sigma_{\mathsf{vec}}(t)u(t)\|_2\|\mathbf{w}'_{\mathsf{o}}(t) \\
&\quad - P_\Omega(\mathbf{w}'_{\mathsf{o}}(t))\|_2 \\
&\leq \|\gamma s(t)\sigma_{\mathsf{vec}}(t)u(t)\|_2\|\mathbf{w}'_{\mathsf{o}}(t) - \mathsf{P}_\Omega(\mathbf{w}'_{\mathsf{o}}(t))\|_2 - \|\gamma s(t)\sigma_{\mathsf{vec}}(t)u(t)\|_2\|\mathbf{w}'_{\mathsf{o}}(t) \\
&\quad - P_\Omega(\mathbf{w}'_{\mathsf{o}}(t))\|_2 \\
&= 0.
\end{aligned}$$

Note that, from $\dot{V}_1(t) = 0$, one has $\mathbf{w}'_{\mathsf{o}}(t) = \mathsf{P}_\Omega(\mathbf{w}'_{\mathsf{o}}(t))$, i.e., $\mathbf{w}'_{\mathsf{o}}(t) \in \Omega$, which is also the largest invariant set. Then, by LaSalle's invariance principle [44], if we have $\sum_{j=1}^{l} w'_{\mathsf{o}j}(0)\sigma_j(\mathbf{w}_{\mathsf{i}j}^{\mathsf{T}}\mathbf{x}(0)) \geq d_0$, i.e., $\mathbf{w}'_{\mathsf{o}}(0) \in \Omega$, then, for any time instant $t \geq 0$, $\mathbf{w}'_{\mathsf{o}}(t)$ of auxiliary system (4.13) satisfies $\mathbf{w}'_{\mathsf{o}}(t) \in \Omega$. The proof is complete. □

*Remark 12:* According to Theorem 1, if $\mathbf{w}'_{\mathsf{o}}(0)$ is properly set such that the inequality $\sum_{j=1}^{l} w'_{\mathsf{o}j}(0)\sigma_j(\mathbf{w}_{\mathsf{i}j}^{\mathsf{T}}\mathbf{x}(0)) \geq d_0 > 0$ is satisfied, then $\sum_{j=1}^{l} w'_{\mathsf{o}j}(t)\sigma_j(\mathbf{w}_{\mathsf{i}j}^{\mathsf{T}}\mathbf{x}(t)) \geq d_0 > 0$ for any $t \geq 0$. It follows that there is no control singularity in LNOC law (4.16).

*Corollary 1:* If $\sum_{j=1}^{l} w'_{\mathsf{o}j}(0)\sigma_j(\mathbf{w}_{\mathsf{i}j}^{\mathsf{T}}\mathbf{x}(0)) \geq d_0$, then auxiliary system (4.13) is equivalent to

$$\begin{cases} \hat{y}^{[\rho]}(t) = \sum_{j=1}^{l} w_{\mathsf{o}j}(t)\sigma_j(\mathbf{w}_{\mathsf{i}j}^{\mathsf{T}}\mathbf{x}(t)) + \sum_{j=1}^{l} w'_{\mathsf{o}j}(t)\sigma_j(\mathbf{w}_{\mathsf{i}j}^{\mathsf{T}}\mathbf{x}(t))u(t) - \sum_{j=0}^{\rho-2}\alpha_j\tilde{y}^{[j+1]}(t) \\ \qquad\qquad - \lambda\phi(s(t)), \\ \dot{w}_{\mathsf{o}j}(t) = -\gamma s(t)\sigma_j(\mathbf{w}_{\mathsf{i}j}^{\mathsf{T}}\mathbf{x}(t)),\ j = 1, 2, \cdots, l, \\ \dot{w}'_{\mathsf{o}j}(t) = -\gamma s(t)\sigma_j(\mathbf{w}_{\mathsf{i}j}^{\mathsf{T}}\mathbf{x}(t))u(t),\ j = 1, 2, \cdots, l. \end{cases} \tag{4.17}$$

*Proof:* From Theorem 1, if $\sum_{j=1}^{l} w'_{\mathsf{o}j}(0)\sigma_j(\mathbf{w}_{\mathsf{i}j}^{\mathsf{T}}\mathbf{x}(0)) \geq d_0 > 0$, then for any time instant $t \geq 0$, $\mathbf{w}'_{\mathsf{o}}(t)$ of auxiliary system (4.13) satisfies $\mathbf{w}'_{\mathsf{o}}(t) \in \Omega$. It follows that $\mathbf{p}(\mathbf{w}'_{\mathsf{o}}(t))$ defined in equation (4.14) satisfies $\mathbf{p}(\mathbf{w}'_{\mathsf{o}}(t)) = 0$, $\forall t \geq 0$, i.e., $p_j(t) \equiv 0$, $\forall j = 1, 2, \cdots, l$. Therefore, auxiliary system (4.13) is equivalent to (4.17). The proof is complete. □

### 4.4.2 Convergence of the auxiliary system

One of the main components of the proposed method is auxiliary system (4.13), which is used to reconstruct the input-to-output property of system (4.6). The following theorem guarantees the convergence of the auxiliary system.

*Theorem 2* (Asymptotical Convergence of the Auxiliary System): If

$$\sum_{j=1}^{l} w'_{\mathsf{o}j}(0)\sigma_j(\mathbf{w}_{\mathsf{i}j}^{\mathsf{T}}\mathbf{x}(0)) \geq d_0 > 0$$

and all the roots of characteristic equation $\sum_{j=0}^{\rho-1}\alpha_j\nu^j = 0$ are located on the left half-plane, then the input-to-output property of auxiliary system (4.13) asymptotically converges to that of nonlinear system (4.6) with $\lim_{t\to+\infty}(\sum_{j=1}^{l}\tilde{w}_{\mathsf{o}j}(t)\sigma_j(\mathbf{w}_{\mathsf{i}j}^{\mathsf{T}}\mathbf{x}(t)) + \sum_{j=1}^{l}\tilde{w}'_{\mathsf{o}j}(t)\sigma_j(\mathbf{w}_{\mathsf{i}j}^{\mathsf{T}}\mathbf{x}(t))u(t)) = 0$, where $\tilde{\mathbf{w}}_{\mathsf{o}}(t) = \mathbf{w}_{\mathsf{o}}(t) - \mathbf{w}_{\mathsf{o}}^{*}$ and $\tilde{\mathbf{w}}'_{\mathsf{o}}(t) = \mathbf{w}'_{\mathsf{o}}(t) - \mathbf{w}'^{*}_{\mathsf{o}}$.

*Proof:* Let $\tilde{\mathbf{w}}_{\mathsf{o}}(t) = \mathbf{w}_{\mathsf{o}}(t) - \mathbf{w}_{\mathsf{o}}^*$ and $\tilde{\mathbf{w}}'_{\mathsf{o}}(t) = \mathbf{w}'_{\mathsf{o}}(t) - \mathbf{w}'^*_{\mathsf{o}}$. Subtracting equation (4.6) from the first equation of auxiliary system (4.13) yields

$$\tilde{y}^{[\rho]}(t) = \sum_{j=1}^{l} \tilde{w}_{\mathsf{o}j}(t)\sigma_j(\mathbf{w}_{\mathsf{i}j}^{\mathsf{T}}\mathbf{x}(t)) + \sum_{j=1}^{l} \tilde{w}'_{\mathsf{o}j}(t)\sigma_j(\mathbf{w}_{\mathsf{i}j}^{\mathsf{T}}\mathbf{x}(t))u(t) - \sum_{j=0}^{\rho-2} \alpha_j \tilde{y}^{[j+1]}(t) - \lambda\phi(s(t)),$$

from which one has

$$\dot{s}(t) = \sum_{j=1}^{l} \tilde{w}_{\mathsf{o}j}(t)\sigma_j(\mathbf{w}_{\mathsf{i}j}^{\mathsf{T}}\mathbf{x}(t)) + \sum_{j=1}^{l} \tilde{w}'_{\mathsf{o}j}(t)\sigma_j(\mathbf{w}_{\mathsf{i}j}^{\mathsf{T}}\mathbf{x}(t))u(t) - \lambda\phi(s(t)), \tag{4.18}$$

where $s(t) = \sum_{j=0}^{\rho-1} \alpha_j \tilde{y}^{[j]}(t)$ with $\tilde{y}(t) = \hat{y}(t) - y(t)$, $\alpha_{\rho-1} = 1$, and $\alpha_j > 0$ for $j = 0, 1, \cdots, \rho-2$.

Consider the following Lyapunov function candidate:

$$V(t) = \frac{1}{2\gamma}\tilde{\mathbf{w}}_{\mathsf{o}}^{\mathsf{T}}(t)\tilde{\mathbf{w}}_{\mathsf{o}}(t) + \frac{1}{2\gamma}\tilde{\mathbf{w}}_{\mathsf{o}}'^{\mathsf{T}}(t)\tilde{\mathbf{w}}'_{\mathsf{o}}(t) + \frac{1}{2}s^2(t). \tag{4.19}$$

Calculating the derivative of $V(t)$ yields

$$\dot{V}(t) = \frac{1}{\gamma}\tilde{\mathbf{w}}_{\mathsf{o}}^{\mathsf{T}}(t)\dot{\tilde{\mathbf{w}}}_{\mathsf{o}}(t) + \frac{1}{\gamma}\tilde{\mathbf{w}}_{\mathsf{o}}'^{\mathsf{T}}(t)\dot{\tilde{\mathbf{w}}}'_{\mathsf{o}}(t) + s(t)\dot{s}(t).$$

From Corollary 1, if $\sum_{j=1}^{l} w'_{\mathsf{o}j}(0)\sigma_j(\mathbf{w}_{\mathsf{i}j}^{\mathsf{T}}\mathbf{x}(0)) \geq d_0$, then auxiliary system (4.13) is equivalent to (4.17). Together with equation (4.18), one further has

$$\begin{aligned}\dot{V}(t) &= -s(t)\sum_{j=1}^{l} \tilde{w}_{\mathsf{o}\mathsf{j}}(t)\sigma_j(\mathbf{w}_{\mathsf{i}j}^{\mathsf{T}}\mathbf{x}(t)) - s(t)\sum_{j=1}^{l} \tilde{w}'_{\mathsf{o}\mathsf{j}}(t)\sigma_j(\mathbf{w}'^{\mathsf{T}}{}_{\mathsf{i}j}\mathbf{x}(t)) + s(t)\Bigg(\sum_{j=1}^{l} \tilde{w}_{\mathsf{o}j}(t) \\ &\quad \times \sigma_j(\mathbf{w}_{\mathsf{i}j}^{\mathsf{T}}\mathbf{x}(t)) + \sum_{j=1}^{l} \tilde{w}'_{\mathsf{o}j}(t)\sigma_j(\mathbf{w}_{\mathsf{i}j}^{\mathsf{T}}\mathbf{x}(t))u(t) - \lambda\phi(s(t))\Bigg) \\ &= -\lambda s(t)\phi(s(t)).\end{aligned}$$

Since $\phi(\cdot)$ is a monotonically increasing odd function, one has

$$\phi(s(t))\begin{cases} > 0 \ \text{ if } \ s(t) > 0, \\ = 0 \ \text{ if } \ s(t) = 0, \\ < 0 \ \text{ if } \ s(t) < 0. \end{cases}$$

Together with $\lambda > 0$, it follows that $\dot{V}(t) \leq 0$, and $\dot{V}(t) = 0$ if and only if $s(t) = 0$.

Consider the largest invariant set $\mathbb{S} = \{s(t)|\dot{V}(t) = 0\} = \{s(t)|s(t) = 0\}$. In the invariant set, by the definition of $s(t)$, one has $\sum_{j=0}^{\rho-1} \alpha_j \tilde{y}^{[j]}(t) = 0$, from which one has $\tilde{y}(t) = \sum_{j=0}^{\rho-1} C_j b_j(t)$, where $C_j$ are constants depending on $\alpha_j$ with $j = 0, 1, \cdots, \rho-1$ and initial values $y^{[j]}(0)$ with $j = 0, 1, \cdots, \rho-1$; $b_j(t)$ belong to fundamental solution set $\{b_1(t), b_2(t), \cdots, b_{\rho-1}(t)\} \subset \{\exp(r_1 t), t\exp(r_1 t), \cdots, t^{d_1-1}\exp(r_1 t), \exp(r_2 t), t\exp(r_2 t), \cdots, t^{d_2-1}\exp(r_2 t), \cdots\}$, where $r_i$ is a root repeated $d_i$ times ($d_i \leq \rho-1$) in the following characteristic equation [138]:

$$\sum_{j=0}^{\rho-1} \alpha_j \nu^j = 0. \tag{4.20}$$

By properly choosing the value of $\alpha_j$ with $j = 0, 1, \cdots, \rho-1$ such that all the roots of (4.20) are located in the left half-plane, one readily has $\lim_{t\to} b_i^{[j]}(t) = 0$, $\forall i \in \{0, 1, \cdots\}$, $\forall j \in \{0, 1, \cdots\}$. It follows that, on the invariant set, $\lim_{t\to+\infty} \tilde{y}^{[j]}(t) = 0$, $\forall j \in \{0, 1, \cdots\}$.

Based on the above results, by LaSalle's invariance principle [44], $\lim_{t\to+\infty} \tilde{y}^{[j]}(t) = 0$, $\forall j \in \{0, 1, \cdots\}$. Then, it follows from equation (4.18) that

$$\lim_{t\to+\infty} (\sum_{j=1}^{l} \tilde{w}_{\mathsf{o}j}(t)\sigma_j(\mathbf{w}_{\mathsf{i}j}^{\mathsf{T}}\mathbf{x}(t)) + \sum_{j=1}^{l} \tilde{w}'_{\mathsf{o}j}(t)\sigma_j(\mathbf{w}_{\mathsf{i}j}^{\mathsf{T}}\mathbf{x}(t))u(t)) = 0.$$

This completes the proof. □

Based on theorem 2 and the persistent excitation (PE) condition [130], we also have the following corollary about the convergence of neural network parameters. Note that the PE condition is generally required for adaptive control methods to achieve parameter convergence [93, 130, 131].

*Corollary 2*: Given that both vector $[\sigma_1(\mathbf{w}_{\mathsf{i}1}^{\mathsf{T}}\mathbf{x}(t)), \sigma_2(\mathbf{w}_{\mathsf{i}2}^{\mathsf{T}}\mathbf{x}(t)), \cdots, \sigma_l(\mathbf{w}_{\mathsf{i}l}^{\mathsf{T}}\mathbf{x}(t))]^{\mathsf{T}}$ and vector $[\sigma_1(\mathbf{w}_{\mathsf{i}1}^{\mathsf{T}}\mathbf{x}(t))u(t), \sigma_2(\mathbf{w}_{\mathsf{i}2}^{\mathsf{T}}\mathbf{x}(t))u(t), \cdots, \sigma_l(\mathbf{w}_{\mathsf{i}l}^{\mathsf{T}}\mathbf{x}(t))u(t)]^{\mathsf{T}}$ satisfy the PE condition, auxiliary system (4.13) satisfies $\lim_{t\to+\infty} \mathbf{w}_{\mathsf{o}}(t) = \mathbf{w}_{\mathsf{o}}^*$ and $\lim_{t\to+\infty} \mathbf{w}'_{\mathsf{o}}(t) = \mathbf{w}'^*_{\mathsf{o}}$.

*Proof:* The proof can be generalized from [130] based on Theorem 2 and is thus omitted. □

Note that the theorems in this chapter do not rely on Corollary 2, and thus the PE con-

dition. This, as we consider, is also one of the differences of the proposed approach compared with some existing neural network based adaptive control methods. In general, it is difficult to verify the PE condition online. To some extent, the PE condition requires a signal to be sufficiently rich [132]. In practice, the PE condition could be guaranteed when using a sufficiently rich reference trajectory for the control system.

### 4.4.3 Stability of the closed-loop system

In this subsection, a theorem is provided to guarantee the stability of the closed-loop system consisting of nonlinear system (4.6) and auxiliary system (4.13).

*Theorem 3* (Asymptotical Stability of the Control System): If $\sum_{j=1}^{l} w'_{\mathsf{o}j}(0)\sigma_j(\mathbf{w}_{\mathsf{i}j}^{\mathsf{T}}\mathbf{x}(0)) \geq d_0 > 0$, all the roots of characteristic equation $\sum_{j=0}^{\rho-1}\alpha_j\nu^j = 0$ are located on the left half-plane, and relative degree $\rho \in \{1,2,3,4\}$, then the closed-loop system consisting of nonlinear system (4.6) and LNOC law (4.16) is asymptotically stable.

*Proof:* First, system (4.6) is rewritten as

$$y^{[\rho]}(t) = \sum_{j=1}^{l} w_{\mathsf{o}j}(t)\sigma_j(\mathbf{w}_{\mathsf{i}j}^{\mathsf{T}}\mathbf{x}(t)) + \sum_{j=1}^{l} w'_{\mathsf{o}j}(t)\sigma_j(\mathbf{w}_{\mathsf{i}j}^{\mathsf{T}}\mathbf{x}(t))u(t) + \eta(t), \tag{4.21}$$

where $\eta(t)$ is defined as follows:

$$\eta(t) = -(\sum_{j=1}^{l} \tilde{w}_{\mathsf{o}j}\sigma_j(\mathbf{w}_{\mathsf{i}j}^{\mathsf{T}}\mathbf{x}(t)) + \sum_{j=1}^{l} \tilde{w}'_{\mathsf{o}j}\sigma_j(\mathbf{w}_{\mathsf{i}j}^{\mathsf{T}}\mathbf{x}(t))u(t))$$

with $\tilde{\mathbf{w}}_{\mathsf{o}}(t) = \mathbf{w}_{\mathsf{o}}(t) - \mathbf{w}_{\mathsf{o}}^*$ and $\tilde{\mathbf{w}}'_{\mathsf{o}}(t) = \mathbf{w}'_{\mathsf{o}}(t) - \mathbf{w}'^*_{\mathsf{o}}$. Recall LNOC law (4.16):

$$\begin{aligned} u(t) = \frac{-1}{\sum_{j=1}^{l} w'_{\mathsf{o}j}(t)\sigma_j(\mathbf{w}_{\mathsf{i}j}^{\mathsf{T}}\mathbf{x}(t))}(\sum_{i=0}^{\rho-1}\frac{T^i(2\rho+1)\rho!}{(\rho+1+i)i!T^\rho}(y_{\mathsf{r}}^{[i]}(t) - y^{[i]}(t)) - y_{\mathsf{r}}^{[\rho]}(t) \\ - \sum_{j=1}^{l} w_{\mathsf{o}j}(t)\sigma_j(\mathbf{w}_{\mathsf{i}j}^{\mathsf{T}}\mathbf{x}(t))). \end{aligned}$$

Substituting the expression of $u(t)$ into (4.21) yields

$$y^{[\rho]}(t) = \sum_{i=0}^{\rho-1}\frac{T^i(2\rho+1)\rho!(y_{\mathsf{r}}^{[i]}(t) - y^{[i]}(t))}{(\rho+1+i)i!T^\rho} - y_{\mathsf{r}}^{[\rho]}(t) + \eta(t). \tag{4.22}$$

Let $e(t) = y_{\mathsf{r}}(t) - y(t)$. Then, from equation (4.22),

$$\sum_{i=0}^{\rho} \frac{T^i(2\rho+1)\rho!}{(\rho+1+i)i!T^\rho} e_i^{[i]}(t) = \eta(t),$$

i.e., the closed-loop system consisting of nonlinear system (4.6) and LNOC law (4.16) can be described as follows:

$$\sum_{i=0}^{\rho} \frac{T^i}{(\rho+1+i)i!} e_i^{[i]}(t) = \eta(t), \tag{4.23}$$

which can be viewed as a linear system with the input being $\eta(t)$. Consider the corresponding autonomous system:

$$\sum_{i=0}^{\rho} \frac{T^i}{(\rho+1+i)i!} e_i^{[i]}(t) = 0.$$

By the Routh-Hurwitz criterion [39], it is readily checked that the all the roots of the corresponding characteristic equation $\sum_{i=0}^{\rho} T^i\nu/((\rho+1+i)i!) = 0$ are located in the left-half plane and thus equilibrium $e(t) = 0$ is exponentially stable, given that $\rho \in \{1,2,3,4\}$. Besides, from Theorem 2, if all the roots of characteristic equation $\sum_{j=0}^{\rho-1} \alpha_j \nu^j = 0$ are located on the left half-plane, then $\lim_{t\to+\infty}(\sum_{j=1}^{l} \tilde{w}_{\mathsf{o}j}(t)\sigma_j(\mathbf{w}_{\mathsf{i}j}^{\mathsf{T}}\mathbf{x}(t)) + \sum_{j=1}^{l} \tilde{w}'_{\mathsf{o}j}(t)\sigma_j(\mathbf{w}_{\mathsf{i}j}^{\mathsf{T}}\mathbf{x}(t))u(t)) = 0$, i.e., $\lim_{t\to+\infty} \eta(t) = 0$. Then, it follows by the bound-input bounded-output property [39] of linear systems that equilibrium $e(t) = 0$ of closed-loop system (4.23) is asymptotically stable. The proof is complete. □

*Remark 13:* As seen from Theorem 3, there are several conditions needed to guarantee the asymptotic stability of the closed-loop system. The condition that

$$\sum_{j=1}^{l} w'_{\mathsf{o}j}(0)\sigma_j(\mathbf{w}_{\mathsf{i}j}^{\mathsf{T}}\mathbf{x}(0)) \geq d_0 > 0$$

is very weak, which can be readily satisfied by choosing values of $w'_{\mathsf{o}j}(0)$. The condition on the roots of characteristic equation $\sum_{j=0}^{\rho-1} \alpha_j \nu^j = 0$ is also very weak. Specifically, one can predefine desired roots $r_i$ with $i = 1, 2, \cdots, \rho-1$. By letting $\sum_{j=0}^{\rho-1} \alpha_j \nu^j = \prod_{j=1}^{\rho-1}(\nu^j - r_j)$, one can readily find the corresponding $\alpha_j$. For example, if $\rho = 2$ and the desired root is $-5$. Then, from $\alpha_0 + \alpha_1\nu = (\nu - (-5))$, one has $\alpha_0 + \alpha_1\nu = \nu + 5$.

It follows that one can simply choose $\alpha_0 = 5$ and $\alpha_1 = 1$. In terms of the condition on the relative degree of the controlled systems, it should be noted that many mechanical systems have a relative degree less than 4 [107, 133, 134].

### 4.4.4 Asymptotic optimality of the performance index

As shown in the previous section, the motivation of the proposed LNOC method is to achieve optimal tracking via minimizing the integral performance index (4.3) for nonlinear systems with fully unknown dynamics. Due to the problem relaxation, the proposed LNOC law cannot be optimal all the time, which is also the reason why it is call a near-optimal one. Fortunately, theoretical analysis shows that the proposed one is asymptotically optimal.

*Theorem 4* (Asymptotical Optimality of the Performance Index): If

$$\sum_{j=1}^{l} w'_{\mathsf{o}j}(0)\sigma_j(\mathbf{w}_{\mathsf{i}j}^{\mathsf{T}}\mathbf{x}(0)) \geq d_0 > 0,$$

given all the roots of characteristic equation $\sum_{j=0}^{\rho-1}\alpha_j\nu^j = 0$ located on the left half-plane and relative degree $\rho \in \{1, 2, 3, 4\}$, then, synthesized by LNOC law (4.16), performance index (4.3) associated with nonlinear system (4.6) asymptotically converges to optimal. In other words, control law (4.16) is asymptotically optimal.

*Proof:* By Taylor expansion, performance index (4.3) is rewritten as

$$J(t) = \frac{1}{2}\int_0^T (\varrho^{\mathsf{T}}(\tau)\check{\Upsilon}(t) - \frac{\tau^\rho}{\rho!}L_g L_f^{\rho-1}h(\mathbf{x}(t))u(t) + \frac{\tau^\rho}{\rho!}\Delta(t))^2\mathsf{d}\tau, \tag{4.24}$$

where $\Delta(t) = (y_{\mathsf{d}}^{[\rho]}(t + \varkappa\tau) - y^{[\rho]}(t + \varkappa\tau) - (y_{\mathsf{d}}^{[\rho]}(t) - y^{[\rho]}(t)))$ and $0 < \varkappa < 1$. By equation (4.6), the performance index defined in equation (4.25) is rewritten as follows:

$$\begin{aligned} J(t) = \frac{1}{2}\int_0^T (&\sum_{i=0}^{\rho-1}\frac{\tau^i}{i!}(y_{\mathsf{r}}^{[i]}(t) - y^{[i]}(t)) + \frac{\tau^\rho}{\rho!}y_{\mathsf{r}}^{\rho}(t) - \frac{\tau^\rho}{\rho!}\sum_{j=1}^{l} w_{\mathsf{o}j}(t)\sigma_j(\mathbf{w}_{\mathsf{i}j}^{\mathsf{T}}\mathbf{x}(t)) \\ &- \frac{\tau^\rho}{\rho!}\sum_{j=1}^{l} w'_{\mathsf{o}j}(t)\sigma_j(\mathbf{w}_{\mathsf{i}j}^{\mathsf{T}}\mathbf{x}(t))u(t) + \frac{\tau^\rho}{\rho!}\Delta(t) - \frac{\tau^\rho}{\rho!}\eta(t))^2\mathsf{d}\tau, \end{aligned} \tag{4.25}$$

where $\eta(t)$ is defined as

$$\eta(t) = -(\sum_{j=1}^{l} \tilde{w}_{\mathsf{o}j}\sigma_j(\mathbf{w}_{\mathsf{i}j}^{\mathsf{T}}\mathbf{x}(t)) + \sum_{j=1}^{l} \tilde{w}'_{\mathsf{o}j}\sigma_j(\mathbf{w}_{\mathsf{i}j}^{\mathsf{T}}\mathbf{x}(t))u(t)).$$

By the triangle inequality [138], from (4.25), one has

$$J(t) \leq \bar{J}(t) + \int_0^T \frac{\tau^{2\rho}}{(\rho!)^2}\Delta^2(t)\mathsf{d}\tau + \int_0^T \frac{\tau^{2\rho}}{(\rho!)^2}\eta^2(t)\mathsf{d}\tau,$$

where

$$\begin{aligned}\bar{J}(t) &= \int_0^T (\sum_{i=0}^{\rho-1} \frac{\tau^i}{i!}(y_{\mathsf{r}}^{[i]}(t) - y^{[i]}(t)) + \frac{\tau^\rho}{\rho!}y_{\mathsf{r}}^{\rho}(t) - \frac{\tau^\rho}{\rho!}\sum_{j=1}^{l} w_{\mathsf{o}j}(t)\sigma_j(\mathbf{w}_{\mathsf{i}j}^{\mathsf{T}}\mathbf{x}(t)) - \frac{\tau^\rho}{\rho!}\sum_{j=1}^{l} w'_{\mathsf{o}j}(t) \\ &\quad \cdot \sigma_j(\mathbf{w}_{\mathsf{i}j}^{\mathsf{T}}\mathbf{x}(t))u(t))^2\mathsf{d}\tau \\ &= \int_0^T (\varrho^{\mathsf{T}}(\tau)\Gamma(t))^2\mathsf{d}\tau - 2\varrho_{\mathsf{int}}\Gamma(t)\sum_{j=1}^{l} w_{\mathsf{o}j}(t)\sigma_j(\mathbf{w}_{\mathsf{i}j}^{\mathsf{T}}\mathbf{x}(t))u(t) \\ &\quad + \frac{T^{2\rho+1}}{(2\rho+1)(\rho!)^2}(\sum_{j=1}^{l} w_{\mathsf{o}j}(t)\sigma_j(\mathbf{w}_{\mathsf{i}j}^{\mathsf{T}}\mathbf{x}(t)))^2u^2(t) \geq 0\end{aligned}$$

is a convex quadratic performance index with respect to $u(t)$; $\Gamma(t)$ is defined as

$$\Gamma(t) = [y_{\mathsf{r}}(t) - y(t), \dot{y}_{\mathsf{r}}(t) - \dot{y}(t), \cdots, y_{\mathsf{r}}^{[\rho-1]}(t) - y^{[\rho-1]}(t), y_{\mathsf{r}}^{[\rho]}(t) - \sum_{j=1}^{l} w_{\mathsf{o}j}(t)\sigma_j(\mathbf{w}_{\mathsf{i}j}^{\mathsf{T}}\mathbf{x}(t))]^{\mathsf{T}};$$

$\varrho(\tau)$ is defined as

$$\varrho(\tau) = [1, \tau, \tau^2/2, \cdots, \tau^\rho/\rho!]^{\mathsf{T}}.$$

In addition, $\varrho_{\mathsf{int}}$ is defined in equation (4.11). Evidently, the sufficient condition for $u(t)$ to be the minimizer of $\bar{J}(t)$ is $\partial\bar{J}(t)/\partial u = 0$, i.e.,

$$\varrho_{\mathsf{int}}\Gamma(t)\sigma_j(\mathbf{w}_{\mathsf{i}j}^{\mathsf{T}}\mathbf{x}(t)) + T^{2\rho+1}(\sigma_j(\mathbf{w}_{\mathsf{i}j}^{\mathsf{T}}\mathbf{x}(t)))^2u(t)/((2\rho+1)(\rho!)^2) = 0,$$

which yields

$$\begin{aligned}&-\sum_{i=0}^{\rho-1} \frac{T^i}{(\rho+1+i)i!}(y_{\mathsf{r}}^{[i]}(t) - y^{[i]}(t)) - \frac{T^\rho}{(2\rho+1)\rho!}y_{\mathsf{r}}^{[\rho]}(t) \\ &+\frac{T^\rho}{(2\rho+1)\rho!}(\sum_{j=1}^{l} w_{\mathsf{o}j}(t)\sigma_j(\mathbf{w}_{\mathsf{i}j}^{\mathsf{T}}\mathbf{x}(t)) + \sum_{j=1}^{l} w'_{\mathsf{o}j}(t)\sigma_j(\mathbf{w}_{\mathsf{i}j}^{\mathsf{T}}\mathbf{x}(t))u(t)) = 0. \quad (4.26)\end{aligned}$$

Evidently, LNOC law (4.16) is the solution of equation (4.26), i.e., control law (4.16) satisfies $\partial \bar{J}(t)/\partial u = 0$. It follows that $\bar{J}(t) = 0$ when control law (4.16) is adopted. Then, one has

$$\begin{aligned} J(t) &\leq \int_0^T \frac{\tau^{2\rho}}{(\rho!)^2}\Delta^2(t)\mathsf{d}\tau + \int_0^T \frac{\tau^{2\rho}}{(\rho!)^2}\eta^2(t)\mathsf{d}\tau \\ &\leq \frac{T^{2\rho+1}}{(\rho!)^2(2\rho+1)}\left(\sup_{0<\varkappa<1, 0\leq\tau\leq T} \Delta^2(t) + \eta^2(t)\right). \end{aligned}$$

From Theorem 2, if $\sum_{j=1}^{l} w'_{\mathsf{o}j}(0)\sigma_j(\mathbf{w}_{ij}^{\mathsf{T}}\mathbf{x}(0)) \geq d_0 > 0$ and all the roots of characteristic equation $\sum_{j=0}^{\rho-1}\alpha_j\nu^j = 0$ are located on the left half-plane, then $\lim_{t\to+\infty}\eta(t) = 0$. Together with the conditions in Theorem 2, with $\rho \in \{1, 2, 3, 4\}$, from Theorem 3, $\lim_{t\to+\infty}(y_{\mathsf{r}}(t) - y(t)) = 0$. By the solution property of linear ordinary differential equations (see the proof of Theorem 2) and the extension of Barbalat's lemma [106], one further has $\lim_{t\to+\infty}(y_{\mathsf{r}}^{[\rho]}(t) - y^{[\rho]}(t)) = 0$. It follows that $\lim_{t\to+\infty}\Delta(t) = 0$. Note that $J(t) \geq 0$. Then, by the pinching theorem [138], $\lim_{t\to+\infty} J(t) = 0$, i.e., performance index (4.3) asymptotically converges to optimal. In other words, control law (4.16) is asymptotically optimal. This completes the proof. □

Intuitively, this work avoids the direct solution of the difficult optimal control problem via Taylor expansion. The uncertainty of the controlled system is addressed via constructing an auxiliary system whose input-output dynamics is asymptotically equivalent to that of the controlled system described by neural networks. As indicated in the theoretical analysis, by the proposed approach, the closed-loop system essentially asymptotically converges to the desired one which guarantees the optimality of the performance index and the stability of the closed-loop system.

## 4.5 Illustrative example

In this section, an illustrative example is presented to show the efficacy of the proposed method, and verify the theoretical results.

Consider the following nonlinear system:

$$\begin{cases} \dot{x}_1(t) = 2x_2(t) + x_1(t), \\ \dot{x}_2(t) = 4x_1^3(t)x_2(t) + \cos(x_2(t)) + \sin(x_1(t))x_2^2(t) \\ \qquad\qquad + 7(\sin(x_2(t)) + 1.1)u(t), \\ y(t) = x_1(t), \end{cases} \tag{4.27}$$

which has a relative degree of 2, i.e., $\rho = 2 \in \{1, 2, 3, 4\}$. The relative degree of the system satisfies the requirement stated on Theorem 3 and Theorem 4. To implement the proposed LNOC method, one only needs to know the relative degree and the state and output information of the system as well as $d_0$. In this example, the reference output is $y_{\mathsf{d}}(t) = 0.5\cos(0.6t) + 0.1$. In addition, $d_0$ is set to $d_0 = 10^{-4}$. It is worth pointing out that other smooth reference outputs can be viewed as the sum of several sinusoidal signals. Therefore, the chosen output is general.

We first present simulation results under the following setups. The predictive period is set to $T = 0.3$ s. The parameters of auxiliary system (4.13) are set to $\lambda = 2$, $\gamma = 5$, $\alpha_1 = 1$, and $\alpha_0 = 2$. Evidently, under this setup, the root of characteristic equation $\alpha_1\nu + \alpha_0 = 0$ is $\nu = -2$, which satisfies the requirement stated in Theorem 2. Without loss of generality, the monotonically increasing odd activation function $\phi(\cdot)$ is chosen as $\phi(z) = (1 - \exp(-3z))/(1 + \exp(-3z))$, i.e., the so-called bipolar sigmoid activation function in [119]. In terms of the setup of sigmoid neural network (4.16), $c_j$ is set to $c_j = 0.0001$ for any $j = 1, 2, \cdots, l$; each element of $\mathbf{w}_{\mathrm{i}j}$ and $b_j$ are randomly set, and $l = 256$, i.e., the hidden-layer of the sigmoid neural network consists of 256 neurons. The initial state $\mathbf{x}(0) = [x_1(0), x_2(0)]^{\mathsf{T}}$ of system (4.27) and $\hat{\mathbf{x}}(0) = [\hat{x}_1(0), \hat{x}_2(0)]^{\mathsf{T}}$ of auxiliary system (4.13), without loss of generality, are set as $\mathbf{x}(0) = \hat{\mathbf{x}}(0) = [0, 0]^{\mathsf{T}}$. In addition, each element of initial output-layer weight vectors $\mathbf{w}_{\mathsf{o}}(0)$ and $\mathbf{w}'_{\mathsf{o}}(0)$ of sigmoid neural network (4.16) is randomly set at interval $(0, 1)$. It is checked that $\mathbf{w}'_{\mathsf{o}}(0)$ satisfies the condition stated in Theorem 1. Fig. 4.3(a) shows the convergence of the input-to-output property of auxiliary system (4.13) to that of nonlinear system (4.27), which verifies

Theorem 2 and also substantiates the efficacy of the proposed sigmoid neural network based auxiliary system for dynamics reconstruction of unknown nonlinear systems. As seen from Fig. 4.3(b) and Fig. 4.3(c), output $y(t)$ of system (4.27) successfully tracks reference output $y_{\mathrm{r}}(t)$ with output error $e(t) = y(t) - y_{\mathrm{r}}(t)$ and its derivatives converging to zero, which verifies Theorem 3. As seen from Fig. 4.3(d), performance index $J(t)$ asymptotically converges to zero, which verifies Theorem 4. In addition, Fig. 4.3(e) and Fig. 4.3(f) show that the time profiles of both state variables $x_i(t)$ (with $i = 1, 2$) and input $u(t)$ are bounded and smooth. In addition, the time profiles of the parameters of sigmoid neural network (4.16) during the control process are shown in Fig. 4.4. As seen from the figures, each parameter is bounded and the magnitude of each parameter does not vary much. Under the above setup, but with different values of predictive period $T$, the time profiles of performance index $J(t)$ and input $u(t)$ are shown in Fig. 4.5. As seen from Fig. 4.5(a), with a smaller value of $T$, the performance index converges to zero faster. In addition, from Fig. 4.5(b), it is observed that the time profiles of input $u(t)$ are smooth under different values of $T$. These results substantiate the efficacy of the proposed LNOC method for nonlinear systems with fully unknown dynamics and verify the theoretical results.

## 4.6 Experimental validation

The performance of the proposed method is also tested via a low-cost DC motor system. DC motor systems are second-order and the output motor angle $\theta$ has a relative degree of 2 with respect to the input voltage $V$. The power for the whole system is provided by a 9-V 1-A AC/DC adaptor through the system power port. As the Arduino UNO board has limited storage, the data during the experiment are sent to a personal computer through the system communication port. In the experiment, the motor angle $\theta$ is directly measured by the encoder. The angle velocity is obtained via the following

tracking differentiator [124]:

$$\begin{cases} \dot{q}_1(t) = q_2(t) - \sqrt{C}|q_1(t) - \theta(t)|^{1/2}\text{sign}(q_1(t) - \theta(t)), \\ \dot{q}_2(t) = -1.1C\text{sign}(q_1(t) - \theta(t)), \\ q_3(t) = q_2(t) - \sqrt{C}|q_1(t) - \theta(t)|^{1/2}\text{sign}(q_1(t) - \theta(t)), \end{cases}$$

where $q_3(t)$ is the output of the tracking differentiator (i.e., the estimation of the angle velocity of the DC motor), which is theoretically guaranteed to converge to $\dot{\theta}(t)$ in finite time given that design parameter $C > 0 \in \mathbb{R}$ is set to be large enough. The proposed method is implemented in the Arduino UNO board, where the calculated control input is converted into PWM signal, which is fed into the motor drive. For the sake of safety, the actual input for the DC motor control system is set as follows:

$$V(t) = \begin{cases} u^+, \text{ if } u(t) > u^+ \\ u(t), \text{ if } u^- \leq u(t) \leq u^+, \\ u^-, \text{ if } u(t) < u^-, \end{cases}$$

where $u(t)$ is the input calculated by the proposed controller (4.16) with $u^+ = 7$ and $u^- = -7$, i.e., the amplitude of the input voltage is limited to be not larger than 7 V. In the experiment, $T$ is set to 0.001 s. The parameters of the auxiliary system is set to $\gamma = 3$, $\lambda = 3$, $\alpha_0 = 2$, and $\alpha_1 = 1$. The parameters of the sigmoid basis functions are set the same as those in the simulative example except that, owing to the computational capability of the Arduino UNO board, we only use ten neurons in the hidden layer, i.e., $l = 10$, and $b_i$ is set to $-5 + i$ which leads to a regular placing of the centers in the region [-4,5]. The parameter of the tracking differentiator is set to $C = 100$. The desired output is set to $\theta_\text{r} = 4\sin(\pi t/5) + 0.2$ rad. The other settings are the same as those in the simulative example. As seen from Fig. 4.6, by the proposed method, the output of the DC motor successfully tracks the desired output, and the value of the performance index is less than $5 \times 10^{-6}$ at the steady state. The experimental data shows that the average tracking error is $1.7110 \times 10^{-4}$ rad. In addition, the input given to the DC motor

does not exceed the limit. The results further verify the efficiency and realizability of the proposed method.

## 4.7 Chapter summary

In this chapter, a novel model-free LNOC method has been proposed for a class of continuous-time nonlinear systems with fully unknown system dynamics. The proposed method can guarantee asymptotic stability of the resultant closed-loop system and the asymptotic optimality of the performance index. An illustrative example and experimental results have substantiated the efficacy of the proposed method and verified the theoretical results. It is worth pointing out that this chapter provides novel results about combining sliding-mode control concept and the universal approximation capability of sigmoid neural networks to tackle the difficulty in designing optimal control laws for nonlinear systems with fully unknown dynamics.

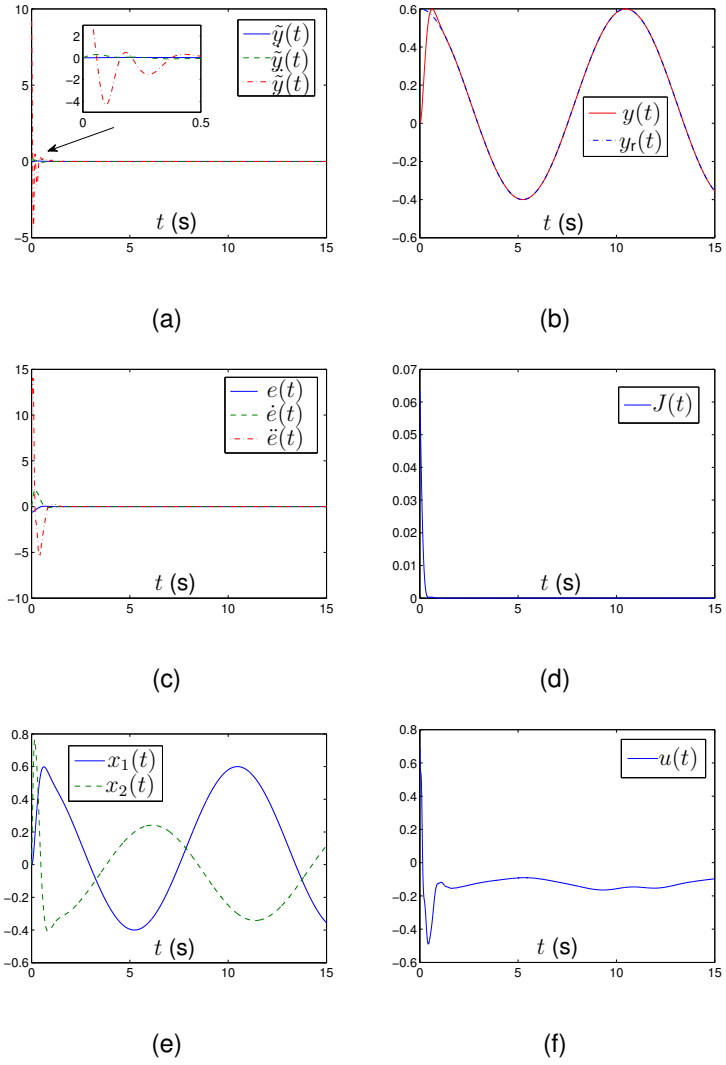
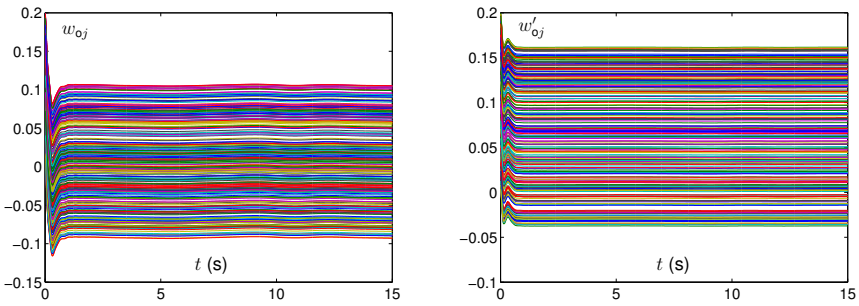
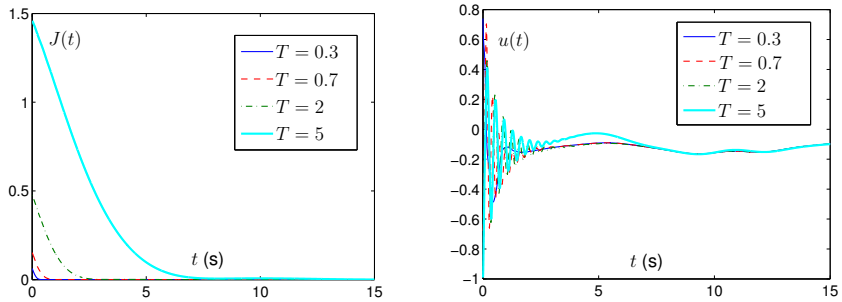



Figure 4.3: Convergence of the input-to-output property of auxiliary system (4.13) to that of nonlinear system (4.27) and performance of nonlinear system (4.27) in tracking time-varying reference output $y_r(t)$ under the control of LNOC law (4.16). (a) Time profiles of $\tilde{y}(t)$, $\dot{\tilde{y}}(t)$, and $\ddot{\tilde{y}}(t)$ with $\tilde{y}(t) = \hat{y}(t) - y(t)$. (b) Time profiles of system output $y(t)$ and reference output $y_r(t)$. (c) Time profiles of tracking errors in different levels, i.e., $e(t)$, $\dot{e}(t)$, and $\ddot{e}(t)$ with $e(t) = y(t) - y_r(t)$. (d) Time profile of performance index $J(t)$ associated with nonlinear system (4.27) with $T = 0.3$ s. (e) Time profiles of state variables $x_1(t)$ and $x_2(t)$ of system (4.27). (f) Time profile of control input $u(t)$.



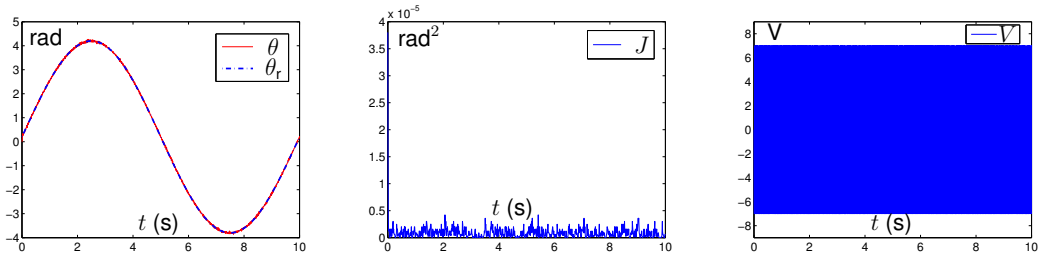

(a) (b)

Figure 4.4: Time profiles of neural network parameters during the control process of nonlinear system (4.27) via LNOC law (4.16). (a) Time profiles of $w_{oj}(t)$ with $j = 1, 2, \cdots, l$. (b) Time profiles of $w'_{oj}(t)$ with $j = 1, 2, \cdots, l$.




(a) (b)

Figure 4.5: Performance comparison of LNOC law (4.16) for the tracking control of nonlinear system (4.27) under different values of $T$ in performance index $J(t)$ defined in equation (4.3). (a) Time profiles of performance index $J(t)$. (b) Time profiles of input $u(t)$ generated by the control law.




(a) (b) (c)

Figure 4.6: Data profiles for the motor control experiment by using the proposed control method. (a) Time profiles of system output $\theta(t)$ and reference output $\theta_{\mathrm{r}}(t)$. (b) Time profile of performance index $J(t)$. (c) Time profile of input voltage $V(t)$.

# Chapter 5

# LNOC for consensus of nonlinear heterogeneous multi-agent systems with learning of parameters

In this chapter, the learning and near-optimal distributed consensus of high-order nonlinear multi-agent systems consisting of heterogeneous agents is investigated.[1] The consensus problem is formulated as a receding-horizon optimal control problem. Under the condition that the dynamics of all agents are fully known, a nominal near-optimal protocol is designed and proposed via making approximation of the performance index. For the situation with fully unknown system parameters, sliding-mode auxiliary systems, which are independent for different agents, are built to reconstruct the input-output properties of agents. Based on the sliding-mode auxiliary systems, an adaptive near-optimal protocol is finally presented to control high-order nonlinear multi-agent systems with fully unknown parameters. Theoretical analysis shows that the proposed protocols can simultaneously guarantee the asymptotic optimality of the performance

[1] The content in this chapter has already been published. Yinyan Zhang and Shuai Li, "Adaptive near-optimal consensus of high-order nonlinear multi-agent systems with heterogeneity," *Automatica*, vol. 85, pp. 426–432, 2017.

index and the asymptotic consensus of multi-agent systems. An illustrative example about a third-order nonlinear multi-agent system consisting of 10 heterogeneous agents with fully unknown parameters further substantiates the efficacy and superiority of the proposed adaptive near-optimal consensus approach.

## 5.1 Problem formulation

Consider a multi-agent system consisting of $n$ nonlinear heterogeneous agents of order $\sigma$, of which the communication topology is described by an undirected connected graph with the Laplacian matrix denoted by $L$. The dynamics of the $i$th ($i = 1, 2, \cdots, n$) agent is described as

$$\begin{aligned}
\dot{x}_i^{(0)} &= x_i^{(1)}, \\
&\vdots \\
\dot{x}_i^{(\sigma-2)} &= x_i^{(\sigma-1)}, \\
\dot{x}_i^{(\sigma-1)} &= f_i(\mathbf{x}_i) + g_i(\mathbf{x}_i)u_i,
\end{aligned}$$

where $x_i^{(j)} \in \mathbb{R}$ with $j = 1, 2, \cdots, \sigma$ denotes the $j$th time derivative of $x_i^{(0)} = x_i$; $\mathbf{x}_i = [x_i, x_i^{(1)}, \cdots, x_i^{(\sigma-1)}]^{\mathsf{T}} \in \mathbb{R}^{\sigma}$ and $u_i \in \mathbb{R}$ denote the state vector and input of the $i$th agent, respectively; $f_i(\cdot) : \mathbb{R}^{\sigma} \to \mathbb{R}$ and $g_i(\cdot) : \mathbb{R}^{\sigma} \to \mathbb{R}$ are continuously differentiable functions which may be different for each agent with $g_i(\mathbf{x}_i) > 0$ for all $i = 1, 2, \cdots, n$. Let $\mathbf{x}_{\mathsf{M}} = [x_1, x_2, \cdots, x_n]^{\mathsf{T}}$, $f_{\mathsf{M}}(\mathbf{x}_{\mathsf{M}}) = [f_1(\mathbf{x}_1), f_2(\mathbf{x}_2), \cdots, f_n(\mathbf{x}_n)]^{\mathsf{T}}$, $\mathbf{u} = [u_1, u_2, \cdots, u_n]^{\mathsf{T}}$ and $g_{\mathsf{M}}(\mathbf{x}_{\mathsf{M}}) = \mathsf{diag}([g_1(\mathbf{x}_1), g_2(\mathbf{x}_2), \cdots, g_n(\mathbf{x}_n)]^{\mathsf{T}})$. The multi-agent system is thus formulated as follows:

$$\begin{cases}
\dot{\mathbf{x}}_{\mathsf{M}}^{(k)} = \mathbf{x}_{\mathsf{M}}^{(k+1)}, i = 0, 1, \cdots, \sigma - 2, \\
\dot{\mathbf{x}}_{\mathsf{M}}^{(\sigma-1)} = f_{\mathsf{M}}(\mathbf{x}_{\mathsf{M}}) + g_{\mathsf{M}}(\mathbf{x}_{\mathsf{M}})\mathbf{u}.
\end{cases} \tag{5.1}$$

To achieve the consensus of multi-agent system (5.1), we propose the following performance index:

$$J(t) = \int_0^T \mathbf{x}_{\mathsf{M}}^{\mathsf{T}}(t+\tau)L_0\mathbf{x}_{\mathsf{M}}(t+\tau)\mathsf{d}\tau + \int_0^T \dot{\mathbf{x}}_{\mathsf{M}}^{\mathsf{T}}(t+\tau)Q\dot{\mathbf{x}}_{\mathsf{M}}(t+\tau)\mathsf{d}\tau, \tag{5.2}$$

where $T > 0 \in \mathbb{R}$ denotes the predictive period; $L_0 \in \mathbb{R}^{n\times n}$ is a matrix to be determined, which satisfies two properties: 1) $L_0$ is symmetric; 2) only one of its eigenvalues is 0 with the corresponding eigenvector being $\mathbf{1} = [1, 1, \cdots, 1]^{\mathsf{T}} \in \mathbb{R}^n$ and the other eigenvalues are strictly bigger than 0. Besides, $Q = \zeta I$ with $\zeta > 0 \in \mathbb{R}$ and $I$ being a $n \times n$ identity matrix.

Remark 1: The two integral terms of $J(t)$ corresponds to the relative potential energy and the kinetic energy of the whole multi-agent system, respectively. When performance index $J(t)$ achieves its minimum, i.e., $J(t) = 0$, the relative potential energy and the kinetic energy become zero. It follows that all the agents of the multiagent system achieve static consensus. The results presented in this chapter can also be extended to velocity consensus, acceleration consensus, etc, by modifying the two integral terms of $J(t)$.

## 5.2 Nominal near-optimal design

In this section, we consider the situation where multi-agent system (5.1) is consisting of agents of fully known dynamics. Let

$$X_{\mathsf{M}} = [\mathbf{x}_{\mathsf{M}}, \mathbf{x}_{\mathsf{M}}^{(1)}, \cdots, \mathbf{x}_{\mathsf{M}}^{(\sigma-1)}, f_{\mathsf{M}}(\mathbf{x}_{\mathsf{M}})].$$

Then, $\mathbf{x}_{\mathsf{M}}(t+\tau)$ of multi-agent system (5.1) is approximated via time-scale Taylor expansion as

$$\mathbf{x}_{\mathsf{M}}(t+\tau) \approx X_{\mathsf{M}}(t)\mathbf{w}_1(\tau) + \tau^{\sigma} g_{\mathsf{M}}(\mathbf{x}_{\mathsf{M}}(t))\mathbf{u}(t)/\sigma!,$$

where

$$\mathbf{w}_1(\tau) = [1, \tau, \cdots, \tau^{\sigma-1}/(\sigma-1)!, \tau^{\sigma}/\sigma!]^{\mathsf{T}}.$$

Similarly,

$$\dot{\mathbf{x}}_{\mathsf{M}}(t+\tau) \approx X_{\mathsf{M}}(t)\mathbf{w}_2(\tau) + \tau^{\sigma-1} g_{\mathsf{M}}(\mathbf{x}_{\mathsf{M}}(t))\mathbf{u}(t)/(\sigma-1)!,$$

where

$$\mathbf{w}_2(\tau) = [0, 1, \tau, \cdots, \tau^{\sigma-1}/(\sigma-1)!]^{\mathsf{T}}.$$

In addition,

$$\mathbf{u}(t+\tau) \approx \mathbf{u}(t).$$

Then, $J(t)$ in (5.2) is approximated as

$$\begin{aligned} J(t) &\approx \hat{J}(t) \\ &= \int_0^T (X_{\mathsf{M}}(t)\mathbf{w}_1(\tau) + \frac{\tau^\sigma}{\sigma!} g_{\mathsf{M}}(\mathbf{x}_{\mathsf{M}}(t))\mathbf{u}(t))^{\mathsf{T}} L_0 (X_{\mathsf{M}}(t)\mathbf{w}_1(\tau) + \frac{\tau^\sigma}{\sigma!} g_{\mathsf{M}}(\mathbf{x}_{\mathsf{M}}(t))\mathbf{u}(t))\mathsf{d}\tau \\ &\quad + \int_0^T (X_{\mathsf{M}}(t)\mathbf{w}_2(\tau) + \frac{\tau^{\sigma-1}}{(\sigma-1)!} g_{\mathsf{M}}(\mathbf{x}_{\mathsf{M}}(t))\mathbf{u}(t))^{\mathsf{T}} \\ &\quad \times Q(X_{\mathsf{M}}(t)\mathbf{w}_2(\tau) + \frac{\tau^{\sigma-1}}{(\sigma-1)!} g_{\mathsf{M}}(\mathbf{x}_{\mathsf{M}}(t))\mathbf{u}(t))\mathsf{d}\tau. \end{aligned} \tag{5.3}$$

Let

$$\mathbf{v}_1 = \int_0^T \tau^\sigma \mathbf{w}_1(\tau)/\sigma!\mathsf{d}\tau,$$

$$\mathbf{v}_2 = \int_0^T \tau^{\sigma-1} \mathbf{w}_2(\tau)/(\sigma-1)!\mathsf{d}\tau,$$

$$\kappa_1 = \int_0^T \tau^{2\sigma}/(\sigma!)^2 \mathsf{d}\tau = T^{2\sigma+1}/((2\sigma+1)(\sigma!)^2),$$

and

$$\kappa_2 = \int_0^T \tau^{2\sigma-2}/((\sigma-1)!)^2 \mathsf{d}\tau = T^{2\sigma-1}/((2\sigma-1)((\sigma-1)!)^2).$$

Then, from (5.3),

$$\begin{aligned} \hat{J}(t) = \int_0^T & \mathbf{w}_1^{\mathsf{T}}(\tau) X_{\mathsf{M}}^{\mathsf{T}}(t) L_0 \times X_{\mathsf{M}}(t)\mathbf{w}_1(\tau) + \mathbf{w}_2^{\mathsf{T}}(\tau) X_{\mathsf{M}}^{\mathsf{T}}(t) Q X_{\mathsf{M}}(t)\mathbf{w}_2(\tau)\mathsf{d}\tau \\ & + 2\mathbf{v}_1^{\mathsf{T}} X_{\mathsf{M}}^{\mathsf{T}}(t) L_0 g_{\mathsf{M}}(\mathbf{x}_{\mathsf{M}}(t))\mathbf{u}(t) + \kappa_1 \mathbf{u}^{\mathsf{T}}(t) g_{\mathsf{M}}^{\mathsf{T}}(\mathbf{x}_{\mathsf{M}}(t)) L_0 g_{\mathsf{M}}(\mathbf{x}_{\mathsf{M}}(t))\mathbf{u}(t) \\ & + 2\mathbf{v}_2^{\mathsf{T}} X_{\mathsf{M}}^{\mathsf{T}}(t) Q g_{\mathsf{M}}(\mathbf{x}_{\mathsf{M}}(t))\mathbf{u}(t) + \kappa_2 \mathbf{u}^{\mathsf{T}}(t) \times g_{\mathsf{M}}^{\mathsf{T}}(\mathbf{x}_{\mathsf{M}}(t)) Q g_{\mathsf{M}}(\mathbf{x}_{\mathsf{M}}(t))\mathbf{u}(t). \end{aligned}$$

In light of the fact that the decision variable is input $\mathbf{u}(t)$, minimizing $\hat{J}(t)$ is then equivalent to minimizing

$$\begin{aligned} \hat{J}_{\mathsf{e}}(t) = & 2\mathbf{v}_1^{\mathsf{T}} X_{\mathsf{M}}^{\mathsf{T}}(t) L_0 g_{\mathsf{M}}(\mathbf{x}_{\mathsf{M}}(t))\mathbf{u}(t) + \kappa_1 \mathbf{u}^{\mathsf{T}}(t) g_{\mathsf{M}}^{\mathsf{T}}(\mathbf{x}_{\mathsf{M}}(t)) L_0 g_{\mathsf{M}}(\mathbf{x}_{\mathsf{M}}(t))\mathbf{u}(t) \\ & + 2\mathbf{v}_2^{\mathsf{T}} X_{\mathsf{M}}^{\mathsf{T}}(t) Q g_{\mathsf{M}}(\mathbf{x}_{\mathsf{M}}(t))\mathbf{u}(t) + \kappa_2 \mathbf{u}^{\mathsf{T}}(t) g_{\mathsf{M}}^{\mathsf{T}}(\mathbf{x}_{\mathsf{M}}(t)) Q g_{\mathsf{M}}(\mathbf{x}_{\mathsf{M}}(t))\mathbf{u}(t). \end{aligned}$$

Note that $\kappa_1 > 0$, $\kappa_2 > 0$, $Q$ is a positive-definite diagonal matrix, $L_0$ is a positive semi-definite symmetric matrix and $g_{\mathsf{M}}(\mathbf{x}_{\mathsf{M}}(t))$ is a diagonal matrix of positive elements. It follows that $\kappa_1 g_{\mathsf{M}}^{\mathsf{T}}(\mathbf{x}_{\mathsf{M}}(t))L_0 g_{\mathsf{M}}(\mathbf{x}_{\mathsf{M}}(t)) + \kappa_2 g_{\mathsf{M}}^{\mathsf{T}}(\mathbf{x}_{\mathsf{M}}(t))Q g_{\mathsf{M}}(\mathbf{x}_{\mathsf{M}}(t))$ is positive-definite, i.e., $\hat{J}_{\mathsf{e}}(t)$ is a convex quadratic performance index. Then, a near-optimal protocol can be obtained by solving $\partial \hat{J}_{\mathsf{e}}(t)/\partial \mathbf{u} = 0$, i.e.,

$$\begin{aligned}&2g_{\mathsf{M}}(\mathbf{x}_{\mathsf{M}}(t))L_0 X_{\mathsf{M}}(t)\mathbf{v}_1 + 2\kappa_1 g_{\mathsf{M}}(\mathbf{x}_{\mathsf{M}}(t))L_0 g_{\mathsf{M}}(\mathbf{x}_{\mathsf{M}}(t))\mathbf{u}(t) + 2g_{\mathsf{M}}(\mathbf{x}_{\mathsf{M}}(t))Q X_{\mathsf{M}}(t)\mathbf{v}_2\\ &\quad + 2\kappa_2 g_{\mathsf{M}}(\mathbf{x}_{\mathsf{M}}(t))Q g_{\mathsf{M}}(\mathbf{x}_{\mathsf{M}}(t))\mathbf{u}(t) = 0.\end{aligned}$$

It follows that

$$L_0 X_{\mathsf{M}}(t)\mathbf{v}_1 + \kappa_1 L_0 g_{\mathsf{M}}(\mathbf{x}_{\mathsf{M}}(t))\mathbf{u}(t) + Q X_{\mathsf{M}}(t)\mathbf{v}_2 + \kappa_2 Q g_{\mathsf{M}}(\mathbf{x}_{\mathsf{M}}(t))\mathbf{u}(t) = 0.$$

Substituting the expressions of $\mathbf{v}_1$ and $\mathbf{v}_2$ into the above equation yields

$$\begin{aligned}&Q\kappa_4 \sum_{j=1}^{\sigma-1} \frac{T^{j-\sigma}(2\sigma-1)(\sigma-1)!}{(\sigma+j-1)(j-1)!}\mathbf{x}_{\mathsf{M}}^{(j)} + (L_0\kappa_3 + Q\kappa_4)(f_{\mathsf{M}}(\mathbf{x}_{\mathsf{M}}) + g_{\mathsf{M}}(\mathbf{x}_{\mathsf{M}}(t))\mathbf{u}(t))\\ &= \sum_{j=0}^{\sigma-1} \frac{-T^{j+2}L_0\mathbf{x}_{\mathsf{M}}^{(j)}}{(\sigma+j+1)\sigma j!},\end{aligned} \tag{5.4}$$

where $\kappa_3 = \kappa_1(\sigma-1)!/T^{\sigma-1} = T^{\sigma+2}/(2\sigma+1)/\sigma/\sigma!$ and $\kappa_4 = \kappa_2(\sigma-1)!/T^{\sigma-1} = T^{\sigma}/(2\sigma-1)/(\sigma-1)!$. Let $(L_0\kappa_3 + Q\kappa_4)^{-1}L_0 = L$. Define $\alpha_j$ and $\beta_j$ as follows:

$$\begin{aligned}\alpha_j &= \frac{T^{j+2}}{(\sigma+j+1)\sigma j!} - \frac{T^{j+2}(2\sigma-1)}{\sigma^2(2\sigma+1)(\sigma+j-1)(j-1)!},\\ \beta_j &= \frac{T^{j-\sigma}(2\sigma-1)(\sigma-1)!}{(\sigma+j-1)(j-1)!}.\end{aligned}$$

Then, by solving equation (5.4), a nominal near-optimal protocol is obtained as follows:

$$\mathbf{u}(t) = g_{\mathsf{M}}^{-1}(\mathbf{x}_{\mathsf{M}}(t))(-\frac{T^2 L\mathbf{x}_{\mathsf{M}}(t)}{\sigma(\sigma+1)} - L\sum_{j=1}^{\sigma-1}\alpha_j\mathbf{x}_{\mathsf{M}}^{(j)}(t) - f_{\mathsf{M}}(\mathbf{x}_{\mathsf{M}}(t)) - \sum_{j=1}^{\sigma-1}\beta_j\mathbf{x}_{\mathsf{M}}^{(j)}(t)). \tag{5.5}$$

From (5.5), for the $i$th agent, the control law is

$$\begin{aligned}u_i(t) = g_i^{-1}(\mathbf{x}_i(t))(&-T^2\sum_{k\in\mathbb{N}(i)}(\mathbf{x}_i(t)-\mathbf{x}_k(t))/(\sigma(\sigma+1)) - \sum_{j=1}^{\sigma-1}\sum_{k\in\mathbb{N}(i)}\alpha_j(\mathbf{x}_i^{(j)}(t)-\mathbf{x}_k^{(j)}(t))\\ &- f_i(\mathbf{x}_i(t)) - \sum_{j=1}^{\sigma-1}\beta_j\mathbf{x}_i^{(j)}(t)),\end{aligned}$$

where $\mathbb{N}(i)$ denotes the set of neighbors of agent $i$ in the undirected connected graph characterized by Laplacian matrix $L$.

*Remark 2*: From $(L_0\kappa_3 + Q\kappa_4)^{-1}L_0 = L$ and $Q = \zeta I$, one has $L_0(I - L\kappa_3) = \kappa_4\zeta L$. By the spectral theorem [147], Laplacian matrix $L$ is orthogonally decomposed as

$$L = \lambda_1\mathbf{a}_1\mathbf{a}_1^\mathsf{T} + \lambda_2\mathbf{a}_2\mathbf{a}_2^\mathsf{T} + \cdots + \lambda_n\mathbf{a}_n\mathbf{a}_n^\mathsf{T},$$

with $\lambda_1 = 0 < \lambda_2 \leq \cdots \leq \lambda_n$ being its eigenvalues and $\mathbf{a}_i$ being the corresponding eigenvectors after normalization with $\mathbf{a}_1 = \mathbf{1}/\sqrt{n}$. Similarly, identity matrix $I$ can be decomposed with the same set of orthogonal eigenvectors as

$$I = \mathbf{a}_1\mathbf{a}_1^\mathsf{T} + \mathbf{a}_2\mathbf{a}_2^\mathsf{T} + \cdots + \mathbf{a}_n\mathbf{a}_n^\mathsf{T}.$$

Noting $\lambda_1 = 0$, it follows that

$$I - \kappa_3 L = \mathbf{a}_1\mathbf{a}_1^\mathsf{T} + (1 - \kappa_3\lambda_2)\mathbf{a}_2\mathbf{a}_2^\mathsf{T} + \cdots + (1 - \kappa_3\lambda_n)\mathbf{a}_n\mathbf{a}_n^\mathsf{T}.$$

Note that $\kappa_3 = T^{\sigma+2}/((2\sigma+1)\sigma\sigma!)$. If none of the eigenvalues of $I - \kappa_3 L$ is 0, i.e., $T \neq ((2\sigma+1)\sigma\sigma!)/(\lambda_i))^{1/(\sigma+2)}$, for all $i = 2, 3, \cdots, n$, then $I - \kappa_3 L$ is invertible. In addition, by the property of spectral decomposition [147],

$$(I - \kappa_3 L)^{-1} = \mathbf{a}_1\mathbf{a}_1^\mathsf{T} + \mathbf{a}_2\mathbf{a}_2^\mathsf{T}/(1 - \kappa_3\lambda_2) + \cdots + \mathbf{a}_n\mathbf{a}_n^\mathsf{T}/(1 - \kappa_3\lambda_n).$$

Noting that $\mathbf{a}_i^\mathsf{T}\mathbf{a}_i = 1$ for all $i$ and $\mathbf{a}_i^\mathsf{T}\mathbf{a}_j = 0$ for $i \neq j$ due to the spectral decomposition property [147], one has

$$L(I - \kappa_3 L)^{-1} = \lambda_1\mathbf{a}_1\mathbf{a}_1^\mathsf{T} + \lambda_2\mathbf{a}_2\mathbf{a}_2^\mathsf{T}/(1 - \kappa_3\lambda_2) + \cdots + \lambda_n\mathbf{a}_n\mathbf{a}_n^\mathsf{T}/(1 - \kappa_3\lambda_n).$$

Thus,

$$L_0 = \zeta\kappa_4 L(I - \kappa_3 L)^{-1} = \zeta\kappa_4\lambda_1\mathbf{a}_1\mathbf{a}_1^\mathsf{T} + \zeta\kappa_4\lambda_2\mathbf{a}_2\mathbf{a}_2^\mathsf{T}/(1 - \kappa_3\lambda_2) + \cdots + \mathbf{a}_n\mathbf{a}_n^\mathsf{T}\zeta\kappa_4\lambda_n/(1 - \kappa_3\lambda_n).$$

Evidently, $L_0$ is symmetric since $\mathbf{a}_i\mathbf{a}_i^\mathsf{T}$ is symmetric for all $i = 1, 2, \cdots, n$. Thus, $L_0$ satisfies property 1). Given that $1 - \kappa_3\lambda_i > 0$, for all $i = 2, \cdots, n$, i.e., $1 - \kappa_3\lambda_n > 0$,

which is equivalent to $0 < T < ((2\sigma+1)\sigma\sigma!/\lambda_n)^{1/(\sigma+2)}$, the least eigenvalue of $L_0$ is $\zeta\kappa_4\lambda_1 = \zeta\kappa_4 \times 0 = 0$, and the other eigenvalues are strictly bigger than 0. From the decomposition expression of $L_0$, $\mathbf{1}$ is thus an eigenvector of $L_0$ corresponding to eigenvalue 0. Therefore, if $0 < T < ((2\sigma+1)\sigma\sigma!/\lambda_n)^{1/(\sigma+2)}$, then $L_0 = \zeta\kappa_4 L(I - \kappa_3 L)^{-1}$ satisfies property 1) and property 2). This guarantees the existence of $L_0$.

*Theorem 1:* If all the roots of the equation $s^\sigma + \sum_{j=1}^{\sigma-1}(\chi_i\alpha_j + \beta_j)s^j + T^2\chi_i/(\sigma(\sigma+1)) = 0$ are located at the left half-plane for all $i = 1, 2, \cdots, n$, where $\chi_1 = n\varepsilon$ with $\varepsilon > 0 \in \mathbb{R}$ and $\chi_k = \lambda_k$ for $k = 2, 3, \cdots, n$ with $\lambda_k$ being the positive eigenvalues of $L$, then nonlinear multi-agent system (5.1) synthesized by nominal near-optimal protocol (5.5) exponentially converges to consensus.

*Proof:* Substituting nominal near-optimal protocol (5.5) into the last equation of (5.1) yields

$$\mathbf{x}_{\mathsf{M}}^{(\sigma)}(t) = -T^2 L\mathbf{x}_{\mathsf{M}}(t)/(\sigma(\sigma+1)) - \sum_{j=1}^{\sigma-1}(L\alpha_j + \beta_j I)\mathbf{x}_{\mathsf{M}}^{(j)}(t).$$

Let $\mathbf{e}(t) = L\mathbf{x}_{\mathsf{M}}(t)$. Thus,

$$\mathbf{e}^{(\sigma)}(t) = -T^2 L\mathbf{e}(t)/(\sigma(\sigma+1)) - \sum_{j=1}^{\sigma-1}(\alpha_j L\mathbf{e}^{(j)}(t) + \beta_j \mathbf{e}^{(j)}(t)).$$

For Laplacian matrix $L$, one has $L = L^T$ and $L\mathbf{1} = 0$ [147]. Thus, $\mathbf{1}^\mathsf{T}L = 0$ and $\varepsilon\mathbf{1}\mathbf{1}^\mathsf{T}\mathbf{e}(t) = 0$ for any $\varepsilon > 0$. Then, $L\mathbf{e}(t) = (L + \varepsilon\mathbf{1}\mathbf{1}^\mathsf{T})\mathbf{e}(t)$. Let $A = L + \varepsilon\mathbf{1}\mathbf{1}^\mathsf{T}$. Then,

$$\mathbf{e}^{(\sigma)}(t) = -\frac{T^2 A\mathbf{e}(t)}{\sigma(\sigma+1)} - \sum_{j=1}^{\sigma-1}(A\alpha_j + \beta_j I)\mathbf{e}^{(j)}(t). \tag{5.6}$$

Since $L$ is symmetric, by the spectral theorem [147], $L$ can be decomposed as

$$L = \lambda_1\mathbf{a}_1\mathbf{a}_1^\mathsf{T} + \lambda_2\mathbf{a}_2\mathbf{a}_2^\mathsf{T} + \cdots + \lambda_n\mathbf{a}_n\mathbf{a}_n^\mathsf{T},$$

where $\lambda_i$ and $\mathbf{a}_i$ denote the $i$th eigenvalue and the $i$th eigenvector after the normalization process, respectively, and $i = 1, 2, \cdots, n$. Note that the least eigenvalue of a Laplacian matrix is 0, which corresponds to eigenvector $\mathbf{1}$, and the second least eigenvalue is greater than 0 [147]. Let $\lambda_1 = 0$. Then, $\lambda_i > 0$ for $i \geq 2$ and $\mathbf{a}_1 = \mathbf{1}/\sqrt{n}$.

Similarly, matrix $A$ can be decomposed as

$$A = (\lambda_1 + n\varepsilon)\mathbf{1}\mathbf{1}^{\mathsf{T}}/n + \lambda_2 \mathbf{a}_2\mathbf{a}_2^{\mathsf{T}} + \cdots, \lambda_n \mathbf{a}_n\mathbf{a}_n^{\mathsf{T}},$$

which indicates that the eigenvalues of $A$ are $\lambda_1 + n\varepsilon = n\varepsilon$, $\lambda_2$, $\cdots$, $\lambda_n$, all of which are positive. Therefore, $A$ is a positive-definite symmetric matrix. By eigenvalue decomposition, there exist orthogonal matrix $P \in \mathbb{R}^{n\times n}$ and diagonal matrix $\Lambda = \mathsf{diag}([\chi_1, \chi_2, \cdots, \chi_n]^{\mathsf{T}}) = \mathsf{diag}([n\varepsilon, \lambda_2, \cdots, \lambda_n]^{\mathsf{T}})$ such that $A = P\Lambda P^{\mathsf{T}}$. Let $\mathbf{y}(t) = P^{\mathsf{T}}\mathbf{e}(t)$. From equation (5.6),

$$\mathbf{y}^{(\sigma)}(t) = -T^2\Lambda\mathbf{y}(t)/(\sigma(\sigma+1)) - \sum_{j=1}^{\sigma-1}(\Lambda\alpha_j + \beta_j I)\mathbf{y}^{(j)}(t),$$

of which the $i$th subsystem is

$$y_i^{(\sigma)}(t) = -T^2\chi_i y_i(t)/(\sigma(\sigma+1)) - \sum_{j=1}^{\sigma-1}(\chi_i\alpha_j + \beta_j)y_i^{(j)}(t)$$

with the characteristic equation being

$$s^\sigma + \sum_{j=1}^{\sigma-1}(\chi_i\alpha_j + \beta_j)s^j + \frac{T^2}{\sigma(\sigma+1)}\chi_i = 0. \tag{5.7}$$

By linear system theory [44], given that all the roots of (5.7) are located at the left half-plane for all $i = 1, 2, \cdots, n$, all the subsystems are exponentially stable. Then, $\mathbf{y}(t)$ exponentially converges to zero. Recall that $P$ is an orthogonal matrix and $\mathbf{y}(t) = P^{\mathsf{T}}\mathbf{e}(t)$. It follows that $\mathbf{e}(t)$ exponentially converges to zero. □

*Theorem 2:* Nominal near-optimal protocol (5.5) for nonlinear multi-agent system (5.1) with fully known dynamics converges to optimal with time, if all the roots of equations $s^\sigma + \sum_{j=1}^{\sigma-1}(\chi_i\alpha_j + \beta_j)s^j + T^2\chi_i/\sigma(\sigma+1) = 0$ and $s^{\sigma-1} + \sum_{j=1}^{\sigma-1}\beta_j s^{j-1} = 0$ are located at the left half-plane for all $i = 1, 2, \cdots, n$, where $\chi_1 = n\varepsilon$ with $\varepsilon > 0 \in \mathbb{R}$, and $\chi_k = \lambda_k$ for $k = 2, 3, \cdots, n$ with $\lambda_k$ being the positive eigenvalues of $L$.

*Proof:* According to Taylor expansion, $J(t)$ in equation (5.2) is rewritten as $J(t) = \int_0^T (X_{\mathsf{M}}(t)\mathbf{w}_1(\tau) + \tau^\sigma g_{\mathsf{M}}(\mathbf{x}_{\mathsf{M}}(t))\mathbf{u}(t)/\sigma! + \Theta_1(t))^{\mathsf{T}} L_0 (X_{\mathsf{M}}(t)\mathbf{w}_1(\tau) + \tau^\sigma g_{\mathsf{M}}(\mathbf{x}_{\mathsf{M}}(t))\mathbf{u}(t)/\sigma! + \Theta_1(t))\mathsf{d}\tau + \int_0^T (X_{\mathsf{M}}(t)\mathbf{w}_2(\tau) + \tau^{\sigma-1} g_{\mathsf{M}}(\mathbf{x}_{\mathsf{M}}(t))\mathbf{u}(t)/(\sigma-1)! + \Theta_2(t))^{\mathsf{T}} Q (X_{\mathsf{M}}(t)\mathbf{w}_2(\tau) +$

$\tau^{\sigma-1}g_{\mathsf{M}}(\mathbf{x}_{\mathsf{M}}(t))\mathbf{u}(t)/(\sigma-1)!+\Theta_2(t))\mathsf{d}\tau$, where $\Theta_1(t)=\tau^{\sigma}(\mathbf{x}_{\mathsf{M}}^{(\sigma)}(t+\varkappa\tau)-\mathbf{x}_{\mathsf{M}}^{(\sigma)}(t))/\sigma!$ and $\Theta_2(t)=\tau^{\sigma-1}(\mathbf{x}_{\mathsf{M}}^{(\sigma)}(t+\varkappa\tau)-\mathbf{x}_{\mathsf{M}}^{(\sigma)}(t))/(\sigma-1)!$ with $\varkappa\in(0,1)$. Recall $\hat{J}(t)$ in (5.3). By triangle inequality [148], one further has

$$\begin{aligned}J(t)&\leq 2\hat{J}(t)+2\int_0^T\Theta_1^{\mathsf{T}}(t)L_0\Theta_1(t)\mathsf{d}\tau+2\int_0^T\Theta_2^{\mathsf{T}}(t)Q\Theta_2(t)\mathsf{d}\tau\\&=2\hat{J}(t)+2T\Theta_1^{\mathsf{T}}(t)L_0\Theta_1(t)+2T\Theta_2^{\mathsf{T}}(t)Q\Theta_2(t).\end{aligned}$$

Since nominal near-optimal protocol (5.5) minimizes convex function $\hat{J}(t)$, one has $\hat{J}(t)=0$ when nominal near-optimal protocol (5.5) is adopted. It follows that

$$J(t)\leq 2T\Theta_1^{\mathsf{T}}(t)L_0\Theta_1(t)+2T\Theta_2^{\mathsf{T}}(t)Q\Theta_2(t).$$

Substituting (5.5) into the last equation of (5.1) yields

$$\mathbf{x}_{\mathsf{M}}^{(\sigma)}(t)+\sum_{j=1}^{\sigma-1}\beta_j\mathbf{x}_{\mathsf{M}}^{(j)}(t)=\delta(t),$$

where $\delta(t)=-T^2L\mathbf{x}_{\mathsf{M}}(t)/(\sigma(\sigma+1))-L\sum_{j=1}^{\sigma-1}\alpha_j\mathbf{x}_{\mathsf{M}}^{(j)}(t)$. Let $\mathbf{z}_{\mathsf{M}}(t)=\dot{\mathbf{x}}_{\mathsf{M}}(t)$. Then, $\mathbf{z}_{\mathsf{M}}^{(\sigma-1)}(t)+\sum_{j=1}^{\sigma-1}\beta_j\mathbf{z}_{\mathsf{M}}^{(j-1)}(t)=\delta(t)$. When $\delta(t)=0$, one has

$$\mathbf{z}_{\mathsf{M}}^{(\sigma-1)}(t)+\sum_{j=1}^{\sigma-1}\beta_j\mathbf{z}_{\mathsf{M}}^{(j-1)}(t)=0,\tag{5.8}$$

of which each subsystem has the same characteristic equation $s^{\sigma-1}+\sum_{j=1}^{\sigma-1}\beta_js^{j-1}=0$. By linear system theory [44], if all the roots of the characteristic equation are located at the half-plane, then system (5.8) is exponentially stable. From Theorem 1, if all the roots of the equation $s^{\sigma}+\sum_{j=1}^{\sigma-1}(\chi_i\alpha_j+\beta_j)s^j+T^2\chi_i/(\sigma(\sigma+1))=0$ are located at the half-plane for all $i=1,2,\cdots,n$, then $\lim_{t\to+\infty}L\mathbf{x}_{\mathsf{M}}(t)=0$ yielding $\lim_{t\to+\infty}\delta(t)=0$. By bounded-input bounded-output stability of linear systems [44], system $\mathbf{x}_{\mathsf{M}}^{(\sigma)}(t)+\sum_{j=1}^{\sigma-1}\beta_j\mathbf{x}_{\mathsf{M}}^{(j)}(t)=\delta(t)$ is asymptotically stable with $\mathbf{z}_{\mathsf{M}}(t)=\dot{\mathbf{x}}_{\mathsf{M}}(t)$ converging to zero. It follows that $\lim_{t\to+\infty}\mathbf{x}_{\mathsf{M}}^{(\sigma)}(t)=0$ and $\lim_{t\to+\infty}\mathbf{x}_{\mathsf{M}}^{(\sigma)}(t+\varkappa\tau)=0$, yielding $\lim_{t\to+\infty}\Theta_1(t)=0$ and $\lim_{t\to+\infty}\Theta_2(t)=0$. Then, $\lim_{t\to+\infty}2T\Theta_1^{\mathsf{T}}(t)L_0\Theta_1(t)+2T\Theta_2^{\mathsf{T}}(t)Q\Theta_2(t)=0$. Note that $J(t)\geq 0$. Recall that $J(t)\leq 2T\Theta_1^{\mathsf{T}}(t)L_0\Theta_1(t)+2T\Theta_2^{\mathsf{T}}(t)Q\Theta_2(t)$. According to the pinching theorem [175], $\lim_{t\to+\infty}J(t)=0$. □

## 5.3 Adaptive near-optimal design

In this section, we further consider the situation where the parameters of multi-agent system (5.1) are unknown, i.e., the parameters in $f_i(\mathbf{x}_i)$ and $g_i(\mathbf{x}_i)$ are unknown. Specifically, the $i$th agent of multi-agent system (5.1) is reformulated as

$$\begin{aligned}
\dot{x}_i^{(0)} &= x_i^{(1)}, \\
&\cdots \\
\dot{x}_i^{(\sigma-2)} &= x_i^{(\sigma-1)}, \\
\dot{x}_i^{(\sigma-1)} &= \boldsymbol{\omega}_i^{\mathsf{T}}\phi_i(\mathbf{x}_i) + \mathbf{b}_i^{\mathsf{T}}\varphi_i(\mathbf{x}_i)u_i,
\end{aligned}$$

where $\boldsymbol{\omega}_i$ and $\mathbf{b}_i$ are unknown constant parameter vectors; $\phi(\mathbf{x}_i)$ and $\varphi(\mathbf{x}_i)$ are known basis function vectors. Then, multi-agent system (5.1) is reformulated as

$$\begin{cases}
\dot{\mathbf{x}}_{\mathsf{M}}^{(k)} = \mathbf{x}_{\mathsf{M}}^{(k+1)}, k = 0, 1, \cdots, \sigma-2, \\
\dot{\mathbf{x}}_{\mathsf{M}}^{(\sigma-1)} = W\phi(\mathbf{x}_{\mathsf{M}}) + B\varphi(\mathbf{x}_{\mathsf{M}})\mathbf{u},
\end{cases} \tag{5.9}$$

with

$$W = [\boldsymbol{\omega}_1^{\mathsf{T}}, 0, \cdots, 0; 0, \boldsymbol{\omega}_2^{\mathsf{T}}, \cdots, 0; \cdots; 0, 0, \cdots, \boldsymbol{\omega}_n^{\mathsf{T}}]$$

and

$$B = [\mathbf{b}_1^{\mathsf{T}}, 0, \cdots, 0; 0, \mathbf{b}_2^{\mathsf{T}}, \cdots, 0; \cdots; 0, 0, \cdots, \mathbf{b}_n^{\mathsf{T}}]$$

being parameter matrices. In addition, $\phi(\mathbf{x}_{\mathsf{M}}) = [\phi_1^{\mathsf{T}}(\mathbf{x}_1), \phi_2^{\mathsf{T}}(\mathbf{x}_2), \cdots, \phi_n^{\mathsf{T}}(\mathbf{x}_n)]^{\mathsf{T}}$ and $\varphi(\mathbf{x}_{\mathsf{M}}) = [\varphi_1(\mathbf{x}_1), 0, \cdots, 0; 0, \varphi_2(\mathbf{x}_2), \cdots, 0; \cdots; 0, 0, \cdots, \varphi_n(\mathbf{x}_n)]$ are basis function matrices. To handle the parameter uncertainty, we design the following auxiliary system to reconstruct the input-output property of system (5.1):

$$\begin{cases}
\hat{\mathbf{x}}_{\mathsf{M}}^{(\sigma)}(t) = \hat{W}(t)\phi(\mathbf{x}_{\mathsf{M}}(t)) + \hat{B}(t)\varphi(\mathbf{x}_{\mathsf{M}}(t))\mathbf{u}(t) - \sum_{c=0}^{\sigma-2}\nu_c\tilde{\mathbf{x}}_{\mathsf{M}}^{(c+1)}(t) - \gamma\mathbf{s}(t), \\
\dot{\hat{W}}(t) = -K_1\mathbf{s}(t)\phi^{\mathsf{T}}(\mathbf{x}_{\mathsf{M}}(t)), \\
\dot{\hat{B}}(t) = -K_2\mathbf{s}(t)\mathbf{u}^{\mathsf{T}}(t)\varphi^{\mathsf{T}}(\mathbf{x}_{\mathsf{M}}(t)),
\end{cases} \tag{5.10}$$

where $\mathbf{s}(t) = \sum_{c=0}^{\sigma-1}\nu_c\tilde{\mathbf{x}}_{\mathsf{M}}^{(c)}(t)$ with $\nu_{\sigma-1} = 1$; $\tilde{\mathbf{x}}_{\mathsf{M}}(t) = \hat{\mathbf{x}}_{\mathsf{M}}(t) - \mathbf{x}_{\mathsf{M}}(t)$ with $\hat{\mathbf{x}}_{\mathsf{M}}(t)$ being the auxiliary state vector; $\hat{W}(t)$ and $\hat{B}(t)$ are auxiliary parameter matrices; $K_1 \in \mathbb{R}^{n\times n}$ and

$K_2 \in \mathbb{R}^{n\times n}$ are diagonal positive-definite gain matrices; $\gamma > 0 \in \mathbb{R}$ is a parameter used to scale the state feedback. The auxiliary system is aided by the concept of sliding mode control [144, 152] with the sliding surface being $\mathbf{s}(t) = 0$, and is thus called sliding-mode auxiliary system.

*Remark 3:* From (5.10), for the $i$th agent, the corresponding auxiliary system is

$$\begin{cases} \hat{x}_i^{(\sigma)}(t) = \hat{\boldsymbol{\omega}}_i^{\mathsf{T}}(t)\phi_i(\mathbf{x}_i(t)) + \hat{\mathbf{b}}_i^{\mathsf{T}}\varphi_i(\mathbf{x}_i(t))u_i(t) - \sum_{c=0}^{\sigma-2}\nu_c\tilde{x}_i^{(c+1)}(t) - \gamma s_i(t), \\ \dot{\hat{\boldsymbol{\omega}}}_i^{\mathsf{T}}(t) = -k_{1ii}s_i(t)\phi_i^{\mathsf{T}}(\mathbf{x}_i(t)), \\ \dot{\hat{\mathbf{b}}}_i^{\mathsf{T}}(t) = -k_{2ii}s_i(t)u_i(t)\varphi_i^{\mathsf{T}}(\mathbf{x}_i(t)), \end{cases} \tag{5.11}$$

where $k_{1ii}$ and $k_{2ii}$ denote the $i$th diagonal elements of matrices $K_1$ and $K_2$ respectively. Evidently, the auxiliary system of each agent is independent from other agents.

*Theorem 3:* The input-output property of sliding-mode auxiliary system (5.10) asymptotically converges to that of multi-agent system (5.1) with fully unknown parameters.

*Proof:* Let $\tilde{W}(t) = \hat{W}(t) - W$ and $\tilde{B}(t) = \hat{B}(t) - B$. Consider candidate Lyapunov function

$$V_1(t) = \mathbf{s}^{\mathsf{T}}(t)\mathbf{s}(t)/2 + \mathrm{tr}(\tilde{W}^{\mathsf{T}}(t)K_1^{-1}\tilde{W}(t))/2 + \mathrm{tr}(\tilde{B}^{\mathsf{T}}(t)K_2^{-1}\tilde{B}(t))/2.$$

The first equation of (5.10) is written as

$$\hat{\mathbf{x}}_{\mathsf{M}}^{(\sigma)}(t) = \hat{W}(t)\phi(\mathbf{x}_{\mathsf{M}}(t)) + \hat{B}(t)\varphi(\mathbf{x}_{\mathsf{M}}(t))\mathbf{u}(t) - \dot{\mathbf{s}}(t) - \gamma\mathbf{s}(t) + \tilde{\mathbf{x}}_{\mathsf{M}}^{(\sigma)}(t).$$

Subtracting equation (5.9) from the above equation yields

$$\dot{\mathbf{s}}(t) = -\gamma\mathbf{s}(t) + \tilde{W}(t)\phi(\mathbf{x}_{\mathsf{M}}(t)) + \tilde{B}(t)\varphi(\mathbf{x}_{\mathsf{M}}(t))\mathbf{u}(t).$$

Then,

$$\dot{V}_1(t) = -\gamma\|\mathbf{s}(t)\|_2^2 \leq 0.$$

Define set

$$\mathbb{S} = \{\tilde{\mathbf{x}}_{\mathsf{M}}(t) \in \mathbb{R}^n | \dot{V}_1(t) = 0\}.$$

From $\dot{V}_1(t) = 0$, one readily has

$$\mathbb{S} = \{\tilde{\mathbf{x}}_{\mathsf{M}}(t) \in \mathbb{R}^n | \mathbf{s}(t) = 0\}.$$

Recall

$$\mathbf{s}(t) = \sum_{c=0}^{\sigma-1} \nu_c \tilde{\mathbf{x}}_{\mathsf{M}}^{(c)}(t).$$

Given that the coefficients $\nu_c$ for $c = 0, 1, 2, \cdots, \sigma - 1$ with $\nu_{\sigma-1} = 1$ satisfy the Routh-Hurwitz stability criterion [44], $\lim_{t\to+\infty} \tilde{\mathbf{x}}_{\mathsf{M}}(t) = 0$. In other words, no solution can stay identically in set $\mathbb{S}$, other than the trivial solution $\tilde{\mathbf{x}}_{\mathsf{M}}(t) = 0$. By LaSalle's invariable set principle [44], equilibrium point $\tilde{\mathbf{x}}_{\mathsf{M}}(t) = 0$ is asymptotic stable. It follows that, $\lim_{t\to+\infty} \mathbf{x}_{\mathsf{M}}^{(\rho)}(t) = \hat{\mathbf{x}}_{\mathsf{M}}^{(\rho)}(t) = \hat{W}(t)\phi(\mathbf{x}_{\mathsf{M}}(t)) + \hat{B}(t)\varphi(\mathbf{x}_{\mathsf{M}}(t))\mathbf{u}(t)$. □

Since parameter matrices $\hat{W}(t)$ and $\hat{B}(t)$ are generated by sliding-mode auxiliary system (5.10), they are totally known. Therefore, $\hat{J}(t)$ in (5.3) is further approximated as follows:

$$\begin{aligned}
\hat{J}(t) &\approx \bar{J}(t) \\
&= \int_0^T (\hat{X}_{\mathsf{M}}(t)\mathbf{w}_1(\tau) + \frac{\tau^\sigma}{\sigma!}\hat{B}(t)\varphi(\mathbf{x}_{\mathsf{M}}(t))\mathbf{u}(t))^{\mathsf{T}} L_0 (\hat{X}_{\mathsf{M}}(t)\mathbf{w}_1(\tau) + \frac{\tau^\sigma}{\sigma!}\hat{B}(t) \\
&\quad \times \varphi(\mathbf{x}_{\mathsf{M}}(t))\mathbf{u}(t))\mathsf{d}\tau + \int_0^T (\hat{X}_{\mathsf{M}}(t)\mathbf{w}_2(\tau) + \frac{\tau^{\sigma-1}}{(\sigma-1)!}\hat{B}(t)\varphi(\mathbf{x}_{\mathsf{M}}(t)) \\
&\quad \times \mathbf{u}(t))^{\mathsf{T}} Q (\hat{X}_{\mathsf{M}}(t)\mathbf{w}_2(\tau) + \frac{\tau^{\sigma-1}}{(\sigma-1)!}\hat{B}(t)\varphi(\mathbf{x}_{\mathsf{M}}(t))\mathbf{u}(t))\mathsf{d}\tau,
\end{aligned} \tag{5.12}$$

where $\hat{X}_{\mathsf{M}}(t) = [\mathbf{x}_{\mathsf{M}}(t), \cdots, \mathbf{x}_{\mathsf{M}}^{(\sigma-1)}(t), \hat{W}(t)\phi(\mathbf{x}_{\mathsf{M}}(t))]$. Following similar steps in Section 5.2, a protocol minimizing performance index $\bar{J}(t)$ is obtained as follows:

$$\begin{aligned}
\mathbf{u}(t) = (\hat{B}(t)\varphi(\mathbf{x}_{\mathsf{M}}(t)))^{-1} (L(-\sum_{j=0}^{\sigma-1} \frac{T^{j+2}\mathbf{x}_{\mathsf{M}}^{(j)}(t)}{(\sigma+j+1)\sigma j!} + \sum_{j=1}^{\sigma-1} \frac{T^{j+2}(2\sigma-1)\mathbf{x}_{\mathsf{M}}^{(j)}(t)}{\sigma^2(2\sigma+1)(\sigma+j-1)(j-1)!}) \\
- \hat{W}(t)\phi(\mathbf{x}_{\mathsf{M}}(t)) - \sum_{j=1}^{\sigma-1} \frac{T^{j-\sigma}(2\sigma-1)(\sigma-1)!}{(\sigma+j-1)(j-1)!}\mathbf{x}_{\mathsf{M}}^{(j)}(t)).
\end{aligned} \tag{5.13}$$

Since $\hat{B}(t)\varphi(\mathbf{x}_{\mathsf{M}}(t)) = \mathsf{diag}([\hat{\mathbf{b}}_1^{\mathsf{T}}\varphi_1(\mathbf{x}_1(t)), \hat{\mathbf{b}}_2^{\mathsf{T}}\varphi_2(\mathbf{x}_2(t)), \cdots, \hat{\mathbf{b}}_n^{\mathsf{T}}\varphi_n(\mathbf{x}_n(t))]^{\mathsf{T}})$ is a diagonal matrix, it is evident that $(\hat{B}(t)\varphi(\mathbf{x}_{\mathsf{M}}(t)))^{-1}$ is also a diagonal matrix. In addition, $\hat{W}(t)\phi(\mathbf{x}_{\mathsf{M}}(t)) = [\hat{\boldsymbol{\omega}}_1^{\mathsf{T}}(t)\phi_1(\mathbf{x}_1(t)), \hat{\boldsymbol{\omega}}_2^{\mathsf{T}}(t)\phi_2(\mathbf{x}_2(t)),$

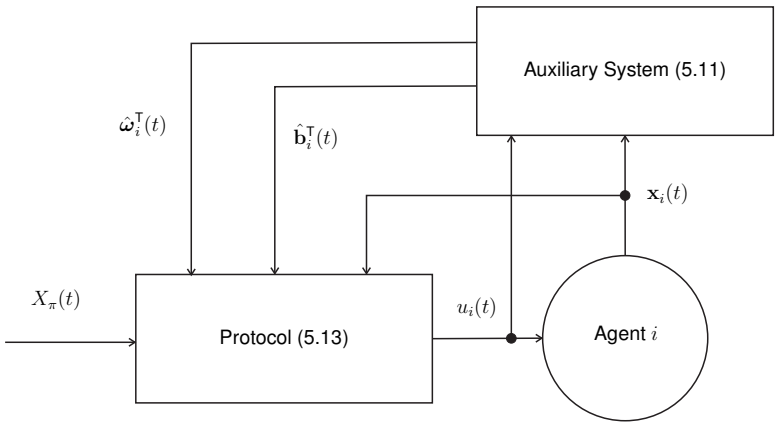



Figure 5.1: Block diagram about the $i$th agent of multi-agent system (5.1) synthesized by adaptive near-optimal protocol (5.13) and auxiliary system (5.11), where $X_\pi(t) = \{\mathbf{x}_j(t)|j \in \mathbb{N}(i)\}$ denotes the state set of neighbors of the $i$th agent.

$\cdots, \hat{\boldsymbol{\omega}}_n^{\mathsf{T}}(t)\phi_n(\mathbf{x}_n(t))]^{\mathsf{T}}$. Then, from (5.13), the control law for the $i$th agent is

$$\begin{aligned} u_i(t) = (\hat{\mathbf{b}}_i^{\mathsf{T}}\varphi_i(\mathbf{x}_i(t)))^{-1}((-\sum_{j=0}^{\sigma-1}\sum_{k\in\mathbb{N}(i)}\frac{T^{j+2}}{(\sigma+j+1)\sigma j!}(\mathbf{x}_i^{(j)}(t)-\mathbf{x}_k^{(j)}(t)) \\ +\sum_{j=1}^{\sigma-1}\sum_{k\in\mathbb{N}(i)}\frac{T^{j+2}(2\sigma-1)}{\sigma^2(2\sigma+1)}\frac{(\mathbf{x}_i^{(j)}(t)-\mathbf{x}_k^{(j)}(t))}{(\sigma+j-1)(j-1)!}-\hat{\boldsymbol{\omega}}_i^{\mathsf{T}}(t)\phi_i(\mathbf{x}_i(t)) \\ -\sum_{j=1}^{\sigma-1}\frac{T^{j-\sigma}(2\sigma-1)(\sigma-1)!}{(\sigma+j-1)(j-1)!}\mathbf{x}_i^{(j)}(t)). \end{aligned}$$

Thus, adaptive near-optimal protocol (5.13) is a distributed protocol, which is also explained by the block diagram shown in Fig. 5.1.

*Theorem 4:* If all the roots of equation (5.7) are located at the left half-plane for all $i = 1, 2, \cdots, n$, then, synthesized by adaptive near-optimal protocol (5.13) and sliding-mode auxiliary system (5.10), nonlinear multi-agent system (5.1) with fully unknown parameters exponentially converges to consensus.

*Proof:* Nonlinear multi-agent system (5.1) can be reformulated as

$$\mathbf{x}_{\mathsf{M}}^{(\sigma)}(t) = W\phi(\mathbf{x}_{\mathsf{M}}) + B\varphi(\mathbf{x}_{\mathsf{M}})\mathbf{u}(t).$$

Let $\tilde{W}(t) = \hat{W}(t) - W$ and $\tilde{B}(t) = \hat{B}(t) - B$. Then,

$$\mathbf{x}_{\mathsf{M}}^{(\sigma)} = \hat{W}(t)\phi(\mathbf{x}_{\mathsf{M}}) + \hat{B}(t)\varphi(\mathbf{x}_{\mathsf{M}})\mathbf{u}(t) + \eta(t),$$

where

$$\eta(t) = -(\tilde{W}(t)\phi(\mathbf{x}_{\mathsf{M}}) + \tilde{B}(t)\varphi(\mathbf{x}_{\mathsf{M}})\mathbf{u}(t)).$$

Then, by following similar steps in the proof of Theorem 1, the closed-loop system consisting of protocol (5.13) and multi-agent system (5.1) with unknown parameters is derived as

$$\mathbf{e}^{(\sigma)}(t) = \frac{-T^2 A\mathbf{e}(t)}{\sigma(\sigma+1)} - \sum_{j=1}^{\sigma-1}(A\alpha_j + \beta_j)\mathbf{e}^{(j)}(t) + \eta(t), \tag{5.14}$$

where

$$\mathbf{e}(t) = L\mathbf{x}_{\mathsf{M}}(t).$$

When $\eta(t) = 0$, closed-loop system (5.14) becomes

$$\mathbf{e}^{(\sigma)}(t) = -T^2 A\mathbf{e}(t)/(\sigma(\sigma+1)) - \sum_{j=1}^{\sigma-1}(A\alpha_j + \beta_j)\mathbf{e}^{(j)}(t),$$

which is exponentially stable, if all the roots of equation (5.7) are located at the left half-plane for all $i = 1, 2, \cdots, n$, according to the proof of Theorem 1. According to Theorem 2,

$$\lim_{t\to+\infty}(\tilde{W}(t)\phi(\mathbf{x}_{\mathsf{M}}(t)) + \tilde{B}(t)\varphi(\mathbf{x}_{\mathsf{M}}(t))\mathbf{u}(t)) = 0.$$

It follows that $\lim_{t\to+\infty}\eta(t) = 0$. By bounded-input bounded-output stability of linear systems [44], closed-loop system (5.14) is thus exponentially stable with $\mathbf{e}(t)$ exponentially converging to zero. □

*Theorem 5:* Adaptive near-optimal protocol (5.13) for nonlinear multi-agent system (5.1) with fully unknown parameters asymptotically converges to optimal, if all the roots of equations (5.7) and $s^{\sigma-1} + \sum_{j=1}^{\sigma-1}\beta_j s^{j-1} = 0$ are located at the left half-plane.

*Proof:* Recall equation (5.9). By Taylor expansion, performance index $J(t)$ in (5.2) is

rewritten as

$$
\begin{aligned}
J(t) = \int_0^T & (\hat{X}_{\mathsf{M}}(t)\mathbf{w}_1(\tau) + \tau^{\sigma}\hat{B}(t)\varphi(\mathbf{x}_{\mathsf{M}}(t))\mathbf{u}(t)/\sigma! + \Theta_1(t) + \Theta_3(t))^{\mathsf{T}} L_0(\hat{X}_{\mathsf{M}}(t)\mathbf{w}_1(\tau) \\
& + \tau^{\sigma}\hat{B}(t)\varphi(\mathbf{x}_{\mathsf{M}}(t))\mathbf{u}(t)/\sigma! + \Theta_1(t) + \Theta_3(t))\mathsf{d}\tau + \int_0^T (\hat{X}_{\mathsf{M}}(t)\mathbf{w}_2(\tau) + \tau^{\sigma-1}\hat{B}(t) \\
& \times \varphi(\mathbf{x}_{\mathsf{M}}(t))\mathbf{u}(t)/(\sigma-1)! + \Theta_2(t) + \Theta_4(t))^{\mathsf{T}} Q(X_{\mathsf{M}}(t)\mathbf{w}_2(\tau) + \tau^{\sigma-1}\hat{B}(t) \\
& \times \varphi(\mathbf{x}_{\mathsf{M}}(t))\mathbf{u}(t)/(\sigma-1)! + \Theta_2(t) + \Theta_4(t))\mathsf{d}\tau,
\end{aligned}
$$

where

$$
\begin{aligned}
\Theta_1(t) &= \tau^{\sigma}(\mathbf{x}_{\mathsf{M}}^{(\sigma)}(t + \varkappa\tau) - \mathbf{x}_{\mathsf{M}}^{(\sigma)}(t))/\sigma!, \\
\Theta_2(t) &= \tau^{\sigma-1}(\mathbf{x}_{\mathsf{M}}^{(\sigma)}(t + \varkappa\tau) - \mathbf{x}_{\mathsf{M}}^{(\sigma)}(t))/(\sigma-1)!, \\
\Theta_3(t) &= -\tau^{\sigma}(\tilde{W}(t)\phi(\mathbf{x}_{\mathsf{M}}(t)) + \tilde{B}(t)\varphi(\mathbf{x}_{\mathsf{M}}(t))\mathbf{u}(t))/\sigma!, \\
\Theta_4(t) &= -\tau^{\sigma-1}(\tilde{W}(t)\phi(\mathbf{x}_{\mathsf{M}}(t)) + \tilde{B}(t)\varphi(\mathbf{x}_{\mathsf{M}}(t))\mathbf{u}(t))/(\sigma-1)!.
\end{aligned}
$$

Recall performance index $\bar{J}(t)$ in (5.12). By triangle inequality [148],

$$
J(t) \leq 2\bar{J}(t) + 2T\Theta_1^{\mathsf{T}}(t)L_0\Theta_1(t) + 2T\Theta_3^{\mathsf{T}}(t)L_0\Theta_3(t) + 2\Theta_3^{\mathsf{T}}(t)Q\Theta_3(t) + 2T\Theta_4^{\mathsf{T}}(t)Q\Theta_4(t).
$$

Based on Theorem 3, one further has $\lim_{t\to+\infty}(\tilde{W}(t)\phi(\mathbf{x}_{\mathsf{M}}(t)) + \tilde{B}(t)\varphi(\mathbf{x}_{\mathsf{M}}(t))\mathbf{u}(t)) = 0$, yielding $\lim_{t\to+\infty}\Theta_3(t) = 0$ and $\lim_{t\to+\infty}\Theta_4(t) = 0$. Then, following similar steps in the proof of Theorem 2, based on Theorem 4, it can be readily proved that $\lim_{t\to+\infty} J(t) = 0$. □

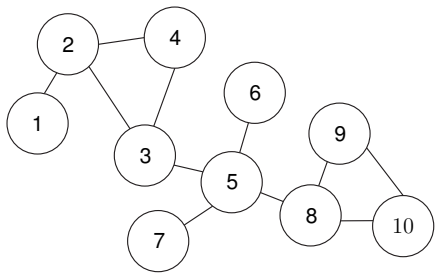



Figure 5.2: Communication topology of multi-agent system (5.15) consisting of 10 heterogeneous agents, where the $i$th node corresponds to the $i$th agent with $i = 1, 2, \cdots, 10$.

## 5.4 Illustrative example

Consider a nonlinear multi-agent system consisting of $10$ agents with the communication graph given in Fig. 5.2 and the following heterogeneous dynamics:

$$\begin{aligned}
x_1^{(3)} &= 3x_1 \sin(x_1) + 8\ddot{x}_1 + 10u_1, \\
x_2^{(3)} &= 8\cos(\dot{x}_2)x_2 - 3\ddot{x}_2 + 12u_2, \\
x_3^{(3)} &= 2x_3\ddot{x}_3 - 2\dot{x}_3 + 4\sin(\ddot{x}_3) + 7u_3, \\
x_4^{(3)} &= 2\sin(x_4) + 9u_4, \\
x_5^{(3)} &= -3\ddot{x}_5 \sin(\dot{x}_5) + 4x_5 + 5u_5, \\
x_6^{(3)} &= 6\dot{x}_6^2 - 2x_6 + 14u_6, \\
x_7^{(3)} &= 2\cos(x_7) - 3\dot{x}_7 + 9u_7, \\
x_8^{(3)} &= 9\sin(x_8) + 18u_8, \\
x_9^{(3)} &= 7\sin(\dot{x}_9) + 4x_9\dot{x}_9 + 15u_9, \\
x_{10}^{(3)} &= 2x_{10}\dot{x}_{10} + 6u_{10}.
\end{aligned} \tag{5.15}$$

The parameters of auxiliary system (5.10) are set to $\gamma = 6$, $K_1 = K_2 = 8I$ with $I$ being the $10 \times 10$ identity matrix, $\nu_0 = 6$, $\nu_1 = 5$, and $\nu_2 = 1$. In the performance index $J(t)$, $Q$ is set to $Q = I$. The initial value of each element of auxiliary parameter vector $\hat{\boldsymbol{\omega}}_i(t)$ and $\hat{\mathbf{b}}_i(t)$ with $i = 1, 2, \cdots, n$ is randomly generated at interval $(0, 5)$. For comparison, the adaptive protocol (4) in [174] is also simulated, of which the parameters defined in [174] are set to $\gamma_2 = 11$, $\gamma_3 = 12$ and $\gamma_4 = 0.1$, $\Gamma_i = K_i = 1$, $c_{i0} = -0.46\gamma_3 x_i(0)/\gamma_4$

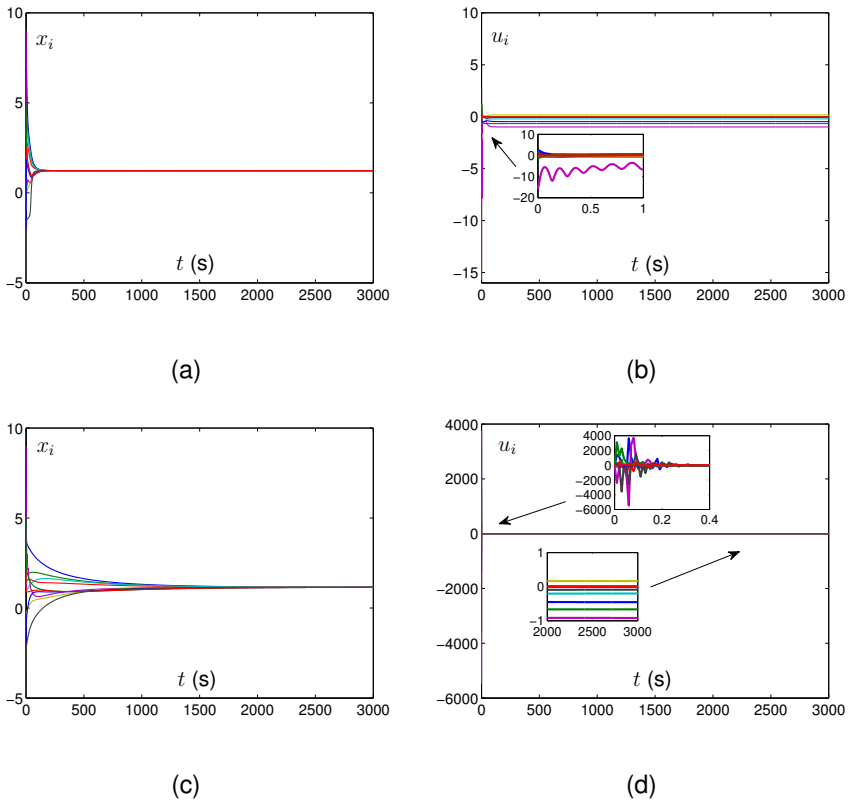



Figure 5.3: Profiles of agent states $x_i$ and inputs $u_i$ of multi-agent system (5.15) with unknown parameters. (a) Profiles of $x_i$ when adaptive near-optimal protocol (5.13) is used. (b) Profiles of $u_i$ when (5.13) is used. (c) Profiles of $x_i$ when protocol (4) in [174] is used. (d) Profiles of $u_i$ when protocol (4) in [174] is used.

according to the chapter. The simulation results are shown in Fig. 5.3 and 5.4. As seen from Fig. 5.3, while both protocols drive the agent states to reach agreement on the same value, when the proposed protocol (5.13) is used, the magnitudes of inputs are much smaller at the first few seconds. Moreover, as seen from Fig. 5.4, the values of performance index $J(t)$ are much smaller when (5.13) is used, showing the optimality of the performance index. These substantiate the superiority of the proposed adaptive near-optimal consensus approach.

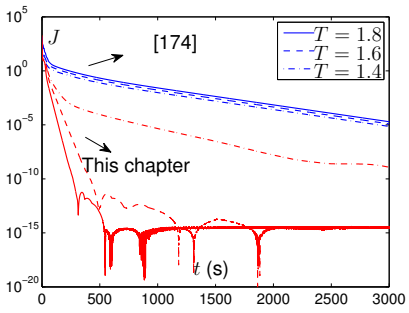



Figure 5.4: Profiles of performance index $J(t)$ defined in (5.2) with respect to different values of $T$, where the blue lines correspond to the cases when protocol (4) in [174] is used and the red ones corresponds to the cases when adaptive near-optimal protocol (5.13) in this chapter is used.

## 5.5 Chapter summary

In this chapter, the learning and near-optimal distributed consensus of high-order nonlinear heterogeneous multi-agent systems has been investigated. Under the condition that the system dynamics of all the agents are fully known, a nominal near-optimal protocol has been firstly designed and proposed. Then, based on sliding-mode auxiliary systems, an adaptive near-optimal protocol has been proposed for high-order nonlinear multi-agent systems with fully-unknown parameters. Theoretical analysis has shown that the proposed protocols can simultaneously guarantee the asymptotic optimality of the performance index with multi-agent systems exponentially converging to consensus. An illustrative example has substantiated the efficacy and superiority of the proposed adaptive near-optimal distributed consensus approach.

# Chapter 6

# LNOC for intelligent redundancy resolution with learning of parameters

In this chapter, inspired by the success of the LNOC method, we consider a special physical system, i.e., redundant manipulators.[1] Redundancy resolution is of great importance in the control of manipulators. Among the existing results for handling this issue, the quadratic program approaches, which are capable of optimizing performance indices subject to physical constraints, are widely used. However, the existing quadratic program approaches require exactly knowing all the physical parameters of manipulators, the condition of which may not hold in some practical applications. This fact motivates us to consider the application of learning and control techniques for simultaneous parameter learning and control. However, the inherent nonlinearity and non-smoothness of the neural model prohibits direct applications of learning and control to this model and there has been no existing result on LNOC of robotic arms using projection neural network (PNN) approaches with parameter convergence. Different from conventional treatments in joint angle space, we investigate the problem from the

[1] The content in this chapter has already been published. Yinyan Zhang, Siyuan Chen, Shuai Li, and Zhijun Zhang, "Adaptive Projection Neural Network for Kinematic Control of Redundant Manipulators With Unknown Physical Parameters," *IEEE Trans. Ind. Electron.*, vol. 65, no. 6, pp. 4909–4920, 2018.

joint speed space and decouple the nonlinear part of the Jacobian matrix from the structural parameters that need to be learnt. Based on the new representation, we establish the first adaptive PNN with online learning for the redundancy resolution of manipulators with unknown physical parameters, which tackles the dilemmas in existing methods. The proposed method is capable of simultaneously optimizing performance indices subject to physical constraints and handling parameter uncertainty. Theoretical results are presented to guarantee the performance of the proposed neural network. Besides, simulations based on a PUMA 560 manipulator with unknown physical parameters together with the comparison with an existing PNN substantiate the efficacy and superiority of the proposed neural network, and verify the theoretical results. The performance is also validated on a physical manipulator.

## 6.1 Introduction

While extensive progress (e.g., [201–203] and the references therein) has been made to address the kinematic uncertainty of non-redundant manipulators, the control of redundant manipulators with unknown kinematics has rarely been considered. It is generally assumed that the kinematics and the associated Jacobian matrix are accurately known in both pseudoinverse-type methods and QP based methods [188–200]. In the presence of uncertainty, these methods may result in errors or even unstable responses in the motion of end effectors. In [204], Cheah *et. al.* adopted a unified framework to address the kinematic uncertainty of both redundant and non-redundant manipulators, by which the end effectors can complete primary tasks in a satisfactory manner. However, Cheah *et. al.*'s work did not consider secondary tasks, which is far from the motivation to introduce redundancy into manipulators. Besides, due to the complexity of inverse kinematics of redundant manipulators [205], which is a one-to-many relationship [206], it is difficult to directly handle kinematic uncertainty via utilizing existing results in adaptive control, e.g., [207–210].

Motivated by the above observations, in this chapter, we propose an adaptive method to solve the redundancy resolution problem in the presence of kinematic uncertainty. The problem is formulated as a QP subject to equality and bounded constraints. Different from the existing results [188–200], in this chapter, some parameters associated with the Jacobian matrix are assumed to be unknown, i.e., the Jacobian matrix is uncertain. The proposed method is capable of simultaneously identifying the Jacobian matrix and handling both parameter uncertainty and physical constraints, while solving the redundancy resolution problem. Note that, although there are versatile methodologies available in the literature for achieving either parameter identification or redundancy resolution, to the best of our knowledge, there is no existing methods for online simultaneous parameter learning and redundancy resolution, except our work presented in this chapter. In addition, the proposed method does not require to perform pseudoinversion.

## 6.2 Preliminary and problem formulation

In this section, the forward kinematics of redundant manipulators is presented. Besides, the problem considered in this chapter is formulated as a QP with an uncertain Jacobian matrix.

### 6.2.1 Forward kinematics

The forward kinematics of redundant manipulators is the theoretical basis for redundancy resolution. Consider an $n$-DOF redundant manipulator with its end-effector working on an $m$-dimensional Cartesian space. By the definition of redundancy, $n > m$. The forward kinematics of the manipulator is analytically described as the following nonlinear function:

$$\mathbf{r}(t) = f(\theta(t)), \tag{6.1}$$

where $\mathbf{r}(t) = [r_1(t), r_2(t), \cdots, r_m]^\mathsf{T} \in \mathbb{R}^m$ and $\theta(t) = [\theta_1(t), \theta_2(t), \cdots, \theta_n(t)]^\mathsf{T} \in \mathbb{R}^n$ denote the Cartesian coordinate of the end-effector on the workspace and the joint angle vector on the joint space, respectively, at time instant $t$; subscript $^\mathsf{T}$ denotes the transpose of a vector or a matrix; nonlinear function $f(\cdot) : \mathbb{R}^n \rightarrow \mathbb{R}^m$ is determined by the mechanical and geometrical properties of a redundant manipulator. In manipulator modeling, $f(\cdot)$ is often derived via the Denavit-Hartenberg (D-H) convention [214].

*Property 1*: The Jacobian matrix $J(\theta(t)) = \partial f(\theta(t))/\partial\theta(t) \in \mathbb{R}^{m\times n}$ of the forward kinematics of a redundant manipulator satisfies the following equation:

$$J(\theta(t)) = W\phi(\theta(t)), \tag{6.2}$$

where $\phi(\theta(t)) \in \mathbb{R}^{k\times n}$ is referred to as the kinematic regressor matrix and $W \in \mathbb{R}^{m\times k}$ is a constant parameter matrix. Besides, each non-null element of $W$ is either a link length or a joint offset of the manipulator.

By calculating time derivatives on both side of (6.1), one has $\dot{\mathbf{r}}(t) = J(\theta(t))\dot{\theta}(t)$. Then, together with (6.2), the forward kinematics of a redundant manipulator at the velocity level is derived as follows:

$$\dot{\mathbf{r}}(t) = W\phi(\theta(t))\dot{\theta}(t). \tag{6.3}$$

Note that, for a redundant manipulator with known twist angles, the analytical expression of $\phi(\cdot)$ can be readily calculated via the D-H convention [214].

In this chapter, we consider the situation that $W$ is unknown, which leads to the uncertainty of the Jacobian matrix. The uncertainty makes the redundancy resolution problem considered in this chapter more difficult than those in [188–200].

### 6.2.2 QP-type problem formulation

QP is the widely adopted redundancy resolution problem formulation since it can deal with various constraints in a unified framework. Let $\mathbf{u} = \dot{\theta}(t)$. At the velocity level, the

general redundancy resolution problem with both equality and bounded constraints is formulated as follows:

$$\min_{\mathbf{u}(t)} \quad \frac{1}{2}\mathbf{u}^{\mathsf{T}}(t)A\mathbf{u}(t) + \mathbf{b}^{\mathsf{T}}\mathbf{u}(t), \tag{6.4a}$$

$$s.t. \quad J(\theta(t))\mathbf{u}(t) = \dot{\mathbf{r}}_{\mathsf{d}}(t) + \zeta(\mathbf{r}_{\mathsf{d}}(t) - f(\theta(t))), \tag{6.4b}$$

$$J(\theta(t)) = W\phi(\theta(t)), \tag{6.4c}$$

$$\mathbf{u}(t) \in \Omega, \tag{6.4d}$$

where positive definite diagonal matrix $A \in \mathbb{R}^{n\times n}$ and vector $\mathbf{b} \in \mathbb{R}^n$ are coefficients of the performance index; $J(\theta(t))$ is the Jacobian matrix; $\mathbf{r}_{\mathsf{d}}(t) \in \mathbb{R}^m$ denotes a smooth desired path of the end-effector; $\zeta > 0 \in \mathbb{R}$ is an adjustable parameter; $\eta^- \in \mathbb{R}^n$ and $\eta^+ \in \mathbb{R}^n$ are the lower bound and upper bound of the allowed velocities in the joint space of the manipulator; $\Omega = \{\mathbf{u} \in \mathbb{R}^n | \eta^- \leq \mathbf{u} \leq \eta^+\}$ is a convex set.

About the performance index shown in (6.4a), we offer the following remark.

*Remark 1:* The performance index shown in (6.4a) includes some widely investigated ones as special cases. For example, the velocity-norm performance index $\|\dot{\theta}(t)\|_2^2/2$ (with $\|\cdot\|_2$ denoting the 2-norm) adopted in [184,186,198,211,213] corresponds to the case that $A$ is an $n \times n$ identity matrix and $\mathbf{b} = 0$. The repetitive-motion performance index $\|\dot{\theta}(t)+\gamma(\theta(t)-\theta(0))\|_2^2/2$ in [188,197,200] aiming at handling the joint-angle drift phenomenon corresponds to the case that $A$ is an $n\times n$ identity matrix and $\mathbf{b} = \gamma(\theta(t)-\theta(0)$, where $\gamma \in \mathbb{R}$ is a constant parameter. By referring to [188, 197, 200], the only requirement on constant parameter $\gamma$ is that $\gamma > 0$. Note that, since the redundancy resolution problem is resolved at the velocity level, i.e., the decision variable is $\dot{\theta}$, the term $\gamma^2(\theta(t) - \theta(0))^{\mathsf{T}}(\theta(t) - \theta(0))/2$ is directly removed from the performance index, which does not affect the optimality of the solution [188, 197, 200]. When $A$ equals to the inertial matrix of the manipulator and $\mathbf{b} = 0$, performance index (6.4a) serves as the kinematic-energy performance index investigated in [198].

About the derivation of the equality constraint shown in (6.4b), we offer the following remark.

*Remark 2:* In the angle level, the equality constraint is

$$f(\theta(t)) = \mathbf{r}_{\mathsf{d}}(t), \tag{6.5}$$

when the end-effector is expected to track a smooth desired path defined by $\mathbf{r}_{\mathsf{d}}(t)$. There are two approaches in the existing literature to deriving the relationship between the desired path $\mathbf{r}_{\mathsf{d}}(t)$ and the joint angle vector $\theta(t)$ at the velocity level. The first one directly computes time derivatives on both sides of (6.5), which yields [184]

$$\dot{\mathbf{r}}_{\mathsf{d}}(t) = J(\theta(t))\dot{\theta}(t). \tag{6.6}$$

Velocity-level redundancy resolution methods based on (6.6) generally require that the initial Cartesian coordinate of the end-effector is the same as the desired one to guarantee asymptotic convergence of end-effector error $\mathbf{e}(t) = \mathbf{r}(t) - \mathbf{r}_{\mathsf{d}}(t)$ to zero and is thus less favorable [215]. The other approach removes this requirement by utilizing the formula $\dot{\beta}(t) = -\zeta\beta(t)$ with $\zeta > 0 \in \mathbb{R}$ being a parameter to scale the convergence rate, which guarantees that $\beta(t)$ asymptotically converges to zero [185]. Let $\beta(t) = \mathbf{r}_{\mathsf{d}}(t) - f(\theta(t))$. Then, by using the formula, the equality constraint (6.4b) is derived.

About the set constraint shown in (6.4d), we offer the following remark.

*Remark 3*: The joint angle limit can be converted into the joint velocity limit. Suppose that the physical joint angle limit is $\theta^- \leq \theta \leq \theta^+$, where $\theta^-$ and $\theta^+$ denote the physical lower bound and upper bound of the joint angle of the manipulator, respectively. Suppose that the physical joint velocity limit is $\dot{\theta}^- \leq \dot{\theta} \leq \dot{\theta}^+$, where $\dot{\theta}^-$ and $\dot{\theta}^+$ denote the physical lower bound and upper bound of the joint angle of the manipulator, respectively. Then, according to [198], the joint angle limit can be incorporated into the set constraint by setting $\eta^- = \max\{\dot{\theta}_{\mathsf{min}}, -k(\theta - \theta_{\mathsf{min}})\}$ and $\eta^+ = \min\{\dot{\theta}_{\mathsf{max}}, -k(\theta - \theta_{\mathsf{max}})\}$ with $k > 0 \in \mathbb{R}$ being a design parameter to scale the strength of negative feedback to comply with the joint angle limit.

## 6.3 Nominal design

For better readability and to lay a basis for latter discussion, in this section, we present the existing nominal design process for solving problem (6.4) under an ideal condition, i.e., $W$ is known (or $J(\theta)$ is known). Under this condition, some special cases of the redundancy resolution problem have been extensively investigated in the existing literature (e.g., [188–198]) with the aid of the Karush-Kuhn-Tucker condition [226].

The nominal design process can be divided into three steps. First, a Lagrange function is defined: $L(\theta,\lambda) = \mathbf{u}^{\mathsf{T}}A\mathbf{u}/2+\mathbf{b}^{\mathsf{T}}\mathbf{u}+\lambda^{\mathsf{T}}(\dot{\mathbf{r}}_{\mathsf{d}}-W\phi(\theta)\mathbf{u}+\zeta(\mathbf{r}_{\mathsf{d}}-f(\theta)))$ where $\lambda \in \mathbb{R}^m$ is called the Lagrange multiplier. Second, the Karush-Kuhn-Tucker condition [226] about the optimal solution of problem (6.4) is written as follows:

$$\begin{aligned} \mathbf{u} &= \mathsf{P}_{\Omega}\left(\mathbf{u}-\frac{\partial L}{\partial \mathbf{u}}\right), \\ W\phi(\theta)\mathbf{u} &= \dot{\mathbf{r}}_{\mathsf{d}}+\zeta(\mathbf{r}_{\mathsf{d}}-f(\theta)), \end{aligned} \tag{6.7}$$

where $\mathsf{P}_{\Omega}(\cdot)$ is a projection function defined as $\mathsf{P}_{\Omega}(\mathbf{u}) = \arg\min_{\mathbf{y}\in\Omega}\|\mathbf{u}-\mathbf{y}\|^2$. Note that $\partial L/\partial\theta = A\mathbf{u}+\mathbf{b}-\phi^{\mathsf{T}}(\theta)W^{\mathsf{T}}\lambda$. Then, a projection neural network (PNN) can be designed as follows to solve (6.4) [227]:

$$\begin{aligned} \varepsilon\dot{\mathbf{u}} &= -\mathbf{u}+\mathsf{P}_{\Omega}\left(\mathbf{u}-\frac{\partial L}{\partial \mathbf{u}}\right) \\ &= -\mathbf{u}+\mathsf{P}_{\Omega}(\mathbf{u}-(A\mathbf{u}+\mathbf{b}-\phi^{\mathsf{T}}(\theta)W^{\mathsf{T}}\lambda)), \\ \varepsilon\dot{\lambda} &= \dot{\mathbf{r}}_{\mathsf{d}}-W\phi(\theta)\mathbf{u}+\zeta(\mathbf{r}_{\mathsf{d}}-\mathbf{r}), \end{aligned} \tag{6.8}$$

where $\varepsilon > 0 \in \mathbb{R}$ is a positive constant design parameter to scale the convergence rate of the PNN. It has been rigorously proved in [227] that the state trajectory of the PNN is exponentially convergent to the optimal solution of (6.4).

About extending PNN (6.8) to solve the problem considered in this chapter, we offer the following remark.

*Remark 4:* The design of PNN (6.8) is based on the ideal condition that the Jacobian matrix $J(\theta) = W\phi(\theta)$ is known. Extending PNN (6.8) to solve problem (6.4) under

the situation that $W$ is unknown is not straightforward, especially when real-time redundancy resolution is required. The difficulty also lies in the corresponding theoretical analysis. Generally, projection neural networks become projected gradient descent in discrete time space. The conversion can be achieved by performing the Euler difference rule for the derivative term. In our work, we consider the case that some parameters are unknown, for which conventional projection neural networks or projected gradient descent cannot address.

## 6.4 Novel adaptive design

In this section, an adaptive PNN is developed to solve the redundancy resolution problem shown in (6.4) with parameter uncertainty.

### 6.4.1 Adaptive projection neural network

Define $\hat{W} \in \mathbb{R}^{m\times k}$. It is expected that $\hat{W}\phi(\theta) - J(\theta)$ converges to zero. The proposed adaptive PNN for solving redundancy resolution problem (6.4) with unknown constant matrix $W$ is described as follows:

$$\varepsilon\dot{\check{\mathbf{u}}} = -\check{\mathbf{u}} + \mathsf{P}_{\Omega}(\check{\mathbf{u}} - (A\check{\mathbf{u}} + \mathbf{b} - \phi^{\mathsf{T}}(\theta)\hat{W}^{\mathsf{T}}\lambda)), \tag{6.9a}$$

$$\varepsilon\dot{\lambda} = \dot{\mathbf{r}}_{\mathsf{d}} - \hat{W}\phi(\theta)\check{\mathbf{u}} + \zeta(\mathbf{r}_{\mathsf{d}} - \mathbf{r}), \tag{6.9b}$$

$$\dot{\hat{W}} = -\nu(\hat{W}\phi(\theta)\mathbf{u} - \dot{\mathbf{r}})\mathbf{u}^{\mathsf{T}}\phi^{\mathsf{T}}(\theta), \tag{6.9c}$$

$$\mathbf{u} = \check{\mathbf{u}} + \rho, \tag{6.9d}$$

where $\check{\mathbf{u}} \in \mathbb{R}^n$, $\hat{W} \in \mathbb{R}^{m\times k}$, and $\lambda \in \mathbb{R}^m$ are state variables of the network; parameter $\nu > 0 \in \mathbb{R}$ is used to scale the strength of parameter-error feedback; $\rho \in \mathbb{R}^n$ is bounded independent and identically distributed (i.i.d.) random noise of zero mean and $\sigma$ deviation, where $\|\rho\|_2 \leq \rho_0$ with $\rho_0 > 0 \in \mathbb{R}$ denoting the bound.

As seen from (6.9), to address the uncertainty of constant matrix $W$, an evolution rule (6.9c) is designed. From (6.3), $\dot{\mathbf{r}} = W\phi(\theta)\dot{\theta} = W\phi(\theta)\mathbf{u}$. It follows that (6.9c) can be rewritten as

$$\begin{aligned}\dot{\hat{W}} =& -\nu(\hat{W}\phi(\theta)\mathbf{u} - W\phi(\theta)\mathbf{u})\mathbf{u}^{\mathsf{T}}\phi^{\mathsf{T}}(\theta) \\ =& -\nu(\hat{W} - W)\phi(\theta)\mathbf{u}\mathbf{u}^{\mathsf{T}}\phi^{\mathsf{T}}(\theta).\end{aligned} \tag{6.10}$$

Evidently, by this rule, the change in the state value of $\hat{W}$ results from the term $\hat{W}\phi(\theta) - W\phi(\theta) = \hat{W}\phi(\theta) - J(\theta)$, i.e., the difference between the estimated value and the true value. When $\hat{W}\phi(\theta)$ converges to $W\phi(\theta)$, i.e., $\hat{W}\phi(\theta) - J(\theta) = 0$, it is evident that $\dot{\hat{W}} = 0$, which means that the dynamical system described by (6.10), i.e., (6.9c), achieves its equilibrium. Besides, in the proposed PNN, the output $\mathbf{u}$ is the sum of the state vector $\breve{\mathbf{u}}$ and additive random noise $\rho$. The reason why noise is introduced into the output of the proposed PNN is to excite the dynamic properties of manipulators.

About how to obtain the values of $\mathbf{r}$ and $\dot{\mathbf{r}}$ in practice, we offer the following remark.

*Remark 5:* The uncertainty in the parameters makes it difficult to acquire the values of $\mathbf{r}$ (i.e., end-effector position) and $\dot{\mathbf{r}}$ (i.e., end-effector velocity) via using the forward kinematics of the manipulator. On the other hand, it is worth pointing out that there are high-accuracy motion capture system available for the measurement of end-effector position $\mathbf{r}$. For example, the Optitrack motion capture system, which consistently produces a positional error less than $3 \times 10^{-4}$ m [216], is becoming more and more popular in the robotics and control community as a reliable tool for position measurement [217–219]. The OptiTrack have been widely used in robotics [220–222]. In the problem considered in this chapter, the output is the position (and, consequently, the speed) of the end-effector of the manipulator. The parameter identification of a dynamical system intrinsically requires knowing the input and output data of the system. Thus, the requirement on knowing the end-effector position is natural. Note that, when conducting parameter identification for manipulators, at least, the end-effector position difference data and input difference data are required [223]. In terms of the end-effector velocity $\dot{\mathbf{r}}$, it can be obtained from the position measurement via using tracking differentia-

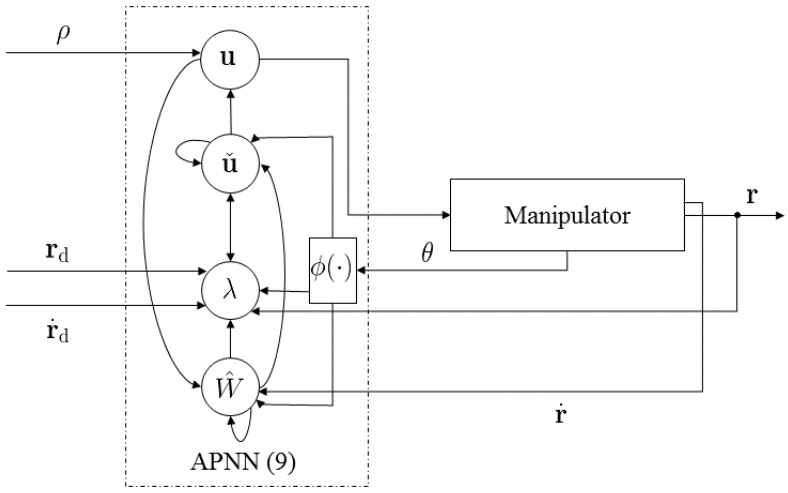



Figure 6.1: The block diagram about the implementation of the proposed adaptive projection neural network (APNN) described in (6.9) for real-time redundancy resolution of manipulators with an uncertain constant matrix $W$ in kinematics.

tors [224, 225]. Since the end-effector acceleration is always bounded, according to Theorem 1 in [225], we can readily configure a real-time first-order tracking differentiator whose output converges to the time-derivative of $\mathbf{r}$, i.e., $\dot{\mathbf{r}}$, in finite time. In other words, mathematically, we have $\mathbf{y}(t) = \dot{\mathbf{r}}(t)$, $\forall t > t_e$, where $\mathbf{y}(t)$ is the output of the tracking differentiator and $t_e > 0$ is a constant.

In terms of the implementation of the proposed adaptive PNN, we offer the following remark.

*Remark 6*: Fig. 6.1 shows a block diagram about the implementation of the proposed adaptive PNN in the real-time redundancy resolution of manipulators. As seen from this block diagram, the joint angle and end-effector information (i.e., $\theta$, $\mathbf{r}$ and $\dot{\mathbf{r}}$) of the redundant manipulator is used as a feedback to the adaptive PNN. Based on the feedback information, the desired path described by $\mathbf{r}_{\mathrm{d}}$ and $\dot{\mathbf{r}}_{\mathrm{d}}$, and artificially added additive noise $\rho$, the network state variables and output are adaptively updated, among which $\mathbf{u}$ is used to control the redundant manipulator.

Regarding the significance and motivation of the proposed method, we provide the following remark.

*Remark 7*: Although there are well established methods for off-line calibration of kinematic parameters of manipulators, there are some cases that require online simultaneous identification and control. In fact, the D-H parameters of a manipulator are affected by the length of the tool it uses. For example, consider the D-H parameters of the PUMA 560 manipulator, which are shown in Table 6.1 of the chapter. When the manipulator is holding a welding rod in a cooperative welding task, the end-effector of the manipulator is the tip of the rod. The length of the rod keeps decreasing during the welding process, which results in the decrease of $d_6$. However, during the welding process, it is unfavorable to conduct off-line calibration of the kinematic parameters once $d_6$ changes. Otherwise, a simple welding process will cost much time due to the time-consuming off-line calibration. In addition, as pointed out by Dixon *et al.*, when a robot picks up tools of uncertain lengths, the overall kinematics becomes uncertain and changes according to different tasks [236–238]. Thus, for the case that a manipulator needs to sequentially conduct different tasks by using different tools, it is still unfavorable to conduct off-line calibration. To sum up, our method serves as an alternative for the case that the kinematic parameters may change during the task execution process.

### 6.4.2 Theoretical analysis

In this subsection, theoretical results about the proposed adaptive PNN are presented.

*Theorem 1*: The state variable $\hat{W}$ of the adaptive projection neural network (6.9) satisfies $\hat{W}\phi(\theta) = J(\theta)$ when $t \to +\infty$ and the state variable $\mathbf{u}$ of (6.9) converges to the optimal solution to the resolution problem shown in (6.4) with an error bounded by $\rho_0$.

*Proof:* Let $\tilde{W}(t) = \hat{W}(t) - W$. Consider the following function: $V_1 = \frac{1}{2}\|\tilde{W}\|_{\mathsf{F}}^2 = \frac{1}{2}\text{trace}(\tilde{W}^{\mathsf{T}}\tilde{W})$, where $\|\cdot\|_{\mathsf{F}}$ and $\text{trace}(\cdot)$ denote the Frobenius norm and the trace of a matrix, respectively. Evidently, $V_1 \geq 0$. In light of (6.2) and (6.9), calculating the time

derivative of $V_1$ gives

$$\begin{aligned}\dot{V}_1 &= \text{trace}(\tilde{W}^{\mathsf{T}}\dot{\tilde{W}}) = -\nu\text{trace}(\tilde{W}^{\mathsf{T}}(\hat{W}\phi(\theta)\mathbf{u} - \dot{\mathbf{r}})\mathbf{u}^{\mathsf{T}}\phi^{\mathsf{T}}(\theta)) \\ &= -\nu\text{trace}(\tilde{W}^{\mathsf{T}}(\hat{W}\phi(\theta)\mathbf{u} - W\phi(\theta)\mathbf{u})\mathbf{u}^{\mathsf{T}}\phi^{\mathsf{T}}(\theta)) = -\nu\text{trace}(\tilde{W}^{\mathsf{T}}\tilde{W}\phi(\theta)\mathbf{u}\mathbf{u}^{\mathsf{T}}\phi^{\mathsf{T}}(\theta)).\end{aligned} \tag{6.11}$$

Note that trace$(XY) =$ trac$(X^{\mathsf{T}}Y^{\mathsf{T}})$ with $X$ and $Y$ being two compatible matrices [228]. Let $X = \tilde{W}^{\mathsf{T}}$ and $Y = \tilde{W}\phi(\theta)\mathbf{u}\mathbf{u}^{\mathsf{T}}\phi^{\mathsf{T}}(\theta)$. Then, it follows from (6.11) and (6.9) that

$$\begin{aligned}\dot{V}_1 &= -\nu\text{trace}(\tilde{W}\phi(\theta)\mathbf{u}\mathbf{u}^{\mathsf{T}}\phi^{\mathsf{T}}(\theta)\tilde{W}^{\mathsf{T}}) = -\nu\text{trace}((\tilde{W}\phi(\theta)\mathbf{u})(\tilde{W}\phi(\theta)\mathbf{u})^{\mathsf{T}}) \\ &= -\nu\|\tilde{W}\phi(\theta)\mathbf{u}\|_{\mathsf{F}}^2 = -\nu\|\tilde{W}\phi(\theta)(\check{\mathbf{u}} + \rho)\|_{\mathsf{F}}^2 \\ &\leq 0.\end{aligned}$$

Then, together with (6.11) and (6.9), the LaSalle's invariance principle [229] is employed, which gives

$$\text{trace}(\tilde{W}^{\mathsf{T}}\tilde{W}\phi(\theta)(\check{\mathbf{u}} + \rho)(\check{\mathbf{u}} + \rho)^{\mathsf{T}}\phi^{\mathsf{T}}(\theta)) = 0, \tag{6.12}$$

when $t \to +\infty$. Calculating expected values on both sides of (6.12) yields

$$E(\text{trace}(\tilde{W}^{\mathsf{T}}\tilde{W}\phi(\theta)(\check{\mathbf{u}} + \rho)(\check{\mathbf{u}} + \rho)^{\mathsf{T}}\phi^{\mathsf{T}}(\theta))) = 0, \tag{6.13}$$

when $t \to +\infty$. Recalling that $\rho$ is i.i.d. zero-mean random noise of deviation $\sigma$, one has

$$
\begin{aligned}
&E(\text{trace}(\tilde{W}^{\mathsf{T}}\tilde{W}\phi(\theta)(\check{\mathbf{u}}+\rho)(\check{\mathbf{u}}+\rho)^{\mathsf{T}}\phi^{\mathsf{T}}(\theta)))\\
&\quad= E(\text{trace}(\tilde{W}^{\mathsf{T}}\tilde{W}\phi(\theta)\check{\mathbf{u}}\check{\mathbf{u}}^{\mathsf{T}}\phi^{\mathsf{T}}(\theta))) + E(\text{trace}(\tilde{W}^{\mathsf{T}}\tilde{W}\phi(\theta)\check{\mathbf{u}}\rho^{\mathsf{T}}\phi^{\mathsf{T}}(\theta)))\\
&\quad\quad + E(\text{trace}(\tilde{W}^{\mathsf{T}}\tilde{W}\phi(\theta)\rho\check{\mathbf{u}}^{\mathsf{T}}\phi^{\mathsf{T}}(\theta))) + E(\text{trace}(\tilde{W}^{\mathsf{T}}\tilde{W}\phi(\theta)\rho\rho^{\mathsf{T}}\phi^{\mathsf{T}}(\theta)))\\
&\quad= E(\text{trace}(\tilde{W}\phi(\theta)\check{\mathbf{u}}\check{\mathbf{u}}^{\mathsf{T}}\phi^{\mathsf{T}}(\theta)\tilde{W}^{\mathsf{T}})) + E(\text{trace}(\tilde{W}\phi(\theta)\rho\check{\mathbf{u}}^{\mathsf{T}}\phi^{\mathsf{T}}(\theta)\tilde{W}^{\mathsf{T}}))\\
&\quad\quad + E(\text{trace}(\phi^{\mathsf{T}}(\theta)\tilde{W}^{\mathsf{T}}\tilde{W}\phi(\theta)\check{\mathbf{u}}\rho^{\mathsf{T}})) + E(\text{trace}(\phi^{\mathsf{T}}\tilde{W}^{\mathsf{T}}\tilde{W}\phi(\theta)\rho\rho^{\mathsf{T}}))\\
&\quad= E(\text{trace}(\tilde{W}\phi(\theta)\check{\mathbf{u}}\check{\mathbf{u}}^{\mathsf{T}}\phi^{\mathsf{T}}(\theta)\tilde{W}^{\mathsf{T}})) + E(\text{trace}(\phi^{\mathsf{T}}(\theta)\tilde{W}^{\mathsf{T}}\tilde{W}\phi(\theta)\check{\mathbf{u}}\rho^{\mathsf{T}}))\\
&\quad\quad + E(\text{trace}(\phi^{\mathsf{T}}(\theta)\tilde{W}^{\mathsf{T}}\tilde{W}\phi(\theta)\check{\mathbf{u}}\rho^{\mathsf{T}})) + E(\text{trace}(\phi^{\mathsf{T}}\tilde{W}^{\mathsf{T}}\tilde{W}\phi(\theta)\rho\rho^{\mathsf{T}}))\\
&\quad= \text{trace}(E(\tilde{W}\phi(\theta)\check{\mathbf{u}}\check{\mathbf{u}}^{\mathsf{T}}\phi^{\mathsf{T}}(\theta)\tilde{W}^{\mathsf{T}})) + \text{trace}(E(\phi^{\mathsf{T}}(\theta)\tilde{W}^{\mathsf{T}}\tilde{W}\phi(\theta)\check{\mathbf{u}})E^{\mathsf{T}}(\rho))\\
&\quad\quad + \text{trace}(E(\phi^{\mathsf{T}}(\theta)\tilde{W}^{\mathsf{T}}\tilde{W}\phi(\theta)\check{\mathbf{u}})E^{\mathsf{T}}(\rho)) + \text{trace}(E(\phi^{\mathsf{T}}\tilde{W}^{\mathsf{T}}\tilde{W}\phi(\theta))E(\rho\rho^{\mathsf{T}}))\\
&\quad= \text{trace}(E(\tilde{W}\phi(\theta)\check{\mathbf{u}}\check{\mathbf{u}}^{\mathsf{T}}\phi^{\mathsf{T}}(\theta)\tilde{W}^{\mathsf{T}})) + \sigma^2\text{trace}(E(\phi^{\mathsf{T}}(\theta)\tilde{W}^{\mathsf{T}}\tilde{W}\phi(\theta)))\\
&\quad= E(\|\tilde{W}\phi(\theta)\check{\mathbf{u}}\|_{\mathsf{F}}^2) + \sigma^2 E(\|\tilde{W}\phi(\theta)\|_{\mathsf{F}}^2),
\end{aligned}
$$

which indicates that $E(\|\tilde{W}\phi(\theta)\check{\mathbf{u}}\|_{\mathsf{F}}^2) + \rho^2 E(\|\tilde{W}\phi(\theta)\|_{\mathsf{F}}^2) = 0$ when $t \to +\infty$ with (6.10) taken into account. Since $E(\|\tilde{W}\phi(\theta)\check{\mathbf{u}}\|_{\mathsf{F}}^2) \geq 0$ and $E(\|\tilde{W}\phi(\theta)\|_{\mathsf{F}}^2) \geq 0$, it is further concluded that $E(\|\tilde{W}\phi(\theta)\|_{\mathsf{F}}^2) = 0$ when $t \to +\infty$. It follows that $\tilde{W}\phi(\theta) = 0$, i.e., $\hat{W}\phi(\theta) = J(\theta)$, when $t \to +\infty$. In other words, the dynamics of the adaptive PNN described in (6.9) asymptotically converges to a invariant set, in which $\hat{W}\phi(\theta) = J(\theta)$. According to LaSalle's invariance principle [229], the following analysis is conducted on the invariant set. Specifically, in the invariant set, based on (6.9), the dynamics of $\lambda$ and $\check{\mathbf{u}}$ becomes

$$
\begin{aligned}
\varepsilon\dot{\check{\mathbf{u}}} =& -\check{\mathbf{u}} + \mathsf{P}_{\Omega}(\check{\mathbf{u}} - (A\check{\mathbf{u}} + \mathbf{b} - J^{\mathsf{T}}(\theta)\lambda)),\\
\varepsilon\dot{\lambda} =& \dot{\mathbf{r}}_{\mathsf{d}} - J(\theta)\check{\mathbf{u}} + \zeta(\mathbf{r}_{\mathsf{d}} - \mathbf{r}),
\end{aligned}
$$

which is further rewritten as

$$
\varepsilon\mathbf{z} = -\mathbf{z} + \mathsf{P}_{\bar{\Omega}}(\mathbf{z} - F(\mathbf{z})) \tag{6.14}
$$

with $\mathbf{z} = [\check{\mathbf{u}}^\mathsf{T}, \lambda^\mathsf{T}]^\mathsf{T}$, $\bar{\Omega} = \{(\check{\mathbf{u}}, \lambda) | \check{\mathbf{u}} \in \Omega, \lambda \in \mathbb{R}^m\}$ and

$$F(\mathbf{z}) = \begin{bmatrix} A\check{\mathbf{u}} + \mathbf{b} - J^\mathsf{T}(\theta)\lambda \\ -\dot{\mathbf{r}}_\mathsf{d} + J(\theta)\check{\mathbf{u}} - \zeta(\mathbf{r}_\mathsf{d} - \mathbf{r}) \end{bmatrix}.$$

Consider a Lyapunov candidate function $V_2 = (\mathbf{z} - P_{\bar{\Omega}}(\mathbf{z}))^\mathsf{T}(\mathbf{z} - P_{\bar{\Omega}}(\mathbf{z}))/2 \geq 0$. Calculating its time derivative along the dynamics (6.14) gives $\dot{V}_2 = (\mathbf{z} - P_{\bar{\Omega}}(\mathbf{z}))\dot{\mathbf{z}} = -(\mathbf{z} - P_{\bar{\Omega}}(\mathbf{z}))(\mathbf{z} - P_{\bar{\Omega}}(\mathbf{z}))/\varepsilon \leq 0$, where the equality holds only when $\mathbf{z} \in \bar{\Omega}$. Therefore, by the Lyapunov theory [229], $\mathbf{z} - P_{\bar{\Omega}}$ asymptotically converges to zero, i.e., $\mathbf{z}$ asymptotically converges to be within set $\bar{\Omega}$.

From the analytical expression of $F(\mathbf{z})$,

$$\nabla F = \frac{\partial F(\mathbf{z})}{\partial \mathbf{z}} = \begin{bmatrix} A & -J^\mathsf{T}(\theta) \\ J(\theta) & 0 \end{bmatrix}.$$

Then,

$$\nabla F + \nabla^\mathsf{T} F = \begin{bmatrix} 2A & 0 \\ 0 & 0 \end{bmatrix},$$

which is positive semi-definite, since $A$ is a positive definite diagonal matrix. In addition, by the mean-value theorem, $\forall \mathbf{z}_1$ and $\mathbf{z}_2$, one has $F(\mathbf{z}_1) - F(\mathbf{y_2}) = \nabla F(\mathbf{z}_3)(\mathbf{z}_1 - \mathbf{z}_2)$, where $\mathbf{z}_3 = \kappa\mathbf{z}_1 + (1-\kappa)\mathbf{z}_2$ and $0 \leq \kappa \leq 1$. It follows that $(\mathbf{z}_1 - \mathbf{z}_2)^\mathsf{T}(F(\mathbf{z}_1) - F(\mathbf{z}_2)) = (\mathbf{z}_1 - \mathbf{z}_2)^\mathsf{T}\nabla F(\mathbf{z}_3)(\mathbf{z}_1 - \mathbf{z}_2) \geq 0$, indicating that $F(\cdot)$ is monotone. Based on Theorem 1 in [230], it is further concluded that dynamical system (6.14) is stable in the sense of Lyapunov and $\mathbf{z}$ globally converges to $\mathbf{z}^* = [\check{\mathbf{u}}^{*\mathsf{T}}, \lambda^{*\mathsf{T}}]^\mathsf{T}$, which satisfies the following inequality:

$$(\mathbf{z} - \mathbf{z}^*)^\mathsf{T} F(\mathbf{z}^*) \geq 0, \ \forall \mathbf{z} \in \bar{\Omega}.$$

In other words, $\forall \check{\mathbf{u}} \in \Omega$ and $\lambda \in \mathbb{R}^m$, the following inequality holds:

$$(\check{\mathbf{u}} - \check{\mathbf{u}}^*)^\mathsf{T}(A\check{\mathbf{u}}^* + \mathbf{b} - J^\mathsf{T}(\theta)\lambda^*) + (\lambda - \lambda^*)^\mathsf{T}(-\dot{\mathbf{r}}_\mathsf{d} + J(\theta)\check{\mathbf{u}}^* - \zeta(\mathbf{r}_\mathsf{d} - \mathbf{r})) = 0.$$

It follows that one can always find a value of $\lambda$ such that $(\lambda - \lambda^*)^\mathsf{T}(-\dot{\mathbf{r}}_\mathsf{d} + J(\theta)\check{\mathbf{u}}^* -$

$\zeta(\mathbf{r}_\mathsf{d} - \mathbf{r}))$ tends to infinity when $-\dot{\mathbf{r}}_\mathsf{d} + J(\theta)\check{\mathbf{u}}^* - \zeta(\mathbf{r}_\mathsf{d} - \mathbf{r}) \neq 0$. As a result,

$$\begin{cases} -\dot{\mathbf{r}}_\mathsf{d} + J(\theta)\check{\mathbf{u}}^* - \zeta(\mathbf{r}_\mathsf{d} - \mathbf{r}) = 0, \\ (\check{\mathbf{u}} - \check{\mathbf{u}}^*)^\mathsf{T}(A\check{\mathbf{u}}^* + \mathbf{b} - J^\mathsf{T}(\theta)\lambda^*) \geq 0, \ \forall \check{\mathbf{u}} \in \Omega. \end{cases} \tag{6.15}$$

Evidently, (6.15) satisfies the solution of the saddle point problem described as follows:

$$\min_{\check{\mathbf{u}} \in \Omega} \max_{\lambda} L(\check{\mathbf{u}} \in \Omega, \lambda) = \check{\mathbf{u}}^\mathsf{T} A\check{\mathbf{u}}/2 + \mathbf{b}^\mathsf{T}\check{\mathbf{u}} + \lambda^\mathsf{T}(\dot{\mathbf{r}}_\mathsf{d} - W\phi(\theta)\check{\mathbf{u}} + \zeta(\mathbf{r}_\mathsf{d} - f(\theta))),$$

of which the solution is identical to the following constrained optimization problem:

$$\begin{aligned} \min_{\check{\mathbf{u}}(t)} \quad & \frac{1}{2}\check{\mathbf{u}}^\mathsf{T}(t)A\check{\mathbf{u}}(t) + \mathbf{b}^\mathsf{T}\check{\mathbf{u}}(t), \\ s.t. \quad & J(\theta(t))\check{\mathbf{u}}(t) = \dot{\mathbf{r}}_\mathsf{d}(t) + \zeta(\mathbf{r}_\mathsf{d}(t) - f(\theta(t))), \\ & \check{\mathbf{u}}(t) \in \Omega. \end{aligned}$$

Thus, $\check{\mathbf{u}}^*$ is the optimal solution of (6.16). Since the actual output of the adaptive PNN (6.9) is $\mathbf{u} = \check{\mathbf{u}} + \rho$ with $\|\rho\|_2 \leq \rho_0$, it follows that $\mathbf{u}$ converges to the optimal solution with an error $\rho$ bounded by $\rho_0$. The proof is complete. □

Theorem 1 shows the identification capability of the proposed adaptive PNN and the convergence of the adaptive PNN to the optimal solution to the redundancy resolution problem shown in (6.4) with an unknown constant matrix $W$.

*Remark 8*: From the proof of Theorem 1, it can be observed that the noise plays a key role to show the convergence. It is well understood in the field of control that the persistent excitation (PE) condition [231] is generally required to guarantee parameter convergence. To some extent, the PE condition requires a signal to be sufficiently rich [232]. To rigorously verify whether a reference trajectory satisfies the PE condition, one can refer to its definition [233, 234]. However, the PE condition is difficult to be checked online [235]. Besides, although the PE condition may be guaranteed via using a sufficiently rich reference trajectory, in applications, the desired reference trajectory may not satisfy the PE condition. For example, in a regulation task, where the end-effector of a manipulator is required to be regulated to a static position, the PE condition

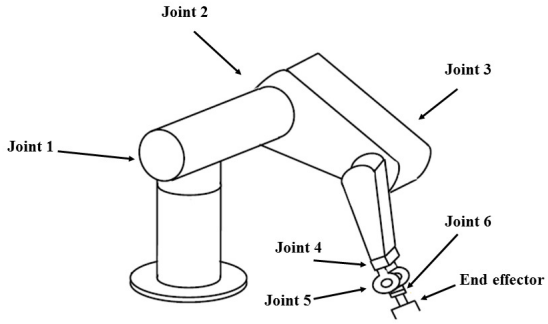



Figure 6.2: The schematic of the physical structure of the PUMA 560 manipulator.

Table 6.1: D-H Parameters of the PUMA 560 Manipulator

| Joint | $a_i$ (m) | $\alpha_i$ (rad) | $d_i$ (m) |
|---|---|---|---|
| 1 | 0 | $\pi/2$ | $d_1$ |
| 2 | $a_2$ | 0 | 0 |
| 3 | $a_3$ | $-\pi/2$ | $d_3$ |
| 4 | 0 | $\pi/2$ | 0 |
| 5 | 0 | $-\pi/2$ | 0 |
| 6 | 0 | 0 | $d_6$ |

is not satisfied. In this chapter, we deliberately introduce a noise, which guarantees the convergence of the parameters and the resultant method does not require online checking of the PE condition. From the proof of Theorem 1, the convergence analysis is not based on the PE condition. In terms of parameter convergence, it can be observed from (6.8) that $\nu$ is a gain parameter to adjust the convergence of $\hat{W}$. By using a larger value of $\nu$, the parameter convergence can be faster.

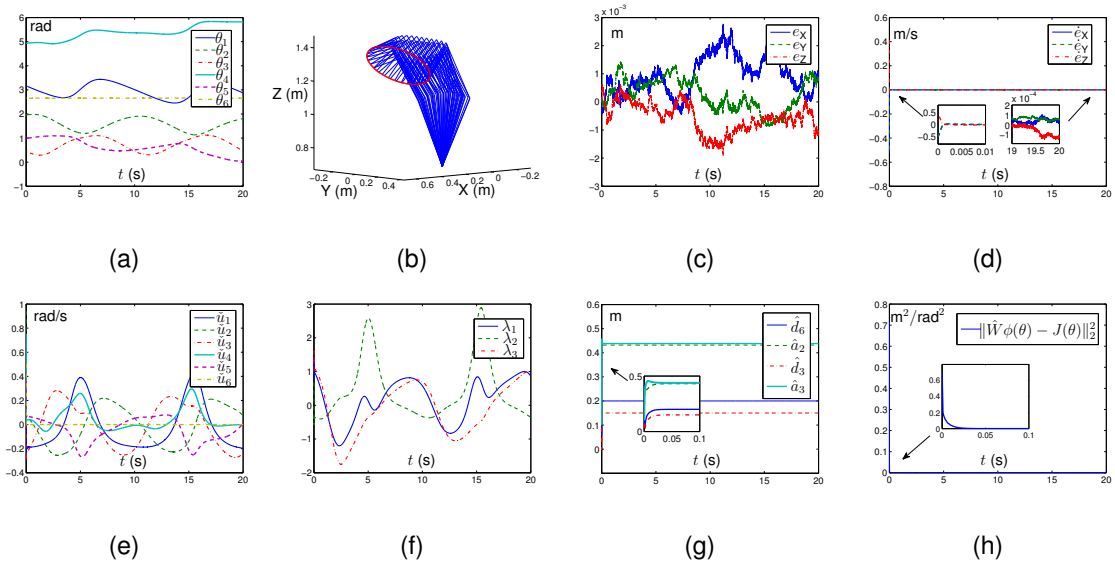



Figure 6.3: Simulation results about the minimum-velocity-norm redundancy resolution of the PUMA 560 manipulator with unknown parameters $a_2$, $a_3$, $d_3$, and $d_6$ via the proposed adaptive PNN (6.9). (a) Joint-angle profiles. (b) Motion trajectory of the manipulator, where the red line denotes the trajectory of the end-effector and the blue lines denote the configurations of the links during the process. (c) Position-error profiles. (d) Velocity-error profiles. (e) $\breve{\mathbf{u}}$ profiles. (f) $\lambda$ profiles. (g) $\hat{W}$ profiles. (h) Estimation profiles. Note that each non-zero element of $\hat{W}$ is identical to one of the four elements (i.e., $\hat{d}_6$, $\hat{d}_3$, $\hat{a}_2$ and $\hat{a}_3$), according to the Appendix.

## 6.5 Simulative verifications and comparisons

In this section, simulation results are presented and compared to verify the theoretical results, and substantiate the efficacy and superiority of the proposed adaptive PNN.

### 6.5.1 PUMA 560 description

In the simulations, a PUMA 560 manipulator, which is a 6-joint 6-DOF spatial manipulator, is used. Note that, when only the position of the end effector is considered for a given task, the PUMA 560 manipulator is redundant, according to the definition of redundancy. The schematic of the physical structure of the PUMA 560 manipulator is shown in Fig. 6.2. The D-H parameters of the PUMA 560 manipulator is shown in

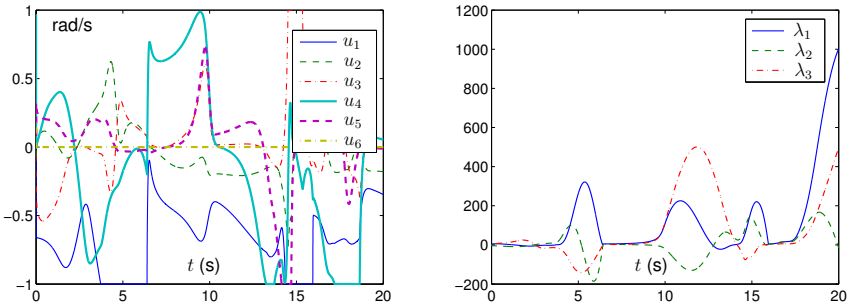
Table 6.1. In the simulations, we assumed that the values of physical parameters $d_1$, $d_3$, $d_6$, $a_3$ and $a_a$ are unknown. The exact values of them are $a_2 = 0.4318$, $a_3 = 0.438$, $d_1 = 0.67$, $d_3 = 0.1505$, and $d_6 = 0.2$. The analytical expression of the forward kinematics and Jacobian matrix of the PUMA 560 manipulator is presented in the Appendix of this chapter. As seen from the Appendix, $W$ is only related to physical parameters $d_3$, $d_6$, $a_2$ and $a_3$.

*Remark 9*: In practice, for each physical parameter of a manipulator, we have two values, i.e., the true value and the nominal value. The nominal value is the one given when designing a manipulator, which often has a error with respect to the true value. After we buy a manipulator, we generally only know the nominal values which are provided by the factory that produces it. As a result, we often use nominal values to conduct controller design or kinematics analysis. In this chapter, $\hat{W}(0)$ corresponds to the nominal values. Thus, we set $W = \hat{W}(0)$ for the nominal PNN to facilitate comparison. Similarly, in practice, when using the proposed adaptive PNN, as the nominal value is an estimation of the true value, we can use it to reduce the initial parameter error so as to enhance the performance of the proposed adaptive PNN. Note that in the simulations shown in this chapter, we also present the case with large parameter errors to show the efficacy of the proposed adaptive PNN.

### 6.5.2 Minimum-velocity-norm redundancy resolution

In this subsection, we present the simulation results when the proposed adaptive PNN (6.9) is employed to the minimum-velocity-norm redundancy resolution of the PUMA 560 manipulator. The performance is then compared with that of the existing PNN (6.8).

The minimum-velocity-norm redundancy resolution problem corresponds to the case that, in (6.4a), $A$ is an $3 \times 3$ identity matrix and $\mathbf{b} = 0$. The parameter in (6.4) is set to $\zeta = 0.1$. Besides, the joint velocity bounds are $\eta^+ = [1, 1, 1, 1, 1, 1]^\mathsf{T}$ rad/s and $\eta^- =$




(a) (b)

Figure 6.4: Simulation results on profiles of state variables of the nominal PNN (6.8) during the process of minimum-velocity-norm redundancy resolution of the PUMA 560 manipulator with unknown parameters $a_2$, $a_3$, $d_3$, and $d_6$. (a) $\mathbf{u}$ profiles. (b) $\lambda$ profiles.

$[-1,-1,-1,-1,-1,-1]^{\mathsf{T}}$ rad/s. In the simulations, the end effector of the manipulator is expected to track the path described as follows:

$$\mathbf{r}_{\mathsf{d}}(t) = \begin{bmatrix} \cos((\pi t)/5)/5 + 7/20 \\ -(\sqrt{3}\sin((\pi t)/5))/10 \\ \sin((\pi t)/5)/10 + 13/10 \end{bmatrix}, \tag{6.17}$$

which is cyclic with the period being 10 s.

In the simulation, the parameters of the adaptive PNN (6.9) are set as $\varepsilon = 0.0001$ and $\nu = 10000$. The deviation of noise $\rho$ is set as $\sigma = 0.001$. As seen from the Appendix, $W$ is determined by the values of $d_3$, $d_6$, $a_2$ and $a_3$. Correspondingly, in state variable $\hat{W}$ of the neural network, they are denoted by $\hat{d}_3$, $\hat{d}_6$, $\hat{a}_2$, and $\hat{a}_3$. Besides, in the initial state, $\hat{W}(0)$ is set by letting $\hat{d}_3(0) = \hat{d}_6(0) = \hat{a}_2(0) = \hat{a}_1(0) = 0.1$ m; $\theta(0) = [3.1649, 1.9548, 0.4584, 4.9330, 0.9870, 2.6527]^{\mathsf{T}}$ rad; each element of $\dot{\theta}(0)$ and $\lambda(0)$ is randomly generated at interval $(0,1)$. The smooth joint-angle profiles and end-effector profiles are presented in Fig. 6.3(a) and Fig. 6.3(b), respectively. Besides, as seen from Fig. 6.3(c) and Fig. 6.3(d), the end-effector of the PUMA 560 manipulator successfully tracks the desired path defined in (6.17) with the position errors being less than $3 \times 10^{-3}$ m (i.e., 3 mm) and the velocity errors being less than $2 \times 10^{-4}$ m/s (i.e., 0.4 mm/s). As seen from Fig. 6.3(e), the profiles of $\hat{\mathbf{u}}$ are smooth and remain in the

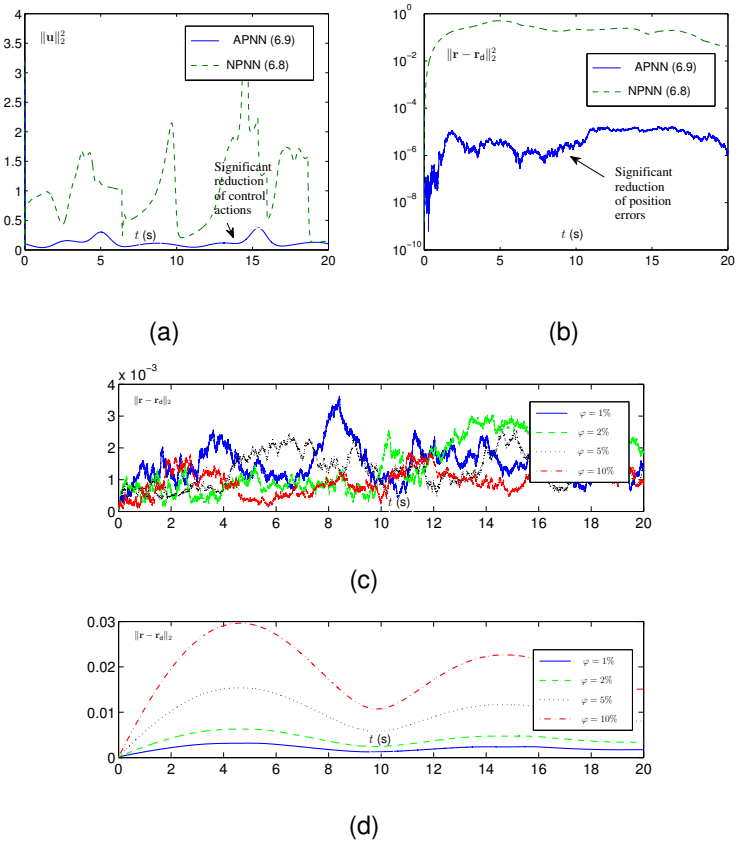



Figure 6.5: Comparison of performances during the process of minimum-velocity-norm redundancy resolution of the PUMA 560 manipulator with unknown parameters $a_2$, $a_3$, $d_3$, and $d_6$ via the proposed adaptive projection neural network (APNN) (6.9) and the nominal projection neural network (NPNN) (6.8). (a) Control actions. (b) Position errors. (c) Profiles of position error norm when APNN (6.9) is adopted. (d) Profiles of position error norm when NPNN (6.8) is adopted.

bounds $\eta^+$ and $\eta^-$. Fig. 6.3(f) shows the smooth evolution of the Lagrange multiplier $\lambda$. Bedsides, as seen from Fig. 6.3(g), the parameters $\hat{d}_3$, $\hat{d}_6$, $\hat{a}_2$ and $\hat{a}_3$ are convergent, which quickly converge to the exact physical parameter values $d_3$, $d_6$, $a_2$ and $a_3$ of the PUMA560 manipulator, respectively. This is also verified by Fig. 6.3(h), from which it is observed that $\|\hat{W}\phi(\theta) - J(\theta)\|_2^2$ quickly converges to zero, substantiating Theorem 1. The quick convergence is due to the parameter setting. In the simulation, we set $\nu = 10000$, which leads to a relative high gain for the feedback of $\dot{\hat{W}}$. It is worth pointing out that, to show the merit of parameter convergence, when $t = 5$ s during the

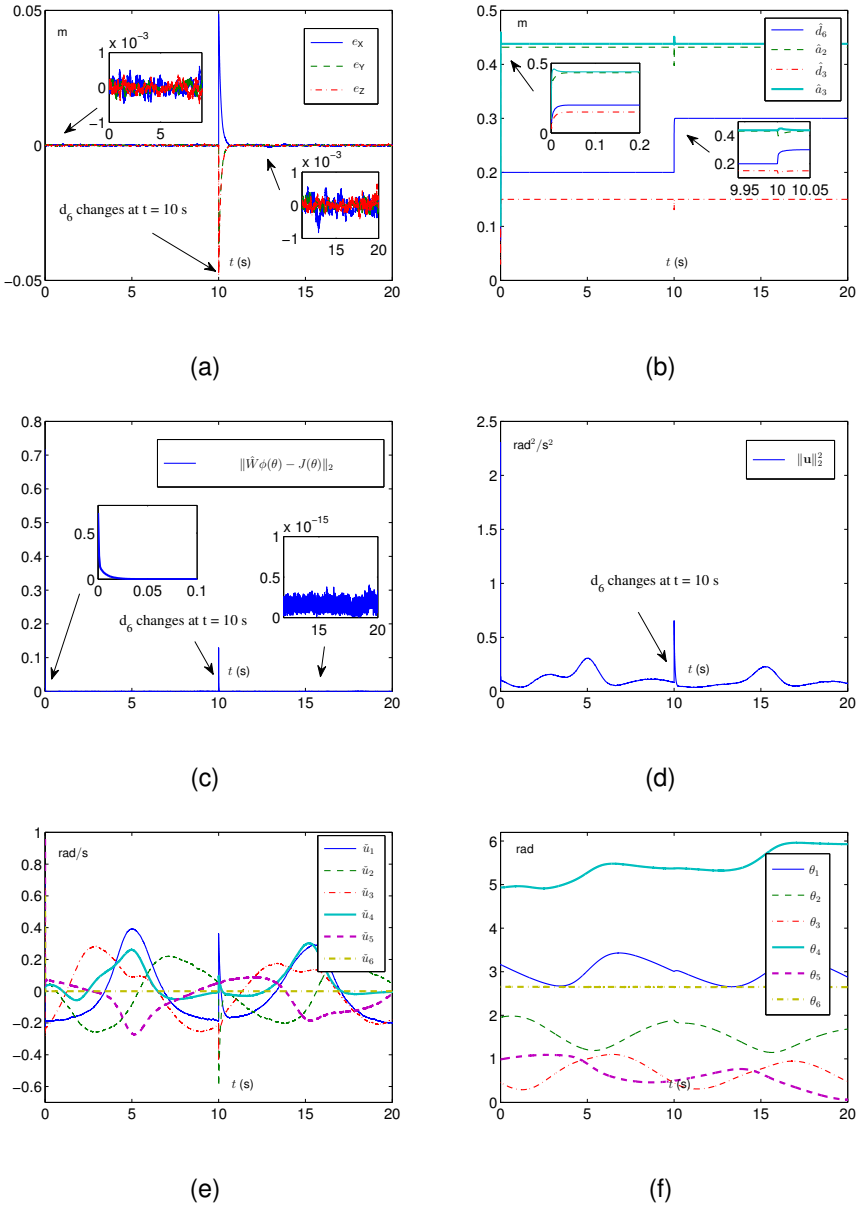



Figure 6.6: Simulation results about the minimum-velocity-norm redundancy resolution of the PUMA 560 manipulator with unknown parameters $a_2$, $a_3$, $d_3$, and $d_6$ via the proposed adaptive PNN (6.9) for the case that $d_6$ suddenly changes from 0.2 m to 0.3 m. (a) Position error profiles. (b) Parameter profiles. (c) Jacobian matrix error profiles. (d) Velocity norm profiles. (e) Joint velocity profiles. (f) Joint angle profiles.

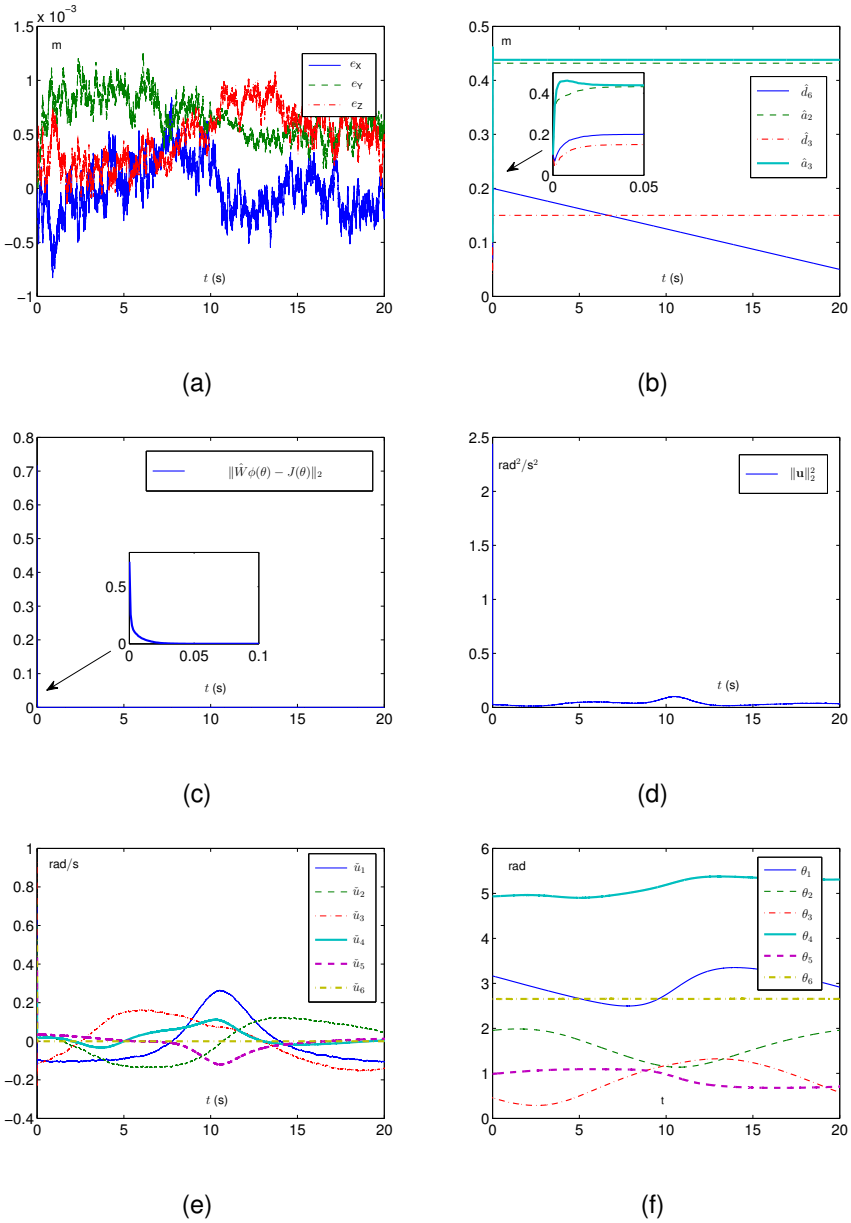
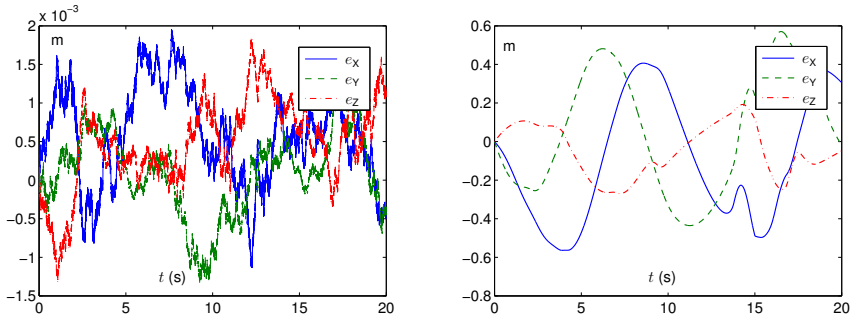



Figure 6.7: Simulation results about the minimum-velocity-norm redundancy resolution of the PUMA 560 manipulator with unknown parameters $a_2$, $a_3$, $d_3$, and $d_6$ via the proposed adaptive PNN (6.9) for the case that $d_6$ keeps on decreasing with $d_6(t) = 0.2 - 0.0075t$ m. (a) Position error profiles. (b) Parameter profiles. (c) Jacobian matrix error profiles. (d) Velocity norm profiles. (e) Joint velocity profiles. (f) Joint angle profiles.




(a) (b)

Figure 6.8: Comparison of position errors during the process of repetitive-motion redundancy resolution of the PUMA 560 manipulator with unknown parameters $a_2$, $a_3$, $d_3$, and $d_6$. (a) Via adaptive projection neural network (6.9). (b) Via nominal projection neural network (6.8) with $W = \hat{W}(0)$.

resolution process, the parameter learning is artificially stopped, i.e., $\dot{W}$ is set to 0 when $t \geq 5$ s. These results substantiate the efficacy of the proposed adaptive PNN in the minimum-velocity-norm redundancy resolution of manipulators with unknown physical parameters and also verify the theoretical results.

For comparison, the nominal PNN (6.8) is also employed to the minimum-velocity-norm redundancy resolution of the PUMA 560 manipulator. For fair comparison, all the shared parameters are set to the same, and the parameter matrix $W$ in (6.8), which is static, is set to the same value of $\hat{W}(0)$ in the adaptive PNN. The state variables of the nominal PNN (6.8) are shown in Fig. 6.4. As seen from Fig. 6.4(a), the magnitude of each element of $\mathbf{u}$ generated by the nominal PNN (6.8) is larger than that in Fig. 6.3(e). In addition, Fig. 6.4(b) shows a divergence of $\lambda$. Moreover, the comparison of control actions and position errors of the manipulator when the two projection neural networks are separately adopted is shown in Fig. 6.5, which shows that better performances (i.e., significantly reduced control actions and position errors) are achieved when the adaptive PNN is used compared with the nominal one. We also conduct simulations with different levels of initial parameter errors. The level of parameter error is

Table 6.2: Joint Displacements $\Delta\theta_i = |\theta_i(20) - \theta_i(0)|$ (rad) of the PUMA560 Manipulator When the End-Effector Finishes the 20-Second Tracking Task of the Cyclic Path Defined in (6.17) Via Different Projection Neural Networks (PNNs).

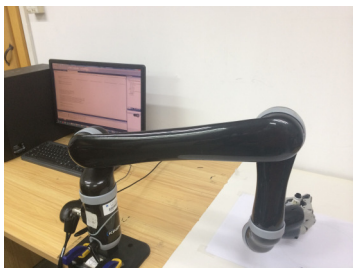
| Joint displacement | Adaptive PNN (6.9) | Nominal PNN (6.8) |
|---|---|---|
| $\Delta\theta_1$ | $1.5704 \times 10^{-3}$ | 9.1862 |
| $\Delta\theta_2$ | $5.1877 \times 10^{-4}$ | 1.8226 |
| $\Delta\theta_3$ | $2.1331 \times 10^{-4}$ | $5.3637 \times 10^{-1}$ |
| $\Delta\theta_4$ | $1.4095 \times 10^{-2}$ | $3.9954$ |
| $\Delta\theta_5$ | $8.1973 \times 10^{-4}$ | $8.98897 \times 10^{-1}$ |
| $\Delta\theta_6$ | $7.5146 \times 10^{-6}$ | 0 |

denoted by $\varphi$, which is calculated by

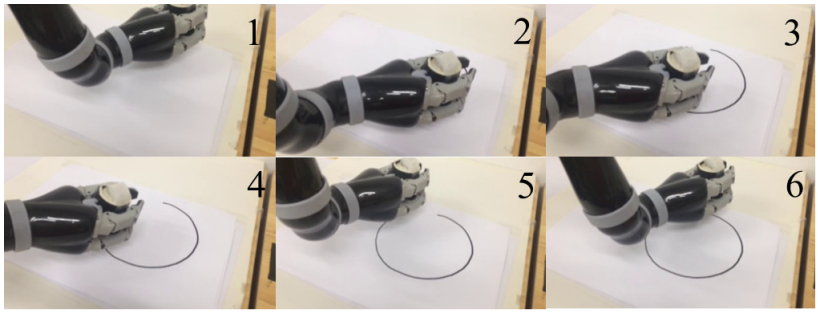
$$\varphi = \frac{\text{nominal parameter value} - \text{true parameter value}}{\text{true parameter value}}.$$

In the simulations, the values of elements in $\hat{W}(0)$ corresponds to the nominal values. Under the same setup, but with different different levels of initial parameter errors, simulations are conducted, for which the error norms are shown in Fig. 6.5(c) and Fig. 6.5(d). As seen from the two subfigures, with the increase of parameter error, the position error norm increases dramatically when the existing PNN is used. For example, when $\varphi = 10\%$, the maximal position error norm is about 3 cm. However, regardless of the parameter errors, the position error norm is always smaller than $4 \times 10^{-3}$ m when the proposed adaptive PNN is adopted. These results substantiate the superiority of the proposed adaptive PNN over the existing one.

To further show the efficacy of the proposed adaptive projection neural network (6.9), we have conducted simulations based on the PUMA560 manipulator used in this chapter for the two cases mentioned in Remark 7. We first consider the case that $d_6$ suddenly changes. Note that, in practice, it may take several seconds to replace the tool of a manipulator with another one. We use this extreme case merely to test the efficacy



(a)




(b)

Figure 6.9: The experiment platform, which includes a personal computer and a Kinova JACO$^2$ manipulator holding a pen and the snapshots during the experiment process. (a) Experiment platform. (b) Snapshots.

of the proposed method. In the simulation, it is assumed that $d_6 = 0.2$ when $t < 10$ and $d_6 = 0.3$ when $t \geq 10$. The other settings are the same as those above except that $\zeta = 8$. The simulation results are shown in Fig. 6.6. Under the same parameter settings, the simulation results for the welding scenario where $d_6$ keeps on decreasing is shown in Fig. 6.7, for which we assume that $d_6(t) = 0.2 - 0.0075t$ m. As seen from Fig. 6.6 and Fig. 6.7, under both cases, the proposed method can effectively handle the parameter uncertainty and guarantees the convergence of end-effector errors to a small neighbor of zero.

### 6.5.3 Repetitive-motion redundancy resolution

In this subsection, we further show the efficacy and superiority of the proposed adaptive PNN in repetitive-motion redundancy resolution. We consider the repetitive-motion

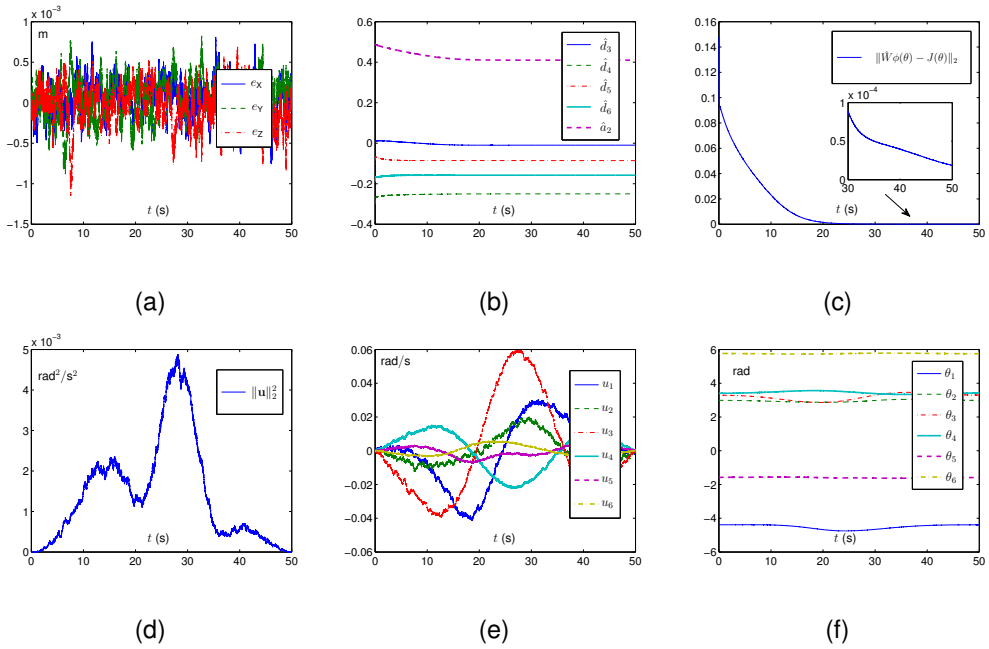



Figure 6.10: Data profiles during the experiment process. (a) Position error profiles. (b) Parameter profiles. (c) Jacobian matrix error profiles. (d) Velocity norm profiles. (e) Joint velocity profiles. (f) Joint angle profiles.

redundancy resolution of the UMA560 manipulator, which corresponds to the case that $A$ is an $6 \times 6$ identity matrix and $\mathbf{b} = \gamma(\theta(t) - \theta(0)$ in the problem formulation (6.4). In the simulations, $\gamma$ is set to 0.1. The results are also compared with the nominal PNN (6.8). Besides, the other setups for both neural networks are the same as those illustrated in the previous simulations. As seen from Fig. 6.8, the tracking accuracy of proposed adaptive PNN is much better than the nominal one. The comparison of the joint displacements defined as $\Delta\theta_i = |\theta_i(20) - \theta_i(0)|$ between the two simulations are shown in Table 6.2. As seen from this Table, the joint displacements are less than 0.02 rad when the adaptive PNN is used, which is much smaller than the maximal joint displacement (i.e., 9.1862) when when the nominal one is used. The above results further substantiate the efficacy and superiority of the proposed adaptive PNN (6.9) in repetitive-motion redundancy resolution of manipulators with unknown parameters.

Table 6.3: D-H Parameters of the Kinova JACO$^2$ Manipulator

| Joint | $a_i$ (m) | $\alpha_i$ (rad) | $d_i$ (m) |
|---|---|---|---|
| 1 | 0 | $\pi/2$ | $d_1$ |
| 2 | $a_2$ | $\pi$ | 0 |
| 3 | $0$ | $\pi/2$ | $d_3$ |
| 4 | 0 | $\pi/3$ | $d_4$ |
| 5 | 0 | $\pi/3$ | $d_5$ |
| 6 | 0 | $\pi$ | $d_6$ |

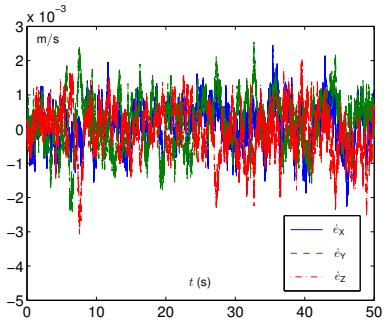



Figure 6.11: End-effector velocity error profiles during the experiment process.

## 6.6 Experimental verification

To further show the efficacy of the proposed adaptive PNN, experiments based on a Kinova JACO$^2$ manipulator have been conducted for the minimum-velocity-norm redundancy resolution aided by the proposed adaptive projection neural network (6.9).

The experiment platform is shown in Fig. 6.9. The D-H parameters of the manipulator is shown in Table. 6.3 with $a_2 = 0.41$, $d_1 = 0.2755$, $d_3 = -0.0098$, $d_4 = -0.2501$, $d_5 = -0.0856$, and $d_6 = -0.1578$. In the experiment, the parameters of the adaptive PNN (9) are set as $\varepsilon = 0.0001$ and $\nu = 4$. The standard deviation of zero-mean noise $\rho$ is set to $0.001$. By D-H convention [214], the analytical expression of $W$ for the Kinova JACO$^2$ manipulator can be readily derived and $W$ does not relate to $d_1$. As we

do not have motion capture devices, such as the OptiTrack, in the experiment, $\mathbf{r}_{\mathrm{d}}(t)$ and $\dot{\mathbf{r}}_{\mathrm{d}}(t)$ are calculated via the forward kinematics derived via the D-H convention by using the exact D-H parameter values. In the experiment, the manipulator is expected to draw a circle with the diameter being 0.2 m. Besides, the joint velocity bounds are $\eta^{+} = [1, 1, 1, 1, 1, 1]^{\mathrm{T}}$ rad/s and $\eta^{-} = [-1, -1, -1, -1, -1, -1]^{\mathrm{T}}$ rad/s. Some snapshots during the experiment process are show in Fig. 6.9(b), from which we can observe that the task is successfully completed. The related profiles are shown in Fig. 6.10. As seen from Fig. 6.10(a), the end-effector errors are less than $1$ mm. From Fig. 6.10(b), it can be observed that the parameters are convergent, which, together with Fig. 6.10(c), shows the convergence of parameter values to the corresponding exact values. The profiles of $\|\mathbf{u}\|_2^2$ and $\mathbf{u}$ are shown in Fig. 6.10(d) and Fig. 6.10(e), respectively. Evidently, the joint angle velocity does not exceed the given bounds. In addition, the joint angle profiles are shown in Fig. 6.10(f), and the end-effector velocity error profiles are shown in Fig. 6.11. The experimental results further show the validity and efficacy of the proposed adaptive PNN in the redundancy resolution of the Kinova JACO$^2$ physical manipulator.

## 6.7 Chapter summary

In this chapter, an adaptive PNN has been proposed for the redundancy resolution of manipulators with unknown physical parameters. Theoretical results have been presented to guarantee the performance of the proposed adaptive PNN. Besides, simulations results for two representative cases (i.e., minimum-velocity-norm redundancy resolution and repetitive-motion redundancy resolution) based on a PUMA 560 manipulator, together with the comparison with the existing nominal PNN, have substantiated the efficacy and superiority of the proposed PNN and verified the theoretical results. Future work of this chapter would be extending the results to acceleration-level redundancy resolution.

# Chapter 7

# Conclusions and future works

In this thesis, we have proposed, analyzed, and validated a learning and near-optimal control (LNOC) method for affine nonlinear systems. Three variants of the LNOC method have been discussed for different scenarios. The method has then been extended to address the consensus of nonlinear multi-agent systems with heterogeneity and the redundancy resolution of redundant manipulators with unknown physical parameters. The proposed scheme applies to both underactuated and overactuated control systems

The future works along the research direction, include but are not limited to the following.

- The LNOC of nonlinear systems with state observers or measurement noises considered can be investigated. For example, robust control methodologies might be incorporated into the control design so as to enhance the efficacy of the proposed control laws.
- In practice, the performance of the proposed LNOC method may be improved via using the predictive smooth variable structure filters [84,85], which can be further investigated.

- In practice, disturbances are unavoidable. The enhancement of the proposed method by incorporating the disturbance observer technique [135–137] is worth investigation.
- Although the dynamics of many physical systems is in the affine form, there are some nonaffine physical systems. Thus, the extension of the LNOC method to nonaffine systems is worth investigation.

# References


[1] F. L. Lewis, D. Vrabie, and V. Syrmos, *Optimal Control*. 3rd ed. New York, USA: Wiley, 2012.

[2] E. K. Kim, F. Mwasilu, H. H. Choi, and J. W. Jung, "An observer-based optimal voltage control scheme for three-phase UPS systems," *IEEE Trans. Ind. Electron.*, vol. 62, no. 4, pp. 2073–2081, Apr. 2015.

[3] S. Moghadasi and S. Kamalasadan, "Optimal fast control and scheduling of power distribution system using integrated receding horizon control and convex conic programming," *IEEE Trans. Ind. Appl.*, vol. 52, no. 3, pp. 2596–2606, May 2016.

[4] T. Dierks, B. Brenner, and S. Jagannathan, "Neural network-based optimal control of mobile robot formations with reduced information exchange," *IEEE Trans. Control Syst. Technol.*, vol. 21, no. 4, pp. 1407–1415, Jul. 2013.

[5] E. Trelat, "Optimal control and applications to aerospace: some results and challenges," *J. Optim. Theory Appl.*, vol. 254, no. 3, pp. 713–758, Sep. 2012.

[6] W. H. Chen, D. J. Ballance, and P. J. Gawthrop, "Optimal control of nonlinear systems: a predictive control approach," *Automatica*, vol. 39, no. 4, pp. 633–641, Apr. 2003.

[7] M. Abu-Khalaf, F. L. Lewis, “Nearly optimal control laws for nonlinear systems with saturating actuators using a neural network HJB approach,” *Automatica*, vol. 41, no. 5, pp. 779–791, May 2005.

[8] Z. Wang, X. Liu, K. Liu, S. Li, and H. Wang, “Backstepping-based Lyapunov function construction using approximate dynamic programming and sum of square techniques,” *IEEE Trans. Cybern.*, in press.

[9] M. Sassano and A. Astolfi, “Dynamic approximate solutions of the HJ inequality and of the HJB equation for input-affine nonlinear systems,” *IEEE Trans. Autom. Control*, vol. 57, no. 10, pp. 2490–2503, Oct. 2012.

[10] K. G. Vamvoudakis and F. L. Lewis, “Online actor-critic algorithm to solve the continuous-time infinite horizon optimal control problem,” *Automatica*, vol. 46, no. 5, pp. 878–888, May 2010.

[11] G. Tao, “Multivariable adaptive control: A survey,” *Automatica*, vol. 50, no. 11, pp. 2737–2764, Nov. 2014.

[12] J. Na, Q. Chen, X. Ren, and Y. Guo, “Adaptive prescribed performance motion control of servo mechanisms with friction compensation,” *IEEE Trans. Control Syst. Technol.*, vol. 61, no. 1, pp. 486–494, Jan. 2014.

[13] G. Li, J. Na, D. P. Stoten, and X. Ren, “Adaptive neural network feedforward control for dynamically substructured systems,” *IEEE Trans. Control Syst. Technol.*, vol. 22, no. 3, pp. 944–954, May 2014.

[14] Z. Jiang, “Global tracking control of underactuated ships by Lyapunov’s direct method,” *Automatica*, vol. 38, no. 2, pp. 301–309, Feb. 2002.

[15] A. Behal, D. M. Dawson, W. E. Dixon, and Y. Fang, “Tracking and regulation control of an underactuated surface vessel with nonintegrable dynamics,” *IEEE Trans. Auotom. Control*, vol. 47, no. 3, pp. 495–500, Mar. 2002.

[16] Z. Yan and J. Wang, “Model predictive control for tracking of underactuated vessels based on recurrent neural networks,” *IEEE J. Ocean. Eng.*, vol. 37, no. 4, pp. 717–726, Oct. 2012.

[17] R.Yu, Q. Zhu, G. Xia, and Z. Liu, “Sliding mode tracking control of an underactuated surface vessel,” *IET Control Theory Appl.*, vol. 6, no. 3, pp. 461–466, Mar. 2012.

[18] H. Ashrafiuon, K. R. Muske, L. C. McNinch, and R. A. Soltan, “Sliding-mode tracking control of surface vessels,” *IEEE Trans. Ind. Electron.*, vol. 55, no. 11, pp. 4004–4012, Nov. 2008.

[19] T. Elmokadem, M. Zribi, and K. Youcef-Toumi, “Trajectory tracking sliding mode control of underactuated AUVs,” *Nonlinear Dyn.*, vol. 84, no. 2, pp. 1079–1091, Apr. 2016.

[20] C. Hu, R. Wang, F. Yan, and N. Chen, “Robust composite nonlinear feedback path following control for underactuated surface vessels with desired-heading amendment,” *IEEE Trans. Ind. Electron.*, vol. 63, no. 10, pp. 6386–6394, Oct. 2016.

[21] C.-Z. Pan, X.-Z. Lai, S. X. Yang, and M. Wu, “A biologically inspired approach to tracking control of underactuated surface vessels subject to unknown dynamics,” *Expert Syst. Appl.*, vol. 42, no. 4, pp. 2153–2161, Mar. 2015.

[22] H. Modares, F. L. Lewis, and M. B. Naghibi-Sistani, “Adaptive optimal control of unknown constrained-input systems using policy iteration and neural networks,” *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 24, no. 10, pp. 1513–1525, Oct. 2013.

[23] F. Y. Wang, H. Zhang, and D. Liu, “Adaptive dynamic programming: an introduction,” *IEEE Control Syst. Mag.*, vol. 4, no. 2, pp. 39–47, May 2009.

[24] D. Liu and Q. Wei, “Policy iteration adaptive dynamic programming algorithm for discrete-time nonlinear systems,” *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 25, no. 3, pp. 621–633, Mar. 2014.

[25] A. Heydari and S. N. Balakrishnan, “Finite-horizon control-constrained nonlinear optimal control using single network adaptive critics,” *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 24, no. 1, pp. 145–157, Jan. 2013.

[26] Y. J. Liu, L. Tang, S. Tong, C. L. P. Chen, and D. J. Li, “Reinforcement learning design-based adaptive tracking control with less learning parameters for nonlinear discrete-time MIMO systems,” *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 26, no. 1, pp. 165–176, Jan. 2015

[27] D. Wang, D. Liu, and H. Li, “Policy iteration algorithm for online design of robust control for a class of continuous-time nonlinear systems,” *IEEE Trans. Autom. Sci. Eng.*, vol. 11, no. 2, pp. 627–632, Apr. 2014.

[28] H. Zhang, Q. Wei, and Y. Luo, “A novel infinite-time optimal tracking control scheme for a class of discrete-time nonlinear systems via the greedy HDP iteration algorithm,” *IEEE Trans. Syst., Man, Cybern., Part B, Cybern.*, vol. 38, no. 4, pp. 937–942, Jul. 2008.

[29] Y. Lv, J. Na, Q. Yang, X. Wu, and Y. Guo, “Online adaptive optimal control for continuous-time nonlinear systems with completely unknown dynamics,” *Int. J. Control*, vol. 89, no. 1, pp. 99–112, Jan. 2016.

[30] R. Kamalapurkar, B. Reish, G. Chowdhary, and W. E. Dixon, “Concurrent learning for parameter estimation using dynamic state-derivative estimators,” *IEEE Trans. Autom. Control*, in press, doi: 10.1109/TAC.2017.2671343.

[31] S. Kersting and M. Buss, “Direct and indirect model reference adaptive control for multivariable piecewise affine systems,” *IEEE Trans. Autom. Control*, in press, doi: 10.1109/TAC.2017.2690060.

[32] D. Q. Mayne and H. Michalska, “Receding horizon control of nonlinear systems,” *IEEE Trans. Autom. Control*, vol. 35, no. 7, pp. 814–824, Jul. 1990.

[33] Y. K. Liu and Y. M. Zhang, “Model-based predictive control of weld penetration in gas tungsten arc welding,” *IEEE Trans. Control Syst. Technol.*, vol. 22, no. 3, pp. 955–966, May 2014.

[34] M. P. Akter, S. Mekhilef, N. M. L. Tan, and H. Akagi, “Modified model predictive control of a bidirectional AC-DC converter based on Lyapunov function for energy storage systems,” *IEEE Trans. Ind. Electron.*, vol. 63, no. 2, pp. 704–715, Feb. 2016.

[35] D. Q. Mayne, “Model predictive control: Recent developments and future promise,” *Automatica*, vol. 50, no. 12, pp. 2967–2986, Dec. 2014.

[36] S. K. Pradhan and B. Subudhi, “Nonlinear adaptive model predictive controller for a flexible manipulator: an experimental study,” *IEEE Trans. Control Syst. Technol.*, vol. 22, no. 5, pp. 1754–1768, Sep. 2014.

[37] Y. Zhang and S. Li, “Predictive suboptimal consensus of multiagent systems with nonlinear dynamics,” *IEEE Trans. Syst., Man, Cybern., Syst.*, in press, doi: 10.1109/TSMC.2017.2668440.

[38] Y. Zhang, D. Chen, L. Jin, Y. Zhang, and Y. Yin, “GD-aided IOL (input-output linearisation) controller for handling affine-form nonlinear system with loose condition on relative degree,” *Int. J. Control*, vol. 89, no. 4, pp. 757-769, 2016.

[39] A. Isidori, *Nonlinear Control Systems: An Introduction*. 3rd ed., New York, USA: Springer, 1995.

[40] W. Sun, S. Tang, H. Gao, and J. Zhao, “Two time-scale tracking control of nonholonomic wheeled mobile robots,” *IEEE Trans. Control Syst. Technol.*, vol. 24, no. 6, pp. 2059–2069, Nov. 2016.

[41] A. R. Mehrabian and K. Khorasani, “Distributed formation recovery control of heterogeneous multiagent Euler-Lagrange systems subject to network switching and

diagnostic imperfections," *IEEE Trans. Control Syst. Technol.*, vol. 24, no. 6, pp. 2158–2166, Nov. 2016.

[42] C. T. Chen, *Linear System Theory and Design*. New York, USA: Oxford Univ. Press, 1999.

[43] C. Yang, Z. Li, R. Cui, and B. Xu, "Neural network-based motion control of underactuated wheeled inverted pendulum models," *IEEE Trans. Neural Netw. Learn. Syst*, vol. 25, no. 11. pp. 2004–2016, Nov. 2014.

[44] H. K. Khalil, *Nonlinear Systems*. 3nd ed., NJ, USA: Prentice-Hall, 2002.

[45] S. Boyd and L. Vandenberghe, *Convex Optimization*. Cambridge: Cambridge University Press, 2004.

[46] B. Liao, Y. Zhang, and L. Jin, "Taylor $O(h^3)$ Discretization of ZNN models for dynamic equality-constrained quadratic programming with application to manipulators," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 27, no. 2, pp. 225–237, Feb. 2016.

[47] H. L. Royden, *Real Analysis*. New York, USA: Macmillan, 1988.

[48] Z. Liu, C. Li, and W. Xu, "Hybrid control of biped robots in the double-support phase $H_\infty$ approach and fuzzy neural networks," *IEE Proc. Control Theory Appl.*, vol. 150, no. 4, pp. 347–354, Jul. 2003.

[49] Y. Liu and S. Tong, "Barrier Lyapunov functions-based adaptive control for a class of nonlinear pure-feedback systems with full state constraints," *Automatica*, vol. 64, pp. 70–75, Feb. 2016.

[50] G. Lai, C. L. P. Chen, and Y. Zhang, "Adaptive asymptotic tracking control of uncertain nonlinear system with input quantization," *Syst. Control Lett.*, vol. 96, pp. 23–29, Oct. 2016.

[51] Y. Liu, Y. Gao, S. Tong, and Y. Li, "Fuzzy approximation-based adaptive backstepping optimal control for a class of nonlinear discrete-time systems with dead-zone," *IEEE Trans. Fuzzy Syst.*, vol. 24, no. 1, pp. 16–28, Feb. 2016.

[52] Y. Liu and S. Tong, "Adaptive fuzzy control for a class of unknown nonlinear dynamical systems," *Fuzzy Set. Syst.*, vol. 263, no. 15, pp. 49–70, Mar. 2015.

[53] K. G. Vamvoudakis, "Q-learning for continuous-time linear systems: A model-free infinite horizon optimal control approach," *Syst. Control Lett.*, vol. 100, pp. 14–20, Feb. 2017.

[54] F. L. Lewis, D. Vrabie, and V. L. Syrmos. *Optimal control*. Hoboken, NJ: Wiley, 2012.

[55] D. Q. Mayne and H. Michalska, "Receding horizon control of nonlinear systems," *IEEE Trans. Autom. Control*, vol. 35, no. 7, pp. 814–824, Jul. 1990.

[56] T. A. Johansen, "Toward dependable embedded model predictive control," *IEEE Syst. J.*, vol. 11, no. 2, pp. 1208–1219, Jun. 2017.

[57] A. Chakrabarty, V. Dinh, M. J. Corless, A. E. Rundell, S. H. Żack, and G. T. Buzzard, "Support vector machine informed explicit nonlinear model predictive control using low-discrepancy sequences," *IEEE Trans. Autom. Control*, vol. 62, no. 1, pp. 135–148, Jan. 2017.

[58] W.-H. Chen, D. J. Ballance, and P. J. Gawthrop, "Optimal control of nonlinear systems: a predictive control approach," *Automatica*, vol. 39, no. 4, pp. 633–641, Apr. 2003.

[59] Y. Zhang and S. Li, "Predictive suboptimal consensus of multiagent systems with nonlinear dynamics," *IEEE Trans. Syst., Man, Cybern., Syst.*, vol. 47, no. 7, pp. 1701–1711, Jul. 2017.

[60] Y. Zhang and S. Li, “Time-scale expansion-based approximated optimal control for underactuated systems using projection neural networks,” *IEEE Trans. Syst., Man, Cybern., Syst.*, in press, doi: 10.1109/TSMC.2017.2703140.

[61] Z. Yu, L. Xiao, H. Li, X. Zhu, and R. Huai, “Model parameter identification for lithium batteries using the coevolutionary particle swarm optimization method,” *IEEE Trans. Ind. Electron.*, vol. 64, no. 7, pp. 5690–5700, Jul. 2017.

[62] S.-H. Lee, A. Yoo, H.-J. Lee, Y.-D. Yoon, and B.-M. Han, “Identification of induction motor parameters at standstill based on integral calculation,” *IEEE Trans. Ind. Appl.*, vol. 53, no. 3, pp. 2130–2139, May 2017.

[63] S. A. Odhano, R. Bojoi, E. Armando, G. Homrich, A. F. Flores Filho, M. Popescu, and D. G. Dorrell, “Identification of three-phase IPM machine parameters using torque tests,” *IEEE Trans. Ind. Appl.*, vol. 53, no. 3, 1883–1891, May 2017.

[64] G. Pin, L. Magni, T. Parisini, and D. M. Raimondo, “Robust receding - horizon control of nonlinear systems with state dependent uncertainties: An input-to-state stability approach,” in *Proc. American Control Conf.*, Jun. 2008, pp. 1667–1672.

[65] X. Zhao, H. Yang, and G. Zong, “Adaptive neural hierarchical sliding mode control of nonstrict-feedback nonlinear systems and an application to electronic circuits”, *IEEE Trans. Syst. Man, Cybern. Syst.*, vol. 47, no. 7, pp. 1394–1404, Jul. 2017.

[66] S. Yin, H. Yang, and O. Kaynak, “Sliding mode observer-based FTC for markovian jump systems with actuator and sensor faults,” *IEEE Trans. Ind. Electron.*, vol. 62, no. 7, pp. 3551–3558, Jul. 2017.

[67] L. Cao, X. Li, X. Chen, and Y. Zhao, “Minimum sliding mode error feedback control for fault tolerant small satellite attitude control,” *Adv. Space Res.*, vol. 53, no. 2, pp. 309–324, Jan. 2014.

[68] L. Cao, X. Chen, and A. K. Misra, “Minimum sliding mode error feedback control for fault tolerant reconfigurable satellite formations with J2 perturbations,” *Acta Astronaut.*, vol. 96, pp. 201–216, Mar. 2014.

[69] W. Sun, S. Tang, H. Gao, and J. Zhao, “Two time-scale tracking control of nonholonomic wheeled mobile robots,” *IEEE Trans. Control Syst. Technol.*, vol. 24, no. 6, pp. 2059–2069, Nov. 2016.

[70] J. Baek, M. Jin, and S. Han, “A new adaptive sliding-mode control scheme for application to robot manipulators,” *IEEE Trans. Ind. Electron.*, vol. 63, no. 5, pp. 3632–3637, Jun. 2016.

[71] J. Han, “From PID to active disturbance rejection control,” *IEEE Trans. Ind. Electron.*, vol. 56, no. 3, pp. 900–906, Mar. 2009.

[72] A. Levant, “Robust exact differentiation via sliding mode technique,” *Automatica*, vol. 34, no. 3, pp. 379–384, Mar. 1998.

[73] J. Davila, “Exact tracking using backstepping control design and high-order sliding modes,” *IEEE Trans. Autom. Control*, vol. 58, no. 8, pp. 2077–2081, Aug. 2013.

[74] H. K. Khalil. *Nonlinear Systems*. 3rd edition. NJ, USA: Prentice-Hall, 2002.

[75] F. M. Atay, “Van der Pol’s oscillator under delayed feedback,” *J. Sound Vib.*, vol. 218, no. 2, pp. 333–339, 1998.

[76] C. A. Chuang, T. T. Lee, C. C. Tien, C. F. Hsu, “Adaptive wavelet neural network control for DC motors via second-order sliding-mode approach,” in *Proc. Int. Conf. Mach. Learn. Cybern.*, May 2011, pp. 1174–1179.

[77] J. Baek, M. Jin, and S. Han, “A new adaptive sliding-mode control scheme for application to robot manipulators,” *IEEE Trans. Ind. Electron.*, vol. 63, no. 6, pp. 3628–3637, Jun. 2016.

[78] F.-J. Lin and P.-H. Chou, “Adaptive control of two-axis motion control system using interval type-2 fuzzy neural network,” *IEEE Trans. Ind. Electron.*, vol. 56, no. 1, pp. 178–193, Jan. 2009.

[79] R. K. Singh and S. Yadav, “Optimized PI controller for an interacting spherical tank system,” in *Proc. 1st Conf. Electron. Mater. Eng. Nano-Technol.*, doi: 10.1109/IEMENTECH.2017.8076977.

[80] R. L. A. Shauri, N. M. Salleh, and A. K. A. Hadi, “PID position control of 7-DOF three-fingered robotic hand for grasping task,” in *Proc. IEEE Int. Conf. Control Syst. Comput. Eng.*, pp. 70–74čň 2014.

[81] P. Anantachaisilp and Z. Lin, “An experimental study on PID tuning methods for active magnetic bearing systems,” *Int. J. Adv. Mechatron. Syst.*, vol. 5, no. 2, pp. 146–154, 2013.

[82] Datasheet. Available at http://www.atmel.com/Images/Atmel-42735-8-bit-AVR-Microcontroller-ATmega328-328P_Datasheet.pdf

[83] A. Hughes and B. Drury. *Electric Motors And Drives: Fundamentals, Types And Applications*. Oxford, UK: Newnes, 2013.

[84] L. Cao and H. Li, “Unscented predictive variable structure filter for satellite attitude estimation with model errors when using low precision sensors,” *Acta Astronaut.*, vol. 127, pp. 505–513, Oct. 2016.

[85] L. Cao, Y. Chen, Z. Zhang, and H. Li, “Predictive smooth variable structure filter for attitude synchronization estimation during satellite formation flying,” *IEEE Trans. Aerosp. Electron. Syst.*, vol. 53, no. 3, pp. 1375–1383, Jun. 2017.

[86] H. Wang, P. Shi, H. Li, and Q. Zhou, “Adaptive neural tracking control for a class of nonlinear systems with dynamic uncertainties,” *IEEE Trans. Cybern.*, vol. 47, no. 10, pp. 3075–3087, Oct. 2017.

[87] Q. Zhou, P. Shi, Y. Tian, and M. Wang, “Aproximation-based adaptive tracking control for MIMO nonlinear systems with input saturation,” *IEEE Trans. Cybern.*, vol. 45, no. 10, pp. 2119-2128, Oct. 2015.

[88] Q. Zhou, H. Li, C. Wu, L. Wang, and C. K. Ahn, “Adaptive fuzzy control of nonlinear systems with unmodeled dynamics and input saturation using small-gain approach,” vol. 47, no. 8, pp. 1979–1989, Aug. 2017.

[89] J. Na, X. Ren, and D. Zheng, “Adaptive control for nonlinear pure-feedback systems with high-order sliding mode observer,” *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 24, no. 3, pp. 370–382, Mar. 2013.

[90] E. Kayacan, E. Kayacan, H. Ramon, and W. Saeys, “Adaptive neuro-fuzzy control of a spherical rolling robot using sliding-mode-control-theory-based online learning algorithm,” *IEEE Trans. Cybern.*, vol. 43, no. 1, pp. 170–179, Feb. 2013.

[91] Q. Shen, P. Shi, and Y. Shi, “Distributed adaptive fuzzy control for nonlinear multiagent systems via sliding mode observers,” *IEEE Trans. Cybern.*, vol. 46, no. 12, pp. 3086–3097, Dec. 2016.

[92] Y. J. Liu, S. C. Tong, D. Wang, T. S. Li, and C. L. P. Chen, “Adaptive neural output feedback controller design with reduced-order observer for a class of uncertain nonlinear SISO systems,” *IEEE Trans. Neural Netw.*, vol. 22, no. 8, pp. 1328–1334, Aug. 2011.

[93] S. L. Dai, M. Wang, and C. Wang, “Neural learning control of marine surface vessels with guaranteed transient tracking performance,” *IEEE Trans. Ind. Electron.*, vol. 63, no. 3, pp. 1717–1727, Mar. 2016.

[94] A. K. Kostarigka and G. A. Rovithakis, “Adaptive dynamic output feedback neural network control of uncertain MIMO nonlinear systems with prescribed performance,” *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 23, no. 1, pp. 138–149, Jan. 2012.

[95] Q. Zhou, H. Li, and P. Shi, “Decentralized adaptive fuzzy tracking control for robot finger dynamics,” *IEEE Trans. Fuzzy Syst.*, vol. 23, no. 3, pp. 501–510, Jun. 2015.

[96] S. Li, M. Zhou, and X. Luo, “Modified primal-dual neural networks for motion control of redundant manipulators with dynamic rejection of harmonic noises,” *IEEE Trans. Neural Netw. Learn. Syst.*, in press, doi: 10.1109/TNNLS.2017.2770172.

[97] C. Yang, X. Wang, L. Cheng, and H. Ma, “Neural-learning-based telerobot control with guaranteed performance,” *IEEE Trans. Cybern.*, vol. 47, no. 10, pp. 3148–3159, Oct. 2017.

[98] W. He, Z. Yan, C. Sun, and Y. Chen, “Adaptive neural network control of a flapping wing micro aerial vehicle with disturbance observer,” *IEEE Trans. Cybern.*, vol. 47, no. 10, pp. 3452–3465, Oct. 2017.

[99] L. Cheng, W. Liu, C. Yang, T. Huang, Z. Hou, and M. Tan, “A neural-network-based controller for piezoelectric-actuated stick-lip devices,” *IEEE Trans. Ind. Electron.*, vol. 65, no. 3, pp. 2598–2607, Mar. 2018.

[100] W. H. Chen, D. J. Balance, P. J. Gawthrop, J. J. Gribble, and J. O. Reill, “Nonlinear PID predictive controller,” *Control Theory Appl.*, vol. vol. 146, no. 6, pp. 603–611, Nov. 1999.

[101] C. Liu, W. H. Chen, and J. Andrews, “Tracking control of small-scale helicopters using explicit nonlinear MPC augmented with disturbance observers,” *Control Eng. Pract.*, vol. 20, no. 3, pp. 258–268, Mar. 2012.

[102] R. Errouissi, A. Al-Durra, and S. M. Muyeen, “Design and implementation of a nonlinear PI predictive controller for a grid-tied photovoltaic inverter,” *IEEE Trans. Ind. Electron.*, vol. 64, no. 2, pp. 1250–1241, Feb. 2017.

[103] R. Errouissi, M. Ouhrouche, W. H. Chen, and A. M. Trzynadlowski, “Robust nonlinear predictive controller for permanent-magnet synchronous motors with an op-

timized cost function," *IEEE Trans. Ind. Electron.*, vol. 59, no. 7, Jul. pp. 2849-2858, 2011.

[104] Y. J. Liu, Y. Gao, S. Tong, and Y. Li, "Fuzzy approximation-based adaptive backstepping optimal control for a class of nonlinear discrete-time systems with deadzone," *IEEE Trans. Fuzzy Syst.*, vol. 24, no. 1, pp. 16–28, Feb. 2016.

[105] K. Sun, Y. Li, and S. Tong, "Fuzzy adaptive output feedback optimal control design for strict-feedback nonlinear systems," *IEEE Trans. Syst. Man Cybern. Syst.*, vol. 47, no. 1, pp. 33–44, Jan. 2017.

[106] Y. Wang, Z. Miao, H. Zhong, and Q. Pan, "Simultaneous stabilization and tracking of nonholonomic mobile robots: A Lyapunov-based approach," *IEEE Trans. Control Syst. Technol.*, vol. 23, no. 4, pp. 1440–1450, Jul. 2015.

[107] Y. Zhang, X. Yu, Y. Yin, C. Peng, and Z. Fan, "Singularity-conquering ZG controllers of z2g1 type for tracking control of the IPC system," *Int. J. Control*, vol. 87, no. 9, pp. 1729–1746, 2014.

[108] J. Baek, M. Jin, and S. Han, "A new adaptive sliding-mode control scheme for application to robot manipulators," *IEEE Trans. Ind. Electron.*, vol. 63, no. 5, pp. 3632–3637, Jun. 2016.

[109] Y. Zhang, D. Chen, L. Jin, Y. Zhang, and Y. Yin, "GD-aided IOL (input-output linearisation) controller for handling affine-form nonlinear system with loose condition on relative degree," *Int. J. Control*, vol. 89, no. 4, pp. 757–769, 2016.

[110] Y. Zhang, G. Tao, and M. Chen, "Adaptive neural network based control of noncanonical nonlinear systems," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 27, no. 9, pp. 1864–1877, Sep. 2016.

[111] J. Davila, "Exact tracking using backstepping control design and high-order sliding modes," *IEEE Trans. Autom. Control*, vol. 58, no. 8, pp. 2077–2081, Aug. 2013.

[112] H. Castaneda, F. Plestan, A. Chriette, and J. D. León-Morales, "Continuous differentiator based on adaptive second-order sliding-mode control for a 3-DOF helicopter," *IEEE Trans. Ind. Electron.*, vol. 63, no. 9, pp. 5786–5793, Sep. 2016.

[113] D. Gu and H. Hu, "Receding horizon tracking control of wheeled mobile robots," *IEEE Trans. Control Syst. Technol.*, vol. 14, no. 4, pp. 743–749, Jul. 2016.

[114] G. Cybenko, "Approximations by superpositions of a sigmoidal function," *Math. Control Signals Syst.*, vol. 2, no. 4, pp. 303–314, 1989.

[115] G. B. Huang, Q. Y. Zhu, and C. K. Siew, "Extreme learning machine: Theory and applications," *Neurocomputing*, vol. 70, pp. 489–501, Dec. 2006.

[116] Z. Zhang and Y. Zhang, "Acceleration-level cyclic-motion generation of constrained redundant robots tracking different paths," *IEEE Trans. Syst., Man, Cybern. B, Cybern.*, vol. 42, no. 4, pp. 1257–1269, Aug. 2012.

[117] I. M. B. Hassine, M. W. Naouar, and N. Mrabet-Bellaaj, "Model predictive-sliding mode control for three-phase grid-connected converters," *IEEE Trans. Ind. Electron.*, vol. 64, no. 2, pp. 1341–1349, Feb. 2017.

[118] A. M. Mohammed and S. Li, "Dynamic neural networks for kinematic redundancy resolution of parallel stewart platforms," *IEEE Trans. Cybern.*, vol. 46, no. 7, pp. 1538–1550, Jul. 2016.

[119] S. Li and Y. Li, "Nonlinearly activated neural network for solving time-varying complex Sylvester equation," *IEEE Trans. Cybern.*, vol. 44, no. 8, pp. 1397–1408, Aug. 2014.

[120] C. T. Chen. *Linear System Theory And Design*. New York: Oxford University Press, 1995.

[121] L. Jin, Y. Zhang, S. Li, and Y. Zhang, "Noise-tolerant ZNN models for solving time-varying zero-finding problems: A control-theoretic approach," *IEEE Trans. Autom. Control*, vol. 62, no. 2, pp. 992–997, Feb. 2017.

[122] G. Joya, M. A. Atencia, and F. Sandoval, "Hopfield neural networks for optimization: Study of the different dynamics," *Neurocomputing*, vol. 43, pp. 219–237, 2002.

[123] A. Cichocki and R. Unbehauen, "Robust neural networks with on-line learning for blind identification and blind sepration of sources," *IEEE Trans. Circuits Syst. I, Fundam. Theory Appl.*, vol. 43, no. 11, pp. 894–906, Nov. 1996.

[124] A. Levant, "Robust exact differentiation via sliding mode technique," *Automatica*, vol. 34, no. 3, pp. 379–384, Mar. 1998.

[125] B. Z. Guo and Z. L. Zhao, "On convergence of tracking differentiator," *Int. J. Control*, vol. 84, no. 4, pp. 693–701, Apr. 2011.

[126] D. Tian, H. Shen, and M. Dai, "Improving the rapidity of nonlinear tracking differentiator via feedforward," *IEEE Trans. Ind. Electron.*, vol. 61, no. 7, pp. 3736–3743, Jul. 2014.

[127] W. He, X. He, and S. S. Ge, "Vibration control of flexible marine riser systems with input saturation," *IEEE/ASME Trans. Mechatron.*, vol. 21, no. 1, pp. 254–265, Feb. 2016.

[128] J. Yu, L. Zhao, H. Yu, C. Lin, and W. Dong, "Fuzzy finite-time command filtered control of nonlinear systems with input saturation," *IEEE Trans. Cybern.*, in press, doi: 10.1109/TCYB.2017.2738648.

[129] B. Jiang, Q. Hu, and M. I. Friswell, "Fixed-time attitude control for rigid spacecraft with actuator saturation and faults," *IEEE Trans. Control Syst. Technol.*, vol. 24, no. 5, pp. 1892–1898, Sep. 2016.

[130] S. Boyd and S. S. Sastry, “Necessary and sufficient conditions for parameter convergence in adaptive control,” *Automatica*, vol. 22, no. 6, pp. 629–639, Nov. 1986.

[131] W. Chen, C. Wen, S. Hua, and C. Sun, “Distributed cooperative adaptive identification and control for a group of continuous-time systems with a cooperative PE condition via consensus,” *IEEE Trans. Autom. Control*, vol. 59, no. 1, pp. 91–106, Jan. 2014.

[132] W. E. Dixon, D. M. Dawson, F. Zhang, and E. Zergeroglu, “Global exponential tracking control of a mobile robot system via a PE condition,” IEEE Trans. Syst., Man, Cybern. B, Cybern., vol. 30, no. 1, pp. 129–142, Feb. 2000.

[133] U. Münz, A. Papachristodoulou, and F. Allgöwer, “Robust consensus controller design for nonlinear relative degree two multi-agent systems with communication constraints,” *IEEE Trans. Autom. Control*, vol. 56, no. 1, pp. 145–151, Jan. 2011.

[134] A. Loría, “Observers are unnecessary for output-feedback control of lagrangian systems,” *IEEE Trans. Autom. Control*, vol. 61, no. 4, pp. 905–920, Apr. 2016.

[135] W. Chen, D. J. Ballance, P. J. Gawthrop, and J. O’Reilly, “A nonlinear disturbance observer for robotic manipulators,” *IEEE Trans. Ind. Electron.*, vol. 47, no. 4, pp. 932–938, Aug. 2000.

[136] W. Chen, “Disturbance observer based control for nonlinear systems,” *IEEE/ASME Trans. Mechatron.*, vol. 9, no. 4, pp. 706–710, Dec. 2004.

[137] J. Yang, S. Li, and X. Yu, “Sliding-mode control for systems with mismatched uncertainties via a disturbance observer,” *IEEE Trans. Ind. Electron.*, vol. 60, no. 1, pp. 160–169, Jan. 2013.

[138] R. K. Nagle, E. Saff, and A. Snider, *Fundamentals of Differential Equations*, 7th ed., Boston, USA: Pearson Addison Wesley, 2008.

[139] Y. Cao and W. Ren, "Finite-time consensus for multi-agent networks with unknown inherent nonlinear dynamics," *Automatica*, vol. 50, pp. 2648–2656, 2014.

[140] W. Chen, X. Li, W. Ren, and C. Wen, "Adaptive consensus of multi-agent systems with unknown identical control directions based on a novel nussbaum-type function," *IEEE Trans. Autom. Control*, vol. 59, no. 7, pp. 1887–1892, 2014.

[141] C. L. P. Chen, G. X. Wen, Y. J. Liu, and Z. Liu, "Observer-based adaptive backstepping consensus tracking control for high-order nonlinear semi-strict-feedback multiagent systems," *IEEE Trans. Cybern.*, vol. 46, no. 7, pp. 1591–1601, 2016.

[142] C. L. P. Chen, G. X. Wen, Y. J. Liu, and F. Y. Wang, "Adaptive consensus control for a class of nonlinear multiagent time-delay systems using neural networks," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 25, no. 6, pp. 1217–1226, 2014.

[143] L. Cheng, Z. G. Hou, and M. Tan, "A mean square consensus protocol for linear multi-agent systems with communication noises and fixed topologies," *IEEE Trans. Autom. Control*, vol. 59, no. 1, pp. 261–267, 2014.

[144] C. Edwards and Y. B. Shtessel, "Adaptive continuous higher order sliding mode control," *Automatica*, vol. 65, pp. 183–190, 2016.

[145] M. C. Fan, Z. Chen, and H. T. Zhang, "Semi-global consensus of nonlinear second-order multi-agent systems with measurement output feedback," *IEEE Trans. Autom. Control*, vol. 59, no. 8, 2222–2227, 2014.

[146] G. Ferrari-Trecate, L. Galbusera, M. P. E. Marciandi, and R. Scattolini, "Model predictive control schemes for consensus in multi-agent systems with single- and double-integrator dynamics," *IEEE Trans. Autom. Control*, vol. 54, no. 11, pp. 2560–2572, 2009.

[147] C. Godsil and G. Royal, *Algebraic Graph Theory*. New York, USA: Springer-Verlag, 2001.

[148] C. Y. Hsu, S. Y. Shaw, and H. J. Wong, “Refinements of generalized triangle inequalities,” *J. Math. Ana. Appl.*, vol. 344, no. 1, pp. 17–31, 2008.

[149] C. C. Hua, X. You, and X. P. Guan, “Leader-following consensus for a class of high-order nonlinear multi-agent systems,” *Automatica*, vol. 73, pp. 138–144, 2016.

[150] J. Huang, C. Wen, W. Wang, and Y. D. Song, “Adaptive finite-time consensus control of a group of uncertain nonlinear mechanical systems,” *Automatica*, vol. 51, pp. 292–301, 2015.

[151] R. Kamalapurkar, H. Dinh, P. Walters, and W. Dixon, “Approximate optimal cooperative decentralized control for consensus in a topological network of agents with uncertain nonlinear dynamics,” *American Control Conf.*, pp. 1320–1325, Washington, DC, USA, 2013.

[152] A. Levant and M. Livne, “Weighted homogeneity and robustness of sliding mode control,” *Automatica*, vol. 72, pp. 186–193, 2016.

[153] S. Li, H. Du, and X. Lin, “Finite-time consensus algorithm for multi-agent systems with double-integrator dynamics,” *Automatica*, vol. 47, no. 8, pp. 1706–1712, 2011.

[154] H. Li and M. Yan, “Receding horizon control based consensus scheme in general linear multi-agent systems,” *Automatica*, vol. 56, pp. 12–18, 2015.

[155] Z. Li, W. Ren, X. Liu, and L. Xie, “Distributed consensus of linear multi-agent systems with adaptive dynamic protocols,” *Automatica*, vol. 49, no. 7, pp. 1986–1995, 2013.

[156] J. J. Hopfield and D. W. Tank, “ ‘Neural’ computation of decisions in optimization problems,” *Bio. Cybern.*, vol. 55, pp. 141–146, 1985.

[157] Y. Xia and J. Wang, "A general methodology for designing globally convergent optimization neural networks," *IEEE Trans. Neural Netw.*, vol. 9, no. 6, pp. 1331–1343, 1998.

[158] Q. Liu and J. Wang, "A one-layer recurrent neural network with a discontinuous hard-limiting activation function for quadratic programming," *IEEE Trans. Neural Netw.*, vol. 19, no. 4, 558–570, 2008.

[159] H. Che and J. Wang, "A two-timescale duplex neurodynamic approach to biconvex optimization," *IEEE Trans. Neural Netw. Learn. Syst.*, in press, doi: 10.1109/TNNLS.2018.2884788.

[160] Z. Peng, J. Wang, and W. Wang, "Distributed maneuvering of autonomous surface vehicles based on neurodynamic optimization and fuzzy approximation," *IEEE Trans. Control Syst. Technol.*, vol. 26, no. 3, pp. 1083–1090, 2018.

[161] Z. Peng and J. Wang, "Output-feedback path-following control of autonomous underwater vehicles based on an extended state observer and projection neural networks," *IEEE Trans. Syst., Man, Cybern., Syst.*, vol. 48, no. 4, pp. 535–544, 2018.

[162] X. Le and J. Wang, "A neurodynamic optimization approach to robust pole assignment for synthesizing linear control systems based on a convex feasibility problem reformulation," *Neural Inform. Process.*, pp. 284–291, 2013.

[163] J. Wang, "Analysis and design of a k-winners-take-all model with a single state variable and the heaviside step activation function," *IEEE Trans. Neural Netw.*, vol. 21, no. 9, pp. 1496–1506, 2010.

[164] L. Jin, S. Li, B. Hu, and M. Liu, "A survey on projection neural networks and their applications," *Appl. Soft Comput.*, vol. 76, pp. 533–544, 2019.

[165] T. Li and J. F. Zhang, “Consensus conditions of multi-agent systems with time-varying topologies and stochastic communication noises,” *IEEE Trans. Autom. Control*, vol. 55, no. 9, pp. 2043–2057, 2010.

[166] Y. J. Liu, Y. Gao, S. Tong, and Y. Li, “Fuzzy approximation-based adaptive back-stepping optimal control for a class of nonlinear discrete-time systems with dead-zone,” *IEEE Trans. Fuzzy Syst.*, vol. 24, no. 1, pp. 16–28, 2016.

[167] Y. J. Liu, Y. Gao, S. Tong, and C. L. P. Chen, “A unified approach to adaptive neural control for nonlinear discrete-time systems with nonlinear dead-zone input,” *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 27, no. 1, pp. 139–150, 2016.

[168] Y. J. Liu and S. Tong, “Barrier Lyapunov functions-based adaptive control for a class of nonlinear pure-feedback systems with full state constraints,” *Automatica*, vol. 64, pp. 70–75, 2016.

[169] K. Liu, G. Xie, W. Ren, and L. Wang, “Consensus for multi-agent systems with inherent nonlinear dynamics under directed topologies,” *Syst. Control Lett.*, vol. 62, pp. 152–162, 2013.

[170] C. Q. Ma, and J. F. Zhang, “Necessary and sufficient conditions for consensus-ability of linear multi-agent systems,” *IEEE Trans. Autom. Control*, vol. 55, no. 5, pp. 1263–1268, 2010.

[171] K. H. Movric and F. L. Lewis, “Cooperative optimal control for multi-agent systems on directed graph topologies,” *IEEE Trans. Autom. Control*, vol. 59, no. 3, pp. 769–774, 2014.

[172] R. Olfati-Saber and R. M. Murray, “Consensus problems in networks of agents with switching topology and time-delays,” *IEEE Trans. Autom. Control*, vol. 49, no. 9, pp. 1520–1533, 2004.

[173] H. Rezaee and F. Abdollahi, “Average consensus over high-order multiagent systems,” *IEEE Trans. Autom. Control*, vol. 60, no. 11, pp. 3047–3052, 2015.

[174] H. Rezaee and F. Abdollahi,“Adaptive stationary consensus protocol for a class of high-order nonlinear multiagent systems with jointly connected topologies,” *Int. J. Robust Nonlin. Control*, vol. 27, no. 9, pp. 1677–1689, 2016.

[175] H. L. Royden, *Real Analysis*. New York, USA: Macmillan, 1988.

[176] E. Semsar-Kazerooni and K. Khorasani, “Optimal consensus algorithms for cooperative team of agents subject to partial information,” *Automatica*, vol. 44, pp. 2766–2777, 2008.

[177] E. Semsar-Kazerooni and K. Khorasani, “Multi-agent team cooperation: A game theory approach,” *Automatica*, vol. 45, pp. 2205–2213, 2009.

[178] G. S. Seyboth, D. V. Dimarogonas, and K. H. Johansson, “Event-based broadcasting for multi-agent average consensus,” *Automatica*, vol. 49, no. 1, pp. 245–252, 2013.

[179] Q. Song, J. Cao, and W. Yu, “Second-order leader-following consensus of nonlinear multi-agent systems via pinning control,” *Syst. Control Lett.*, vol. 59, no. 9, pp. 553–562, 2010.

[180] W. Yu, W. Ren, W. X. Zheng, G. Chen, and J. Lü, “Distributed control gains design for consensus in multi-agent systems with second-order nonlinear dynamics,” *Automatica*, vol. 49, no. 7, pp. 2107–2115, 2013.

[181] X. Zhang, L. Liu, and G. Feng, “Leader-follower consensus of time-varying nonlinear multi-agent systems,” *Automatica*, vol. 52, pp. 8–14, 2015.

[182] H. Zhang, J. Zhang, G. H. Yang, and Y. Luo, “Leader-based optimal coordination control for the consensus problem of multiagent differential games via fuzzy adaptive dynamic programming,” *IEEE Trans. Fuzzy Syst.*, vol. 23, no. 1, pp. 152–163, 2015.

[183] L. Zhu and Z. Chen, “Robust homogenization and consensus of nonlinear multi-agent systems,” *Syst. Control Lett.*, vol. 65, pp. 50–55, 2014.

[184] Z. Zhang, A. Beck, and N. Magnenat-Thalmann, “Human-like behavior generation based on head-arms model for robot tracking external targets and body parts,” *IEEE Trans. Cybern.*, vol. 45, no. 8, pp. 1390–1400, Aug. 2015.

[185] L. Jin and Y. Zhang, “G2-Type SRMPC scheme for synchronous manipulation of two redundant robot arms,” *IEEE Trans. Cybern.*, vol. 45, no. 2, pp. 153–164, Feb. 2015.

[186] D. Guo and Y. Zhang, “A new inequality-based obstacle-avoidance MVN scheme and its application to redundant robot manipulators,” *IEEE Trans. Syst., Man, Cybern. C, Appl. Rev.*, vol. 42, no. 6, pp. 1326–1340, Nov. 2012.

[187] W. Xu, J. Zhang, B. Liang, and B. Li, “Singularity analysis and avoidance for robot manipulators with nonspherical wrists,” *IEEE Trans. Ind. Electron.*, vol. 63, no. 1, pp. 277–290, Jan. 2016.

[188] Y. Zhang, J. Wang, and Y. Xia, “A Dual neural network for redundancy resolution of kinematically redundant manipulators subject to joint limits and joint velocity limits,” *IEEE Trans. Neural Netw.*, vol. 14, no. 3, pp. 658–667, May 2003.

[189] Y. Zhang, J. Wang, and Y. Xu, “A dual neural network for bi-criteria kinematic control of redundant manipulators,” *IEEE Trans. Robot. Autom.*, vol. 18, no. 6, pp. 923–931, Dec. 2002.

[190] B. Liao and W. Liu, “Pseudoinverse-type bi-criteria minimization scheme for redundancy resolution of robot manipulators,” *Robotica*, vol. 33, no. 10, pp. 2100–2113, Dec. 2015.

[191] H. Ding and S. K. Tso, “Redundancy resolution of robotic manipulators with neural computation,” *IEEE Trans. Ind. Electron.*, vol. 46, no. 1 , pp. 230–233, Feb. 1999.

[192] C. A. Klein and C. H. Huang, “Review of pseudoinverse control for use with kinematically redundant manipulators,” *IEEE Trans. Syst., Man, Cybern.*, no. 2, pp. 245-50, Mar. 1983.

[193] A. Cherubini, R. Passama, A. Crosnier, A. Lasnier, and P. Fraisse, “Collaborative manufacturing with physical human-robot interaction,” *Robot. Cim-Int. Manuf.*, vol. 31, pp. 1-13, Aug. 2016.

[194] L. Xiao and Y. Zhang, “Acceleration-level repetitive motion planning and its experimental verification on a six-link planar robot manipulator,” *IEEE Trans. Control Syst. Techno.*, vol. 21, no. 3, pp. 906–914, May 2013.

[195] L. Jin, S. Li, H. M. La, and X. Luo, “Manipulability optimization of redundant manipulators using dynamic neural networks,” *IEEE Trans. Ind. Electron.*, vol. 64, no. 6, pp. 4710–4720, Jun. 2017.

[196] F.-T. Cheng, R.-J. Sheu, and T.-H. Chen, “The improved compact QP method for resolving manipulator redundancy,” *IEEE Trans. Syst., Man, Cybern.*, vol. 25, pp. 1521–1530, Nov. 1995.

[197] Z. Zhang, Z. Li, Y. Zhang, Y. Luo, and Y. Li, “Neural-dynamic-method-based dual-arm CMG scheme with time-varying constraints applied to humanoid robots,” *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 26, no. 12, pp. 3251–3262, Dec. 2015.

[198] Y. Zhang, S. S. Ge, and T. H. Lee, “A unified quadratic-programming-based dynamical system approach to joint torque optimization of physically constrained redundant manipulators,” *IEEE Trans. Syst., Man, Cybern. B, Cybern.*, vol. 34, no. 5, pp. 2126–2132, Oct. 2004.

[199] Z.-G. Hou, L. Cheng, and M. Tan, “Multicriteria optimization for coordination of redundant robots using a dual neural network,” *IEEE Trans. Syst., Man, Cybern. B, Cybern.*, vol. 40, no. 4, pp. 1075–1087, Aug. 2010.

[200] D. Chen and Y. Zhang, “A hybrid multi-objective scheme applied to redundant robot manipulators,” *IEEE Trans. Autom. Sci. Eng.*, vol. 14, no. 3, pp. 1337–1350, Jul. 2017.

[201] M. Li, Y. Li, S. S. Ge, and T. H. Lee, “Adaptive control of robotic manipulators with unified motion constraints,” *IEEE Trans. Syst., Man, Cybern. Syst.*, vol. 47, no. 1, pp. 184–194, Jan. 2017.

[202] H. Wang, “Adaptive control of robot manipulators with uncertain kinematics and dynamics,” *IEEE Trans. Autom. Control*, vol. 62, no. 2, pp. 948–954, Feb. 2017.

[203] F. Aghili, “Adaptive control of manipulators forming closed kinematic chain with inaccurate kinematic model,” *IEEE/ASME Trans. Mechatronics*, vol. 18, no. 5, pp. 1544–1554, Oct. 2013.

[204] C. C. Cheah, M. Hirano, S. Kawamura, and S. Arimoto, “Approximate Jacobian control for robots with uncertain kinematics and dynamics,” *IEEE Trans. Robot. Autom.*, vol. 19, no. 4, pp. 192–702, Aug. 2003.

[205] M. Shimizu, H. Kakuya, W.-K. Yoon, K. Kitagaki, and K. Kosuge, “Analytical inverse kinematic computation for 7-DOF redundant manipulators with joint limits and its application to redundancy resolution,” *IEEE Trans. Robot.*, vol. 24, no. 5, pp. 1131–1141, Oct. 2008.

[206] P. K. Patchaikani, L. Behera, and G. Prasad, “A single network adaptive critic-based redundancy resolution scheme for robot manipulators,” *IEEE Trans. Ind. Electron.*, vol. 59, no. 8, pp. 3241–3253, Aug. 2012.

[207] H. Wang, P. Shi, H. Li, and Q. Zhou, “Adaptive neural tracking control for a class of nonlinear systems with dynamic uncertainties,” *IEEE Trans. Cybern.*, vol. 47, no. 10, pp. 3075–3087, Oct. 2017.

[208] J. Na, Q. Chen, X. Ren, and Y. Guo, "Adaptive prescribed performance motion control of servo mechanisms with friction compensation," *IEEE Trans. Ind. Electron.*, vol. 61, no. 1, pp. 486–494, Jan. 2014.

[209] Z. Li, Z. Huang, W. He, and C.-Y. Su, "Adaptive impedance control for an upper limb robotic exoskeleton using biological signals," *IEEE Trans. Ind. Electron.*, vol. 64, no. 2, pp. 1664–1674, Feb. 2017.

[210] F. Jaramillo-Lopez, G. Kenne, and F. Lamnabhi-Lagarrigue, "Adaptive control for a class of uncertain nonlinear systems: application to photovoltaic control systems," *IEEE Trans. Autom. Control*, vol. 62, no. 1, pp. 393–398, Jan. 2017.

[211] J. Fang, J. Zhao, T. Mei, and J. Chen, "Online optimization scheme with dual-mode controller for redundancy-resolution with torque constraints," *Robot. Cim-Int. Manuf.*, vol. 40, pp. 44–54, Aug. 2016.

[212] Y. Zhang, S. Li, and X. Zhou, "Recurrent-neural-network-based velocity-level redundancy resolution for manipulators subject to a joint acceleration limit," *IEEE Trans. Ind. Electron.*, vol. 65, no. 5, pp. 3573–3582.

[213] Y. Zhang, S. Li, J. Gui, and X. Luo, "Velocity-level control with compliance to acceleration-level constraints: A novel scheme for manipulator redundancy resolution," *IEEE Trans. Ind. Inform.*, to be published, doi: 10.1109/TII.2017.2737363.

[214] M. W. Spong, S. Hutchinson, and M. Vidyasagar, *Robot Modeling And Control*. New York, USA: Wiley, 2006.

[215] S. Li, Y. Zhang, and L. Jin, "Kinematic control of redundant manipulators using neural networks," *IEEE Trans. Neurl Netw. Learn. Syst.*, vol. 28, no. 10, pp. 2243–2254, Oct. 2017.

[216] OptiTrack Motion Capture Systems, 2017. [Online]. Available: http://optitrack.com/

[217] S. S. M. Salehian, M. Khoramshahi, and A. Billard, "A dynamical system approach for softly catching a flying object: Theory and experiment," *IEEE Trans. Robot.*, vol. 32, no. 2, pp. 462–471, Apr. 2016.

[218] E. Davis and P. E. I. Pounds, "Direct sensing of thrust and velocity for a quadrotor rotor array," *IEEE Robot. Autom. Lett.*, vol. 2, no. 3, pp. 1360–1366, Jul. 2017.

[219] A. Wang, B. Mu, and Y. Shi, "Consensus control for a multi-agent system with integral-type event-triggering condition and asynchronous periodic detection," *IEEE Trans. Ind. Electron.,* vol. 64, no. 7, pp. 5629–5639, Jul. 2017.

[220] T. Bartelds, A. Capra, S. Hamaza, S. Stramigioli, and M. Fumagalli, "Compliant aerial manipulators: Toward a new generation of aerial robotic workers," *IEEE Robot. Autom. Lett*, vol. 1, no. 1, pp. 477–483, Jan. 2016.

[221] A. Ajoudani, N. G. Tsagarakis, and A. Bicchim, "Tele-impedance: Towards transferring human impedance regulation skills to robots," in *Proc. IEEE Int. Conf. Robot. Autom.*, pp. 382–388, 2012.

[222] G. P. R. Papini, M. Fontana, and M. Bergamasco, "Desktop haptic interface for simulation of hand-tremor," *IEEE Trans. Haptics*, vol. 9, no. 1, pp. 33–42, Jan. 2016.

[223] G. Du and P. Zhang, "Online serial manipulator calibration based on multisensory process via extended kalman and particle filters," *IEEE Trans. Ind. Electron.*, vol. 61, no. 12, pp. 6852–6859, Dec. 2014.

[224] J. Han, "From PID to active disturbance rejection control," *IEEE Trans. Ind. Electron.,* vol. 56, no. 3, pp. 900–906, Mar. 2009.

[225] A. Levant, "Robust exact differentiation via sliding mode technique," *Automatica*, vol. 34, no. 3, pp. 379–384, Mar. 1998.

[226] S. Boyd and L. Vandenberghe, *Convex Optimization*. England: Cambridge University Press, 2004.

[227] Y. Xia and G. Feng, "On convergence rate of projection neural networks," *IEEE Trans. Autom. Control*, vol. 49, no. 1, pp. 91–96, Jan. 2004.

[228] R. Bellman, *Introduction to Matrix Analysis*. Philadelphia, USA: Society for Industrial and Applied Mathematics, 1997.

[229] H. K. Khalil, *Nonlinear Systems*. New Jersey, USA: Prentice-Hall, 2002.

[230] X. Gao, "Exponential stability of globally projected dynamic systems," *IEEE Trans. Neural Netw.*, vol. 14, no. 2, pp. 426–431, Mar. 2003.

[231] V. Adetola and M. Guay, "Finite-time parameter estimation in adaptive control of nonlinear systems," *IEEE Trans. Autom. Control*, vol. 53, no. 3, pp. 807–811, Apr. 2008.

[232] R. Mishkov and S. Darmonsk, "Nonlinear adaptive control system design with asymptotically stable parameter estimation error," *Int. J. Control*, to be published, doi: 10.1080/00207179.2016.1276631.

[233] E.W. Bai and S.S. Sastry, "Persistency of excitation, sufficient richness and parameter convergence in discrete time adaptive control," *Syst. Contrl Lett.*, vol. 6, pp. 153–163, Aug. 1985.

[234] W. E. Dixon, D. M. Dawson, F. Zhang, and E. Zergeroglu, "Global exponential tracking control of a mobile robot system via a PE condition," *IEEE Trans. Syst., Man, Cybern. B, Cybern.*, vol. 30, no. 1, pp. 129–142, Feb. 2000.

[235] H. Modares, F. L. Lewis, and M. B. Naghibi-Sistani, "Integral reinforcement learning and experience replay for adaptive optimal control of partially-unknown constrained-input continuous-time systems," *Automatica*, vol. 50, no. 1, pp. 193–202, Jan. 2014.

[236] W. E. Dixon, “Adaptive regulation of amplitude limited robot manipulators with uncertain kinematics and dynamics,” *IEEE Trans. Autom. Control*, vol. 52, no. 3, pp. 488–493, Mar. 2007.

[237] C. C. Cheah, C. Liu, and J. J. E. Slotine, “Adaptive jacobian tracking control of robots with uncertainties in kinematic, dynamic and actuator models,” *IEEE Trans. Autom. Control*, vol. 51, no. 6, pp. 1024–1029, Jun. 2006.

[238] H. Wang and Y. Xie, “Passivity based adaptive Jacobian tracking for free-floating space manipulators without using spacecraft acceleration,” *Automatica*, vol. 45, pp. 1510–1517, Jun. 2009.