

Yinyan Zhang
Shuai Li
Xuefeng Zhou

Deep Reinforcement Learning with Guaranteed Performance

A Lyapunov-Based Approach

Studies in Systems, Decision and Control

Volume 265

Series Editor

Janusz Kacprzyk, Systems Research Institute, Polish Academy of Sciences,
Warsaw, Poland

The series “Studies in Systems, Decision and Control” (SSDC) covers both new developments and advances, as well as the state of the art, in the various areas of broadly perceived systems, decision making and control—quickly, up to date and with a high quality. The intent is to cover the theory, applications, and perspectives on the state of the art and future developments relevant to systems, decision making, control, complex processes and related areas, as embedded in the fields of engineering, computer science, physics, economics, social and life sciences, as well as the paradigms and methodologies behind them. The series contains monographs, textbooks, lecture notes and edited volumes in systems, decision making and control spanning the areas of Cyber-Physical Systems, Autonomous Systems, Sensor Networks, Control Systems, Energy Systems, Automotive Systems, Biological Systems, Vehicular Networking and Connected Vehicles, Aerospace Systems, Automation, Manufacturing, Smart Grids, Nonlinear Systems, Power Systems, Robotics, Social Systems, Economic Systems and other. Of particular value to both the contributors and the readership are the short publication timeframe and the world-wide distribution and exposure which enable both a wide and rapid dissemination of research output.

** Indexing: The books of this series are submitted to ISI, SCOPUS, DBLP, Ulrichs, MathSciNet, Current Mathematical Publications, Mathematical Reviews, Zentralblatt Math: MetaPress and Springerlink.

More information about this series at <http://www.springer.com/series/13304>

Yinyan Zhang · Shuai Li · Xuefeng Zhou

Deep Reinforcement Learning with Guaranteed Performance

A Lyapunov-Based Approach



Springer

Yinyan Zhang
College of Cyber Security
Jinan University
Guangzhou, China

Shuai Li
School of Information Science and
Engineering
Lanzhou University
Lanzhou, China

Xuefeng Zhou
Guangdong Institute of Intelligent
Manufacturing
Guangdong Academy of Science
Guangzhou, China

ISSN 2198-4182 ISSN 2198-4190 (electronic)
Studies in Systems, Decision and Control
ISBN 978-3-030-33383-6 ISBN 978-3-030-33384-3 (eBook)
<https://doi.org/10.1007/978-3-030-33384-3>

© Springer Nature Switzerland AG 2020

This work is subject to copyright. All rights are reserved by the Publisher, whether the whole or part of the material is concerned, specifically the rights of translation, reprinting, reuse of illustrations, recitation, broadcasting, reproduction on microfilms or in any other physical way, and transmission or information storage and retrieval, electronic adaptation, computer software, or by similar or dissimilar methodology now known or hereafter developed.

The use of general descriptive names, registered names, trademarks, service marks, etc. in this publication does not imply, even in the absence of a specific statement, that such names are exempt from the relevant protective laws and regulations and therefore free for general use.

The publisher, the authors and the editors are safe to assume that the advice and information in this book are believed to be true and accurate at the date of publication. Neither the publisher nor the authors or the editors give a warranty, expressed or implied, with respect to the material contained herein or for any errors or omissions that may have been made. The publisher remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.

This Springer imprint is published by the registered company Springer Nature Switzerland AG
The registered company address is: Gewerbestrasse 11, 6330 Cham, Switzerland

To our ancestors and parents, as always

Preface

In the past decades, both optimal control and adaptive control have been widely investigated to solve control problems of nonlinear systems arising from engineering applications. Optimal control aims at finding a control law to drive a control system to a desired state while optimizing certain performance index with or without constraints. Adaptive control is a tool to handle parameter uncertainty or structure uncertainty of control systems. In most works, the two types of control methods are separated. Reinforcement learning and, in particular, deep reinforcement learning have attracted more and more research interest in recent years. Such type of learning methods could be a powerful tool for the control of nonlinear systems.

In this book, we present our systematic investigations on a near-optimal adaptive control method based on the Taylor expansion, neural networks, estimator design approaches, and the idea of sliding mode control. We mainly handle the tracking control problem of nonlinear systems under different scenarios, for which the output of a controlled nonlinear system is expected to track a desired reference output trajectory with respect to time t . The performance of the presented approach is theoretically guaranteed. Specifically, the performance index is asymptotically convergent to the optimal and the tracking error is asymptotically convergent to 0. Issues like actuator saturations are also considered in this book. By combining the design method of the near-optimal adaptive control method and the novel ideas in the developments of methods for the redundancy resolution of redundant manipulators, two new redundancy resolution methods are also presented.

To make the contents clear and easy to follow, in this book, each part (and even each chapter) is written in a relatively self-contained manner.

This book is divided into the following seven chapters.

Chapter 1—In this chapter, a brief review is provided about recent advancements in the field of optimal control.

Chapter 2—In this chapter, a time-scale expansion-based scheme is presented for approximately solving the optimal control problem of continuous-time underactuated nonlinear systems subject to input constraints and system dynamics. By time-scale Taylor approximation of the original performance index, the optimal control problem is relaxed into an approximated optimal control problem. Based on

the system dynamics, the problem is further reformulated as a quadratic program, which is solved by a projection neural network. Theoretical analysis on the closed-loop system synthesized by the controlled system and the projection neural network is conducted, which reveals that, under certain conditions, the closed-loop system possesses exponential stability and the original performance index converges to zero as time tends to infinity.

Chapter 3—In this chapter, a unified online adaptive near-optimal control framework is presented 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. 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 presented adaptive near-optimal control law asymptotically converges to the optimal. The efficacy of the presented framework and the theoretical results are validated by an application to underactuated surface vessels.

Chapter 4—In this chapter, an adaptive near-optimal control law, which is inherently real time, is designed to tackle the contradiction 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 presented adaptive 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 presented 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 presented method to a van der Pol oscillator are presented to validate the efficacy of the presented adaptive 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 presented method.

Chapter 5—In this chapter, the receding-horizon near-optimal tracking control problem about a class of continuous-time nonlinear systems with fully unknown dynamics is considered. A novel model-free adaptive near-optimal control method is presented to solve this problem via utilizing the Taylor expansion-based problem relaxation, the universal 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 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 adaptive near-optimal control 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 asymptotically converges to optimal. An illustrative example and experimental results are presented, which substantiate the efficacy of the presented method and verify the theoretical results.

Chapter 6—In this chapter, an adaptive projection neural network (PNN) with online learning is presented for the redundancy resolution of manipulators with unknown physical parameters, which tackles the dilemmas in existing methods. The presented 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 presented neural network. Besides, simulations based on a PUMA 560 manipulator with unknown physical parameters together with the comparison of an existing PNN substantiate the efficacy and superiority of the presented neural network and verify the theoretical results.

Chapter 7—In this chapter, a novel recurrent neural network is presented to simultaneously address the periodic input disturbance, joint angle constraint, and joint velocity constraint, and optimize a general quadratic performance index. The presented recurrent neural network applies to both regulation and tracking tasks. Theoretical analysis shows that, with the presented neural network, the end-effector tracking and regulation errors asymptotically converge to zero in the presence of both input disturbance and the two constraints. Simulation examples and comparisons with an existing controller are also presented to validate the effectiveness and superiority of the presented controller.

In summary, this book mainly presents methods and algorithms for the near-optimal adaptive control of nonlinear systems, together with the corresponding theoretical analysis and simulative examples. Based on these methods, two new methods for the redundancy resolution of redundant manipulators with consideration of parameter uncertainty and periodic disturbances are also presented. This book is written for graduate students and academic and industrial researchers in the field of adaptive/optimal control, robotics, and dynamic neural networks. We hope that this book will benefit the readers and could give them some inspirations in related fields. In particular, we hope that the results presented in this book could help develop deep reinforcement learning approaches for the control of nonlinear systems with performance guarantee through Lyapunov-based approaches, for which we have the current title of this book.

Any comments or suggestions are welcome, and the authors can be contacted via e-mails: zhangyinyan12@163.com (Yinyan Zhang), shuaili@ieee.org (Shuai Li), and xuefengzhou@vip.qq.com (Xuefeng Zhou).

Guangzhou, China
Lanzhou, China
Guangzhou, China
August 2019

Yinyan Zhang
Shuai Li
Xuefeng Zhou

Acknowledgements

During the work on this book, we have had the pleasure of discussing its various aspects and results with many cooperators and students. We highly appreciate their contributions, which help improve the quality and presentation of this book.

Guangzhou, China
Lanzhou, China
Guangzhou, China
August 2019

Yinyan Zhang
Shuai Li
Xuefeng Zhou

Contents

1 A Survey of Near-Optimal Control of Nonlinear Systems	1
1.1 Introduction	1
1.2 Dynamic Programming	3
1.2.1 Example for Continuous-Time Nonlinear System	3
1.2.2 Example for Discrete-Time Nonlinear System	4
1.2.3 Adaptive Dynamic Programming for Approximately Solving HJB Equations	5
1.2.4 Numerical Methods Without NNA for Approximately Solving HJB Equations	7
1.3 Near-Optimal Control via Nonlinear Programming	8
1.3.1 Methods via Control Parameterization	8
1.3.2 Model Predictive Control	9
1.3.3 Method via Taylor Expansion	10
1.4 Summary	13
References	14
2 Near-Optimal Control with Input Saturation	21
2.1 Introduction	21
2.2 Preliminary and Notations	23
2.3 Problem Formulations	25
2.3.1 Original Formulation	25
2.3.2 Reformulation	26
2.4 Projection Neural Network Design	27
2.5 Theoretical Analysis	29
2.6 Numerical Investigation	34
2.6.1 Performance Comparison for Manipulator Control	35
2.6.2 Application to Underactuated Ships	37
2.7 Questions and Answers	41
2.8 Summary	44
References	44

3 Adaptive Near-Optimal Control with Full-State Feedback	49
3.1 Introduction	49
3.2 Preliminary	51
3.3 General Linear Systems	52
3.3.1 Problem Formulation	52
3.3.2 Nominal Design	53
3.3.3 Adaptive Design	55
3.4 Extension to Nonlinear Systems	56
3.4.1 Problem Formulation	56
3.4.2 Nominal Design	58
3.4.3 Adaptive Design	59
3.4.4 Computational Complexity Analysis	63
3.5 Theoretical Results	64
3.5.1 Convergence of Auxiliary Systems	64
3.5.2 Stability of Closed-Loop Systems	66
3.5.3 Asymptotic Optimality	69
3.6 Application to Uncertain Underactuated Surface Vessel	72
3.6.1 Without Measurement Noises	72
3.6.2 With Measurement Noises	74
3.6.3 Capability for Real-Time Control	75
3.7 Questions and Answers	79
3.8 Summary	92
References	92
4 Adaptive Near-Optimal Control Using Sliding Mode	97
4.1 Introduction	97
4.2 Problem Formulation and Preliminary	99
4.3 Nominal Near-Optimal Design	100
4.4 Adaptive Near-Optimal Design	101
4.5 Illustrative Examples	107
4.5.1 Example 1	108
4.5.2 Example 2	111
4.6 Application to van der Pol Oscillator	112
4.7 Experimental Validation	113
4.8 Questions and Answers	119
4.9 Summary	125
References	125
5 Model-Free Adaptive Near-Optimal Tracking Control	129
5.1 Introduction	129
5.2 Problem Description and Preliminary	131
5.2.1 Problem Description	131
5.2.2 Sigmoid Neural Network	133
5.2.3 Problem Reformulation	134

5.3	Control Design	134
5.3.1	Problem Relaxation	135
5.3.2	Reconstruction of Input-to-Output Dynamics	137
5.3.3	Adaptive Near-Optimal Control Law	139
5.4	Theoretical Analysis	142
5.4.1	Confirmation of No Singularity Problems	142
5.4.2	Convergence of the Auxiliary System	144
5.4.3	Stability of the Closed-Loop System	147
5.4.4	Asymptotic Optimality of Performance Index	149
5.5	Illustrative Example	151
5.6	Experimental Validation	152
5.7	Questions and Answers	157
5.8	Summary	160
	References	161
6	Adaptive Kinematic Control of Redundant Manipulators	167
6.1	Introduction	167
6.2	Preliminary and Problem Formulation	169
6.2.1	Forward Kinematics	169
6.2.2	QP-Type Problem Formulation	170
6.3	Nominal Design	171
6.4	Adaptive Design	172
6.4.1	Adaptive Projection Neural Network	172
6.4.2	Theoretical Analysis	175
6.5	Simulative Verifications and Comparisons	179
6.5.1	PUMA 560 Description	179
6.5.2	Minimum-Velocity-Norm Redundancy Resolution	180
6.5.3	Repetitive-Motion Redundancy Resolution	185
6.6	Experimental Verification	186
6.7	Questions and Answers	188
6.8	Summary	192
	References	195
7	Redundancy Resolution with Periodic Input Disturbance	199
7.1	Introduction	199
7.2	Preliminary and Problem Description	201
7.2.1	Manipulator Kinematics Model	201
7.2.2	Problem Description	201
7.2.3	Problem Reformulation as Quadratic Program	203
7.3	Neural Network Design	204
7.3.1	Step 1: Nominal Design Without Disturbance	204
7.3.2	Step 2: Modified Controller with Disturbance Rejection	206

7.4	Simulation Examples and Comparisons	209
7.4.1	Simulation Setup	210
7.4.2	End-Effector Regulation	210
7.4.3	End-Effector Tracking	215
7.5	Questions and Answers	221
7.6	Summary	222
	References	222

Abbreviations

ADP	Adaptive dynamic programming
APNN	Adaptive projection neural network
D-H	Denavit–Hartenberg
DOF	Degrees of freedom
FPGA	Field-programmable gate array
HJB	Hamilton–Jacobi–Bellman
MPC	Model predictive control
NNA	Neural network approximation
PE	Persistent excitation
PID	Proportional–integral–derivative
PNN	Projection neural network
PSO	Particle swarm optimization
QP	Quadratic program

Chapter 1

A Survey of Near-Optimal Control of Nonlinear Systems



Abstract For nonlinear dynamical systems, the optimal control problem generally requires solving a partial differential equation called the Hamilton–Jacobi–Bellman equation, of which the analytical solution generally cannot be obtained. However, the demand for optimal control keeps increasing, with the goal to save energy, reduce transient time, minimize error accumulation, etc. Consequently, methods were reported to approximately solve the problem leading to the so-called near-optimal control although their technical details differ. This research direction has experienced great progress in recent years but a timely review of them is still missing. This chapter serves as a brief survey for existing methods in this research direction.

1.1 Introduction

Optimal control is a control methodology aiming at finding a control policy that optimizes a given performance index. The classical optimal control theory is on the basis of the Pontryagin's maximum principle or the dynamic programming principle [1]. The former provides a necessary condition for optimality and often gives an open-loop control law while the latter provides a sufficient condition by solving a so-called Hamilton–Jacobi–Bellman (HJB) equation, which is a partial differential equation and its analytical solution is difficult to derive for nonlinear dynamical systems. The optimal control theory for linear dynamical systems are in general well established with a strong emphasis on the linear quadratic regulator and the linear quadratic tracking [2–7]. Kindly note that for linear dynamical systems, the HJB equation degrades to the Riccati equation, which is much easier to handle. Almost all linear optimal control problems for systems with fully known dynamics have easily computable solutions [5]. However, in practice, control systems of interest are more or less nonlinear [8–11]. As a result, a systematic and implementable approach to achieve the optimal or near-optimal control of nonlinear dynamical systems is demanding.

In the past two decades, many results have been reported about the optimal control of nonlinear systems, including finite-horizon optimal control and infinite-horizon optimal control. Because the analytical solution of an HJB equation generally cannot

be obtained, most of the existing methods focus on finding an approximate solution for it, e.g., [12], although the assumptions or scenarios for the investigated problems may be different. There are other methods that try to relax the original optimal control problem and then find an analytical control law, e.g., [13], which also provide an approximate solution to the original problem. As a result, for nonlinear dynamical systems, in general, only near-optimal control can be realized in practice.

There are surveys for some specific methods aiming at the near-optimal regulation or tracking control of dynamical systems with nonlinearity. For example, in 2005, Bertsekas [14] presented a survey for dynamic programming and suboptimal control for discrete-time control systems with a focus on adaptive dynamic programming (ADP) and model predictive control (MPC). In 2014, Mayne [15] gave a survey for the developments of MPC, which is also referred to as receding horizon control. MPC is a control method widely used in process control, which generates control policies via solving a constrained discrete-time optimal control problem online. Surveys for MPC were also reported in [16–18]. In the past decades, reinforcement learning based optimal control, which is often referred to ADP, is also widely investigated. This method utilizes neural networks as a tool to approximate the solution of the corresponding HJB equation. Surveys for the developments of ADP in the past two decades were given in [19–21] although with different perspectives. It should be noted that, although there are a large number of publications about MPC and ADP, there are also some other methods for solving the problem, such as [12, 22, 23]. While many results have been reported about the near-optimal control of nonlinear dynamical systems, no systematic survey has been reported on its developments in the past two decades. Such a survey, however, is very important for new comers or interested researchers to have a general understanding or knowledge about this field.

An important new trend in the development of optimal control could be the synthesis with neural networks, which is substantiated by the related publications in the past two decades [24–26]. The universal approximation property makes neural networks suitable to learn any smooth function [27]. As a result, for traditional optimal control, which deals with the case with fully known systems' dynamics, the solution of the HJB equation can be learned by neural networks. Specifically, we can first assume that the solution can be represented by a neural network with known structure and connections but unknown weights. Then, we can apply some classical methods to learn the weights of the neural network. This kind of idea is used in ADP. Meanwhile, with neural networks, we can also deal with system uncertainty, leading to the development of the so-called adaptive optimal control or even model-free optimal control [28–31].

In light of the above observations, this chapter aims at providing a brief survey for the major developments in the near-optimal control for nonlinear systems during the past two decades. Note that, the Pontryagin's maximum principle and stochastic optimal control will not be covered. Specifically, we focus on dynamic programming based methods and nonlinear programming based methods for realizing the near-optimal control of nonlinear dynamical systems.

1.2 Dynamic Programming

In this section, we first provide a basic optimal control problem about two types of nonlinear dynamical systems [1] to illustrate the HJB equations given by the dynamic programming principle. Although the two examples are basic, they are thought to be competent to show the difficulties in this field of research. Then, we discuss some near-optimal control methods on the basis of approximately solving the HJB equations.

1.2.1 Example for Continuous-Time Nonlinear System

As an example, consider the following deterministic continuous-time nonlinear affine dynamical system:

$$\dot{x}(t) = f(x(t)) + g(x(t))u(t), \quad (1.1)$$

in which $x \in \mathbb{R}^n$ denotes the state variable and $u \in \mathbb{R}^m$ denotes the input variable; $\dot{x} = dx/dt$ with t denoting the time instant. It is generally assumed that (1) $f(0) = 0$; (2) function $f(x(t)) + g(x(t))u(t)$ is locally Lipschitz; (3) system (1.1) is stabilizable.

The optimal control problem of dynamical system (1.1) can be classified according to the corresponding performance index, which is also called cost function. For example, the infinite-horizon optimal regulation problem in general is concerned with the minimization of the following cost function [32]:

$$\begin{aligned} J(x, u) &= \int_0^\infty r(x(\tau), u(\tau))d\tau \\ &= \int_0^\infty (Q(x(\tau)) + u^T(\tau)Ru(\tau))d\tau, \end{aligned}$$

in which $Q(x) \in \mathbb{R}$ is positive definite and $R \in \mathbb{R}^{m \times m}$ is symmetric and positive definite. An important requirement here is that the input should be admissible so as to guarantee a finite cost function [28]. The value function is

$$V(x, u) = \int_t^\infty (Q(x(\tau)) + u^T(\tau)Ru(\tau))d\tau,$$

which is equivalent to

$$r(x, u) + \frac{\partial V^T}{\partial x}(f(x) + g(x)u) = 0, \quad V(0) = 0.$$

Consequently, the optimal value function can be determined via solving a continuous-time HJB equation given as follows:

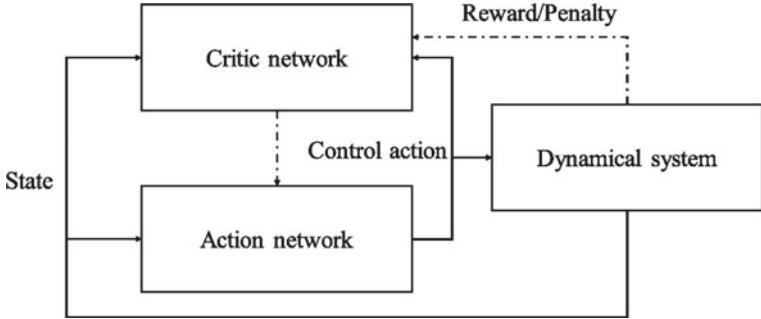


Fig. 1.1 Basic structure of ADP for the near-optimal control of nonlinear dynamical systems

$$r(x, u^*) + \left(\frac{\partial V^*}{\partial x} \right)^T (f(x) + g(x)u^*) = 0$$

by which the corresponding optimal control law is

$$\begin{aligned} u^* &= \underset{u}{\operatorname{argmin}} \left\{ r(x, u^*) + \left(\frac{\partial V^*}{\partial x} \right)^T (f(x) + g(x)u^*) \right\} \\ &= -\frac{1}{2} R^{-1} g^T(x) \frac{\partial V^*}{\partial x}. \end{aligned}$$

1.2.2 Example for Discrete-Time Nonlinear System

Corresponding to system (1.1), the deterministic discrete-time nonlinear affine dynamical system is given as follows:

$$x_{k+1} = f(x_k) + g(x_k)u_k, \quad (1.2)$$

where $k = 0, 1, 2, \dots$ denotes the sampling index. The assumptions for system (1.1) are also adopted here. The infinite-horizon optimal regulation problem for system (1.2) is in general concerned with the minimization of the following cost function:

$$J(x, u) = \sum_{i=0}^{\infty} r(x(i), u(i)) = \sum_{i=0}^{\infty} (Q(x(k)) + u^T(k) R u(k)).$$

The value function is then given as

$$V(x(k), u(k)) = \sum_{i=k}^{\infty} (Q(x(i)) + u^T(i) R u(i)),$$

which is equivalent to

$$V(x(k), u(k)) = r(x(k), u(k)) + V(x(k+1), u(k+1)).$$

By the principle of dynamic programming, the optimal value function is given via a discrete-time HJB equation as follows:

$$V^*(x(k), u(k)) = \min_u \{r(x(k), u(k)) + V(x(k+1), u(k+1))\}$$

by which we have

$$\begin{aligned} u^*(x(k)) &= \operatorname{argmin}_u \{r(x(k), u(k)) + V(x(k+1), u(k+1))\} \\ &= -\frac{1}{2} R^{-1} g^T(x(k)) \frac{\partial V^*(x(k+1))}{\partial x(k+1)}. \end{aligned}$$

From the above two examples, it can be found that the HJB equations is important to the optimal control of nonlinear dynamical systems. Unfortunately, they are partial differential equations, of which the analytical solutions are generally difficult and even impossible to derive for systems of interest in practice. It should be noted here that the cost functions can be different from those in the above two examples and there can be other constraints in the problem formulations.

Since the explicit solution of the HJB equations are difficult to derive, an intuitive way is to provide approximate solutions for them. In the next subsection, the near-optimal control methods through approximately solving the HJB equations is discussed.

1.2.3 Adaptive Dynamic Programming for Approximately Solving HJB Equations

ADP is a neural network-based method for the near-optimal control of nonlinear dynamical systems on the basis of the universal approximation property of neural networks and dynamic programming. A basic structure of ADP is shown in Fig. 1.1. It mainly consists of the critic network and the action network. The two networks are parameterized, for which the weights need to be trained in either online or off-line manner. The former is utilized to evaluate the current control action based on the performance index, which provides guidance for control action improvement. The latter is used to implement the control action. In other words, the critic network serves as a tool to realize the approximation of the value function while the other is adopted to achieve the approximation of the optimal control policy. This idea is natural as

in the HJB equations there are two unknown functionals. One is the value function and the other is the control policy. When the system dynamics is unknown, another network called the identification network can be included. The additional network estimates the system dynamics based on available measurement data.

The development of ADP for nonlinear dynamical systems in the past 20 years could be discussed from three perspectives, i.e., control structures, problems of interest, and applications.

There are several control structures of ADP. According to the design of the critic network, they could be roughly classified into the following categories, heuristic dynamic programming (HDP) [33–42], dual heuristic dynamic programming (DHDP) [26, 43–48], globalized dual heuristic dynamic programming (GDHDP) [49–53], and the action-dependent extensions of the above [54–56]. In HDP, control actions are generated by the action network and the value function is approximated by the critic network. For DHDP, the core is the usage of the critic network as a tool to realize the approximation of the derivatives of the value function regarding state vectors. GDHP is a novel synthesis of HDP and DHP, for which the critic network is adopted to realize the approximation of the value function as well as the derivatives. It should be noted that while all the structures are able to generate the near-optimal control policy, the computational efficiency and the value of the performance index could be different.

Recent developments about ADP for nonlinear dynamical systems can also be discussed by the problems of interest. For the optimal regulation problem, Dierks et al. [57] proposed a heuristic dynamic programming scheme for affine nonlinear discrete-time dynamical systems without explicitly knowing the system dynamics. The scheme consists of online system identification and off-line training for control law parameters. The optimal regulation problem was also investigated by using other ADP methods with different control structures and assumptions [58–61]. For example, Wei and Liu [58] proposed a novel ADP scheme for a class of general discrete-time nonlinear systems with the initial state being not fixed, which can guarantee the convergence to the optimum with an arbitrary initial admissible control sequence. Wang et al. [61] proposed an ADP method for the near-optimal regulation of unknown continuous-time affine nonlinear systems with disturbance attenuation. The approximate optimal tracking control of nonlinear dynamical systems under certain assumptions was also addressed by ADP [36–38, 40, 42, 47, 62, 63]. In particular, Zhang et al. [62] proposed an ADP method for robust state tracking of general nonlinear continuous-time systems, for which it is theoretically guaranteed that the input to the system is near-optimal. Recently, event-triggered ADP was also proposed [39, 64–66]. In event-triggered ADP, the update of control actions depends on a certain condition called the triggering condition. When the condition is satisfied, the update of control actions is executed. As a result, it can save the computation cost and transmission load. Note that ADP generally requires using neural networks, such as all the ADP literature mentioned above, for which the computational burden is still high when it is applied to systems with high-dimensional state spaces. In [68], a novel ADP scheme is developed for an uncertain continuous

-time affine nonlinear dynamical system, which does not require the approximation using neural networks and guarantees suboptimal stabilization via online learning.

In the past two decades, some applications of ADP were reported, which include power systems [67], air-breathing hypersonic vehicles [69], wearable robots [70], inverted pendulum systems [71], and turbogenerators [72].

1.2.4 Numerical Methods Without NNA for Approximately Solving HJB Equations

In the past two decades, for nonlinear dynamical systems, numerical methods without using the neural network approximation (NNA) were also proposed for approximately solving HJB equations to achieve near-optimal control. Beard et al. [73] proposed to approximately solving the HJB equation via successive Galerkin approximations. Chen et al. [74] investigated an infinite-time nonlinear quadratic optimal control problem and presented a method to calculate a near-optimal solution via solving a Riccati equation and a series of algebraic equations. Song et al. [75] proposed a finite difference based numerical algorithm with sigmoidal transformation for approximately solving the HJB equation associated with the receding-horizon optimal control of nonlinear dynamical systems. The method of Fakharian et al. [76] relies on the Adomian decomposition method. Nik et al. [77] proposed an analytical approximate solution for the HJB equation by using He's polynomials and homotopy perturbation. To solve the problem, a modified homotopy perturbation method was proposed in [78] by using the Padé approximant, which outperforms the method in [77]. Zhu [79] proposed a computational method for solving the HJB equation associated with the optimal control of affine nonlinear systems. Govindarajan et al. [80] proposed a sparse collocation method to calculate a near-optimal solution in the context of the optimal control of continuous-time nonlinear systems with a finite-horizon integral cost functions. Sun and Guo [81] proposed an upwind finite-difference scheme for approximately solving the HJB equation and presented the convergence result. In [82], an iterative method was proposed to solve a generalized HJB equation (GHJB) which is an approximation of the HJB equation in the optimal control of nonlinear systems. The method converts the GHJB equation to a set of nonlinear equations. The linearization of the nonlinear equations are employed to obtain the solutions and a good initial control guess is required. Michailidis et al. [83] proposed a novel cognitive adaptive optimization based method to approximately solve the HJB equation for the adaptive optimal control of large-scale nonlinear systems. In their case, the smooth function $f(x)$ in the system dynamics is unknown. The method does not need a parameterized model for the system dynamics and it optimizes the control action and the corresponding value function in real time. Although many numerical algorithms were proposed in the past two decades to obtain the approximate solution of the HJB equations, they are subject to certain assumptions on the system forms and the problems of interest. According to the literature, no

accurate and computational efficient numerical method exists in terms of approximately solving HJB equations regardless of the problem formulations.

1.3 Near-Optimal Control via Nonlinear Programming

As seen from the problem formulations, optimal control problems of nonlinear dynamical systems are essentially optimization problems. Thus, some results were also reported in the existing literature about approximately solving optimal control problems from the perspective of nonlinear programming. Specifically, some treatments are adopted to approximate the control variables, states, performance index, or even the continuous-time system dynamics, by which the original problems are relaxed to corresponding nonlinear programming problems.

1.3.1 *Methods via Control Parameterization*

In this subsection, we discuss some numerical methods that approximately solve optimal control problems of a nonlinear dynamical system via approximating states or control variables.

In the direct shooting methods, which include the direct multiple shooting method and the direct single shooting method, an optimal control problem is relaxed to a finite-dimensional nonlinear programming problem through control approximation and state approximation [84–87]. The discretizations are conducted by defining a shooting grid and the solution continuity is guaranteed via adding matching conditions. Its performance is affected by the number of grid points, the location of the shooting points, and the type of the base functions. It should be noted that the resultant nonlinear optimization problems are often large-scale. In direct collocation methods, trial functions are used to approximate control variables and states of the controlled system while system dynamics and constraints are collocated at some points within the solution domain. Among them, pseudospectral methods received much attention [88–93]. It has been recognized that compared with the shooting method and the traditional direct collocation methods, pseudospectral methods have higher accuracy, lower sensitivity to initial value, and faster convergence [93, 94]. In pseudospectral methods, the problem is first relaxed to a nonlinear programming problem via parameterizing the states and control variables by the usage of global polynomials together with collocating differential algebraic equations [88–93]. Based on the dual variational principle, Peng et al. [95] proposed an adaptive symplectic pseudospectral method. According to the method, an original optimal control problem is converted into several nonlinear equations. The equations were solved by Newton–Raphson iteration methods. Tabrizidooz et al. [96] proposed a composite pseudospectral method. Ge et al. [97] proposed to address an optimal attitude control problem of coupled-rigid-body spacecrafts by using the Chebyshev

–Gauss pseudospectral method, in which the barycentric Lagrange interpolation was used when approximating states and control variables. The pseudospectral method has been applied in energy systems [98, 99].

1.3.2 Model Predictive Control

In this subsection, we discuss MPC. At each sampling instant, the control action is the first one of a sequence of control actions calculated through solving a discrete finite-horizon open-loop optimal control problem based on the current state of the control plant [15, 100]. It should be noted that the optimization problem is generally convex for MPC when a linear system dynamics is adopted to predict the system behavior, while in nonlinear MPC (i.e., when the nonlinear system dynamics is directly used to predict the system behavior) the problem is generally not convex, which makes the optimization problem difficult to solve [101, 102]. By performing an online open-loop optimal control calculation based on the state feedback, i.e., online solving the corresponding optimization problems, MPC does not rely on (approximately) solving the HJB equation [16]. As a result, the computational efficiency of MPC is better than that of traditional optimal control methods. Meanwhile, MPC can deal with both state and input constraints. However, as MPC requires solving a large-scale nonlinear optimization problem at each sampling instant when it is applied to nonlinear dynamical systems, it is still computational intensive for control systems with a high sampling frequency. This is also the reason why MPC found most of its applications in petrochemical and related industries. One of the main topics in the recent developments of MPC for nonlinear systems is how to enhance the computational efficiency so as to make it applicable to systems with a high control frequency [103–105]. An intuitive method is to approximate the nonlinear system dynamics by linearization, by which one can solve the problem under the framework of MPC for linear dynamical systems [104]. Evolutionary computation methods were also applied to solve the problem. Xu et al. [105] proposed a high-efficiency hardware implementation method of MPC based on field-programmable gate array (FPGA), which employs a particle swarm optimization (PSO) algorithm to solve the nonlinear optimization problem. Mozaffari et al. [106] proposed a modified dynamic PSO algorithm for enhancing the computational efficiency of the MPC for nonlinear dynamical systems, where the reproduction and aging operators were removed so as to find near-optimal solutions in a very short period. The recurrent neural network approach was also reported for computationally efficient MPC for nonlinear dynamical systems [107–111]. This approach, in general, first converts the MPC problem into a constrained quadratic program and then solves it via recurrent neural networks. In [110], a robust MPC method was proposed for nonlinear dynamical systems with unmodeled dynamics and bounded uncertainties. The problem was solved by using extreme learning machine for system identification and recurrent neural networks for solving the resultant minimax optimization problem. The enhancement of MPC via neural networks has been widely investigated in

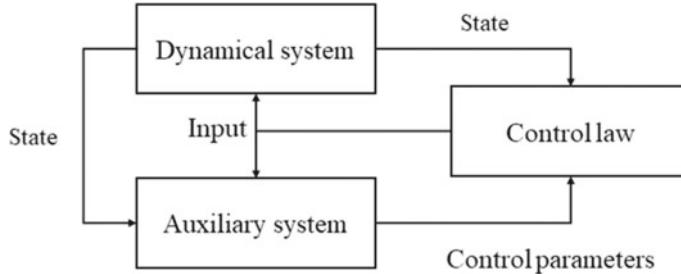


Fig. 1.2 Control structure of adaptive Taylor expansion-based near-optimal control via the auxiliary system approach

recent years. Cheng et al. [112] proposed a nonlinear model predictive control method aiming at piezoelectric actuators, where multilayer neural networks were used for system identification. As the MPC approach relies on the system model to predict the system behavior, proper system modeling is required. However, in practice, there are some phenomena or effects that are difficult to model. As a result, efforts have been devoted to enhance the robustness of MPC, leading to the so-called robust MPC [113–117]. There are generally two approaches to achieve robust MPC. One is to use a min-max objective function. In particular, Liu et al. [117] proposed a self-triggered robust MPC approach with a min-max objective function, for which the next triggering time is precomputed based on current system states and the dynamics of the controlled nonlinear dynamical system. The other is to add constraints in the MPC formulation to guarantee robustness [116]. The combination of MPC with another robust control method called sliding mode control was also reported [118]. MPC has been applied to the control of several nonlinear physical systems, such as unmanned aerial vehicles [119, 120], polystyrene batch reactors [121], overhead cranes [122], inverted pendulums [123], spark-ignition engines [124], DC/DC converters [125], and articulated unmanned ground vehicles [126].

1.3.3 Method via Taylor Expansion

There is also a Taylor expansion-based method suitable for the receding-horizon optimal control of continuous-time affine nonlinear systems [23, 127]. This method is also referred to as explicit MPC since a closed-form control law is derived and it also relies on predicting the system behavior via the system dynamics. The advantage of the method lies in the computational efficiency. Unlike MPC discussed above, the discretization for the system dynamics is not adopted in the Taylor expansion-based method. The main feature of this method is the usage of the time-scale Taylor expansion to the variables in the performance index, which makes it possible to relax the problem into a quadratic program, and, as a result, the explicit solution can be

directly obtained when there is no constraint. The order of the Taylor expansion depends on the relative degree of the system.

Here, we give an example of the Taylor expansion-based method [13]. The details of this method will be discussed in latter chapters. Consider the following receding-horizon optimal tracking control problem with respect to an affine nonlinear dynamical system:

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

where $J(t)$ denoting the performance index is given as follows:

$$J(t) = \int_0^T (y_d(t + \tau) - y(t + \tau))^T Q (y_d(t + \tau) - y(t + \tau)) d\tau, \quad (1.3)$$

in which the constant $T > 0 \in \mathbb{R}$ denotes the prediction period; the parameter matrix Q in the performance index is symmetric positive definite; the desired output function denoted by $y_d(t)$ with respect to time t is continuously differentiable. By using the notations of Lie derivatives and the definition of relative degree [13], if the relative degree of the controlled system is ρ , we have

$$\begin{cases} \dot{y}(t) = L_f h(x(t)), \\ \vdots \\ y^{[\rho-1]}(t) = L_f^{\rho-1} h(x(t)), \\ y^{[\rho]}(t) = L_f^\rho h(x(t)) + L_g L_f^{\rho-1} h(x(t)) u(t), \end{cases}$$

with $y^{[i]}(t)$ denoting the i th-order derivative of $y(t)$, for $i = 1, 2, \dots, \rho$. Let $Y(t) = [y(t), L_f h(x(t)), \dots, L_f^{\rho-1} h(x(t)), L_f^\rho h(x(t))]$. With the aid of the time-scale Taylor expansion, output $y(t + \tau)$ is predicted as follows by using the information at time instant t :

$$y(t + \tau) \approx Y(t)w(\tau) + \frac{\tau^\rho}{\rho!} L_g L_f^{\rho-1} h(x(t)) u(t),$$

in which $w(\tau) = [1, \tau, \dots, \tau^{\rho-1}/(\rho-1)!, \tau^\rho/\rho!]^T$. Similarly, $y_d(t + \tau) \approx Y_d(t)w(\tau)$, where $Y_d(t) = [y_d(t), \dot{y}_d(t), \dots, y_d^{[\rho]}(t)]$. As a result, for $J(t)$ in (1.3), we have

$$\begin{aligned} J(t) & \approx \hat{J}(t) \\ & = \int_0^T \left(E(t)w(\tau) - \frac{\tau^\rho}{\rho!} L_g L_f^{\rho-1} h(x(t)) u(t) \right)^T Q \left(E(t) \right) \end{aligned}$$

$$\begin{aligned}
& \times w(\tau) - \frac{\tau^\rho}{\rho!} L_g L_f^{\rho-1} h(x(t)) u(t) \Big) d\tau \\
&= \int_0^T w^T(\tau) E^T(t) Q E(t) w(\tau) d\tau - 2 \int_0^T \frac{\tau^\rho}{\rho!} \\
&\quad \times w^T(\tau) d\tau E^T(t) Q L_g L_f^{\rho-1} h(x(t)) u(t) + \int_0^T \frac{\tau^{2\rho}}{(\rho!)^2} \\
&\quad \times d\tau u^T(t) (L_g L_f^{\rho-1} h(x(t)))^T Q L_g L_f^{\rho-1} h(x(t)) u(t),
\end{aligned}$$

where $E = Y_d(t) - Y(t)$. Let

$$\begin{aligned}
v &= \int_0^T \frac{\tau^\rho}{\rho!} w^T(\tau) d\tau \\
&= \left[\frac{T^{\rho+1}}{(\rho+1)\rho!}, \frac{T^{\rho+2}}{(\rho+2)\rho!1!}, \dots, \frac{T^{2\rho+1}}{(2\rho+1)(\rho!)^2} \right]
\end{aligned}$$

and

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

Because the decision variable is $u(t)$, the solution to the minimization problem of $\hat{J}(t)$ is the same as the solution to the minimization of the quadratic function as follows:

$$\Psi(t) = u^T(t) \Theta u(t) + p^T u(t),$$

where $\Theta = \kappa (L_g L_f^{\rho-1} h(x(t)))^T Q L_g L_f^{\rho-1} h(x(t))$ and $p = -2(L_g L_f^{\rho-1} h(x(t)))^T Q^T E v^T$. Given that Θ is positive definite, the problem becomes an unconstrained quadratic program and the solution that minimizes $\Phi(t)$ is derived by solving $\partial\Psi(t)/\partial u = 0$, by which

$$u(t) = (L_g L_f^{\rho-1} h(x(t)))^{-1} \frac{1}{\kappa} (Y_d(t) - Y(t)) v^T. \quad (1.4)$$

Since the earlier work of Chen et al. [23, 127], this method has attracted much attention. Merabet et al. [128] applied the method to the speed control of induction motors. Wang et al. [129] presented the application of the method to the current control of a shunt active power filter. The performance of the method was also validated in a ball and beam system [130] and permanent magnet synchronous motors [131, 132]. To enhance the robustness of the method, the combinations with disturbance observers were also reported [133–139]. In particular, the method proposed by Yang and Zheng [135] is able to compensate for mismatched disturbances. To address the input constraint, projection neural networks were adopted in [9], by which the control action is given by the network output in an online manner. In [140, 141], the method was further extended to the case with parameter uncertainty. Specifically, a state-feedback auxiliary system was designed to compensate for the parameter

uncertainty. The block diagram of the resultant control structure can be shown in Fig. 1.2. An output-feedback auxiliary system was also proposed to solve the problem by using the technique in sliding mode control [13]. To address the problem of the measurement of the time derivatives of the system output, tracking differentiators were used. It was also shown that, for affine nonlinear dynamical systems, under certain conditions, the method can be applied to the case with fully unknown system dynamics by using neural networks [29]. The extensions of the Taylor expansion-based method to the consensus of multi-agent systems where the agent dynamics are nonlinear were reported in [142, 143]. This is achieved via a proper design of the performance index. However, the consensus protocol is not fully distributed.

1.4 Summary

This chapter has briefly reviewed some recent advancements in the near-optimal control of systems with nonlinear dynamics. Particular attention has been paid on the near-optimal control methods based on dynamic programming and nonlinear programming. Based on the survey, some related issues for future research works are outlined as follows.

- (1) For both dynamic programming and nonlinear programming based methods, the computational efficiency of existing near-optimal control methods for nonlinear dynamical systems are still not satisfactory for general cases. Although there have been some attempts to deal with this issue, e.g., the event-triggering approach [39, 64–66], how to design a computationally efficient near-optimal control method for nonlinear dynamical systems is still considered to be an open question.
- (2) Only a few results have been reported about the near-optimal control of nonlinear time-delay dynamical systems [36, 45]. However, time-delays exist ubiquitously in practice. In fact, there are many existing results for the optimal control of linear time-delay dynamical systems, such as [2, 6, 36]. Thus, the near-optimal control law design for nonlinear time-delay dynamical systems is worth investigating.
- (3) The near-optimal cooperative control of nonlinear multi-agent systems is seldom investigated. This problem is very challenging, especially when we need to design a fully distributed control law. In fact, even for linear multi-agent systems, the near-optimal control is difficult. There are only some results that can guarantee inverse optimality [144].
- (4) Most of the existing near-optimal control methods require certain model information of the controlled nonlinear dynamical systems. It would be better if a fully model-free method can be proposed. This is also very challenging. For nonlinear dynamical systems, the data for learning the system dynamics should be sufficient so as to ensure the learning performance. Meanwhile, in practice, the measurement data are usually polluted by measurement noises.
- (5) The combination of near-optimal control method with other modern control methods for nonlinear dynamical systems is interesting. For example, there are

some results about the co-design of optimal control and the backstepping design, such as [145]. The combination of the sliding mode control with optimal control was also investigated [146]. It would be better if the resultant control design can have the advantages of both near-optimal control methods and other modern control methods.

References

1. Lewis, F.L., Vrabie, D., Syrmos, V.L.: Optimal Control. Wiley, New York (2012)
2. Zhang, H., Li, L., Xu, J., Fu, M.: Linear quadratic regulation and stabilization of discrete-time systems with delay and multiplicative noise. *IEEE Trans. Autom. Control* **60**(10), 2599–2613 (2015)
3. Grieder, P., Borrelli, F., Torrisi, F., Morari, M.: Computation of the constrained infinite time linear quadratic regulator. *Automatica* **40**, 701–708 (2004)
4. Modares, H., Lewis, F.L., Jiang, Z.: Optimal output-feedback control of unknown continuous-time linear systems using off-policy reinforcement learning. *IEEE Trans. Autom. Control* **46**(11), 2401–2410 (2016)
5. Anderson, B.D.O., Moore, J.B.: Optimal Control: Linear Quadratic Methods. Courier Corporation (2007)
6. Zhang, H., Duan, G., Xie, L.: Linear quadratic regulation for linear time-varying systems with multiple input delays. *Automatica* **42**, 1465–1476 (2006)
7. Modares, H., Lewis, F.L.: Linear quadratic tracking control of partially-unknown continuous-time systems using reinforcement learning. *IEEE Trans. Autom. Control* **59**(11), 3051–3056 (2014)
8. Zhou, J., Wen, C., Zhang, Y.: Adaptive output control of nonlinear systems with uncertain dead-zone nonlinearity. *IEEE Trans. Autom. Control* **51**(3), 504–511 (2006)
9. Zhang, Y., Li, S.: Time-scale expansion-based approximated optimal control for underactuated systems using projection neural networks. *IEEE Trans. Syst. Man, Cybern., Syst.* in press. <https://doi.org/10.1109/TSMC.2017.2703140>
10. Liang, X., Fang, Y., Sun, N., Lin, H.: Nonlinear hierarchical control for unmanned quadrotor transportation systems. *IEEE Trans. Ind. Electron.* **65**(4), 3395–3405 (2018)
11. Xiao, B., Yin, S.: Velocity-free fault-tolerant and uncertainty attenuation control for a class of nonlinear systems. *IEEE Trans. Autom. Control* **63**(7), 4400–4411 (2016)
12. Sassano, M., Astolfi, A.: Dynamic approximate solutions of the HJ inequality and of the HJB equation for input-affine nonlinear systems. *IEEE Trans. Autom. Control* **57**(10), 2490–2503 (2012)
13. Zhang, Y., Li, S., Jiang, X.: Near-optimal control without solving HJB equations and its applications. *IEEE Trans. Ind. Electron.* **65**(9), 7173–7184 (2018)
14. Bertsekas, D.P.: Dynamic programming and suboptimal control: a survey from ADP to MPC. *Eur. J. Control* **11**(4), 310–334 (2005)
15. Mayne, D.Q.: Model predictive control: recent developments and future promise. *Automatica* **50**(12), 2967–2986 (2014)
16. Lee, J.H.: Model predictive control: review of the three decades of development. *Int. J. Control. Autom. Syst.* **9**(3), 415–424 (2011)
17. Mesbah, A.: Stochastic model predictive control with active uncertainty learning: a survey on dual control. *Annu. Rev. Control* **45**, 107–117 (2018)
18. Vazquez, S., Leon, J.I., Franquelo, L.G., Rodriguez, J., Young, H.A., Marquez, A., Zanchetta, P.: Model predictive control: a review of its applications in power electronics. *IEEE Ind. Electron. Mag.* **8**(1), 16–31 (2014)

19. Kiumarsi, B., Vamvoudakis, K.G., Modares, H., Lewis, F.L.: Optimal and autonomous control using reinforcement learning: a survey. *IEEE Trans. Neural Netw. Learn. Syst.* **29**(6), 2042–2062 (2018)
20. Khan, S.G., Herrmann, G., Lewis, F.L., Pipe, T., Melhuish, C.: Reinforcement learning and optimal adaptive control: an overview and implementation examples. *Annu. Rev. Control* **36**, 42–59 (2012)
21. Wang, D., He, H., Liu, D.: Adaptive critic nonlinear robust control: a survey. *IEEE Trans. Cybern.* **47**(10), 3429–3451 (2017)
22. Kang, W., Wilcox, L.C.: Mitigating the curse of dimensionality: sparse grid characteristics method for optimal feedback control and HJB equations. *Optimal Control Appl.* **68**(2), 289–315 (2017)
23. Chen, W., Ballanceb, D.J., Gawthrop, P.J.: Optimal control of nonlinear systems: a predictive control approach. *Automatica* **39**(4), 633–641 (2003)
24. Liu, Y., Tong, S.: Optimal control-based adaptive NN design for a class of nonlinear discrete-time block-triangular systems. *IEEE Trans. Cybern.* **46**(11), 2670–2680 (2016)
25. Wang, D., Liu, D., Li, H., Ma, H.: Neural-network-based robust optimal control design for a class of uncertain nonlinear systems via adaptive dynamic programming. *Infom. Sci.* **282**, 167–179 (2014)
26. Zhang, H., Qin, C., Luo, Y.: Neural-network-based constrained optimal control scheme for discrete-time switched nonlinear system using dual heuristic programming. *IEEE Trans. Autom. Sic. Eng.* **11**(3), 839–849 (2014)
27. Park, J., Sandberg, I.W.: Universal approximation using radial-basis-function networks. *Neural comput.* **3**(2), 246–257 (1991)
28. Vrabie, D., Lewis, F.: Neural network approach to continuous-time direct adaptive optimal control for partially unknown nonlinear systems. *Neural Netw.* **22**(3), 237–246 (2009)
29. Zhang, Y., Li, S., Liu, X.: Neural network-based model-free adaptive near-optimal tracking control for a class of nonlinear systems. *IEEE Trans. Neural Netw. Learn. Syst.* in press. <https://doi.org/10.1109/TNNLS.2018.2828114>
30. Luo, B., Liu, D., Huang, T., Wang, D.: Model-free optimal tracking control via critic-only Q-learning. *IEEE Trans. Neural Netw. Learn. Syst.* **27**(10), 2134–2144 (2016)
31. Tassa, Y., Erez, T.: Least squares solutions of the HJB equation with neural-network value-function approximators. *IEEE Trans. Neural Netw.* **18**(4), 1–10 (2007)
32. Dierks, T., Jagannathan, S.: Optimal control of affine nonlinear continuous-time systems. In: *Proceedings of the ACC*, pp. 1568–1573 (2010)
33. Wang, F., Zhang, H., Liu, D.: Adaptive dynamic programming: an introduction. *IEEE Comput. Mag.* **4**(2), 39–47 (2009)
34. Al-Tamimi, A., Lewis, F.L., Abu-Khalaf, M.: Discrete-time nonlinear HJB solution using approximate dynamic programming: Convergence proof. *IEEE Trans. Syst., Man, Cybern. B, Cybern.* **38**(4), 943–949 (2008)
35. Liu, D., Wei, Q.: Policy iteration adaptive dynamic programming algorithm for discrete-time nonlinear systems. *IEEE Trans. Neural Netw. Learn. Syst.* **25**(3), 621–634 (2014)
36. Zhang, H., Song, R., Wei, Q., Zhang, T.: Optimal tracking control for a class of nonlinear discrete-time systems with time delays based on heuristic dynamic programming. *IEEE Trans. Neural Netw.* **22**(12), 1851–1862 (2011)
37. Zhang, H., Wei, Q., Luo, Y.: 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. B, Cybern.* **38**(4), 937–942 (2008)
38. Yang, L., Si, J., Tsakalis, K.S., Rodriguez, A.A.: Direct heuristic dynamic programming for nonlinear tracking control with filtered tracking error. *IEEE Trans. Syst., Man, Cybern. B, Cybern.* **39**(6), 1617–1622 (2009)
39. Dong, L., Zhong, X., Sun, C., He, H.: Adaptive event-triggered control based on heuristic dynamic programming for nonlinear discrete-time systems. *IEEE Trans. Neural Netw. Learn. Syst.* **28**(7), 1594–1605 (2017)

40. Luo, B., Liu, D., Huang, T., Liu, J.: Output tracking control based on adaptive dynamic programming with multistep policy evaluation. *IEEE Trans. Syst., Man., Cybern., Syst.* to be published. <https://doi.org/10.1109/TSMC.2017.2771516>
41. He, H., Ni, Z., Fu, J.: A three-network architecture for on-line learning and optimization based on adaptive dynamic programming. *Neurocomputing* **78**(1), 3–13 (2012)
42. Ni, Z., He, H., Wen, J.: Adaptive learning in tracking control based on the dual critic network design. *IEEE Trans. Neural Netw. Learn. Syst.* **24**(6), 913–928 (2013)
43. Ni, Z., He, H., Zhao, D., Xu, X., Prokhorov, D.V.: GrDHP: a general utility function representation for dual heuristic dynamic programming. *IEEE Trans. Neural Netw. Learn. Syst.* **26**(3), 614–627 (2015)
44. Ni, Z., He, H., Zhong, X., Prokhorov, D.V.: Model-free dual heuristic dynamic programming. *IEEE Trans. Neural Netw. Learn. Syst.* **26**(8), 1834–1839 (2015)
45. Wang, B., Zhao, D., Alippi, C., Liu, D.: Dual heuristic dynamic programming for nonlinear discrete-time uncertain systems with state delay. *Neurocomputing* **134**, 222–229 (2014)
46. Lin, W., Yang, P.: Adaptive critic motion control design of autonomous wheeled mobile robot by dual heuristic programming. *Automatica* **44**(11), 2716–2723 (2008)
47. Lian, C., Xu, X., Chen, H., He, H.: Near-optimal tracking control of mobile robots via receding-horizon dual heuristic programming. *IEEE Trans. Cybern.* **46**(11), 2484–2496 (2016)
48. Fairbank, M., Alonso, E., Prokhorov, D.: An equivalence between adaptive dynamic programming with a critic and backpropagation through time. *IEEE Trans. Neural Netw. Learn. Syst.* **24**(12), 2088–2100 (2013)
49. Zhong, X., Ni, Z., He, H.: Gr-GDHP: a new architecture for globalized dual heuristic dynamic programming. *IEEE Trans. Cybern.* **47**(10), 3318–3330 (2017)
50. Yen, G.G., DeLima, P.G.: Improving the performance of globalized dual heuristic programming for fault tolerant control through an online learning supervisor. *IEEE Trans. Autom. Sci. Eng.* **2**(2), 121–131 (2005)
51. Wang, D., Liu, D., Wei, Q., Zhao, D., Jin, N.: Optimal control of unknown nonaffine nonlinear discrete-time systems based on adaptive dynamic programming. *Automatica* **48**(8), 1825–1832 (2012)
52. Fairbank, M., Alonso, E., Prokhorov, D.: Simple and fast calculation of the second-order gradients for globalized dual heuristic dynamic programming in neural networks. *IEEE Trans. Neural Netw. Learn. Syst.* **23**(10), 1671–1676 (2012)
53. Liu, D., Wang, D., Zhao, D., Wei, Q., Jin, N.: Neural-network-based optimal control for a class of unknown discrete-time nonlinear systems using globalized dual heuristic programming. *IEEE Trans. Autom. Sci. Eng.* **9**(3), 628–634 (2012)
54. Zhao, D., Xia, Z., Wang, D.: Model-free optimal control for affine nonlinear systems with convergence analysis. *IEEE Trans. Autom. Sci. Eng.* **12**(4), 1461–1468 (2015)
55. Si, J., Wang, Y.: On-line learning control by association and reinforcement. *IEEE Trans. Neural Netw.* **12**(2), 264–276 (2001)
56. Wei, Q., Zhang, H., Dai, J.: Model-free multiobjective approximate dynamic programming for discrete-time nonlinear systems with general performance index functions. *Neurocomputing* **72**(7), 1839–1848 (2009)
57. Dierks, T., Thumati, B.T., Jagannathan, S.: Optimal control of unknown affine nonlinear discrete-time systems using offline-trained neural networks with proof of convergence. *Neural Netw.* **22**(5), 851–860 (2009)
58. Wei, Q., Liu, D.: An iterative ϵ -optimal control scheme for a class of discrete-time nonlinear systems with unfixed initial state. *Neural Netw.* **32**, 236–244 (2012)
59. Dierks, T., Jagannathan, S.: Online optimal control of nonlinear discrete-time systems using approximate dynamic programming. *J. Control Theory Appl.* **9**(3), 361–369 (2011)
60. Zhao, Q., Xu, H., Jagannathan, S.: Neural network-based finite-horizon optimal control of uncertain affine nonlinear discrete-time systems. *IEEE Trans. Neural Netw. Learn. Syst.* **26**(3), 486–499 (2015)
61. Wang, D., He, H., Mu, C., Liu, D.: Intelligent critic control with disturbance attenuation for affine dynamics including an application to a microgrid system. *IEEE Trans. Ind. Electron.* **64**(6), 4935–4944 (2017)

62. Zhang, H., Cui, L., Zhang, X., Luo, Y.: Data-driven robust approximate optimal tracking control for unknown general nonlinear systems using adaptive dynamic programming method. *IEEE Trans. Neural Netw.* **22**(12), 2226–2236 (2011)
63. Wei, Q., Liu, D.: Adaptive dynamic programming for optimal tracking control of unknown nonlinear systems with application to coal gasification. *IEEE Trans. Autom. Sci. Eng.* **11**(4), 1020–1036 (2014)
64. Zhong, X., He, H.: An event-triggered adp control approach for continuous-time system with unknown internal states. *IEEE Trans. Cybern.* **47**(3), 683–694 (2017)
65. Yang, X., He, H.: Adaptive critic designs for event-triggered robust control of nonlinear systems with unknown dynamics. *IEEE Trans. Cybern.* to be published. <https://doi.org/10.1109/TCYB.2018.2823199>
66. Wang, D., Mu, C., Liu, D., Ma, H.: On mixed data and event driven design for adaptive-critic-based nonlinear H_∞ control. *IEEE Trans. Neural Netw. Learn. Syst.* **29**(4), 993–1005 (2018)
67. Liu, W., Venayagamoorthy, G.K., Wunsch II, D.C.: A heuristic-dynamic-programming-based power system stabilizer for a turbogenerator in a single-machine power system. *IEEE Trans. Ind. Appl.* **41**(5), 1377–1385 (2005)
68. Jiang, Y., Jiang, Z.: Global adaptive dynamic programming for continuous-time nonlinear systems. *IEEE Trans. Autom. Control* **60**(11), 2917–2929 (2015)
69. Mu, C., Ni, Z., Sun, C., He, H.: Air-breathing hypersonic vehicle tracking control based on adaptive dynamic programming. *IEEE Trans. Neural Netw. Learn. Syst.* **28**(3), 584–598 (2017)
70. Wen, Y., Si, J., Gao, X., Huang, S., Huang, H.: A new powered lower limb prosthesis control framework based on adaptive dynamic programming. *IEEE Trans. Neural Netw. Learn. Syst.* **28**(9), 2215–2220 (2017)
71. Xu, X., Lian, C., Zuo, L., He, H.: Kernel-based approximate dynamic programming for real-time online learning control: an experimental study. *IEEE Trans. Control Syst. Technol.* **22**(1), 146–156 (2014)
72. Venayagamoorthy, G.K., Harley, R.G., Wunsch, D.C.: Comparison of heuristic dynamic programming and dual heuristic programming adaptive critics for neurocontrol of a turbogenerator. *IEEE Trans. Neural Netw.* **13**(3), 764–773 (2002)
73. Beard, R.W., Saridis, G.N., Wen, J.T.: Approximate solutions to the time-invariant Hamilton-Jacobi-Bellman equation. *J. Opt. Theory Appl.* **96**(3), 589–626 (1998)
74. Chen, Y., Edgar, T., Manousiouthakis, V.: On infinite-time nonlinear quadratic optimal control. *Syst. Control Lett.* **51**(3), 259–268 (2004)
75. Song, C., Ye, J., Liu, D., Kang, Q.: Generalized receding horizon control of fuzzy systems based on numerical optimization algorithm. *IEEE Trans. Fuzzy Syst.* **17**(6), 1336–1352 (2009)
76. Fakharian, A., Beheshti, M.T.H., Davari, A.: Solving the Hamilton-Jacobi-Bellman equation using Adomian decomposition method. *Int. J. Comp. Math.* **87**(12), 2769–2785 (2010)
77. Nik, H.S., Effati, S., Shirazian, M.: An approximate-analytical solution for the Hamilton-Jacobi-Bellman equation via homotopy perturbation method. *App. Math. Model.* **36**(11), 5614–5623 (2012)
78. Ganjefar, S., Rezaei, S.: Modified homotopy perturbation method for optimal control problems using the Padé approximant. *App. Math. Model.* **40**(15), 7062–7081 (2016)
79. Zhu, J.: A feedback optimal control by Hamilton-Jacobi-Bellman equation. *Eur. J. Control* **37**, 70–74 (2017)
80. Govindarajan, N., de Visser, C.C., Krishnakumar, K.: A sparse collocation method for solving time-dependent HJB equations using multivariate B-splines. *Automatica* **50**(9), 2234–2244 (2014)
81. Sun, B., Guo, B.: Convergence of an upwind finite-difference scheme for Hamilton-Jacobi-Bellman equation in optimal control. *IEEE Trans. Autom. Control* **60**(11), 3012–3017 (2015)
82. Chen, X., Chen, X.: An iterative method for optimal feedback control and generalized HJB equation. *IEEE/CAA J. Autom. Sinica* **5**(5), 999–1006 (2018)

83. Michailidis, I., Baldi, S., Kosmatopoulos, E.B., Ioannou, P.A.: Adaptive optimal control for large-scale nonlinear systems. *IEEE Trans. Autom. Control* **62**(11), 5567–5577 (2017)
84. Kirches, C., Bock, H.G., Schlöer, J.P., Sager, S.: Block-structured quadratic programming for the direct multiple shooting method for optimal control. *Opt. Methods Soft.* **26**(2), 239–257 (2011)
85. Schäfer, A., Kühl, P., Diehl, M., Schlöder, J., Bock, H.G.: Fast reduced multiple shooting methods for nonlinear model predictive control. *Chem. Eng. Process.* **46**, 1200–1214 (2007)
86. Gerdts, M.: Direct shooting method for the numerical solution of higher-index DAE optimal control problems. *J. Opt. Theory Appl.* **117**(2), 267–294 (2003)
87. Hannemann-Tamás, R., Marquardt, W.: How to verify optimal controls computed by direct shooting methods? - A tutorial. *J. Process Control* **22**, 494–507 (2012)
88. Garg, D., Hager, W.W., Rao, A.V.: Pseudospectral methods for solving infinite-horizon optimal control problems. *Automatica* **47**(4), 829–837 (2011)
89. Wang, X., Peng, H., Zhang, S., Chen, B., Zhong, W.: A symplectic pseudospectral method for nonlinear optimal control problems with inequality constraints. *ISA Trans.* **68**, 335–352 (2017)
90. Garg, D., Patterson, M., Hager, W.W., Rao, A.V., Benson, D.A., Huntington, G.T.: A unified framework for the numerical solution of optimal control problems using pseudospectral methods. *Automatica* **46**(11), 1843–1851 (2010)
91. Darby, C.L., Hager, W.W., Rao, A.V.: An hp-adaptive pseudospectral method for solving optimal control problems. *Optim. Control Appl. Meth.* **32**(4), 476–502 (2011)
92. Ruths, J., Zlotnik, A., Li, J.: Convergence of a pseudospectral method for optimal control of complex dynamical systems. In: Proceedings of the CDC-ECC, pp. 5553–5558. Orlando (2011)
93. Xu, S., Li, S.E., Deng, K., Li, S., Cheng, B.: A unified pseudospectral computational framework for optimal control of road vehicles. *IEEE/ASME Trans. Mechatron.* **20**(4), 1499–1510 (2015)
94. Fahroo, F., Ross, I.M.: Advances in pseudospectral methods for optimal control. Proceeding of the AIAA Guidance. Navigation, and Control Conference and Exhibition, pp. 18–21. Honolulu, Hawaii (2008)
95. Peng, H., Wang, X., Li, M., Chen, B.: An hp symplectic pseudospectral method for nonlinear optimal control. *Commun. Nonlinear Sci. Numer. Simulat.* **42**, 623–644 (2017)
96. Tabrizidooz, H.R., Marzban, H.R., Pourbabae, M., Hedayati, M.: A composite pseudospectral method for optimal controlproblems with piecewise smooth solutions. *J. Frankl. Inst.* **354**(5), 2393–2414 (2017)
97. Ge, X., Yi, Z., Chen, L.: Optimal control of attitude for coupled-rigid-body spacecraft via Chebyshev-Gauss pseudospectral method. *Appl. Math. Mech. -Eng. Ed.* **38**(9), 1257–1272 (2017)
98. Genest, R., Ringwood, J.V.: Receding horizon pseudospectral control for energy maximization with application to wave energy devices. *IEEE Trans. Control Syst. Technol.* **25**(1), 29–38 (2017)
99. Merigaud, A., Ringwood, J.V.: Improving the computational performance of nonlinear pseudospectral control of wave energy converters. *IEEE Trans. Sustain. Energy* **9**(3), 1419–1426 (2018)
100. Mayne, D.Q., Rawlings, J.B., Rao, C.V., Scokaert, P.O.M.: Constrained model predictive control: stability and optimality. *Automatica* **36**(6), 789–814 (2010)
101. Gros, S.: A distributed algorithm for NMPC-based wind farm control. In: Proceedings of the CDS, pp. 4844–4849. Los Angeles, California (2014)
102. Cannon, M.: Efficient nonlinear model predictive control algorithms. *Annu. Rev. Control* **28**(2), 229–237 (2004)
103. Ohtsuka, T.: A continuation/GMRES method for fast computation of nonlinear receding horizon control. *Automatica* **40**(4), 563–574 (2004)
104. Wan, Z., Kothare, M.V.: Efficient scheduled stabilizing output feedback model predictive control for constrained nonlinear systems. *IEEE Trans. Autom. Control* **49**(7), 1172–1177 (2004)

105. Xu, F., Chen, H., Gong, X., Mei, Q.: Fast nonlinear model predictive control on fpga using particle swarm optimization. *IEEE Trans. Ind. Electron.* **63**(1), 310–321 (2016)
106. Mozaffari, A., Azad, N.L., Hedrick, J.K.: A nonlinear model predictive controller with multiagent online optimizer for automotive cold-start hydrocarbon emission reduction. *IEEE Trans. Veh. Technol.* **65**(6), 4548–4563 (2016)
107. Li, Z., Deng, J., Lu, R., Xu, Y., Bai, J., Su, C.: Trajectory-tracking control of mobile robot systems incorporating neural-dynamic optimized model predictive approach. *IEEE Trans. Syst., Man, Cybern., Syst.* **46**(6), 740–749 (2016)
108. Pan, Y., Wang, J.: Model predictive control of unknown nonlinear dynamical systems based on recurrent neural networks. *IEEE Trans. Ind. Electron.* **59**(8), 3089–3101 (2012)
109. Yan, Z., Wang, J.: Model predictive control of nonlinear systems with unmodeled dynamics based on feedforward and recurrent neural networks. *IEEE Trans. Ind. Inform.* **8**(4), 746–756 (2012)
110. Yan, Z., Wang, J.: Robust model predictive control of nonlinear systems with unmodeled dynamics and bounded uncertainties based on neural networks. *IEEE Trans. Neural Netw. Learn. Syst.* **25**(3), 457–469 (2014)
111. Yan, Z., Wang, J.: Nonlinear model predictive control based on collective neurodynamic optimization. *IEEE Trans. Neural Netw. Learn. Syst.* **26**(4), 840–850 (2015)
112. Cheng, L., Liu, W., Hou, Z., Yu, J., Tan, M.: Neural-network-based nonlinear model predictive control for piezoelectric actuators. *IEEE Trans. Ind. Electron.* **62**(12), 7717–7727 (2015)
113. Bayer, F., Burger, M., Allgöwer, F.: Discrete-time incremental ISS: a framework for robust NMPC. In: Proceedings of the ECC, Zürich, pp. 2068–2073. Switzerland (2013)
114. Garcia, G.A., Keshmiri, S.S., Stastny, T.: Robust and adaptive nonlinear model predictive controller for unsteady and highly nonlinear unmanned aircraft. *IEEE Trans. Control Syst. Technol.* **23**(4), 1620–1627 (2015)
115. Pin, G., Raimondo, D.M., Magni, L., Parisini, T.: Robust model predictive control of nonlinear systems with bounded and state-dependent uncertainties. *IEEE Trans. Autom. Control* **54**(7), 1681–1687 (2009)
116. Li, H., Shi, Y.: Robust distributed model predictive control of constrained continuous-time nonlinear systems: a robustness constraint approach. *IEEE Trans. Autom. Control* **59**(6), 1673–1678 (2014)
117. Liu, C., Li, H., Gao, J., Xu, D.: Robust self-triggered min-max model predictive control for discrete-time nonlinear systems. *Automatica* **89**, 333–339 (2018)
118. Rubagotti, M., Raimondo, D.M., Ferrara, A., Magni, L.: Robust model predictive control with integral sliding mode in continuous-time sampled-data nonlinear systems. *IEEE Trans. Autom. Control* **56**(3), 556–570 (2011)
119. Kang, Y., Hedrick, J.K.: Linear tracking for a fixed-wing UAV using nonlinear model predictive control. *IEEE Trans. Control Syst. Technol.* **17**(5), 1202–1210 (2009)
120. Kim, S., Oh, H., Suk, J., Tsourdos, A.: Coordinated trajectory planning for efficient communication relay using multiple UAVs. *Control Eng. Prac.* **29**, 42–49 (2014)
121. Hosen, M.A., Hussain, M.A., Mjalli, F.S.: Control of polystyrene batch reactors using neural network based model predictive control (NNMPC): an experimental investigation. *Control Eng. Prac.* **19**, 454–467 (2011)
122. Vukov, M., Loock, W.V., Houska, B., Ferreau, H.J., Swevers, J., Diehl, M.: Experimental validation of nonlinear MPC on an overhead crane using automatic code generation. In: Proceedings of the ACC, pp. 6264–6269. Fairmont Queen Elizabeth, Montréal, Canada (2012)
123. Mills, A., Wills, A., Ninness, B.: Nonlinear model predictive control of an inverted pendulum. In: Proceedings of the ACC, pp. 2335–2340. Hyatt Regency Riverfront, St. Louis (2009)
124. Zhu, Q., Onori, S., Prucka, R.: An economic nonlinear model predictive control strategy for SI engines: model-based design and real-time experimental validation. *IEEE Trans. Control Syst. Technol.*, to be published, <https://doi.org/10.1109/TCST.2017.2769660>
125. Xie, Y., Ghaemi, R., Sun, J., Freudenberg, J.S.: Model predictive control for a full bridge DC/DC converter. *IEEE Trans. Control Syst. Technol.* **20**(1), 164–172 (2012)

126. Kayacan, E., Saeys, W., Ramon, H., Belta, C., Peschel, J.: Experimental validation of linear and nonlinear MPC on an articulated unmanned ground vehicle. *IEEE/ASME Trans. Mechatron.* to be published. <https://doi.org/10.1109/TMECH.2018.2854877>
127. Chen, W.: Closed-form nonlinear MPC for multivariable nonlinear systems with different relative degree. In: *Proceedings of the ACC*, pp. 4887–4892. Denver (2003)
128. Merabet, A., Arioui, H., Ouhrouche, M.: Cascaded predictive controller design for speed control and load torque rejection of induction motor. In: *Proceedings of the ACC*, pp. 1139–1144. Seattle (2008)
129. Wang, X., Xie, Y., Shuai, D.: Three-phase four-leg active power filter based on nonlinear optimal predictive control. In: *Proceedings of the CCC*, pp. 217–222. Kunming, Yunnan, China (2008)
130. Dong, Z., Zheng, G., Liu, G.: Networked nonlinear model predictive control of the ball and beam system. In: *Proceedings of the CCC*, pp. 469–473. Kunming, Yunnan, China (2008)
131. Errouissi, R., Ouhrouche, M.: Nonlinear predictive controller for a permanent magnet synchronous motor drive. *Math. Comput. Simul.* **81**, 394–406 (2010)
132. Errouissi, R., Ouhrouche, M., Chen, W., Trzynadlowski, A.M.: Robust nonlinear predictive controller for permanent-magnet synchronous motors with an optimized cost function. *IEEE Trans. Ind. Electron.* **59**(7), 2849–2858 (2012)
133. Yang, J., Zhao, Z., Li, S., Zheng, W.X.: Composite predictive flight control for airbreathing hypersonic vehicles. *Int. J. Control.* **87**(9), 1970–1984 (2014)
134. Liu, C., Chen, W., Andrews, J.: Tracking control of small-scale helicopters using explicit nonlinear MPC augmented with disturbance observers. *Control Eng. Prac.* **20**(3), 258–268 (2012)
135. Yang, J., Zheng, W.X.: Offset-free nonlinear MPC for mismatched disturbance attenuation with application to a static var compensator. *IEEE Trans. Circuits Syst. II, Exp. Briefs* **61**(1), 49–53 (2014)
136. Ouari, K., Rekioua, T., Ouhrouche, M.: Real time simulation of nonlinear generalized predictive control for wind energy conversion system with nonlinear observer. *ISA Trans.* **53**(1), 76–84 (2014)
137. Errouissi, R., Muyeen, S.M., Al-Durra, A., Leng, S.: Experimental validation of a robust continuous nonlinear model predictive control based grid-interlinked photovoltaic inverter. *IEEE Trans. Ind. Electron.* **63**(7), 4495–4505 (2016)
138. Errouissi, R., Al-Durra, A., Muyeen, S.M.: Design and implementation of a nonlinear PI predictive controller for a grid-tied photovoltaic inverter. *IEEE Trans. Ind. Electron.* **64**(2), 1241–1250 (2017)
139. Errouissi, R., Ouhrouche, M., Chen, W., Trzynadlowski, A.M.: Robust cascaded nonlinear predictive control of a permanent magnet synchronous motor with antiwindup compensator. *IEEE Trans. Ind. Electron.* **59**(8), 3078–3088 (2012)
140. Zhang, Y., Li, S., Liu, X.: Adaptive near-optimal control of uncertain systems with application to underactuated surface vessels. *IEEE Trans. Control Syst. Technol.* **26**(4), 1204–1218 (2018)
141. Zhang, Y., Li, S., Luo, X., Shang, M.: A dynamic neural controller for adaptive optimal control of permanent magnet DC motors. In: *Proceedings of the IJCNN*, pp. 839–844. Anchorage, USA (2017)
142. Zhang, Y., Li, S.: Predictive suboptimal consensus of multiagent systems with nonlinear dynamics. *IEEE Trans. Syst., Man, Cybern., Syst.* **47**(7), 1701–1711 (2017)
143. Zhang, Y., Li, S.: Adaptive near-optimal consensus of high-order nonlinear multi-agent systems with heterogeneity. *Automatica* **85**, 426–432 (2017)
144. Movric, K.H., Lewis, F.L.: Cooperative optimal control for multi-agent systems on directed graph topologies. *IEEE Trans. Autom. Control* **59**(3), 769–774 (2014)
145. Liu, Y., Gao, Y., Tong, S., Li, Y.: Fuzzy approximation-based adaptive backstepping optimal control for a class of nonlinear discrete-time systems with dead-zone. *IEEE Trans. Fuzzy Syst.* **24**(1), 16–28 (2016)
146. Chen, S., Wang, J., Yao, M., Kim, Y.: Improved optimal sliding mode control for a non-linear vehicle active suspension system. *J. Sound Vib.* **395**, 1–25 (2017)

Chapter 2

Near-Optimal Control with Input Saturation



Abstract In this chapter, a time-scale expansion-based scheme is presented for approximately solving the optimal control problem of continuous-time underactuated nonlinear systems subject to input constraints and system dynamics. By time-scale Taylor approximation of the original performance index, the optimal control problem is relaxed into an approximated optimal control problem. Based on the system dynamics, the problem is further reformulated as a quadratic program, which is solved by a projection neural network. Theoretical analysis on the closed-loop system synthesized by the controlled system and the projection neural network is conducted, which reveals that, under certain conditions, the closed-loop system possesses exponential stability and the original performance index converges to zero as time tends to infinity. In addition, two illustrative examples, which are based on a flexible joint manipulator and an underactuated ship, are provided to validate the theoretical results and demonstrate the efficacy and superiority of the presented control scheme.

2.1 Introduction

Optimal control has long been a heated research topic in the control community, which often deals with the problem of minimizing a given performance index subject to the dynamics of a controlled system and some constraints. The optimal control of nonlinear systems often requires the solution of Hamilton–Jacobi–Bellman partial differential equations, which are often difficult to solve directly and some indirect methods have been proposed [1–4].

Viewed as a promising approach, model-based predictive control has drawn much attention, which avoids solving Hamilton–Jacobi–Bellman equations and generally obtains the control action by solving a constrained discrete-time finite-horizon optimal control problem [5]. Specifically, an optimal control law needs to be found to minimize a discrete-time finite-horizon performance index by using the information obtained from the discrete-time model of the controlled system. As a special case of model-based predictive control approach, by adopting the moving-horizon control concept, Chen et al. [6] defined a receding-horizon continuous-time integral-type

performance index for an affine nonlinear system and proposed an explicit control law, which can guarantee the stability of the closed-loop system under some conditions. Following their method, Hedjar et al. [7] further considered the optimal control of a special type of affine nonlinear systems and proved the stability of the closed-loop system by Lyapunov theory. To reach analytical conclusions, the works by Chen et al. [6] and Hedjar et al. [7] ignore the constraint on the control inputs. This simplification also happens in some finite-horizon control literature [8–12]. However, for practical systems, the input constraints exist, for example, the input power provided to an electrohydraulic actuator is limited in practice [13, 14]. There are also some control methods for the optimal control of discrete-time nonlinear systems with input constraints taken into account, which adopt a discrete-time performance index and are based on discrete-time system models, such as the model predictive control investigated in [5, 15–19]. In [20], the optimal control problem for continuous-time nonlinear systems with a nonintegral type performance index subject to bounded-control constraints is considered, where an iterative algorithm based on fixed-point iterations was proposed for calculating the optimal control law.

Neural networks are widely investigated in recent decades [21–28]. One of the main applications of neural networks is solving optimization problems [24–27]. In [24], a noise-tolerant recurrent neural network was proposed for solving time-varying quadratic programs. In [26], a two-layer recurrent neural network was proposed for solving non-smooth convex optimization problems with convex inequality and linear equality constraints, which does not need penalty parameters. Besides, recurrent neural networks have been applied to the redundancy resolution of robot manipulators [24, 29–59], which are often formulated as an optimization problem subject to physical and kinematic constraints of the manipulators. For example, in [52], a dual neural network was proposed for redundancy resolution of kinematically redundant manipulators subject to joint limits and joint velocity limits. By utilizing the capability for solving optimization problems, adaptivity, or the global approximation property, neural networks have also been applied to the field of control [17–19, 60–65]. For example, Chen et al. [60] proposed a scheme to achieve the adaptive consensus control for a class of nonlinear multi-agent time-delay systems using neural networks.

In this chapter, the optimal control of continuous-time underactuated nonlinear systems with input constraints is investigated, which cover a large number of mechanical systems [66–69]. A finite-horizon integral-type performance index of the original optimal control problem about an underactuated system is approximated by time-scale Taylor expansion, which is converted into an approximated optimal control problem without the linearization of the continuous-time system. This is different from the existing results [17–19] on handling control problems of nonlinear systems via recurrent neural networks. This chapter adopts a time-scale approximation, the resultant approximated optimal control problem has less variables to be computed than that based on model predictive control, which employs a discrete-time performance index. The approximated control problem is further formulated as a quadratic program and a projection neural network is designed to solve it.

2.2 Preliminary and Notations

In this section, some necessary definitions, theoretical basis, and assumptions are presented.

Definition 2.1 The underactuated nonlinear system considered in this chapter is described as follows:

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

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

Definition 2.2 ([69]) 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 2.3 ([69]) System (2.1) is said to have a relative degree of ρ in the region of interest \mathbb{U} if the following properties hold

- $\forall \mathbf{x} \in \mathbb{U}$, $L_g L_f^i h(\mathbf{x}) = 0$, for $0 \leq i < \rho - 1$;
- $\forall \mathbf{x} \in \mathbb{U}$, $L_g L_f^{\rho-1} h(\mathbf{x}) \neq 0$.

Lemma 2.1 ([70]) If $\nabla F(\mathbf{s})$ is symmetric and positive definite, then the projection neural network

$$\lambda \dot{\mathbf{s}} = -\mathbf{s} + P_{\Omega}(\mathbf{s} - F(\mathbf{s})) \quad (2.2)$$

is globally and exponentially stable, where $\mathbf{s} \in \mathbb{R}^\sigma$ is the state variable, parameter $\lambda > 0 \in \mathbb{R}$ is used to scale the convergence of the projection neural network, mapping $F(\cdot) : \mathbb{R}^\sigma \rightarrow \mathbb{R}^\sigma$ is continuously differentiable, $\Omega = \{\zeta \in \mathbb{R}^\sigma | \zeta^- < \zeta < \zeta^+\}$ is a closed convex subset of \mathbb{R}^σ with ζ^- and ζ^+ denoting the lower bound and the upper

bound, respectively, and each element of projection function $P_{\Omega}(\mathbf{s}) : \mathbb{R}^{\sigma} \rightarrow \Omega$ is defined as follows:

$$P_{\Omega}(s_i) = \begin{cases} \zeta_i^+, & s_i > \zeta_i^+, \\ s_i, & \zeta_i^- \leq s_i \leq \zeta_i^+, \\ \zeta_i^-, & s_i < \zeta_i^-. \end{cases} \quad (2.3)$$

Lemma 2.2 ([66]) Consider the following singularly perturbed system:

$$\begin{cases} \dot{\varpi} = \mu(\varpi, \rho, \varepsilon), \\ \varepsilon \dot{\rho} = \vartheta(\varpi, \rho, \varepsilon), \end{cases} \quad (2.4)$$

where $\varpi \in \mathbb{R}^v$ and $\rho(t) \in \mathbb{R}^x$ denote state vectors of the system, and $\varepsilon > 0 \in \mathbb{R}$ is a constant parameter; functions $\mu : \mathbb{R}^{v+x+1} \rightarrow \mathbb{R}^v$ and $\vartheta : \mathbb{R}^{v+x+1} \rightarrow \mathbb{R}^x$ are smooth.

Assume that the following conditions are satisfied for all $(\varpi, \varepsilon) \in B_v \times [0, \varepsilon_0]$.

- (1) $\mu(0, 0, \varepsilon) = 0$ and $\vartheta(0, 0, \varepsilon) = 0$.
- (2) The equation $\vartheta(\varpi, \rho, 0) = 0$ has an isolated root $\rho = \phi(\varpi)$ such that $\phi(\varpi)$ is constant, where $\phi : \mathbb{R}^v \rightarrow \mathbb{R}^x$ is a smooth function.
- (3) The functions μ , ϑ , ϕ , and their partial derivatives up to the second order are bounded for $\rho - \phi(\varpi) \in B_x$.
- (4) The origin of the reduced system $\dot{\varpi} = \mu(\varpi, \phi(\varpi), 0)$ is exponentially stable.
- (5) The origin of the bounded-layer system

$$\frac{d\omega}{d\hat{\tau}} = \vartheta(\varpi, \omega + \phi(\varpi), 0) \quad (2.5)$$

is exponentially stable, where $\omega = \rho - \phi(\varpi)$ and $\hat{\tau} = t/\varepsilon$.

Then, there exists $\varepsilon^* > 0 \in \mathbb{R}$ such that for all $\varepsilon < \varepsilon^*$, the origin $(\varpi = 0, \rho = 0)$ of system (2.4) is exponentially stable.

In this chapter, the following general assumptions are imposed on system (2.1) [69, 71–76]: (1) the zero dynamics of system (2.1) is stable; (2) all state variables of system (2.1) are available; (3) system (2.1) has a well-defined relative degree ρ ; (4) output $y(t)$ of system (2.1) and desired output $y_d(t)$ are ρ times continuously differentiable with respect to time t .

Throughout this chapter, the following notations are used. $|A|$ denotes the determinant of a matrix A . $\|A\|_2$ denotes the Euclidean norm when A is a matrix or a vector, which reduces to the absolute value when A is a real number. A^T denotes the transpose when A is a matrix or a vector. A^* and A^{-1} denote the adjoint matrix and the inverse of a matrix A , respectively. In addition, $n!$ denotes the factorial of an integer n .

2.3 Problem Formulations

In this section, the optimal control problem about (2.1) with an input constraint is presented. By the time-scale Taylor series expansion, it is then relaxed to an approximated optimal control problem.

2.3.1 Original Formulation

The finite-horizon optimal control problem about system (2.1) with an input constraint is defined as follows:

$$\text{minimize} \quad J(t) \quad (2.6)$$

$$\text{subject to} \quad \dot{\mathbf{x}} = f(\mathbf{x}) + g(\mathbf{x})\mathbf{u}(t), \quad (2.7)$$

$$\mathbf{y}(t) = h(\mathbf{x}), \quad (2.8)$$

$$\mathbf{u}(t) \in \Omega, \quad (2.9)$$

where convex set $\Omega = \{\mathbf{u} \in \mathbb{R}^r | \mathbf{u}^- \leq \mathbf{u} \leq \mathbf{u}^+\}$ with constant vectors $\mathbf{u}^- \in \mathbb{R}^r$ and $\mathbf{u}^+ \in \mathbb{R}^r$ denoting the lower bound and upper bound of the constraint imposed on control input $\mathbf{u}(t)$, and the performance index $J(t)$ is defined as follows:

$$J(t) = \int_0^T (\mathbf{y}_d(t + \tau) - \mathbf{y}(t + \tau))^T Q (\mathbf{y}_d(t + \tau) - \mathbf{y}(t + \tau)) d\tau, \quad (2.10)$$

where matrices $Q \in \mathbb{R}^{r \times r}$ is symmetric and positive definite. Constant parameter $T > 0 \in \mathbb{R}$ is termed predictive period in the sense of predictive control [6]. Intuitively, the optimal control problem is to find an input $\mathbf{u}(t)$ belonging to set Ω such that the performance index $J(t)$ about the system is minimized. Thus, the optimal control law must be feasible first, which means that it must belong to Ω . Note that throughout this chapter, by $\mathbf{u}^- \leq \mathbf{u}$ and $\mathbf{u} \leq \mathbf{u}^+$, we mean that $u_i^- \leq u_i$ and $u_i \leq u_i^+$, respectively, for all i . At each time instant t , it is required to solve for the optimal control law $\mathbf{u}_o^*(t)$ to problem (2.6)–(2.9). It is assumed that there exists an optimal control law for optimal control problem (2.6), i.e., the problem is solvable.

We offer the following remarks about optimal control problem (2.6)–(2.9) with respect to underactuated system (2.1).

Remark 2.1 As mentioned in the Introduction part, projection neural networks have been successfully adopted to solve redundancy resolution problems of redundant robot manipulators, which are often formulated as a constrained optimization problem. However, problem (2.6)–(2.9) differs from redundancy resolution problems of robot manipulators in terms of the performance index. More importantly, the system considered in problem (2.6)–(2.9) is underactuated since the number of inputs of

the system is less than the number of freedom degrees of the system, i.e., $r < n$, which is quite different from redundant robot manipulator systems. In this sense, the methodology used in redundancy resolution of manipulators cannot be directly used to solve problem (2.6)–(2.9).

Remark 2.2 In light of the problem formulation, it is difficult to obtain an analytic solution to optimal control problem (2.6)–(2.9) since it is a constrained dynamic optimization problem with nonlinearity. Therefore, an approximation of the performance index is employed in this chapter. Intuitively, this is, to some extent, to obtain a feasible solution at the cost of optimality.

2.3.2 Reformulation

To relax the original optimal control problem shown in (2.6)–(2.9), the time-scale Taylor series expansion is employed in this subsection and the original optimal control problem is then reformulated as an approximated one. Such a relaxation not only makes the problem solvable but also provides an opportunity to fully use the information included in the system dynamics.

By the time-scale Taylor series expansion, we have

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

Letting $\mathbf{w} = [1, \tau, \dots, \tau^{\rho-1}/(\rho-1)!, \tau^\rho/\rho!]^T$ and $Y_d(t) = [\mathbf{y}_d(t), \dots, \mathbf{y}_d^{[\rho-1]}, \mathbf{y}_d^{[\rho]}(t)]$, we further have

$$\mathbf{y}_d(t + \tau) \approx Y_d(t)\mathbf{w}(\tau).$$

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

$$\mathbf{y}(t + \tau) \approx \mathbf{y}(t) + \tau \dot{\mathbf{y}}(t) + \cdots + \tau^\rho \frac{\mathbf{y}^{[\rho]}(t)}{\rho!}. \quad (2.11)$$

Let $Y_a(t) = [\mathbf{y}(t), \dots, \mathbf{y}^{[\rho-1]}, \mathbf{y}^{[\rho]}(t)]$. The approximation of $\mathbf{y}(t + \tau)$ is written in a more compact form:

$$\mathbf{y}(t + \tau) \approx Y_a(t)\mathbf{w}(\tau).$$

In terms of the control input $\mathbf{u}(t + \tau)$, for a small value of τ , it can be approximated by $\mathbf{u}(t)$, i.e., $\mathbf{u}(t + \tau) \approx \mathbf{u}(t)$. Then, we have the following approximation of the performance index $J(t)$ defined in Eq. (2.10):

$$J(t) \approx \int_0^T (Y_d(t)\mathbf{w}(\tau) - Y_a(t)\mathbf{w}(\tau))^T Q (Y_d(t)\mathbf{w}(\tau) - Y_a(t)\mathbf{w}(\tau)) d\tau, \quad (2.12)$$

which is denoted as $J_a(t)$. Therefore, the optimal control problem shown in (2.6)–(2.9) is relaxed to the following approximated optimal control problem:

$$\text{minimize} \quad J_a(t) \quad (2.13)$$

$$\text{subject to} \quad \dot{\mathbf{x}} = f(\mathbf{x}) + g(\mathbf{x})\mathbf{u}(t), \quad (2.14)$$

$$\mathbf{y}(t) = h(\mathbf{x}), \quad (2.15)$$

$$\mathbf{u}(t) \in \Omega. \quad (2.16)$$

The optimal control law for approximated optimal control problem (2.13)–(2.16) is denoted as $\mathbf{u}_a^*(t)$, which is also termed the approximated optimal control law to the original optimal control problem depicted in (2.6)–(2.9). To lay a basis for latter illustration, the optimal control law for optimal control problem (2.13)–(2.15) [i.e., without considering input constraint (2.16)] is denoted as $\mathbf{u}_n^*(t)$.

Assumption 1 It is assumed that $\mathbf{u}_n^*(t)$ exists. It is also assumed that there exists a constant $0 \leq t_c < \infty$ such that for all $t > t_c$, $\mathbf{u}_n^*(t) \in \Omega$. In other words, $\mathbf{u}_a^*(t) = \mathbf{u}_n^*(t)$ for all $t > t_c$.

The existence of $\mathbf{u}_n^*(t)$ guarantees that $\mathbf{u}_n^*(t)$ is bounded. Therefore, in practice, for a control task within a time interval, e.g., $[0, t_f]$, where t_f denotes the final time instant, there always exist a \mathbf{u}^+ such that $\mathbf{u}^+ > \sup_{t \in [0, t_f]} \mathbf{u}_n^*(t)$ and a \mathbf{u}^- such that $\mathbf{u}^- < \inf_{t \in [0, t_f]} \mathbf{u}_n^*(t)$.

2.4 Projection Neural Network Design

In this section, a projection neural network is designed to solve the optimal control problem. Based on the system dynamics, the approximated optimal control problem described in (2.13)–(2.16) is further converted into a convex quadratic program. Then, a projection neural network is constructed based on the quadratic program for handling the approximated optimal control of underactuated system (2.1).

From system (2.1) with relative degree ρ , via differentiating output $\mathbf{y}(t)$ with respect to time t for ρ times and using the Lie derivative notation, based on Definitions 2.2 and 2.3, the following equations based on the system dynamics are obtained

$$\begin{cases} \dot{\mathbf{y}}(t) = \frac{\partial h}{\partial \mathbf{x}} \dot{\mathbf{x}} = \frac{\partial h}{\partial \mathbf{x}} f(\mathbf{x}) = L_f h(\mathbf{x}), \\ \vdots \\ \mathbf{y}^{[\rho-1]}(t) = L_f^{\rho-1} h(\mathbf{x}), \\ \mathbf{y}^{[\rho]}(t) = L_f^\rho h(\mathbf{x}) + \frac{\partial L_f^{\rho-1} h(\mathbf{x})}{\partial \mathbf{x}} g(\mathbf{x}) \mathbf{u}(t) \\ = L_f^\rho h(\mathbf{x}) + L_g L_f^{\rho-1} h(\mathbf{x}) \mathbf{u}(t). \end{cases} \quad (2.17)$$

Substituting Eq.(2.17) into Eq.(2.11) results in the following approximation of $\mathbf{y}(t + \tau)$:

$$\mathbf{y}(t + \tau) \approx \mathbf{y}(t) + \tau L_f h(\mathbf{x}) + \cdots + \frac{\tau^\rho}{\rho!} L_f^\rho h(\mathbf{x}) + \frac{\tau^\rho}{\rho!} L_g L_f^{\rho-1} h(\mathbf{x}) \mathbf{u}(t),$$

which is actually predicting the system output via the system dynamics. With $Y(t) = [\mathbf{y}(t), L_f h(\mathbf{x}), \dots, L_f^{\rho-1} h(\mathbf{x})]$, a more compact form of the above equation is obtained

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

By substituting the above expression of $\mathbf{y}(t + \tau)$ predicted by the system dynamics into (2.12), performance index $J_a(t)$ is then rewritten as

$$\begin{aligned} & \int_0^T \left(Y_d(t) \mathbf{w}(\tau) - Y(t) \mathbf{w}(\tau) - \frac{\tau^\rho}{\rho!} L_g L_f^{\rho-1} h(\mathbf{x}) \mathbf{u}(t) \right)^T Q \\ & \cdot \left(Y_d(t) \mathbf{w}(\tau) - Y(t) \mathbf{w}(\tau) - \frac{\tau^\rho}{\rho!} L_g L_f^{\rho-1} h(\mathbf{x}) \mathbf{u}(t) \right) d\tau \\ &= \int_0^T (\mathbf{w}^T(\tau) E^T - \frac{\tau^\rho}{\rho!} \mathbf{u}^T(\tau) (L_g L_f^{\rho-1} h(\mathbf{x}))^T) Q (E \mathbf{w}(\tau) - \frac{\tau^\rho}{\rho!} L_g L_f^{\rho-1} h(\mathbf{x}) \mathbf{u}(t)) d\tau \\ &= \int_0^T (\mathbf{w}^T(\tau) E^T Q E \mathbf{w}(\tau) - 2 \frac{\tau^\rho}{\rho!} \mathbf{w}^T(\tau) E^T Q L_g L_f^{\rho-1} h(\mathbf{x}) \mathbf{u}(t) \\ & \quad + \frac{\tau^{2\rho}}{(\rho!)^2} \mathbf{u}^T(\tau) (L_g L_f^{\rho-1} h(\mathbf{x}))^T Q L_g L_f^{\rho-1} h(\mathbf{x}) \mathbf{u}(t)) d\tau \\ &= \int_0^T \mathbf{w}^T(\tau) E^T Q E \mathbf{w}(\tau) d\tau - 2 \int_0^T \frac{\tau^\rho}{\rho!} \mathbf{w}^T(\tau) d\tau E^T Q L_g L_f^{\rho-1} h(\mathbf{x}) \mathbf{u}(t) \\ & \quad + \int_0^T \frac{\tau^{2\rho}}{(\rho!)^2} d\tau \mathbf{u}^T(\tau) (L_g L_f^{\rho-1} h(\mathbf{x}))^T Q L_g L_f^{\rho-1} h(\mathbf{x}) \mathbf{u}(t), \end{aligned}$$

where $E = Y_d(t) - Y(t)$.

Let

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

and

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

Since the decision variable is $\mathbf{u}(t)$, minimizing performance index J_a of approximated optimal control problem (2.13)–(2.16) is then equivalent to minimizing $-2\mathbf{v}^T Q L_g L_f^{\rho-1} h(\mathbf{x}) \mathbf{u}(t) + \kappa \mathbf{u}^T(t) (L_g L_f^{\rho-1} h(\mathbf{x}))^T Q L_g L_f^{\rho-1} h(\mathbf{x}) \mathbf{u}(t)$ ([77]).

Let $\Theta = \kappa(L_g L_f^{\rho-1} h(\mathbf{x}))^\top Q L_g L_f^{\rho-1} h(\mathbf{x})$ and $\mathbf{p} = -2(L_g L_f^{\rho-1} h(\mathbf{x}))^\top Q^\top E \mathbf{v}^\top$. Performance index J_a is thus reformulated as the following quadratic performance index:

$$\Psi(t) = \mathbf{u}^\top(t) \Theta \mathbf{u}(t) + \mathbf{p}^\top \mathbf{u}(t),$$

which further converts approximated optimal control problem (2.13)–(2.16) into the following quadratic program:

$$\text{minimize} \quad \Psi(t) = \mathbf{u}^\top(t) \Theta \mathbf{u}(t) + \mathbf{p}^\top \mathbf{u}(t) \quad (2.18)$$

$$\text{subject to} \quad \mathbf{u}(t) \in \Omega, \quad (2.19)$$

where the system dynamics has already been incorporated into the performance index Ψ . Therefore, designing a projection neural network to solve the approximated optimal control problem of system (2.1) becomes designing a projection neural network to solve the above quadratic program. By following [70], a competent projection neural network to solve quadratic program (2.18)–(2.19) is designed as follows:

$$\lambda \dot{\mathbf{u}}(t) = -\mathbf{u}(t) + P_\Omega \left(\mathbf{u}(t) - \frac{\partial \Psi(t)}{\partial \mathbf{u}} \right), \quad (2.20)$$

where

$$\frac{\partial \Psi(t)}{\partial \mathbf{u}} = 2\Theta \mathbf{u}(t) + \mathbf{p} \quad (2.21)$$

with λ , \mathbf{u} , $\Theta(t)$, and \mathbf{p} are defined as the aforementioned.

2.5 Theoretical Analysis

In this section, theoretical analysis is provided about the time-scale expansion-based approximated optimal control scheme for underactuated system (2.1) by using a projection neural network (2.20).

Theorem 2.1 *Given that matrices Q and $L_g L_f^{\rho-1} h(\mathbf{x})$ are symmetric and positive definite, the projection neural network (2.20) globally and exponentially converges to optimal solution $\mathbf{u}^*(t)$ to quadratic program (2.18)–(2.19), i.e., the optimal control law $\mathbf{u}_a^*(t)$ to the approximated optimal control problem (2.13)–(2.16).*

Proof Since matrices Q and $L_g L_f^{\rho-1} h(\mathbf{x})$ are symmetric and positive definite, $\Theta = \kappa(L_g L_f^{\rho-1} h(\mathbf{x}))^\top Q L_g L_f^{\rho-1} h(\mathbf{x})$ is also symmetric and positive definite. Based on Lemma 2.1, since $\nabla \Psi = \partial \Psi / \partial \mathbf{u} = 2\Theta$ is symmetric and positive definite, projection neural network (2.20) is globally and exponentially stable, i.e., projection neural network (2.20) globally and exponentially converges to optimal solution $\mathbf{u}^*(t)$ to quadratic program (2.18)–(2.19), which is also the optimal control law $\mathbf{u}_a^*(t)$ to the approximated optimal control problem (2.13)–(2.16). The proof is complete. \square

Theorem 2.2 Given that matrices Q and $L_g L_f^{\rho-1} h(\mathbf{x})$ are symmetric and positive definite, with relative degree $\rho < 5$, the closed-loop system, which consists of system (2.1) and approximated optimal control law \mathbf{u}_a^* , is exponentially stable.

Proof For the convenience of presentation and readability, let $M = L_g L_f^{\rho-1} h(\mathbf{x})$. From (2.21), we have

$$\begin{aligned}\mathbf{u}_n^* &= -\frac{1}{2}\Theta^{-1}\mathbf{p} \\ &= -\frac{1}{2}(\kappa M^T Q M)^{-1}(-2M^T Q^T E \mathbf{v}^T) \\ &= (\kappa M^T Q M)^{-1} M^T Q^T E \mathbf{v}^T.\end{aligned}\quad (2.22)$$

Based on Assumption 1, the proof falls into two situations, which are specified as follows.

Situation 1: When $t \in (t_c, +\infty)$, based on Assumption 1, we have $\mathbf{u}_a^*(t) = \mathbf{u}_n^*(t)$. Substituting $\mathbf{u}_a^*(t)$ into Eq. (2.17), i.e.,

$$\mathbf{y}^{[\rho]}(t) = L_f^\rho h(\mathbf{x}) + L_g L_f^{\rho-1} h(\mathbf{x}) \mathbf{u}(t) \quad (2.23)$$

yields

$$\mathbf{y}^{[\rho]}(t) = L_f^\rho h(\mathbf{x}) + M(\kappa M^T Q M)^{-1} M^T Q^T E \mathbf{v}^T. \quad (2.24)$$

Let $(\kappa M^T Q M)^*$ and $|\kappa M^T Q M|$ denote the adjoint matrix and the determinant of $\kappa M^T Q M$, respectively. Then, Eq. (2.24) is written as

$$\mathbf{y}^{[\rho]}(t) = L_f^\rho h(\mathbf{x}) + M \frac{(\kappa M^T Q M)^*}{|\kappa M^T Q M|} M^T Q^T E \mathbf{v}^T.$$

In view of the definitions of E and \mathbf{v} , we further have

$$\begin{aligned}|\kappa M^T Q M| \mathbf{y}^{[\rho]}(t) &= |\kappa M^T Q M| L_f^\rho h(\mathbf{x}) + M(\kappa M^T Q M)^* M^T Q^T E \mathbf{v}^T \\ &= |\kappa M^T Q M| L_f^\rho h(\mathbf{x}) + M(\kappa M^T Q M)^* M^T Q^T \sum_{i=0}^{\rho-1} \frac{T^{\rho+i+1}}{(\rho+i+1)\rho!i!} \mathbf{e}^{[i]}(t) \\ &\quad + M(\kappa M^T Q M)^* M^T Q^T \kappa \mathbf{y}_d^{[\rho]}(t) - M(\kappa M^T Q M)^* M^T Q^T \kappa L_f^\rho h(\mathbf{x}).\end{aligned}$$

Note that, for matrix $A \in \mathbb{R}^{r \times r}$, vector $\mathbf{x} \in \mathbb{R}^r$, and number $c \in \mathbb{R}$, $A\mathbf{x} + c\mathbf{x} = (A + cI)\mathbf{x}$, where I is the $r \times r$ identity matrix. It follows that the following equation holds

$$|\kappa M^T Q M| \mathbf{e}^{[\rho]}(t) + M(\kappa M^T Q M)^* M^T Q^T \sum_{i=0}^{\rho-1} \frac{T^{\rho+i+1}}{(\rho+i+1)\rho!i!} \mathbf{e}^{[i]}(t)$$

$$\begin{aligned}
& + (M(\kappa M^T Q M)^* M^T Q^T \kappa - |\kappa M^T Q M| I) \mathbf{y}_d^{[\rho]}(t) + (|\kappa M^T Q M| I \\
& - M(\kappa M^T Q M)^* M^T Q^T \kappa) L_f^\rho h(\mathbf{x}) \\
& = 0.
\end{aligned}$$

Then, one further has

$$\kappa^r |M^T Q M| \mathbf{e}^{[\rho]} + \kappa^{r-1} |M^T Q M| \sum_{i=0}^{\rho-1} \frac{T^{\rho+i+1}}{(\rho+i+1)\rho!i!} \mathbf{e}^{[i]} = 0,$$

i.e.,

$$\sum_{i=0}^{\rho} \frac{T^i}{(\rho+i+1)i!} \mathbf{e}^{[i]} = 0, \quad (2.25)$$

of which the characteristic polynomial is $P(\gamma) = \sum_{i=0}^{\rho} T^i \gamma^i / ((\rho+i+1)i!)$. According to the Routh–Hurwitz criterion [78], the real parts of all the roots of the characteristic polynomial $P(\gamma)$ are negative if $\rho < 5$, from which, it is further concluded that the closed-loop system governed by Eq. (2.25) is exponentially stable.

Situation 2: When $t \in [0, t_c]$, define a constant vector $\hat{\mathbf{e}} \in \mathbb{R}^r$, of which $\hat{e}_k = \max_{0 \leq t \leq t_c} \{\max\{\|u_{nk}^*(t) - u_{ak}^*(t)\|_2, \|u_{ak}^*(t) - u_{nk}^*(t)\|_2\}\}$ with $k = 1, 2, \dots, r$. Then, we have $\mathbf{u}_a^*(t) - \hat{\mathbf{e}} \leq \mathbf{u}_a^*(t) \leq \mathbf{u}_n^*(t) + \hat{\mathbf{e}}$. On one hand, from $\mathbf{u}_a^*(t) \geq \mathbf{u}_n^*(t) - \hat{\mathbf{e}}$, we have

$$\mathbf{u}_a^*(t) \geq (\kappa M^T Q M)^{-1} M^T Q^T E \mathbf{v}^T - \hat{\mathbf{e}}.$$

Substituting the above inequality into Eq. (2.23) yields

$$\mathbf{y}^{[\rho]}(t) \geq L_f^\rho h(\mathbf{x}) + M(\kappa M^T Q M)^{-1} M^T Q^T E \mathbf{v}^T - M \hat{\mathbf{e}}.$$

Then, similar to the proof for situation 1, the following inequality is obtained

$$\begin{aligned}
& - |\kappa M^T Q M| M \hat{\mathbf{e}} \leq \kappa^r |M^T Q M| \mathbf{e}^{[\rho]} \\
& + \kappa^{r-1} |M^T Q M| \sum_{\varsigma=0}^{\rho-1} \frac{T^{\rho+\varsigma+1}}{(\rho+\varsigma+1)\rho!\varsigma!} \mathbf{e}^{[\varsigma]} \leq |\kappa M^T Q M| M \hat{\mathbf{e}},
\end{aligned}$$

i.e.,

$$-c M \hat{\mathbf{e}} \leq \sum_{i=0}^{\rho} \frac{T^i}{(\rho+i+1)i!} \mathbf{e}^{[i]} \leq c M \hat{\mathbf{e}},$$

where $c = \kappa \rho! / T^{\rho+1} > 0$. When $\rho < 5$, based on the Routh–Hurwitz criterion [78] and ordinary differential equation theory, the solution to the above inequality is bounded for all $t \in [0, t_c]$.

By summarizing the analysis on the two situations, the closed-loop system consisting of system (2.1) and the approximated optimal control law \mathbf{u}_a^* is exponentially stable. The proof is thus complete. \square

Theorem 2.2 shows the stability of the closed-loop system consisting of system (2.1) and approximated optimal control law \mathbf{u}_a^* . Note that, the approximated optimal control law \mathbf{u}_a^* is obtained via projection neural network (2.20). In other words, the actual control is implemented via the projection neural network. Since there is a transient before the projection neural network converges to \mathbf{u}_a^* , it is worth analyzing the stability of the closed-loop system consisting of system (2.1) and projection neural network (2.20), which is conducted in the following theorem.

Theorem 2.3 *Given that matrices Q and $L_g L_f^{\rho-1} h(\mathbf{x})$ are symmetric and positive definite, with $\rho < 5$ and $\lambda > 0 \in \mathbb{R}$ being small enough, the equilibrium of closed-loop system consisting of system (2.1) and projection neural network (2.20) is exponentially stable with tracking error $\mathbf{e}(t) = \mathbf{y}_d(t) - \mathbf{y}(t)$ exponentially converging to zero.*

Proof Let $\psi = [\mathbf{e}, \dot{\mathbf{e}}, \dots, \mathbf{e}^{[\rho-1]}]^T$. The closed-loop system consisting of underactuated system (2.1) and projection neural network (2.20) is described as follows:

$$\left\{ \begin{array}{l} \dot{\psi}(t) = \gamma(\psi(t), \mathbf{u}(t), \lambda) \\ = \begin{bmatrix} \dot{\mathbf{e}} \\ \mathbf{e}^{[2]} \\ \dots \\ \mathbf{y}_d^{[\rho]} - L_f^\rho h(\mathbf{x}) - L_g L_f^{\rho-1} h(\mathbf{x}) \mathbf{u}(t) \end{bmatrix}, \\ \lambda \dot{\mathbf{u}} = F(\psi(t), \mathbf{u}(t), \lambda) = -\mathbf{u}(t) + P_{\mathcal{Q}} \left(\mathbf{u}(t) - \frac{\partial \Psi}{\partial \mathbf{u}} \right), \end{array} \right. \quad (2.26)$$

of which the equilibrium point is $(\psi(t) = 0, \mathbf{u}(t) = \mathbf{u}_n^*(t))$. Evidently, the above system is a special case of singularly perturbed system (2.4). Therefore, the stability of the equilibrium point is evaluated via Lemma 2.2.

Situation 1: When $t \in [0, t_c]$, based on Theorem 2.2, tracking error $\mathbf{e}(t)$ is bounded, and based on Theorem 2.1, projection neural network (2.20) is exponentially convergent.

Situation 2: When $t \in (t_c, +\infty)$, based on Assumption 1 and Theorem 2.2, we have $\mathbf{u}_a^*(t) = \mathbf{u}_n^*(t)$, and, given that matrices Q and $L_g L_f^{\rho-1} h(\mathbf{x})$ are symmetric and positive definite, with $\rho < 5$, the following equation holds

$$\dot{\psi}(t) = \begin{bmatrix} \dot{\mathbf{e}} \\ \mathbf{e}^{[2]} \\ \dots \\ -\frac{1}{\kappa} \sum_{i=0}^{\rho-1} K_i \mathbf{e}^{[i]} \end{bmatrix}. \quad (2.27)$$

Table 2.1 Parameter values of the Single-Link Flexible Joint manipulator

Parameter	K_s	J_h	J_l	K_m	K_g	R_m	m	g	h
Value	1.61	0.0021	0.0059	0.00767	70	2.6	0.403	-9.81	0.06

Then, we have $\gamma(0, \mathbf{u}_n^*, \lambda) = 0$. Since $\mathbf{u}_a^*(t) = \mathbf{u}_n^*(t)$, in terms of $F(0, \mathbf{u}_n^*, \lambda)$, we have

$$F(0, \mathbf{u}_n^*, \lambda) = -\mathbf{u}_a^* + P_\Omega \left(\mathbf{u}_a^* - \frac{\partial \Psi}{\partial \mathbf{u}}|_{\mathbf{u}=\mathbf{u}_a^*} \right) = 0. \quad (2.28)$$

Therefore, condition (1) is satisfied. From Theorem 2.1, it is evident that $F(\psi, \mathbf{u}, \lambda)$ has an isolated root $\mathbf{u} = \mathbf{u}_a^*(t)$, i.e., condition (2) is satisfied. From Eq.(2.26), Theorem 2.1, and the assumptions imposed on system (2.1), functions γ , F , and their partial derivative up to the second order are bounded for $\mathbf{u}(t) - \mathbf{u}_a^*(t)$, i.e., condition (3) is satisfied. From Theorem 2.2, given that matrices Q and $L_g L_f^{\rho-1} h(\mathbf{x})$ are symmetric and positive definite, with relative degree $\rho < 5$, equilibrium point $(\psi = 0, \mathbf{u}(t) = \mathbf{u}_a^*(t))$ of system $\dot{\psi}(t) = \gamma(\psi(t), \mathbf{u}(t), \lambda)$ is exponentially stable, i.e., condition (4) is satisfied. In addition, the origin (i.e., $\mathbf{u}(t) - \mathbf{u}_a^*(t) = 0$) of the bounded-layer system for system (2.26),

$$\frac{d(\mathbf{u} - \mathbf{u}^*)}{d\hat{\tau}} = F(\psi(t), \mathbf{u}(t), \lambda), \quad (2.29)$$

where $\hat{\tau} = t/\lambda$ is exponentially stable in view of Theorem 2.1 and condition (5) is thus satisfied.

Since all the five conditions are satisfied for all $t \in (t_c, +\infty)$, according to Lemma 2.2, there exists $\lambda^* > 0 \in \mathbb{R}$ such that for all $\lambda < \lambda^*$, the equilibrium point $(\psi(t) = 0, \mathbf{u} = \mathbf{u}_a^*)$ of the closed-loop system is exponentially stable. By summarizing the analysis on Situation 1 and Situation 2, the equilibrium point $(\psi(t) = 0, \mathbf{u}(t) = \mathbf{u}_n^*(t))$ of the closed-loop system consisting of system (2.1) and projection neural network (2.20) is exponentially stable. It follows that the tracking error $\mathbf{e}(t) = \mathbf{y}_d(t) - \mathbf{y}(t)$ exponentially converges to zero. The proof is thus complete. \square

Remark 2.3 Based on Theorem 2.3, parameter λ of the projection neural network adopted in the control scheme presented in this chapter should be set as small as possible to guarantee the overall stability of the closed-loop system consisting of underactuated system (2.1) and projection neural network (2.20). Our simulation experience indicates that the order of parameter λ should not be larger than 10^{-7} .

To validate the feasibility of the approximation for the original optimal control problem via time-scale series expansion, we offer the following theorem concerning the optimality of the performance index.

Theorem 2.5 Given that matrices Q and $L_g L_f^{\rho-1} h(\mathbf{x})$ are symmetric and positive definite, $\rho < 5$ and $\lambda > 0 \in \mathbb{R}$ is small enough, performance index $J(t)$ of system (2.1) synthesized by projection neural network (2.20) satisfies $\lim_{t \rightarrow \infty} J(t) = 0$.

Proof Based on Taylor series expansion and in view of Eqs. (2.10) and (2.12) about $J(t)$ and $J_a(t)$, we have

$$J(t) = \int_0^T (Y_d(t)\mathbf{w}(\tau) - Y_a(t)\mathbf{w}(\tau) + \Delta)^T Q (Y_d(t)\mathbf{w}(\tau) - Y_a(t)\mathbf{w}(\tau) + \Delta) d\tau,$$

where $\Delta = \tau^\rho (\mathbf{y}_d^{[\rho]}(t + \varkappa_1 \tau) - \mathbf{y}^{[\rho]}(t + \varkappa_1 \tau) - (\mathbf{y}_d^{[\rho]}(t) - \mathbf{y}^{[\rho]}(t))) / (\rho!)$ with $\varkappa \in (0, 1)$. Based on triangle inequality, we further have

$$\begin{aligned} J(t) &\leq 2J_a(t) + 2 \int_0^T \frac{\tau^{2\rho}}{(\rho!)^2} \Delta^T Q \Delta d\tau \\ &= 2J_a(t) + 2 \frac{T^{2\rho+1}}{(2\rho+1)(\rho!)^2} \Delta^T Q \Delta \\ &\leq 2J_a(t) + 2 \frac{T^{2\rho+1}}{(2\rho+1)(\rho!)^2} \sup_{\varkappa \in (0,1)} \|\Delta\|_2^2 \|Q\|_2, \end{aligned}$$

where $\|\cdot\|_2$ denotes the Euclidean norm of a vector or a matrix. Given that matrices Q and $L_g L_f^{\rho-1} h(\mathbf{x})$ are symmetric and positive definite, $\rho < 5$ and $\lambda > 0$ is small enough, from Theorem 2.3, the output of system (2.1) synthesized by neural network (2.20) satisfies $\mathbf{y}(t) \rightarrow \mathbf{y}_d(t)$ when $t \rightarrow +\infty$, indicating $\mathbf{y}^{[\rho]}(t) \rightarrow \mathbf{y}_d^{[\rho]}(t)$, i.e., $\Delta \rightarrow 0$. Then, we have

$$0 \leq \lim_{t \rightarrow \infty} J(t) \leq \lim_{t \rightarrow 0} 2J_a(t) = 0.$$

As a result, $\lim_{t \rightarrow \infty} J(t) = 0$. The proof is thus complete. \square

Remark 2.4 Generally speaking, there are two approaches to introduce a constraint to input \mathbf{u} in the problem formulation. The first one is to introduce a bounded constraint as $\mathbf{u}(t) \in \Omega$, which yields an optimization problem subject to a bounded constraint. The second one is to introduce a penalty term about input $\mathbf{u}(t)$ in the performance index $J(t)$. For the method presented in this chapter, the input constraint is considered by the first approach, i.e., adding a bounded constraint as $\mathbf{u}(t) \in \Omega$ to the optimal control problem.

2.6 Numerical Investigation

In this section, we compare the projection neural network-based approximated optimal control scheme aided by time-scale expansion with an existing finite-horizon controller based on a single-link flexible joint manipulator, which shows the superiority and efficacy of the presented scheme. The application to an underactuated ship further verifies the efficacy of the scheme.

2.6.1 Performance Comparison for Manipulator Control

In this example, a single-input single-output underactuated nonlinear system (i.e., a single-link flexible joint manipulator presented in [79]) is considered, which is described as follows:

$$\begin{cases} \dot{x}_1 = x_2, \\ \dot{x}_2 = \frac{K_s}{J_h}x_3 - \frac{K_m^2 K_g^2}{R_m J_h}x_2 + \frac{K_m K_g}{R_m J_h}u, \\ \dot{x}_3 = x_4, \\ \dot{x}_4 = -\frac{K_s}{J_h}x_3 + \frac{K_m^2 K_g^2}{R_m J_h}x_2 - \frac{K_m K_g}{R_m J_h}u - \frac{K_s}{J_l}x_3 + \frac{mgh}{J_l} \sin(x_1 + x_3), \\ y = x_1 + x_3, \end{cases} \quad (2.30)$$

where $\mathbf{x} = [x_1, x_2, x_3, x_4]^T \in \mathbb{R}^4$ is the state vector, $u \in \mathbb{R}$ is the control input, and $y \in \mathbb{R}$ is the system output. Values of parameters $K_s > 0 \in \mathbb{R}$, $J_h > 0 \in \mathbb{R}$, $J_l > 0 \in \mathbb{R}$, $K_m > 0 \in \mathbb{R}$, $K_g > 0 \in \mathbb{R}$, $R_m > 0 \in \mathbb{R}$, $m > 0 \in \mathbb{R}$, $g < 0 \in \mathbb{R}$, and $h > 0 \in \mathbb{R}$ of the single-link flexible joint manipulator are listed in Table 2.1 and the units of the parameters can be found in [79]. The desired output is denoted as $y_d(t)$.

For system (2.30), we have the following equations:

$$\begin{cases} \dot{y} = x_2 + x_4 = L_f h(\mathbf{x}), \\ y^{[2]} = -\frac{K_s}{J_l}x_3 + \frac{mgh}{J_l} \sin(x_1 + x_3) = L_f^2 h(\mathbf{x}), \\ y^{[3]} = -\frac{K_s}{J_l}x_4 + \frac{mgh}{J_l}x_2 \cos(x_1 + x_3) + \frac{mgh}{J_l} \cos(x_1 + x_3)x_4 = L_f^3 h(\mathbf{x}), \\ y^{[4]} = \frac{K_s^2}{J_l J_h}x_3 - \frac{K_s K_m^2 K_g^2}{J_l R_m J_h}x_2 + \frac{K_s^2}{J_l^2}x_3 - \frac{mgh K_s}{J_l^2} \sin(x_1 + x_3) \\ \quad - \frac{mgh}{J_l} \sin(x_1 + x_3)(x_2^2 + 2x_2 x_4 + x_4^2) - \frac{mgh K_s}{J_l^2} \cos(x_1 + x_3)x_3 \\ \quad + \left(\frac{mgh}{J_l}\right)^2 \cos(x_1 + x_3) \sin(x_1 + x_3) + \frac{K_s K_m K_g}{J_l R_m J_h}u = L_f^4 h(\mathbf{x}) + L_g L_f^3 h(\mathbf{x})u, \end{cases}$$

which indicate that this system has a relative degree of 4, i.e., $\rho = 4$.

In the simulation, the output of system (2.30) is expected to track desired output $y_d(t) = \pi \sin(t)/6 + 0.1$ with $Q = 1$. The lower bound and upper bound for control input u are specified as $u^- = -0.8$ and $u^+ = 0.8$, respectively. In addition, the parameters of the presented control scheme are set to $\lambda = 10^{-7}$ and $T = 0.1$ s. The initial values of the state variables of system (2.30) and projection neural network (2.20) are set to $x_1(0) = x_2(0) = x_3(0) = x_4(0) = u(0) = 0$. For a 20 s simulation of system (2.30) synthesized by projection neural network (2.20), the results are shown in Fig. 2.1. As seen from Fig. 2.1b, d, output y of the single-link flexible joint manipulator system (2.30) successfully tracks the desired output and tracking error $e(t) = y_d(t) - y(t)$ exponentially converges to zero. In addition, Fig. 2.1c shows that

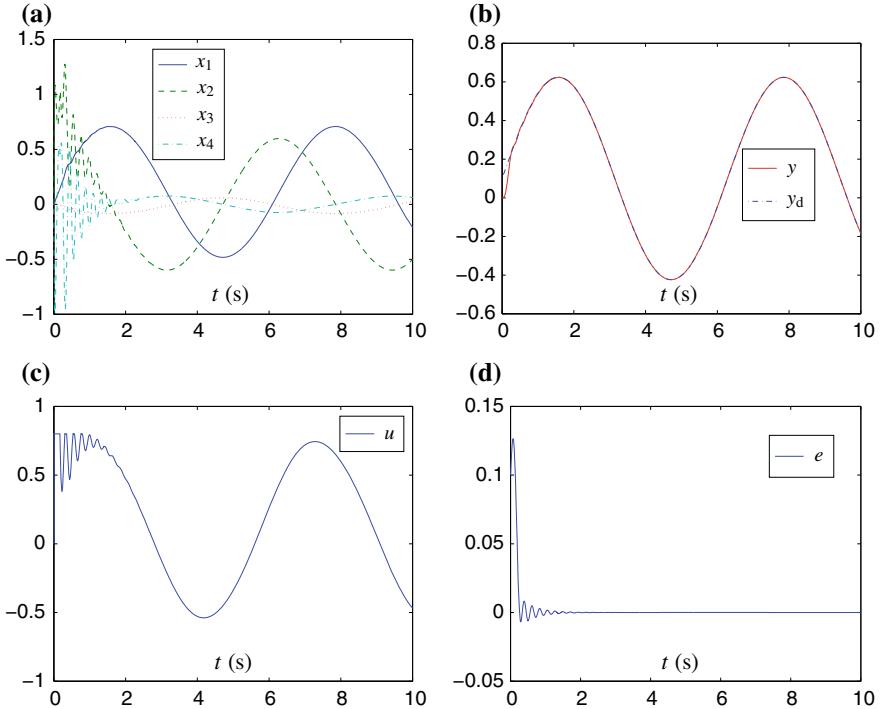


Fig. 2.1 Tracking performance of single-link flexible joint manipulator system (2.30) synthesized by projection neural network (2.20) with $y_d = \pi \sin(t)/6 + 0.1$, $Q = 1$, $u^- = -0.8$, $u^+ = 0.8$, $\lambda = 10^{-7}$, and $T = 0.1$ s. **a** Trajectories of state variables. **b** Trajectories of actual output y and desired output y_d . **c** Trajectory of control input u

control input u always lies between u^- and u^+ . In other words, the tracking control task is successfully completed with the requirement on the control input satisfied, which substantiates the efficacy of the presented control scheme and verifies the theoretical results.

For comparison, with the same setup, simulation results of the single-link flexible joint manipulator system (2.30) synthesized by controller (7) in [7], which deals with the optimal control problem with the same performance index, are shown in Fig. 2.2. As seen from Fig. 2.2, although output $y(t)$ of the single-link flexible joint manipulator system successfully tracks the desired output $y_d(t)$, it is evident that the magnitude of control input u generated via controller (7) in [7] is larger than u^+ at some time instants $t \in (0, 2]$ s. In addition, from Figs. 2.2b, d and 2.1b, d, it is seen that controller (7) in [7] causes more overshooting than that of the presented control scheme. It is also seen that error convergence of the system synthesized by neural network (2.20) is faster than that synthesized by controller (7) in [7]. These results substantiate the superiority of the presented control scheme in terms of the error convergence speed, overshooting, and input magnitude.

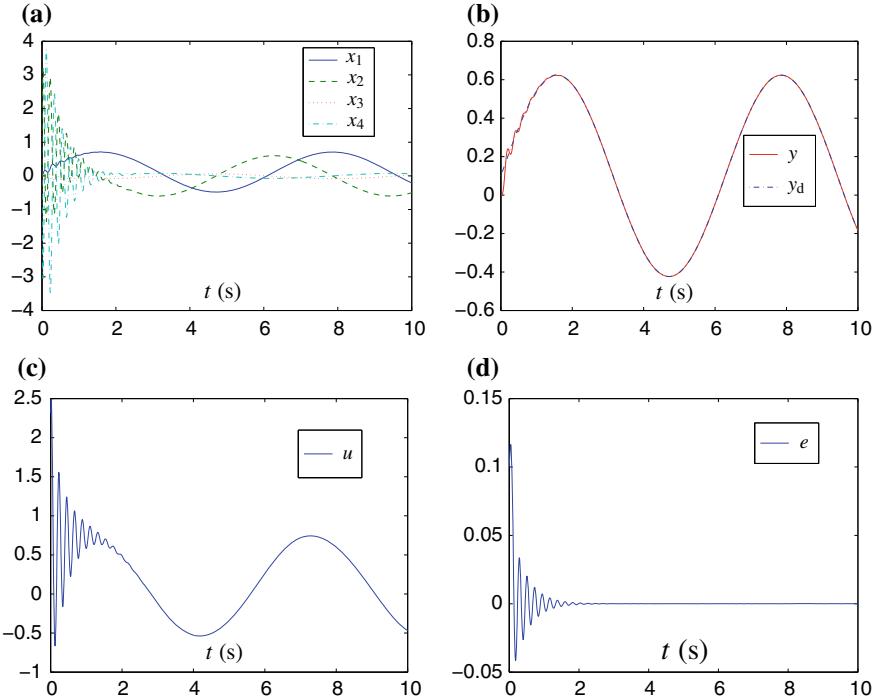


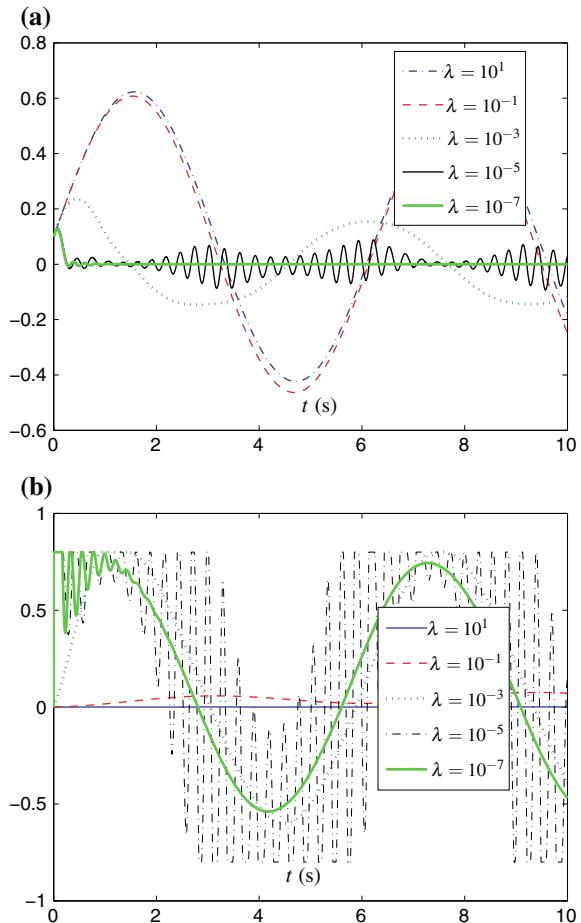
Fig. 2.2 Tracking performance of single-link flexible joint manipulator system (2.30) synthesized by controller (7) in [7] with $y_d = \pi \sin(t)/6 + 0.1$, $Q = 1$, $T = 0.1$ s. **a** Trajectories of state variables **b** Trajectories of actual output y and desired output y_d . **c** Trajectory of control input u

To show the effect of the value of parameter λ on the performance of the presented projection neural network (2.20), simulations are performed with different values of λ and the other setups are the same as the aforementioned. As seen from Fig. 2.3a, given a smaller value of parameter $\lambda > 0$, better error convergence is achieved. Besides, as seen from Fig. 2.3b, the inputs generated by projection neural network (2.20) always satisfy the input constraint, i.e., the value of $u(t)$ is always within the interval $[-0.8, 0.8]$.

2.6.2 Application to Underactuated Ships

In this example, we consider a surface ship 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 ship are governed by the following ordinary differential equations [80]:

Fig. 2.3 Trajectories of tracking error $e = y_d - y$ and control input u when the presented projection neural network (2.20) is adopted for the single-link flexible joint manipulator system (2.30) with different values of λ . **a** Trajectories of tracking error e . **b** Trajectories of control input u



$$\begin{cases} \dot{x}_1 = x_4 \cos x_3 - x_5 \sin x_3, \\ \dot{x}_2 = x_4 \sin x_3 + x_5 \cos x_3, \\ \dot{x}_3 = x_6, \\ \dot{x}_4 = \frac{m_{22}}{m_{11}}x_5x_6 - \frac{d_{11}}{m_{11}}x_4 + \frac{1}{m_{11}}u_1, \\ \dot{x}_5 = -\frac{m_{11}}{m_{22}}x_4x_6 - \frac{d_{22}}{m_{22}}x_5, \\ \dot{x}_6 = \frac{m_{11} - m_{22}}{m_{33}}x_4x_5 - \frac{d_{33}}{m_{33}}x_6 + \frac{1}{m_{33}}u_2, \end{cases} \quad (2.31)$$

where (x_1, x_2) denotes the coordinate of the mass center of the ship in the earth-fix 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.

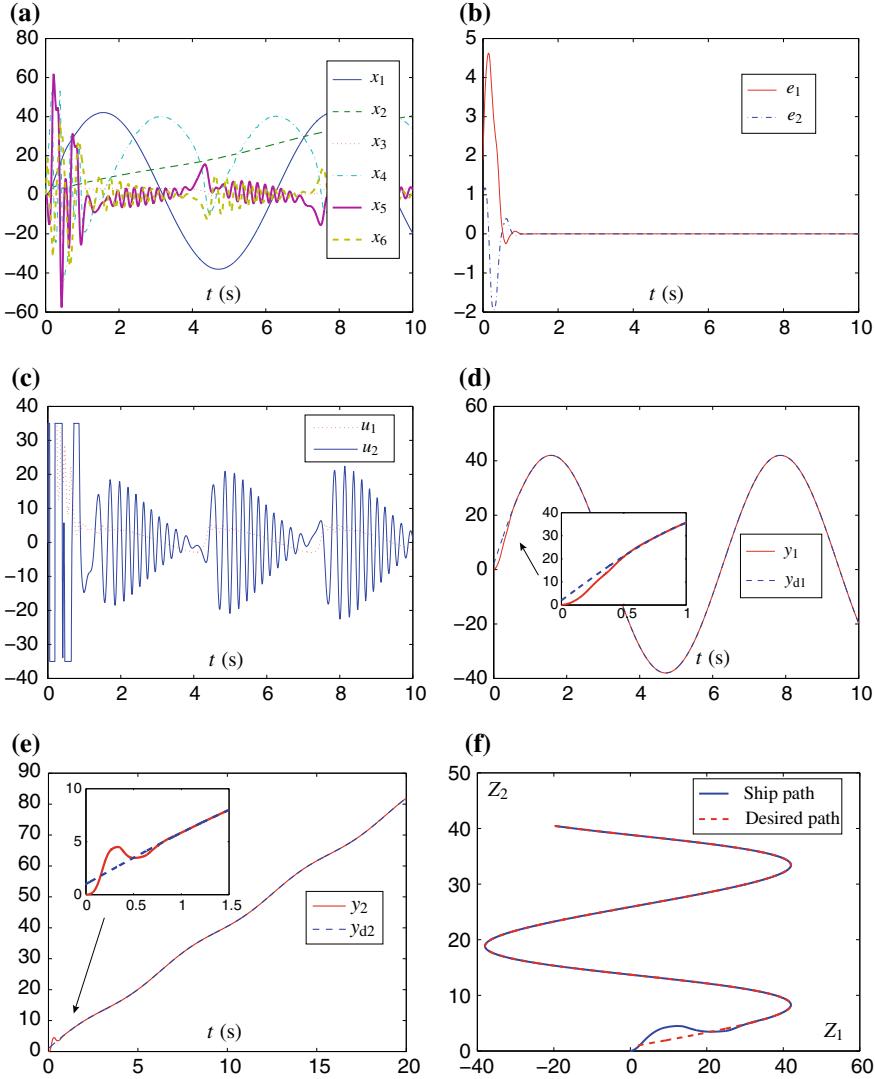


Fig. 2.4 Tracking performance of ship system (2.31) synthesized by projection neural network (2.20) with desired output $\mathbf{y}_d = [2 \sin(t) + 2, t + 1]^T$, neural network parameter $\lambda = 10^{-9}$, predictive period $T = 0.1$ s, and bounds of control input \mathbf{u} given as $\mathbf{u}^- = [-35, -35]^T$ and $\mathbf{u}^+ = [35, 35]^T$. **a** Trajectories of state variables. **b** Trajectories of tracking errors $e_1 = y_{d1} - y_1$ and $e_2 = y_{d2} - y_2$. **c** Trajectory of control inputs u_1 and u_2 . **d** Trajectories of output $y_1 = x_1 + l \cos x_3$ and desired output y_{d1} . **e** Trajectories of output $y_2 = x_2 + l \sin x_3$ and desired output y_{d2} . **f** The actual path of the ship and the desired path

In addition, parameters $m_{11}, m_{22}, m_{33}, d_{11}, d_{22}$, and d_{33} are positive, which are assumed to be constant and are given by the ship inertia and damping matrices [80]. Evidently, the state vector is $\mathbf{x} = [x_1, x_2, x_3, x_4, x_5, x_6]^T$; the input vector is

$\mathbf{u} = [u_1, u_2]^T$. To avoid the singularity problem [80], the considered output of the ship system is defined as follows:

$$\mathbf{y} = \begin{bmatrix} y_1 \\ y_2 \end{bmatrix} = \begin{bmatrix} x_1 + l \cos x_3 \\ x_2 + l \sin x_3 \end{bmatrix}, \quad (2.32)$$

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 ship.

For the ship system considered in this example, we have

$$\dot{\mathbf{y}} = \begin{bmatrix} x_4 \cos x_3 - x_5 \sin x_3 - lx_6 \sin x_3 \\ x_4 \sin x_3 + x_5 \cos x_3 + lx_6 \cos x_3 \end{bmatrix} = L_f h(\mathbf{x}).$$

Then, the following equations are derived

$$\begin{aligned} y_1^{[2]} = & \frac{m_{22}}{m_{11}} x_5 x_6 \cos x_3 - \frac{d_{11}}{m_{11}} x_4 \cos x_3 - x_4 x_6 \sin x_3 + \frac{m_{11}}{m_{22}} x_4 x_6 \sin x_3 + \frac{d_{22}}{m_{22}} x_5 \sin x_3 \\ & - x_5 x_6 \cos x_3 - \frac{m_{11} - m_{22}}{m_{33}} l x_4 x_5 \sin x_3 + \frac{d_{33}}{m_{33}} l x_6 \sin x_3 - l x_6^2 \cos x_3 \\ & + \frac{1}{m_{11}} \cos x_3 u_1 - \frac{1}{m_{33}} l \sin x_3 u_2, \end{aligned}$$

$$\begin{aligned} y_2^{[2]} = & \frac{m_{22}}{m_{11}} x_5 x_6 \sin x_3 - \frac{d_{11}}{m_{11}} x_4 \sin x_3 + x_4 x_6 \cos x_3 - \frac{m_{11}}{m_{22}} x_4 x_6 \cos x_3 - \frac{d_{22}}{m_{22}} x_5 \cos x_3 \\ & - x_5 x_6 \sin x_3 + \frac{m_{11} - m_{22}}{m_{33}} l x_4 x_5 \cos x_3 - \frac{d_{33}}{m_{33}} l x_6 \cos x_3 - l x_6^2 \sin x_3 \\ & + \frac{1}{m_{11}} \sin x_3 u_1 + \frac{1}{m_{33}} l \cos x_3 u_2, \end{aligned}$$

which indicate that ship system (2.31) considered in this example has a relative degree of 2, i.e., $\rho = 2$.

In the simulation, according to [80], the parameters of ship system (2.30) are set to $m_{11} = m_{22} = m_{33} = 0.1$, $d_{11} = d_{33} = 0$, and $d_{22} = 0.2$. The output of the ship system is expected to track desired output $\mathbf{y}_d = [2 \sin(t) + 2, t + 1]^T$ with $Q = [5, 2; 2, 5]$. The lower bound and upper bound for control input \mathbf{u} is specified as $\mathbf{u}^- = [-30, -30]^T$ and $\mathbf{u}^+ = [30, 30]^T$, respectively. In addition, the parameters of the presented control scheme are set to $\lambda = 10^{-9}$ and $T = 0.1$ s. The initial values of the state variables of system (2.30) and projection neural network (2.20) are set to $x_1(0) = x_2(0) = x_3(0) = x_4(0) = u(0) = 0$. In addition, the value of parameter l is set to 0.1. The results of a 20 s simulation with the above setup are shown in Fig. 2.4. As seen from this figure, the considered output $\mathbf{y}(t)$ successfully tracks the desired output $\mathbf{y}_d(t)$ with the tracking error $\mathbf{e}(t) = \mathbf{y}_d(t) - \mathbf{y}(t)$ exponentially converging to zero. Besides, during the simulation, the control input $\mathbf{u}(t)$ imposed on the ship

system always satisfies the constraint on it. In addition, Fig. 2.4f shows the ship path plotted by (y_1, y_2) data and the desired path plotted by (y_{d1}, y_{d2}) data during the simulation process, which further validates the convergence of position error during the control process. These results further validate the efficacy of the approximated optimal control scheme.

2.7 Questions and Answers

In this section, we present some discussions about the contents in this chapter in the form of questions and answers.

Question 2.1 “What is $\mathbf{e}^{[i]}$? Is it the i th column vector of matrix E ?”

Answer: $\mathbf{e}^{[i]}(t)$ is defined as $\mathbf{e}^{[i]}(t) = \mathbf{y}_d^{[i]}(t) - \mathbf{y}^{[i]}(t)$, where $\mathbf{y}_d^{[i]}(t)$ and $\mathbf{y}^{[i]}(t)$ denote the i th-order time derivatives of $\mathbf{y}_d(t)$ and $\mathbf{y}(t)$, respectively.

Question 2.2 “If the optimal control law does not fall into the constraint set, is it valid for the proposed scheme?”

Answer: Consider the optimal control problem:

$$\text{minimize} \quad J(t) \quad (2.33)$$

$$\text{subject to} \quad \dot{\mathbf{x}} = f(\mathbf{x}) + g(\mathbf{x})\mathbf{u}(t), \quad (2.34)$$

$$\mathbf{y}(t) = h(\mathbf{x}), \quad (2.35)$$

$$\mathbf{u}(t) \in \Omega, \quad (2.36)$$

where convex set $\Omega = \{\mathbf{u} \in \mathbb{R}^r | \mathbf{u}^- \leq \mathbf{u} \leq \mathbf{u}^+\}$ with constant vectors $\mathbf{u}^- \in \mathbb{R}^r$ and $\mathbf{u}^+ \in \mathbb{R}^r$ denoting the lower bound and upper bound of the constraint imposed on control input $\mathbf{u}(t)$, and the performance index $J(t)$ is defined as follows:

$$\begin{aligned} J(t) &= \int_0^T (\mathbf{y}_d(t + \tau) - \mathbf{y}(t + \tau))^T Q (\mathbf{y}_d(t + \tau) - \mathbf{y}(t + \tau)) d\tau \\ &\quad + \int_0^T \mathbf{u}^T(t + \tau) R \mathbf{u}(t + \tau) d\tau. \end{aligned} \quad (2.37)$$

An intuitive explanation about the optimal control problem is to find an input $\mathbf{u}(t)$ belonging to set Ω such that the performance index $J(t)$ about the system is minimized. Thus, the optimal control law must be feasible first, which means that it must belong to Ω . Consider the underactuated ship system shown in the manuscript as an example. The force in surge and the control torque in yaw (i.e., the inputs) of a ship are physically constrained, which means that their magnitude cannot be extremely large, which is captured by the set constraint $\mathbf{u}(t) \in \Omega$. The input that minimizes the performance index $J(t)$ cannot exceed the physical bounds. In conclusion, an

optimal control law must be feasible first, which means that it must fall into the constraint set.

Question 2.3 “*Why rho is taken less than 5?*”

Answer: By using the Routh–Hurwitz criterion to analyze the characteristic polynomial of the closed-loop system, it is found that when $\rho < 5$, the real parts of all the roots of the polynomial are negative. Then, by Lemma 4, the stability result is thus concluded. In other words, this condition (i.e., $\rho < 5$) guarantees the stability of the system.

Question 2.4 “*This work stresses on the presence of the input constraints. However, we remark that the matrix R is taken equal to zero, which means that the constraint on the control inputs is ignored.*”

Answer: For the convenience of illustration, the problem formulation is repeated as follows:

$$\text{minimize} \quad J(t) \quad (2.38)$$

$$\text{subject to} \quad \dot{\mathbf{x}} = f(\mathbf{x}) + g(\mathbf{x})\mathbf{u}(t), \quad (2.39)$$

$$\mathbf{y}(t) = h(\mathbf{x}), \quad (2.40)$$

$$\mathbf{u}(t) \in \Omega, \quad (2.41)$$

where convex set $\Omega = \{\mathbf{u} \in \mathbb{R}^r | \mathbf{u}^- \leq \mathbf{u} \leq \mathbf{u}^+\}$ with constant vectors $\mathbf{u}^- \in \mathbb{R}^r$ and $\mathbf{u}^+ \in \mathbb{R}^r$ denoting the lower bound and upper bound of the constraint imposed on control input $\mathbf{u}(t)$, and the performance index $J(t)$ is defined as follows:

$$\begin{aligned} J(t) = & \int_0^T (\mathbf{y}_d(t + \tau) - \mathbf{y}(t + \tau))^T Q (\mathbf{y}_d(t + \tau) \\ & - \mathbf{y}(t + \tau)) d\tau + \int_0^T \mathbf{u}^T(t + \tau) R \mathbf{u}(t + \tau) d\tau. \end{aligned} \quad (2.42)$$

Generally speaking, there are two approaches to introduce a constraint to input \mathbf{u} in the problem formulation. The first one is to introduce a bounded constraint as $\mathbf{u}(t) \in \Omega$, which yields an optimization problem subject to a bounded constraint. The second one is to introduce a nonzero R in the performance index $J(t)$. For the method proposed in this chapter, when $R = 0$, the input constraint is still considered by the first approach, i.e., adding a bounded constraint as $\mathbf{u}(t) \in \Omega$ to the optimization problem. This is also verified by Theorem 1 and the simulation results. For example, Fig. 2.1c in this chapter shows that, when $R = 0$, the control input generated via the proposed method satisfies the input constraint $u(t) \in [-0.8, 0.8]$ while that generated via an existing method violates the bound (please see Fig. 2.2c in this chapter). In addition, in the performance index, R and Q are used to scale the trade-off between tracking accuracy and control effort. When $R = 0$, it means that tracking accuracy is considered to be more important than control effort. In other words, the $R = 0$ case

in the chapter corresponds to achieving optimal tracking accuracy subject to limited input described by $\mathbf{u}(t) \in \Omega$.

Question 2.5 “*In the proof, why do you introduce the identity matrix in some terms and not in other terms?*”

Answer: In this chapter, the identity matrix is introduced somewhere based on a calculation rule about matrices and vectors. Specifically, for matrix $A \in \mathbb{R}^{r \times r}$, vector $\mathbf{x} \in \mathbb{R}^r$ and number $c \in \mathbb{R}$, $A\mathbf{x} + c\mathbf{x} = \mathbf{A}\mathbf{x} + cI\mathbf{x} = (A + cI)\mathbf{x}$, where I is the $r \times r$ identity matrix. According to this rule, from

$$\begin{aligned}|TR + \kappa M^T QM|\mathbf{y}^{[\rho]}(t) &= |TR + \kappa M^T QM|L_f^\rho h(\mathbf{x}) + M(TR + \kappa M^T QM)^* M^T Q^T E \mathbf{v}^T \\&= |TR + \kappa M^T QM|L_f^\rho h(\mathbf{x}) + M(TR + \kappa M^T QM)^* M^T Q^T \\&\quad \times \sum_{i=0}^{\rho-1} \frac{T^{\rho+i+1}}{(\rho+i+1)\rho i!} \mathbf{e}^{[i]}(t) + M(TR + \kappa M^T QM)^* M^T Q^T \\&\quad \times \kappa \mathbf{y}_d^{[\rho]}(t) - M(TR + \kappa M^T QM)^* M^T Q^T \kappa L_f^\rho h(\mathbf{x}),\end{aligned}$$

the following equation is readily obtained

$$\begin{aligned}|TR + \kappa M^T QM|\mathbf{e}^{[\rho]}(t) + M(TR + \kappa M^T QM)^* M^T Q^T \sum_{i=0}^{\rho-1} \frac{T^{\rho+i+1}}{(\rho+i+1)\rho i!} \mathbf{e}^{[i]}(t) \\+ (M(TR + \kappa M^T QM)^* M^T Q^T \kappa - |TR + \kappa M^T QM|I)\mathbf{y}_d^{[\rho]}(t) + (|TR + \kappa M^T QM|I \\- M(TR + \kappa M^T QM)^* M^T Q^T \kappa)L_f^\rho h(\mathbf{x}) = 0.\end{aligned}$$

Question 2.6 “*In the proof of Theorem 3, why do you introduce the exponent r ? I think it is just equal to one.*”

Answer: Note that, in this chapter, $|\cdot|$ denotes the determinant of a matrix, for which the explanation has been marked in blue color in the proof of Theorem 1. In the derivations, a property about the determinant of a matrix is used. Specifically, for $\kappa \in \mathbb{R}$ and $A \in \mathbb{R}^{r \times r}$, $|\kappa A| = \kappa^r |A|$. Thus, $|\kappa M^T QM| = \kappa^r |M^T QM|$.

Question 2.7 “*What is the effect of the value of λ while respecting the condition less than 10^{-7} ?*”

Answer: The value of λ scales the convergence speed of the projection neural network. Specifically, a smaller value of $\lambda > 0$ leads to faster convergence, which results in better convergence of the tracking error. For example, we have conducted simulations based on the single-link flexible joint manipulator system with respect to different values of parameter λ and the other setups are the same. As seen from the figures in the corresponding sections of this chapter, when the value of λ decreases from 10 to 10^{-7} , the tracking error converges better. Meanwhile, the value of $u(t)$

generated by the proposed projection neural network is always within the interval $[-0.8, 0.8]$.

Question 2.8 “What is $\mathbf{e}^{[i]}$? Is it the i th column vector of matrix E ?”

Answer: $\mathbf{e}^{[i]}(t)$ is defined as $\mathbf{e}^{[i]}(t) = \mathbf{y}_d^{[i]}(t) - \mathbf{y}^{[i]}(t)$, where $\mathbf{y}_d^{[i]}(t)$ and $\mathbf{y}^{[i]}(t)$ denote the i th-order time derivatives of $\mathbf{y}_d(t)$ and $\mathbf{y}(t)$, respectively.

2.8 Summary

In this chapter, with the input constraint taken into account, a time-scale expansion-based approximated optimal control scheme for underactuated systems via a projection neural network has been presented. Theoretical analysis has proved that, under some conditions, the closed-loop system consisting of the controlled underactuated system and the projection neural network is exponentially stable and the integral cost function converges to time-horizon optimal with time. The simulation comparison between the scheme and existing finite-horizon optimal controller for the control of a single-link flexible joint robot manipulator has validated the efficacy and superiority of the scheme. The application of the presented scheme to an underactuated ship has further substantiated the efficacy of the scheme and validated the theoretical conclusions drawn in this chapter.

References

1. Liu, Y.-J., Tong, S.: Optimal control-based adaptive NN design for a class of nonlinear discrete-time block-triangular systems. *IEEE Trans. Cybern.* **46**(11), 2670–2680 (2016)
2. Liu, Y.-J., Gao, Y., Tong, S., Li, Y.: Fuzzy approximation-based adaptive backstepping optimal control for a class of nonlinear discrete-time systems with dead-zone. *IEEE Trans. Fuzzy Syst.* **24**(1), 16–28 (2016)
3. Lewis, F.L., Syrmos, V.L.: *Optimal Control*. Wiley, New York (1995)
4. Chen, Z., Jagannathan, S.: Generalized Hamilton–Jacobi–Bellman formulation-based neural network control of affine nonlinear discrete-time systems. *IEEE Trans. Neural Netw.* **19**(1), 90–106 (2008)
5. Mayne, D.Q.: Model predictive control: recent developments and future promise. *Automatica* **50**(12), 2967–2986 (2014)
6. Chen, W.H., Ballance, D.J., Gawthrop, P.J.: Optimal control of nonlinear systems: a predictive control approach. *Automatica* **39**(4), 633–641 (2003)
7. Hedjar, R., Toumi, R., Boucher, P., Dumur, D.: Finite horizon nonlinear predictive control by the Taylor approximation application to robot tracking trajectory. *Int. J. Appl. Math. Comput. Sci.* **15**(4), 527–540 (2005)
8. Mayne, D.Q., Michalska, H.: Receding horizon control of nonlinear systems. *IEEE Trans. Autom. Control* **35**(7), 814–824 (1990)
9. Han, D., Shi, L.: Guaranteed cost control of affine nonlinear systems via partition of unity method. *Automatica* **49**(2), 660–666 (2013)

10. Lian, C., Xu, X., Chen, H., He, H.: Near-optimal tracking control of mobile robots via receding-horizon dual heuristic programming. *IEEE Trans. Cybern.* **46**(11), 2484–2496 (2016)
11. Li, H., Shi, Y., Yan, W.: On neighbor information utilization in distributed receding horizon control for consensus-seeking. *IEEE Trans. Cybern.* **46**(9), 2019–2027 (2016)
12. Yang, Q., Jagannathan, S.: Reinforcement learning controller design for affine nonlinear discrete-time systems using online approximators. *IEEE Trans. Syst. Man Cybern. B Cybern.* **42**(2), 377–390 (2012)
13. Du, H., Zhang, N.: Fuzzy control for nonlinear uncertain electrohydraulic active suspensions with input constraint. *IEEE Trans. Fuzzy Syst.* **17**(2), 343–356 (2009)
14. Liu, Y.-J., Gao, Y., Tong, S., Philip Chen, C.L.: A unified approach to adaptive neural control for nonlinear discrete-time systems with nonlinear dead-zone input. *IEEE Trans. Neural Netw. Learn. Syst.* **27**(1), 39–150 (2016)
15. Ma, Y., Borrelli, F., Hencey, B., Coffey, B., Bengea, S., Haves, P.: Model predictive control for the operation of building cooling systems. *IEEE Trans. Control Syst. Technol.* **20**(3), 796–803 (2012)
16. Han, H.G., Wu, X.L., Qiao, J.F.: Real-time model predictive control using a self-organizing neural network. *IEEE Trans. Neural Netw. Learn. Syst.* **24**(9), 1425–1436 (2013)
17. Lu, C.H.: Design and application of stable predictive controller using recurrent wavelet neural networks. *IEEE Trans. Ind. Electron.* **56**(9), 3733–3742 (2009)
18. Peng, H., Nakano, K., Shioya, H.: Nonlinear predictive control using neural nets-based local linearization ARX model-stability and industrial application. *IEEE Trans. Control Syst. Technol.* **15**(1), 130–143 (2017)
19. Li, Z., Xiao, H., Yang, C., Zhao, Y.: Model predictive control of nonholonomic chained systems using general projection neural networks optimization. *IEEE Trans. Syst. Man Cybern. Syst.* **45**(10), 1313–1321 (2015)
20. Lu, P.: Constrained tracking control of nonlinear systems. *Automatica* **27**(5), 305–314 (1996)
21. Li, X., Rakkiyappan, R., Velmurugan, G.: Dissipativity analysis of memristor-based complex-valued neural networks with time-varying delays. *Inform. Sci.* **294**, 645–665 (2015)
22. Zhang, Y., Ge, S.S., Lee, T.H.: A unified quadratic-programming-based dynamical system approach to joint torque optimization of physically constrained redundant manipulators. *IEEE Trans. Syst. Man Cybern. B* **34**(5), 2126–2132 (2004)
23. Li, X., Regan, D.O., Akca, H.: Global exponential stabilization of impulsive neural networks with unbounded continuously distributed delays. *IMA J. Appl. Math.* **80**(1), 85–99 (2015)
24. Jin, L., Zhang, Y., Li, S., Zhang, Y.: Modified ZNN for time-varying quadratic programming with inherent tolerance to noises and its application to kinematic redundancy resolution of robot manipulators. *IEEE Trans. Ind. Electron.* **63**(11), 6978–6988 (2016)
25. Xiao, L.: A nonlinearly-activated neurodynamic model and its finite-time solution to equality-constrained quadratic optimization with nonstationary coefficients. *Appl. Soft Comput.* **40**, 252–259 (2016)
26. Qin, S., Xue, X.: A two-layer recurrent neural network for nonsmooth convex optimization problems. *IEEE Trans. Neural Netw. Learn. Syst.* **26**(6), 1149–1160 (2015)
27. Zhang, Z., Zhang, Y.: Variable joint-velocity limits of redundant robot manipulators handled by quadratic programming. *IEEE/ASME Trans. Mechatronics* **18**(2), 674–686 (2013)
28. Li, X., Song, S.: Impulsive control for existence, uniqueness and global stability of periodic solutions of recurrent neural networks with discrete and continuously distributed delays. *IEEE Trans. Neural Netw.* **24**, 868–877 (2013)
29. Li, S., Chen, S., Liu, B., Li, Y., Liang, Y.: Decentralized kinematic control of a class of collaborative redundant manipulators via recurrent neural networks. *Neurocomputing* **91**, 1–10 (2012)
30. Li, S., Cui, H., Li, Y., Liu, B., Lou, Y.: Decentralized control of collaborative redundant manipulators with partial command coverage via locally connected recurrent neural networks. *Neural Comput. Appl.* **23**(3), 1051–1060 (2013)
31. Li, S., He, J., Li, Y., Rafique, M.U.: Distributed recurrent neural networks for cooperative control of manipulators: A game-theoretic perspective. *IEEE Trans. Neural Netw. Learn. Syst.* **28**(2), 415–426 (2017)

32. Jin, L., Li, S., La, H.M., Luo, X.: Manipulability optimization of redundant manipulators using dynamic neural networks. *IEEE Trans. Ind. Electron.* **64**(6), 4710–4720 (2017)
33. Li, Y., Li, S., Hannaford, B.: A novel recurrent neural network for improving redundant manipulator motion planning completeness. *ICRA*, 2956–2961 (2018)
34. Zhang, Y., Li, S.: A neural controller for image-based visual servoing of manipulators with physical constraints. *IEEE Trans. Neural Netw. Learning Syst.* **29**(11), 5419–5429 (2018)
35. Li, S., Zhou, M., Luo, X.: Modified primal-dual neural networks for motion control of redundant manipulators with dynamic rejection of harmonic noises. *IEEE Trans. Neural Netw. Learning Syst.* **29**(10), 4791–4801 (2018)
36. Li, S., Wang, H., Rafique, M.U.: A novel recurrent neural network for manipulator control with improved noise tolerance. *IEEE Trans. Neural Netw. Learning Syst.* **29**(5), 1908–1918 (2018)
37. Jin, L., Li, S., Luo, X., Li, Y., Qin, B.: Neural dynamics for cooperative control of redundant robot manipulators. *IEEE Trans. Ind. Inform.* **14**(9), 3812–3821 (2018)
38. Li, J., Zhang, Y., Li, S., Mao, M.: New discretization-formula-based zeroing dynamics for real-time tracking control of serial and parallel manipulators. *IEEE Trans. Ind. Inform.* **14**(8), 3416–3425 (2018)
39. Chen, D., Zhang, Y., Li, S.: Tracking control of robot manipulators with unknown models: A jacobian-matrix-adaption method. *IEEE Trans. Ind. Inform.* **14**(7), 3044–3053 (2018)
40. Zhang, Y., Li, S., Gui, J., Luo, X.: Velocity-level control with compliance to acceleration-level constraints: A novel scheme for manipulator redundancy resolution. *IEEE Trans. Ind. Inform.* **14**(3), 921–930 (2018)
41. Xiao, L., Liao, B., Li, S., Zhang, Z., Ding, L., Jin, L.: Design and analysis of FTZNN applied to the real-time solution of a nonstationary Lyapunov equation and tracking control of a wheeled mobile manipulator. *IEEE Trans. Ind. Inform.* **14**(1), 98–105 (2018)
42. Zhang, Y., Chen, S., Li, S., Zhang, Z.: Adaptive projection neural network for kinematic control of redundant manipulators with unknown physical parameters. *IEEE Trans. Ind. Electron.* **65**(6), 4909–4920 (2018)
43. Zhang, Z., Lin, Y., Li, S., Li, Y., Yu, Z., Luo, Y.: Tricriteria optimization-coordination motion of dual-redundant-robot manipulators for complex path planning. *IEEE Trans. Control Syst. Technol.* **26**(4), 1345–1357 (2018)
44. Jin, L., Li, S., Hu, B., Yi, C.: Dynamic neural networks aided distributed cooperative control of manipulators capable of different performance indices. *Neurocomputing* **291**, 50–58 (2018)
45. Jin, L., Li, S., Yu, J., He, J.: Robot manipulator control using neural networks: a survey. *Neurocomputing* **285**, 23–34 (2018)
46. Chen, D., Zhang, Y., Li, S.: Zeroing neural-dynamics approach and its robust and rapid solution for parallel robot manipulators against superposition of multiple disturbances. *Neurocomputing* **275**, 845–858 (2018)
47. Li, S., Shao, Z., Guan, Y.: A dynamic neural network approach for efficient control of manipulators. *IEEE Trans. Syst. Man Cybern. Syst.* **49**(5), 932–941 (2019)
48. Zhang, Y., Li, S., Zhou, X.: Recurrent-neural-network-based velocity-level redundancy resolution for manipulators subject to a joint acceleration limit. *IEEE Trans. Ind. Electron.* **66**(5), 3573–3582 (2019)
49. Zhang, Z., Chen, S., Li, S.: Compatible convex-nonconvex constrained QP-based dual neural networks for motion planning of redundant robot manipulators. *IEEE Trans. Control Syst. Technol.* **27**(3), 1250–1258 (2019)
50. Xu, Z., Li, S., Zhou, X., Yan, W., Cheng, T., Huang, D.: Dynamic neural networks based kinematic control for redundant manipulators with model uncertainties. *Neurocomputing* **329**, 255–266 (2019)
51. Li, S., Zhang, Y., Jin, L.: Kinematic control of redundant manipulators using neural networks. *IEEE Trans. Neural Netw. Learn. Syst.* (In Press)
52. Zhang, Y., Wang, J., Xia, Y.: A dual neural network for redundancy resolution of kinematically redundant manipulators subject to joint limits and joint velocity limits. *IEEE Trans. Neural Netw.* **14**(3), 658–667 (2003)

53. Zhang, Z., Li, Z., Zhang, Y., Luo, Y., Li, Y.: Neural-dynamic-method-based dual-arm CMG scheme with time-varying constraints applied to humanoid robots. *IEEE Trans. Neural Netw. Learn. Syst.* **26**(12), 3251–3262 (2015)
54. Zhang, Y., Li, W., Zhang, Z.: Physical-limits-constrained minimum velocity norm coordinating scheme for wheeled mobile redundant manipulators. *Robotica* **33**(6), 1325–1350 (2015)
55. Mohammed, A.M., Li, S.: Dynamic neural networks for kinematic redundancy resolution of parallel stewart platforms. *IEEE Trans. Cybern.* **46**(7), 1538–1550 (2016)
56. Jin, L., Zhang, Y.: G2-type SRMPC scheme for synchronous manipulation of two redundant robot Arms. *IEEE Trans. Cybern.* **45**(2), 153–164 (2015)
57. Jin, L., Li, S., Xiao, L., Lu, R., Liao, B.: Cooperative motion generation in a distributed network of redundant robot manipulators with noises. *IEEE Trans. Syst. Man Cybern. Syst.* (In Press). <https://doi.org/10.1109/TSMC.2017.2693400>
58. Li, S., Shao, Z., Guan, Y.: A dynamic neural network approach for efficient control of manipulators. *IEEE Trans. Syst. Man Cybern. Syst.* (In Press). <https://doi.org/10.1109/TSMC.2017.2690460>
59. Jin, L., Li, S.: Distributed task allocation of multiple robots: a control perspective. *IEEE Trans. Syst. Man Cybern. Syst.* (In Press). <https://doi.org/10.1109/TSMC.2016.2627579>
60. Chen, C.L.P., Wen, G.-X., Liu, Y.-J., Wang, F.-Y.: Adaptive consensus control for a class of nonlinear multiagent time-delay systems using neural networks. *IEEE Trans. Neural Netw. Learn. Syst.* **25**(6), 1217–1226 (2014)
61. Wang, H.Q., Liu, X.P., Liu, K.F.: Robust adaptive neural tracking control for a class of stochastic nonlinear interconnected systems. *IEEE Trans. Neural Netw. Learn. Syst.* **27**(3), 510–523 (2016)
62. Chen, C.L.P., Liu, Y.-J., Wen, G.-X.: Fuzzy neural network-based adaptive control for a class of uncertain nonlinear stochastic systems. *IEEE Trans. Cybern.* **44**(5), 583–593 (2014)
63. Liu, Y.-J., Tang, L., Tong, S., Chen, C.L.P.: Adaptive NN controller design for a class of nonlinear MIMO discrete-time systems. *IEEE Trans. Neural Netw. Learn. Syst.* **26**(5), 1007–1018 (2015)
64. Liu, Y.-J., Tang, L., Tong, S., Chen, C.L.P., Li, D.-J.: Reinforcement learning design-based adaptive tracking control with less learning parameters for nonlinear discrete-time MIMO systems. *IEEE Trans. Neural Netw. Learn. Syst.* **26**(1), 165–176 (2015)
65. Li, T., Duan, S., Liu, J., Wang, L., Huang, T.: A spintronic memristor-based neural network with radial basis function for robotic manipulator control implementation. *IEEE Trans. Syst. Man Cybern. Syst.* **46**(4), 582–588 (2016)
66. Khalil, H.K.: Nonlinear Systems, 3rd edn. Prentice-Hall, New Jersey (2002)
67. Gao, Y., Wang, H., Liu, Y.-J.: Adaptive fuzzy control with minimal leaning parameters for electric induction motors. *Neurocomputing* **156**, 143–150 (2015)
68. Wang, H., Chen, B., Liu, X., Liu, K., Lin, C.: Robust adaptive fuzzy tracking control for pure-feedback stochastic nonlinear systems with input constraints. *IEEE Trans. Cybern.* **43**(6), 2093–2104 (2013)
69. Isidori, A.: Nonlinear Control Systems: An Introduction, 3rd edn. Springer, New York (1995)
70. Xia, Y.: Further results on global convergence and stability of globally projected dynamical systems. *J. Optim. Theory Appl.* **122**(3), 627–649 (2004)
71. Wang, H., Shi, P., Li, H., Zhou, Q.: Adaptive neural tracking control for a class of nonlinear systems with dynamic uncertainties. *IEEE Trans. Cybern.* (In Press)
72. Chen, C.L.P., Wen, G.-X., Liu, Y.-J., Liu, Z.: Observer-based adaptive backstepping consensus tracking control for high-order nonlinear semi-strict-feedback multiagent systems. *IEEE Trans. Cybern.* **46**(7), 1591–1601 (2016)
73. Liu, Y.-J., Tong, S.: Barrier Lyapunov functions-based adaptive control for a class of nonlinear pure-feedback systems with full state constraints. *Automatica* **64**, 70–75 (2016)
74. He, W., Ge, W., Li, Y., Liu, Y.-J., Yang, C., Sun, C.: Model identification and control design for a humanoid robot. *IEEE Trans. Syst. Man Cybern. Syst.* **47**(1), 45–57 (2017)
75. Wang, H., Liu, X., Liu, K.: Adaptive fuzzy tracking control for a class of pure-feedback stochastic nonlinear systems with non-lower triangular structure. *Fuzzy Set Syst.* **302**, 101–120 (2016)

76. Chien, T., Chen, C., Huang, C.: Feedback linearization control and its application to mimo cancer immunotherapy. *IEEE Trans. Control Syst. Technol.* **18**(4), 953–961 (2010)
77. Zhang, Y., Li, S.: Predictive suboptimal consensus of multiagent systems with nonlinear dynamics. *IEEE Trans. Syst. Man Cybern. Syst.* (In Press). <https://doi.org/10.1109/TSMC.2017.2668440>
78. Allen, L.J.S.: *An Introduction to Mathematical Biology*. Pearson Education, New Jersey (2007)
79. Groves, K., Serrani, A.: Modeling and nonlinear control of a single-link flexible joint manipulator (2004)
80. Jiang, Z.: Global tracking control of underactuated ships by Lyapunov's direct method. *Automatica* **38**(2), 301–309 (2002)

Chapter 3

Adaptive Near-Optimal Control with Full-State Feedback



Abstract In this chapter, a unified online adaptive near-optimal control framework is presented 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. 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 presented adaptive near-optimal control law is asymptotically optimal. The efficacy of the presented framework and the theoretical results are validated by an application to underactuated surface vessels.

3.1 Introduction

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]. 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 control techniques were developed [11–13]. However, traditional adaptive control is generally far from optimal [14]. 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 [14–21]. Modares et al. [14] presented an online policy iteration algorithm for the adaptive optimal control of unknown constrained-input systems. Liu et al. [16] extended the work in [14] 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. [21] 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 [22]. A special case of optimal control could be the redundancy resolution problem of manipulators [23–53]. Control laws of receding horizon optimal control can be updated discretely in time yielding discrete-time receding horizon control [54–56]. Results about adaptive discrete-time receding horizon control have also been reported, e.g., [56, 57]. A major concern of discrete-time receding horizon control is computational burden [56]. It is worth pointing out that Chen et al. [6] presented an explicit control law for the continuous-time receding horizon optimal control of nonlinear systems with known dynamics, which significantly reduces the computational burden. In [58], the method in [6] was extended to the consensus of second-order nonlinear multi-agent systems.

The tracking control of underactuated surface vehicles has attracted significant attention from both academia and industry [59–66]. For example, the global tracking control of underactuated surface vehicles with fully known parameters was addressed in [59] by Lyapunov's direct method. Under perfect knowledge on the system dynamics, in [61], 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 [62–64]. In addition, a neural network-based backstepping approach was proposed in [66] for the tracking control of underactuated surface vehicles with unknown system parameters.

This chapter aims at proposing a unified online adaptive near-optimal control framework for uncertain linear and nonlinear systems. The parameters of the presented adaptive near-optimal control laws are updated in a real-time manner and there is no need to do off-line training for the parameters of the presented adaptive near-optimal control laws. The presented framework deals with the online adaptive

receding horizon optimal control and possesses a simpler structure compared with the existing adaptive optimal control methods mentioned above. Specifically, the framework presented in this chapter mainly consists of constructing auxiliary systems and making a 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 adaptive near-optimal control of linear systems. Then, we extend the results to nonlinear systems with fully unknown parameters. To reduce overshooting that 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 adaptive control loop and grants great enhancements in performance optimality, and tolerance to uncertainty. Theoretical analysis shows that closed-loop systems using the presented control laws are asymptotically stable. It is also proved that the presented near-optimal control laws asymptotically converge to the optimal. An application of the presented near-optimal control scheme to an underactuated surface vessel validates the efficacy of the presented framework and the theoretical results.

The rest of this chapter is organized as follows: In Sect. 3.2, some definitions, theoretical basis and assumptions are provided. In Sect. 3.3, we present online adaptive near-optimal control laws for general linear systems. In Sect. 3.4, we further present online adaptive near-optimal control laws for nonlinear systems. Theoretical results are presented in Sect. 3.5. In Sect. 3.6, the presented online adaptive near-optimal control is applied to an underactuated surface vessel with parameter uncertainty. Then, in Sect. 3.8, we conclude this chapter with final remarks.

3.2 Preliminary

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

Definition 3.1 The nonlinear system considered in this chapter is described as follows:

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

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 $f : \mathbb{R}^n \rightarrow \mathbb{R}^n$, $g : \mathbb{R}^n \rightarrow \mathbb{R}^{n \times m}$ and $h : \mathbb{R}^n \rightarrow \mathbb{R}^m$ are continuously differentiable.

Definition 3.2 ([67]) 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 3.3 ([67]) System (3.1) is said to have a relative degree of ρ in the region of interest \mathbb{U} if the following properties hold true:

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

In this chapter, the following general assumptions are imposed on system (3.1).

- (1) The zero dynamics of system (3.1) are stable [68].
- (2) All state variables of system (3.1) are available [69, 70].
- (3) System (3.1) has a well-defined relative degree ρ [68].
- (4) The output $\mathbf{y}(t)$ of system (3.1) and the desired output $\mathbf{y}_d(t)$ are ρ times continuously differentiable with respect to time t [68].

3.3 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.

3.3.1 Problem Formulation

Consider the following linear system:

$$\begin{cases} \dot{\mathbf{x}}(t) = A\mathbf{x}(t) + B\mathbf{u}(t), \\ \mathbf{y}(t) = C\mathbf{x}(t), \end{cases} \quad (3.2)$$

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 (3.1), one can calculate the relative degree ρ of the linear system by Definition 3.2

and Definition 3.3. In this chapter, we consider the situation that linear system (3.2) is controllable [71]. In addition, throughout this chapter, $\mathbf{y}_d(t)$ is used to denote the desired output.

The finite-horizon optimal control problem about system (3.2) is formulated as

$$\begin{aligned} & \text{minimize} && J_l(t) \\ & \text{subject to} && \dot{\mathbf{x}}(t) = A\mathbf{x}(t) + B\mathbf{u}(t), \\ & && \mathbf{y}(t) = C\mathbf{x}(t), \end{aligned} \quad (3.3)$$

where $J_l(t)$ denotes the performance index. In this chapter, $J_l(t)$ is given as

$$\begin{aligned} J_l(t) = & \int_0^T (\mathbf{y}_d(t + \tau) - \mathbf{y}(t + \tau))^T Q (\mathbf{y}_d(t + \tau) - \mathbf{y}(t + \tau)) d\tau \\ & + \int_0^T \mathbf{u}^T(t + \tau) R \mathbf{u}(t + \tau) d\tau \end{aligned} \quad (3.4)$$

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 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.

3.3.2 Nominal Design

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

$$\begin{cases} \mathbf{y}_d(t + \tau) \approx Y_d(t)\mathbf{w}(\tau), \\ \mathbf{u}(t + \tau) \approx \mathbf{u}(t). \end{cases} \quad (3.5)$$

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

$$\mathbf{y}(t + \tau) \approx \mathbf{y}(t) + \tau \dot{\mathbf{y}}(t) + \dots + \tau^\rho \frac{\mathbf{y}^{[\rho]}(t)}{\rho!}. \quad (3.6)$$

For linear system (3.2) of relative ρ , according to Definitions 3.2 and 3.3, we have $CB = 0$, $CAB = 0, \dots, CA^{\rho-2}B = 0$, $CA^{\rho-1}B \neq 0$. It follows that

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

Let $Y_1(t) = [\mathbf{y}(t), CA\mathbf{x}(t), \dots, CA^\rho\mathbf{x}(t)]$. Equation (3.6) is then written as

$$\mathbf{y}(t + \tau) \approx Y_1(t)\mathbf{w}(\tau) + \frac{\tau^\rho}{\rho!}CA^{\rho-1}B\mathbf{u}(t). \quad (3.7)$$

Substituting Eqs. (3.5) and (3.7) into Eq. (3.4) yields

$$\begin{aligned} J_1(t) &\approx \int_0^T \left(E_1(t)\mathbf{w}(\tau) - \frac{\tau^\rho}{\rho!}CA^{\rho-1}B\mathbf{u}(t) \right)^T Q \left(E_1(t)\mathbf{w}(\tau) - \frac{\tau^\rho}{\rho!}CA^{\rho-1}B\mathbf{u}(t) \right) \\ &\quad + \int_0^T \mathbf{u}^T(t)R\mathbf{u}(t)d\tau \\ &= \int_0^T \mathbf{w}^T(\tau)E_1^T(t)QE_1(t)\mathbf{w}(\tau)d\tau - 2 \int_0^T \frac{\tau^\rho}{\rho!}\mathbf{w}^T(\tau)d\tau E_1^T(t)CA^{\rho-1}B\mathbf{u}(t) \\ &\quad + \int_0^T \frac{\tau^{2\rho}}{(\rho!)^2}d\tau \mathbf{u}^T(t)(CA^{\rho-1}B)^T Q CA^{\rho-1}B\mathbf{u}(t) + T\mathbf{u}^T(t)R\mathbf{u}(t), \\ &= \hat{J}_1(t), \end{aligned}$$

where $E_1(t) = Y_d(t) - Y_1(t)$, and $\hat{J}(t)$ denotes the approximation of $J(t)$. Let

$$\begin{aligned} \mathbf{v} &= \int_0^T \frac{\tau^\rho}{\rho!}\mathbf{w}^T(\tau)d\tau \\ &= \left[\frac{T^{\rho+1}}{(\rho+1)\rho!}, \frac{T^{\rho+2}}{(\rho+2)\rho!1!}, \dots, \frac{T^{2\rho+1}}{(2\rho+1)(\rho!)^2} \right], \end{aligned}$$

and

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

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

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

where $\Theta = TR + \kappa(CA^{\rho-1}B)^T Q CA^{\rho-1}B$ and $\mathbf{p} = (-2\mathbf{v}E_1^T Q CA^{\rho-1}B)^T$. Given that Θ is positive-definite, $J_e(t)$ is convex and $\mathbf{u}(t)$ is thus obtained by solving $\partial J_e(t)/\partial \mathbf{u} = 0$, which gives

$$\mathbf{u}(t) = (TR + \kappa(CA^{\rho-1}B)^T Q CA^{\rho-1}B)^{-1} (QCA^{\rho-1}B)^T (Y_d(t) - Y_1(t)) \mathbf{v}^T. \quad (3.8)$$

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 (3.3) is presented as follows [72]:

$$\mathbf{u}(t) = -R^{-1}(t)B^T(P(t)\mathbf{x}(t) + \mathbf{s}(t)), \quad (3.9)$$

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

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

with $P(t+T) = 0$ and $Q' = C^T QC$. Besides, $\mathbf{s}(t)$ is determined by the following differential equation [72]:

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

with $C\mathbf{x}_d(t) = \mathbf{y}_d(t)$ and $\mathbf{s}(t+T) = 0$. Evidently, compared with traditional solution (3.9), near-optimal control law (3.8) does not need to solve any differential equation.

3.3.3 Adaptive Design

In this subsection, we design an online adaptive near-optimal controller 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:

$$\dot{\hat{\mathbf{x}}}(t) = \hat{A}\mathbf{x}(t) + \hat{B}\mathbf{u}(t) - K_x(\hat{\mathbf{x}}(t) - \mathbf{x}(t)), \quad (3.10)$$

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 $K_x = \text{diag}([k_{x_1}, k_{x_2}, \dots, k_{x_n}]) \in \mathbb{R}^{n \times n}$ is a positive-definite gain matrix, and the others are defined as the aforementioned. It is desired that linear auxiliary system (3.10) possesses the same dynamic behavior as linear system (3.2). To achieve such an objective, we define the following evolutions for matrices \hat{A} and \hat{B} :

$$\begin{cases} \dot{\hat{A}} = -K_A(\hat{\mathbf{x}}(t) - \mathbf{x}(t))\mathbf{x}^T(t), \\ \dot{\hat{B}} = -K_B(\hat{\mathbf{x}}(t) - \mathbf{x}(t))\mathbf{u}^T(t), \end{cases} \quad (3.11)$$

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

$$\begin{cases} \dot{\hat{\mathbf{x}}}(t) = \hat{A}\hat{\mathbf{x}}(t) + \hat{B}\mathbf{u}(t) - K_x(\hat{\mathbf{x}}(t) - \mathbf{x}(t)), \\ \dot{\hat{A}} = -K_A(\hat{\mathbf{x}}(t) - \mathbf{x}(t))\mathbf{x}^T(t), \\ \dot{\hat{B}} = -K_B(\hat{\mathbf{x}}(t) - \mathbf{x}(t))\mathbf{u}^T(t). \end{cases} \quad (3.12)$$

Gain matrices in linear auxiliary system (3.12) are chosen in the following manner. The values of the diagonal elements in K_x are much smaller than those in K_A and K_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_x at least 10 times smaller than those in K_A and K_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 (3.2), linear auxiliary system (3.10) 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 adaptive near-optimal control law for linear system (3.2) is thus obtained as follows:

$$\mathbf{u}_a(t) = (TR + \kappa(C\hat{A}^{\rho-1}\hat{B})^TQC\hat{A}^{\rho-1}\hat{B})^{-1}(QC\hat{A}^{\rho-1}\hat{B})^T(Y_d(t) - Y_l(t))\mathbf{v}^T. \quad (3.13)$$

The block diagram of linear system (3.2) synthesized by online adaptive near-optimal control law (3.13) and linear auxiliary system (3.12) is shown in Fig. 3.1. As seen from this figure, by utilizing state information and input information of linear system (3.2), auxiliary system (3.12) estimates the parameter matrices of linear system (3.2). The parameter matrices generated by auxiliary system (3.12) are passed to control law (3.13), which yields adaptive near-optimal control action to linear system (3.2).

3.4 Extension to Nonlinear Systems

Based on the previous result on linear systems, in this section, we present the result about online adaptive near-optimal control of nonlinear systems.

3.4.1 Problem Formulation

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

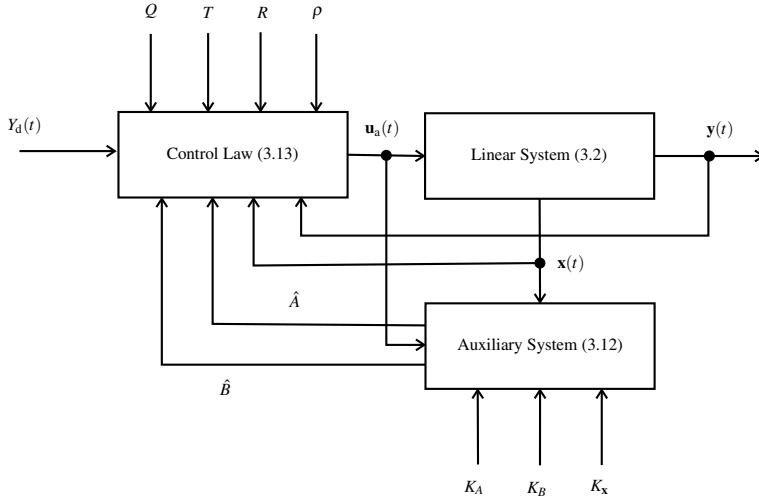


Fig. 3.1 Block diagram of linear system (3.2) with parameter uncertainty synthesized by online adaptive near-optimal control law (3.13) and linear auxiliary system (3.12)

$$\begin{aligned} & \text{minimize} && J_n(t) \\ & \text{subject to} && \dot{\mathbf{x}} = f(\mathbf{x}) + g(\mathbf{x})\mathbf{u}(t), \\ & && \mathbf{y}(t) = h(\mathbf{x}), \end{aligned} \quad (3.14)$$

where $J_n(t)$ denotes the performance index and is given as

$$\begin{aligned} J_n(t) = & \int_0^T (\mathbf{y}_d(t + \tau) - \mathbf{y}(t + \tau))^T Q (\mathbf{y}_d(t + \tau) - \mathbf{y}(t + \tau)) d\tau \\ & + \int_0^T \mathbf{u}^T(t + \tau) R \mathbf{u}(t + \tau) d\tau. \end{aligned} \quad (3.15)$$

Compared with linear system (3.2), the finite-horizon optimal control problem of the nonlinear system (3.1) 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 the nonlinear system (3.1).

Remark 3.1 Directly solving finite-horizon optimal control problem (3.14) 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.

3.4.2 Nominal Design

In this subsection, the design process of a near-optimal nominal control law for nonlinear system (3.1) is presented under the assumption that all the parameters of the nonlinear system (3.1) are known.

Similar to the case of linear system (3.2), one has

$$\begin{cases} \mathbf{y}_d(t + \tau) \approx Y_d(t)\mathbf{w}(\tau), \\ \mathbf{u}(t + \tau) \approx \mathbf{u}(t), \\ \mathbf{y}(t + \tau) \approx \mathbf{y}(t) + \tau \dot{\mathbf{y}}(t) + \dots + \tau^\rho \frac{\mathbf{y}^{[\rho]}(t)}{\rho!}. \end{cases}$$

For nonlinear system (3.1) with relative degree ρ , based on Definitions 3.2 and 3.3, one has

$$\begin{cases} \dot{\mathbf{y}}(t) = \frac{\partial h}{\partial \mathbf{x}} \dot{\mathbf{x}} = \frac{\partial h}{\partial \mathbf{x}} f(\mathbf{x}) = L_f h(\mathbf{x}), \\ \vdots \\ \mathbf{y}^{[\rho-1]}(t) = L_f^{\rho-1} h(\mathbf{x}), \\ \mathbf{y}^{[\rho]}(t) = L_f^\rho h(\mathbf{x}) + \frac{\partial L_f^{\rho-1} h(\mathbf{x})}{\partial \mathbf{x}} g(\mathbf{x}) \mathbf{u}(t) \\ = L_f^\rho h(\mathbf{x}) + L_g L_f^{\rho-1} h(\mathbf{x}) \mathbf{u}(t). \end{cases} \quad (3.16)$$

Let $Y_n(t) = [\mathbf{y}(t), L_f h(\mathbf{x}), \dots, L_f^{\rho-1} h(\mathbf{x}), \dots, L_f^\rho h(\mathbf{x})]$. Then, $\mathbf{y}(t + \tau)$ of nonlinear system (3.1) is approximated as

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

Performance index $J_n(t)$ shown in Eq. (3.15) is thus approximated as

$$\begin{aligned} J_n(t) &\approx \hat{J}_n(t) \\ &= \int_0^T \left(E_n(t)\mathbf{w}(\tau) - \frac{\tau^\rho}{\rho!} L_g L_f^{\rho-1} h(\mathbf{x}) \mathbf{u}(t) \right)^T Q(E_n(t)\mathbf{w}(\tau) \\ &\quad - \frac{\tau^\rho}{\rho!} L_g L_f^{\rho-1} h(\mathbf{x}) \mathbf{u}(t)) d\tau + T \mathbf{u}^T(t) R \mathbf{u}(t) \\ &= \int_0^T \mathbf{w}^T(\tau) E_n^T(t) Q E_n(t) \mathbf{w}(\tau) d\tau - 2 \int_0^T \frac{\tau^\rho}{\rho!} \mathbf{w}^T(\tau) d\tau E_n^T(t) Q L_g L_f^{\rho-1} h(\mathbf{x}) \mathbf{u}(t) \\ &\quad + \int_0^T \frac{\tau^{2\rho}}{(\rho!)^2} d\tau \mathbf{u}^T(t) (L_g L_f^{\rho-1} h(\mathbf{x}))^T Q L_g L_f^{\rho-1} h(\mathbf{x}) \mathbf{u}(t) + T \mathbf{u}^T(t) R \mathbf{u}(t), \end{aligned} \quad (3.17)$$

where $E_n = Y_d(t) - Y_n(t)$. Similar to the control law design for linear system (3.2), since the decision variable is $\mathbf{u}(t)$, minimizing performance index $\hat{J}_n(t)$ is equivalent to minimizing the following quadratic performance index:

$$\Psi_n(t) = \mathbf{u}^T(t)\Theta_n\mathbf{u}(t) + \mathbf{p}_n^T\mathbf{u}(t), \quad (3.18)$$

where

$$\Theta_n = TR + \kappa(L_g L_f^{\rho-1} h(\mathbf{x}))^T Q L_g L_f^{\rho-1} h(\mathbf{x})$$

and

$$\mathbf{p}_n = -2(L_g L_f^{\rho-1} h(\mathbf{x}))^T Q^T E_n \mathbf{v}^T.$$

Given that Θ_n is positive-definite, performance index Ψ_n shown in (3.18) is convex and the optimal solution is thus obtained by solving $\partial\Psi_n(t)/\partial\mathbf{u} = 0$, which gives the following nominal near-optimal control law for nonlinear system (3.1):

$$\mathbf{u}(t) = (TR + \kappa(L_g L_f^{\rho-1} h(\mathbf{x}))^T Q L_g L_f^{\rho-1} h(\mathbf{x}))^{-1} (Q L_g L_f^{\rho-1} h(\mathbf{x}))^T (Y_d(t) - Y_n(t)) \mathbf{v}^T, \quad (3.19)$$

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

3.4.3 Adaptive Design

In this subsection, we design online adaptive near-optimal control laws for nonlinear system (3.1) 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; $\phi_f(\cdot) : \mathbb{R}^n \rightarrow \mathbb{R}^{N_f}$ and $\phi_g(\cdot) : \mathbb{R}^n \rightarrow \mathbb{R}^{N_g \times m}$ denote suitable basis functions; N_f and N_g denote the numbers of neurons. Therefore, nonlinear system (3.1) can be rewritten as the following parameterized system:

$$\begin{cases} \dot{\mathbf{x}}(t) = W_f \phi_f(\mathbf{x}) + W_g \phi_g(\mathbf{x}) \mathbf{u}(t), \\ \mathbf{y}(t) = h(\mathbf{x}). \end{cases} \quad (3.20)$$

In this sense, the parameter uncertainty of nonlinear system (3.1) corresponds to the uncertainty of weight matrices W_f and W_g . Based on the parameterized system (3.20), we design an adaptive near-optimal control law for nonlinear system (3.1) 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 (3.20):

$$\dot{\hat{\mathbf{x}}}(t) = \hat{W}_f \phi_f(\mathbf{x}) + \hat{W}_g \phi_g(\mathbf{x}) \mathbf{u}(t) - K_x (\hat{\mathbf{x}}(t) - \mathbf{x}(t)), \quad (3.21)$$

where parameter matrices \hat{W}_f and \hat{W}_g need to be obtained with K_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:

$$\begin{cases} \dot{\hat{W}}_f = -K_f(\hat{\mathbf{x}}(t) - \mathbf{x}(t))\phi_f^T(\mathbf{x}), \\ \dot{\hat{W}}_g = -K_g(\hat{\mathbf{x}}(t) - \mathbf{x}(t))\mathbf{u}^T(t)\phi_g^T(\mathbf{x}), \end{cases} \quad (3.22)$$

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

$$\begin{cases} \dot{\hat{\mathbf{x}}}(t) = \hat{W}_f\phi_f(\mathbf{x}) + \hat{W}_g\phi_g(\mathbf{x})\mathbf{u}(t) - K_x(\hat{\mathbf{x}}(t) - \mathbf{x}(t)), \\ \dot{\hat{W}}_f = -K_f(\hat{\mathbf{x}}(t) - \mathbf{x}(t))\phi_f^T(\mathbf{x}), \\ \dot{\hat{W}}_g = -K_g(\hat{\mathbf{x}}(t) - \mathbf{x}(t))\mathbf{u}^T(t)\phi_g^T(\mathbf{x}). \end{cases} \quad (3.23)$$

Note that the values of gain matrices K_x , K_f , and K_g in nonlinear auxiliary system (3.23) are similarly chosen as those in linear auxiliary system (3.12).

For special case when $\hat{\mathbf{x}}(t) = \mathbf{x}(t)$: When nonlinear auxiliary system (3.21) has reconstructed the states of nonlinear system (3.20), i.e., $\hat{\mathbf{x}}(t) = \mathbf{x}(t)$, nonlinear system (3.1) with parameter uncertainty becomes

$$\begin{cases} \dot{\mathbf{x}}(t) = \hat{W}_f\phi_f(\mathbf{x}) + \hat{W}_g\phi_g(\mathbf{x})\mathbf{u}(t), \\ \mathbf{y}(t) = h(\mathbf{x}). \end{cases} \quad (3.24)$$

We perform the controller design based on (3.24). 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 ρ of nonlinear system (3.1) with parameter uncertainty is a prior knowledge, we thus have

$$\begin{cases} \dot{\mathbf{y}}(t) = \frac{\partial h}{\partial \mathbf{x}}\hat{f}(\mathbf{x}) = L_{\hat{f}}h(\mathbf{x}), \\ \vdots \\ \mathbf{y}^{[\rho-1]}(t) = L_{\hat{f}}^{\rho-1}h(\mathbf{x}), \\ \mathbf{y}^{[\rho]}(t) = L_{\hat{f}}^\rho h(\mathbf{x}) + \frac{\partial L_{\hat{f}}^{\rho-1}h(\mathbf{x})}{\partial \mathbf{x}}\hat{g}(\mathbf{x})\mathbf{u}(t) \\ = L_{\hat{f}}^\rho h(\mathbf{x}) + L_{\hat{g}}L_{\hat{f}}^{\rho-1}h(\mathbf{x})\mathbf{u}(t). \end{cases} \quad (3.25)$$

Let $\hat{Y}_n(t) = [\mathbf{y}(t), L_{\hat{f}}h(\mathbf{x}), \dots, L_{\hat{f}}^\rho h(\mathbf{x})]$. Then, following similar steps in Sect. 3.4.2, based on system (3.24), an online adaptive near-optimal control law is obtained as follows for nonlinear system (3.1) with parameter uncertainty:

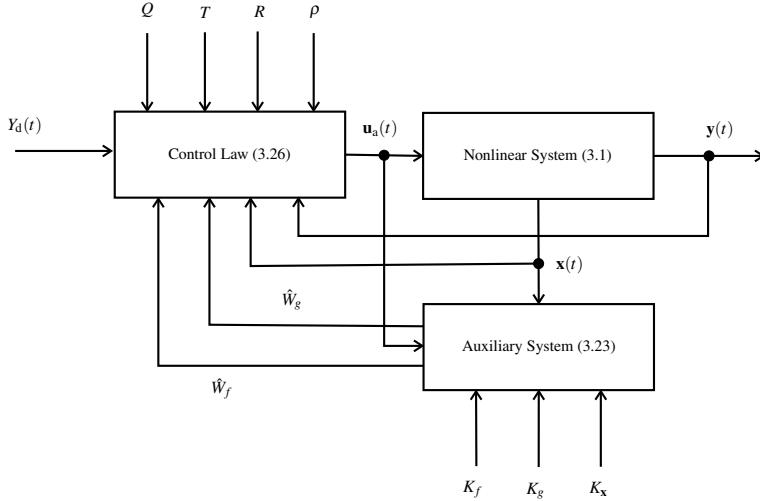


Fig. 3.2 Block diagram of nonlinear system (3.1) with parameter uncertainty synthesized by online adaptive near-optimal control law (3.26) and nonlinear auxiliary system (3.23)

$$\mathbf{u}_a(t) = (TR + \kappa(L_{\hat{g}}L_{\hat{f}}^{\rho-1}h(\mathbf{x}))^\top Q L_{\hat{g}}L_{\hat{f}}^{\rho-1}h(\mathbf{x}))^{-1}(QL_{\hat{g}}L_{\hat{f}}^{\rho-1}h(\mathbf{x}))^\top(Y_d(t) - \hat{Y}_n(t))\mathbf{v}^\top. \quad (3.26)$$

The block diagram of nonlinear system (3.1) with parameter uncertainty synthesized by adaptive near-optimal control law (3.26) and nonlinear auxiliary system (3.23) is shown in Fig. 3.2.

For general situation without $\hat{\mathbf{x}}(t) = \mathbf{x}(t)$: For the general situation, nonlinear system (3.1) with parameter uncertainty is represented as

$$\begin{cases} \dot{\mathbf{x}}(t) = \hat{W}_f \phi_f(\mathbf{x}) + \hat{W}_g \phi_g(\mathbf{x}) \mathbf{u}(t) - (\tilde{W}_f \phi_f(\mathbf{x}) + \tilde{W}_g \phi_g(\mathbf{x}) \mathbf{u}(t)), \\ \mathbf{y}(t) = h(\mathbf{x}), \end{cases} \quad (3.27)$$

where $\tilde{W}_f = \hat{W}_f - W_f$ and $\tilde{W}_g = \hat{W}_g - W_g$. Given that the relative degree ρ of nonlinear system (3.1) with parameter uncertainty is a prior knowledge, we thus have

$$\begin{cases} \dot{\mathbf{y}}(t) = L_{\hat{f}}h(\mathbf{x}) - L_{\hat{f}}h(\mathbf{x}), \\ \vdots \\ \mathbf{y}^{[\rho-1]}(t) = L_{\hat{f}}^{\rho-1}h(\mathbf{x}) - L_{\hat{f}}^{\rho-1}h(\mathbf{x}), \\ \mathbf{y}^{[\rho]}(t) = L_{\hat{f}}^\rho h(\mathbf{x}) + L_{\hat{g}}L_{\hat{f}}^{\rho-1}h(\mathbf{x})\mathbf{u}(t) - (L_{\hat{f}}^\rho h(\mathbf{x}) + L_{\hat{g}}L_{\hat{f}}^{\rho-1}h(\mathbf{x})\mathbf{u}(t)), \end{cases} \quad (3.28)$$

where $L_{\hat{f}}^i h(\mathbf{x}) = L_{\hat{f}}^i h(\mathbf{x}) - L_f^i h(\mathbf{x})$. Since W_g and W_f are unknown, control design cannot be done based on (3.28).

Remark 3.2 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 (3.26) is a good choice since the difference between controller (3.26) and nominal controller (3.19) is small. Without a good prior knowledge, the difference may be extremely large before auxiliary system (3.23) fully reconstructs the states and dynamics of nonlinear system (3.1), which may lead to extremely large magnitude of adaptive control input $\mathbf{u}_a(t)$. 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 $\mathbf{u}_a(t)$ generated by adaptive near-optimal control law (3.26), the following modified online adaptive near-optimal controller can be adopted:

$$\mathbf{u}_s(t) = \text{sat}_\beta(\mathbf{u}_a(t)), \quad (3.29)$$

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

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

This approach can also be adopted to deal with the adaptive near-optimal control of linear system (3.2) to avoid very large magnitude of control inputs.

Remark 3.3 In terms of control structure complexity and computational burden, the presented adaptive near-optimal control is compared with ADP [14–21] as follows:

- (1) For the adaptive near-optimal control, ADP generally needs actor and critic networks to guarantee near-optimality and identifier networks to handle system uncertainty. In the presented approach, there is only one network associated with the auxiliary system to handle system uncertainty. In this sense, compared with ADP, the presented adaptive near-optimal control 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 presented adaptive near-optimal control does not need the training of the weights of the networks. Besides, the presented control law is presented in an analytical form, of which the parameters are directly obtained from the auxiliary system.

3.4.4 Computational Complexity Analysis

Since nonlinear systems are more complicated than linear systems, in this subsection, we analyze the computational complexity of the presented adaptive near-optimal control 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 [73].

- (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 (3.23) is conducted by firstly discretizing it. By using the Euler difference formula [74], the discrete-time nonlinear auxiliary system can be obtained as follows:

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

where $\tau > 0 \in \mathbb{R}$ is the sampling period and $k = 1, 2, \dots$ is the updating index. To utilize the presented adaptive near-optimal control 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 presented adaptive near-optimal control law (3.26), for which $T\mathbf{R} \in \mathbb{R}^{m \times m}$, $\mathbf{v}^T \in \mathbb{R}^{\rho+1}$, $Q \in \mathbb{R}^{m \times m}$, $\kappa \in \mathbb{R}$ are constant and do not need to be

calculated at each sampling time instant; $Y_d(t) \in \mathbb{R}^{m \times (\rho+1)}$ is calculated beforehand; $L_{\hat{g}} L_{\hat{f}}^{\rho-1} h(\mathbf{x}) \in \mathbb{R}^{m \times m}$ and $\hat{Y}_n(t) \in \mathbb{R}^{m \times (\rho+1)}$ 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 $\hat{Y}_n(t)$. Therefore, based on (3.26), computing $\mathbf{u}_a(t)$ 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 presented adaptive near-optimal control method consists of two parts: updating the auxiliary system and computing the adaptive near-optimal control law based on the parameter matrices generated by the auxiliary system. Therefore, the total computational complexity of the presented adaptive near-optimal control method for nonlinear system (3.1) is the sum of the two parts. It follows that the presented adaptive near-optimal control method for nonlinear system (3.1) 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 presented 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.

3.5 Theoretical Results

In this section, considering that linear system (3.2) can be viewed as a special case of nonlinear system (3.1), we mainly present theoretical results about the performance of auxiliary system (3.23) and nonlinear system (3.1) with parameter uncertainty synthesized by online adaptive near-optimal control law (3.26).

3.5.1 Convergence of Auxiliary Systems

In this subsection, we present theoretical results about the convergence of auxiliary systems (3.23) and (3.12).

Theorem 3.1 *The states and dynamics of nonlinear auxiliary system (3.23) converge to those of nonlinear system (3.1) with time.*

Proof Recall that nonlinear system (3.1) can be described by parameterized system (3.20):

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

and nonlinear auxiliary system (3.23):

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

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

$$\begin{cases} \dot{\tilde{\mathbf{x}}}(t) = \tilde{W}_f \phi_f(\mathbf{x}) + \tilde{W}_g \phi_g(\mathbf{x}) \mathbf{u}(t) - K_{\mathbf{x}} \tilde{\mathbf{x}}(t), \\ \dot{\tilde{W}}_f = -K_f \tilde{\mathbf{x}}(t) \phi_f^T(\mathbf{x}), \\ \dot{\tilde{W}}_g = -K_g \tilde{\mathbf{x}}(t) \mathbf{u}^T(t) \phi_g^T(\mathbf{x}). \end{cases} \quad (3.30)$$

Define the following candidate Lyapunov function:

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

where $\text{tr}(\cdot)$ 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 $V_1(t) \geq 0$ always holds. Calculating the time derivative of $V_1(t)$ along the system dynamics yields

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

Substituting Eq. (3.30) into $\dot{V}_1(t)$, one further obtains

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

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 [75], $\tilde{\mathbf{x}}(t) = 0$ is asymptotically stable. Therefore, when $t \rightarrow \infty$, one has $\hat{\mathbf{x}} = \mathbf{x}$ and auxiliary system (3.23) 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

(3.23) is the same as nonlinear system (3.1). It follows that $\lim_{t \rightarrow +\infty} (\tilde{W}_f \phi_f(\mathbf{x}) + \tilde{W}_g \phi_g(\mathbf{x}) \mathbf{u}) = 0$. In other words, the states and dynamics of nonlinear auxiliary system (3.23) converge to those of nonlinear system (3.1) with time. The proof is complete. \square

In Theorem 3.1, it is proved that a special case is achieved in an asymptotic manner, i.e., $\lim_{t \rightarrow +\infty} \hat{\mathbf{x}}(t) = \mathbf{x}(t)$. The underlying intuition is that, under the conditions specified in Theorem 3.1, 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 presented control law valid for general cases. Based on Theorem 3.1, we offer the following corollary about auxiliary system (3.12).

Corollary 3.1 *The states and dynamics of linear auxiliary system (3.12) converge to those of linear system (3.2) with time.*

Proof It can be generalized from Theorem 3.1 and is thus omitted. \square

3.5.2 Stability of Closed-Loop Systems

In terms of the stability of the closed-loop systems consisting of the controlled systems and the presented online adaptive near-optimal control laws, we offer the following theoretical results.

Lemma 3.1 *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. \square

Theorem 3.2 *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 (3.1) with parameter uncertainty and online adaptive near-optimal control law (3.26) is asymptotically stable in the sense of attraction and Lyapunov stability.*

Proof Substituting $\mathbf{u}(t) = \mathbf{u}_a(t)$ into the last equation of (3.16) with $\mathbf{u}_a(t)$ defined in Eq. (3.26) yields

$$\mathbf{y}^{[\rho]}(t) = L_f^\rho h(\mathbf{x}) + L_g L_f^{\rho-1} h(\mathbf{x}) \mathbf{u}_a(t). \quad (3.31)$$

Let

$$\begin{cases} L_{\hat{f}}^\rho h(\mathbf{x}) = L_{\hat{f}}^\rho h(\mathbf{x}) - L_f^\rho h(\mathbf{x}), \\ L_{\hat{g}} L_{\hat{f}}^{\rho-1} h(\mathbf{x}) = L_{\hat{g}} L_{\hat{f}}^{\rho-1} h(\mathbf{x}) - L_g L_f^{\rho-1} h(\mathbf{x}), \\ \eta(t) = L_{\hat{f}}^\rho h(\mathbf{x}) + L_{\hat{g}} L_{\hat{f}}^{\rho-1} h(\mathbf{x}) \mathbf{u}_a(t). \end{cases} \quad (3.32)$$

Equation (3.31) is thus written as

$$\mathbf{y}^{[\rho]}(t) = L_f^\rho h(\mathbf{x}) + L_{\hat{f}} L_{\hat{f}}^{\rho-1} h(\mathbf{x}) \mathbf{u}_a(t) - \eta(t). \quad (3.33)$$

Given that $R = 0$, adaptive near-optimal control law (3.26) becomes

$$\mathbf{u}(t) = (\kappa(L_{\hat{g}} L_{\hat{f}}^{\rho-1} h(\mathbf{x}))^\top Q L_{\hat{g}} L_{\hat{f}}^{\rho-1} h(\mathbf{x}))^{-1} (Q L_{\hat{g}} L_{\hat{f}}^{\rho-1} h(\mathbf{x}))^\top (Y_d(t) - \hat{Y}_n(t)) \mathbf{v}^\top. \quad (3.34)$$

Substituting Eq. (3.34) into Eq. (3.33) yields

$$\begin{aligned} \mathbf{y}^{[\rho]}(t) &= L_f^\rho h(\mathbf{x}) + L_{\hat{g}} L_{\hat{f}}^{\rho-1} h(\mathbf{x}) (\kappa(L_{\hat{g}} L_{\hat{f}}^{\rho-1} h(\mathbf{x}))^\top Q L_{\hat{g}} L_{\hat{f}}^{\rho-1} h(\mathbf{x}))^{-1} (Q L_{\hat{g}} L_{\hat{f}}^{\rho-1} h(\mathbf{x}))^\top \\ &\quad \times (Y_d(t) - \hat{Y}_n(t)) \mathbf{v}^\top - \eta(t). \end{aligned}$$

Given that $L_{\hat{g}} L_{\hat{f}}^{\rho-1} h(\mathbf{x})$ is invertible and based on Lemma 3.1, one further has

$$\begin{aligned} \mathbf{y}^{[\rho]}(t) &= L_f^\rho h(\mathbf{x}) + \frac{1}{\kappa} ((L_{\hat{g}} L_{\hat{f}}^{\rho-1} h(\mathbf{x}))^\top Q)^{-1} (L_{\hat{g}} L_{\hat{f}}^{\rho-1} h(\mathbf{x}))^\top Q^\top (Y_d(t) - \hat{Y}_n(t)) \mathbf{v}^\top \\ &\quad - \eta(t). \end{aligned}$$

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

$$\mathbf{y}^{[\rho]}(t) = L_f^\rho h(\mathbf{x}) + \frac{1}{\kappa} (Y_d(t) - \hat{Y}_n(t)) \mathbf{v}^\top - \eta(t). \quad (3.35)$$

Let $\tilde{Y}_n(t) = \hat{Y}_n(t) - Y_n(t)$. Equation (3.35) is then rewritten as

$$\mathbf{y}^{[\rho]}(t) = L_f^\rho h(\mathbf{x}) + \frac{1}{\kappa} (Y_d(t) - Y_n(t)) \mathbf{v}^\top + L_{\hat{f}}^\rho h(\mathbf{x}) - \frac{1}{\kappa} \tilde{Y}_n(t) \mathbf{v}^\top - \eta(t). \quad (3.36)$$

Let $\delta(t) = -L_{\hat{f}}^\rho h(\mathbf{x}) + \tilde{Y}_n(t) \mathbf{v}^\top / \kappa + \eta(t)$. Recalling the definition of $\eta(t)$ in (3.32), we further have $\delta(t) = L_{\hat{g}} L_{\hat{f}}^{\rho-1} h(\mathbf{x}) \mathbf{u}_a(t) + \tilde{Y}_n(t) \mathbf{v}^\top / \kappa$. Then, Eq. (3.36) can be written as

$$\mathbf{y}^{[\rho]}(t) = L_f^\rho h(\mathbf{x}) + \frac{1}{\kappa} (Y_d(t) - Y_n(t)) \mathbf{v}^\top - \delta(t). \quad (3.37)$$

Recall that $\hat{Y}_n(t) = [\mathbf{y}(t), L_{\hat{f}} h(\mathbf{x}), \dots, L_{\hat{f}}^{\rho-1} h(\mathbf{x})]$, $Y_d(t) = [\mathbf{y}_d(t), \dots, \mathbf{y}_d^{[\rho-1]}, \mathbf{y}_d^{[\rho]}(t)]$,

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

and

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

As a result, from Eq. (3.37), one has

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

Let $\mathbf{e}(t) = \mathbf{y}_d(t) - \mathbf{y}(t)$. It follows that the closed-loop system consisting of nonlinear system (3.1) and adaptive near-optimal control law (3.26) is

$$\mathbf{e}^{[\rho]}(t) = -\frac{1}{\kappa} \sum_{j=0}^{\rho-1} \frac{T^{\rho+1+j}}{(\rho+1+j)\rho!j!} \mathbf{e}^{[j]}(t) + \delta(t). \quad (3.38)$$

Note that $\delta(t)$ can be viewed as a perturbation or an input. When $\delta(t) = 0$, from Eq. (3.38), the unforced system is

$$\frac{1}{\kappa} \sum_{i=0}^{\rho} \frac{T^{\rho+1+i}}{(\rho+1+i)\rho!i!} \mathbf{e}^{[i]}(t) = 0. \quad (3.39)$$

When $\rho \in \{1, 2, 3, 4\}$, according to the Routh–Hurwitz criterion [71], it can be readily proved that system (3.39) is exponentially asymptotically stable. Therefore, linear system (3.38) satisfies the bounded-input bounded-output property [71].

From the proof of Theorem 3.1, with $\mathbf{u}(t) = \mathbf{u}_a(t)$, we have

$$\lim_{t \rightarrow +\infty} (\tilde{W}_f \phi_f(\mathbf{x}) + \tilde{W}_g \phi_g(\mathbf{x}) \mathbf{u}_a(t)) = 0.$$

Given that $\phi_f(\mathbf{x})$ and $\phi_g(\mathbf{x}) \mathbf{u}_a(t)$ satisfy the persistent excitation condition [21, 76, 77], one further has $\lim_{t \rightarrow +\infty} \tilde{W}_f(t) = 0$ and $\lim_{t \rightarrow +\infty} \tilde{W}_g(t) = 0$. In light of the definitions of $L_f^i h(\mathbf{x})$ and $L_{\tilde{f}} L_{\tilde{f}}^{\rho-1} h(\mathbf{x})$ in Eq. (3.25), one further has $\lim_{t \rightarrow +\infty} L_f^i h(\mathbf{x}) = 0$, $\lim_{t \rightarrow +\infty} \tilde{Y}_n(t) \lim_{t \rightarrow +\infty} L_{\tilde{g}} L_{\tilde{f}}^{\rho-1} h(\mathbf{x}) \mathbf{u}_a(t) = 0$. Therefore,

$$\lim_{t \rightarrow +\infty} \delta(t) = \lim_{t \rightarrow +\infty} (L_f^\rho h(\mathbf{x}) + L_{\tilde{g}} L_{\tilde{f}}^{\rho-1} h(\mathbf{x}) \mathbf{u}_a(t) + L_{\tilde{f}}^\rho h(\mathbf{x}) - \tilde{Y}_n(t) \mathbf{v}^T / \kappa) = 0.$$

According to the bounded-input bounded-output property [71], $\lim_{t \rightarrow +\infty} \|\mathbf{e}(t)\|_2 = \lim_{t \rightarrow +\infty} \|\delta(t)\|_2 = 0$, i.e., the equilibrium $\mathbf{e}(t) = 0$ is asymptotically stable. In other words, closed-looped system (3.38) consisting of nonlinear system (3.1) and online adaptive near-optimal control law (3.26) is asymptotically stable. The proof is thus complete. \square

Now we have the following remarks about the online adaptive near-optimal control of nonlinear system (3.1) with parameter uncertainty.

Remark 3.4 Intuitively, the philosophy of the presented online adaptive control scheme for nonlinear system (3.1) 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 (3.23), of which the input-to-state and input-to-output properties converge to the same as nonlinear system (3.1) 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, 76, 77].

Remark 3.5 Based on Theorems 3.1 and 3.2, it can be proved that the closed-looped system consisting of nonlinear system (3.1) and saturated online adaptive near-optimal control law (3.29) is also asymptotically stable. Based on Theorem 3.1, \hat{W}_f and \hat{W}_g are always bounded. Together with Eq.(3.26) and Theorem 3.2, $\mathbf{u}_a(t)$ is bounded and continuous at each time instant t , when weight matrix $R = 0$, relative degree $\rho \in \{1, 2, 3, 4\}$, and matrix $L_g L_f^{\rho-1} h(\mathbf{x})$ is invertible. Since $\mathbf{u}_a(t)$ is bounded and continuous at each time instant t , there always exist a time instant t_i and a constant saturation vector β such that, for any $t > t_i$, $-\beta_j \leq u_{aj}(t) \leq \beta_j$ with $j = 1, 2, \dots, m$. It follows that $\text{sat}_\beta(\mathbf{u}_a(t)) = \mathbf{u}_a(t)$ for all $t > t_i$. Let $\mathbf{u}_e(t) = \text{sat}_\beta(\mathbf{u}_a(t)) - \mathbf{u}_a(t)$ and $\eta_1(t) = \eta(t) + L_g L_f^{\rho-1} h(\mathbf{x})\mathbf{u}_e(t)$. Evidently, $\lim_{t \rightarrow +\infty} \eta_1(t) = \lim_{t \rightarrow +\infty} \eta(t)$, since $\mathbf{u}_e(t) = 0$ for $t > t_i$. Following the steps in the proof of Theorem 3.2, one can thus readily prove that the closed-looped system consisting of nonlinear system (3.1) and saturated online adaptive near-optimal control law (3.29) is asymptotically stable.

Remark 3.6 Theorem 3.2 is derived under the condition that $R = 0$. In view of performance index (3.15), matrices Q and R are used to scale the trade-off between tracking accuracy described by $\mathbf{y}_d(t) - \mathbf{y}(t)$ 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 [71], via the presented method, the tracking error $\mathbf{y}_d(t) - \mathbf{y}(t)$ can be guaranteed to be bounded.

In light of the fact that the online adaptive near-optimal control of linear system (3.2) is a special case of that of nonlinear system (3.1), we have the following corollary.

Corollary 3.2 *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 (3.2) with parameter uncertainty and online adaptive near-optimal control law (3.13) is asymptotically stable.*

Proof It can be generalized from the proof of Theorem 3.2 and is thus omitted. \square

3.5.3 Asymptotic Optimality

In this chapter, a unified approximation of the performance indices is utilized for the optimal control of both linear system (3.2) and nonlinear system (3.1) to avoid

solving Hamilton equations and to obtain analytical control laws. In this subsection, theoretical results about the optimality of the presented control laws are presented.

Theorem 3.3 *If matrix Q is symmetric and positive definite, matrix $R = 0$, matrix $L_g L_f^{\rho-1} h(\mathbf{x})$ is invertible, and relative degree $\rho \in \{1, 2, 3, 4\}$, then performance index $J_n(t)$ of nonlinear system (3.1) with parameter uncertainty is bounded and asymptotically converges to the optimal, i.e., online adaptive near-optimal control law (3.26) asymptotically converges to the optimal.*

Proof Based on Taylor expansion and in view of Eqs.(3.15) and (3.17) about $J_n(t)$ and $\hat{J}_n(t)$, with $R = 0$, one has

$$\begin{aligned} J_n(t) &= \int_0^T \left(E_n(t) \mathbf{w}(\tau) - \frac{\tau^\rho}{\rho!} L_g L_f^{\rho-1} h(\mathbf{x}) + \frac{\tau^\rho}{\rho!} \Delta(t) \right)^T Q \\ &\quad \times \left(E_n(t) \mathbf{w}(\tau) - \frac{\tau^\rho}{\rho!} L_g L_f^{\rho-1} h(\mathbf{x}) + \frac{\tau^\rho}{\rho!} \Delta(t) \right) d\tau, \end{aligned}$$

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

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

where $\|\cdot\|_2$ defines the two-norm of a vector or a matrix. According to Theorem 3.2, given that matrix Q is symmetric and positive definite, matrix $R = 0$, and matrix $L_g L_f^{\rho-1} h(\mathbf{x})$ is invertible, the closed-loop system consisting of nonlinear system (3.1) and adaptive near-optimal control law (3.26) 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

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

where $\mathbf{s}_i(t) = [e_i(t), \dot{e}_i(t), \dots, e_i^{[\rho-1]}(t)]^T \in \mathbb{R}^\rho$, $v_i(t) = [0, \dots, 0, \delta_i(t)]^T \in \mathbb{R}^\rho$, 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 $\lim_{t \rightarrow +\infty} \mathbf{s}_i(t) = 0$ for all $i = 1, 2, \dots, m$. From Theorem 3.2, $\lim_{t \rightarrow +\infty} \delta(t) = 0$, i.e., $\lim_{t \rightarrow +\infty} v_i(t) = 0$ for all $i = 1, 2, \dots, m$. By the bounded-input bounded-output property of linear systems [71], the equilibrium $\mathbf{s}_i(t) = 0$ of linear system $\dot{\mathbf{s}}_i(t) = W\mathbf{s}_i(t) + v_i$ is thus asymptotically stable, i.e., $\lim_{t \rightarrow +\infty} \mathbf{s}_i(t) = 0$. Together with $\dot{\mathbf{s}}_i(t) = W\mathbf{s}_i(t) + v_i(t)$, it follows that $\lim_{t \rightarrow +\infty} \dot{\mathbf{s}}_i(t) = 0$. Recalling that $\mathbf{s}_i(t) = [e_i(t), \dot{e}_i(t), \dots, e_i^{[\rho-1]}(t)]^T$, one further has $\lim_{t \rightarrow +\infty} e_i^{[\rho]}(t) = 0$ for all $i = 1, 2, \dots, m$. It follows that $\lim_{t \rightarrow +\infty} \mathbf{e}^{[\rho]}(t) = 0$, i.e., $(\mathbf{y}^{[\rho]}(t) - \mathbf{y}_d^{[\rho]}(t)) = 0$. It follows that $\lim_{t \rightarrow +\infty} \sup_{0 < \psi < 1} \|\Delta(t)\|_2^2 \|Q\|_2 = 0$. It is also obtained in the proof of Theorem 3.2 that $\lim_{t \rightarrow +\infty} \tilde{W}_f(t) = 0$ and $\lim_{t \rightarrow +\infty} \tilde{W}_g(t) = 0$, given that $\phi_f(\mathbf{x})$ and $\phi_g(\mathbf{x})\mathbf{u}_a(t)$ satisfy the persistent excitation condition [21, 76, 77]. It follows that $\lim_{t \rightarrow +\infty} (\mathbf{u}(t) - \mathbf{u}_a(t)) = 0$, where $\mathbf{u}(t)$ is defined in Eq. (3.19) and minimizes the convex quadratic performance index $\hat{J}_n(t)$ shown in Eq. (3.17), for which, in the adaptive case, $\mathbf{u}(t)$ is replaced by $\mathbf{u}_a(t)$. It follows that $\lim_{t \rightarrow +\infty} \hat{J}_n(t) = 0$ when adaptive near-optimal control law (3.26) is employed. Note that $J_n(t) \geq 0$. Then, according to the pinching theorem [78], $\lim_{t \rightarrow +\infty} J_n(t) = 0$. The proof is complete. \square

Similarly, we have the following corollary about the online adaptive near-optimal control of linear system (3.2).

Corollary 3.3 *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 $J_l(t)$ of linear system (3.1) with parameter uncertainty is bounded and asymptotically converges to the optimal, i.e., online adaptive near-optimal control law (3.13) asymptotically converges to the optimal.*

Proof It can be generalized from the proof of Theorem 3.3 and is thus omitted. \square

Before ending this section, 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 3.7 In online adaptive near-optimal control law (3.26), when the term $\kappa(L_{\hat{g}}L_{\hat{f}}^{\rho-1}h(\mathbf{x}))^T Q L_{\hat{g}}L_{\hat{f}}^{\rho-1}h(\mathbf{x})$ is singular and $R = 0$, a regulation term can be used. Specifically, one may use $vI_l + \kappa(L_{\hat{g}}L_{\hat{f}}^{\rho-1}h(\mathbf{x}))^T Q L_{\hat{g}}L_{\hat{f}}^{\rho-1}h(\mathbf{x})$ to replace $\kappa(L_{\hat{g}}L_{\hat{f}}^{\rho-1}h(\mathbf{x}))^T Q L_{\hat{g}}L_{\hat{f}}^{\rho-1}h(\mathbf{x})$, where $v > 0 \in \mathbb{R}$ is a small positive parameter, e.g., 10^{-6} , with I_l being an identity matrix of suitable dimension. Given that v is small enough, the control performance is still satisfactory due to the bounded-input

bounded-output property [71]. This approach also works for online adaptive near-optimal control law (3.13).

3.6 Application to Uncertain Underactuated Surface Vessel

The control of surface vessels has long been regarded as a challenging problem due to the inherent underactuation of the system. To test the efficacy and generality of the presented control scheme, in this section, the presented online adaptive near-optimal control 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 presented control scheme for the underactuated surface vessel.

3.6.1 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 equations [59]:

$$\begin{cases} \dot{x}_1 = x_4 \cos x_3 - x_5 \sin x_3, \\ \dot{x}_2 = x_4 \sin x_3 + x_5 \cos x_3, \\ \dot{x}_3 = x_6, \\ \dot{x}_4 = \frac{m_{22}}{m_{11}} x_5 x_6 - \frac{d_{11}}{m_{11}} x_4 + \frac{1}{m_{11}} u_1, \\ \dot{x}_5 = -\frac{m_{11}}{m_{22}} x_4 x_6 - \frac{d_{22}}{m_{22}} x_5, \\ \dot{x}_6 = \frac{m_{11} - m_{22}}{m_{33}} x_4 x_5 - \frac{d_{33}}{m_{33}} x_6 + \frac{1}{m_{33}} u_2, \end{cases} \quad (3.40)$$

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 [59]. Evidently, the state vector is $\mathbf{x} = [x_1, x_2, x_3, x_4, x_5, x_6]^T$; the input vector is $\mathbf{u} = [u_1, u_2]^T$. In this application, the actual inertia and damping parameters of surface vessel system (3.40) is $m_{11} = m_{22} = m_{33} = 0.1$, $d_{11} = d_{33} = 0$ and $d_{22} = 0.2$ [59].

Surface vessel system (3.40) can be rewritten in the form of parameterized system (3.20), where $\phi_f(\mathbf{x}) \in \mathbb{R}^8$ with $\phi_{f1}(\mathbf{x}) = x_4 \cos x_3$, $\phi_{f2}(\mathbf{x}) = x_5 \sin x_3$, $\phi_{f3}(\mathbf{x}) = x_4 \sin x_3$, $\phi_{f4}(\mathbf{x}) = x_5 \cos x_3$, $\phi_{f5}(\mathbf{x}) = x_6$, $\phi_{f6}(\mathbf{x}) = x_5 x_6$, $\phi_{f7}(\mathbf{x}) = x_4 x_6$, and $\phi_{f8}(\mathbf{x}) = x_5$; the non-null elements of matrix $W_f \in \mathbb{R}^{6 \times 8}$ are $w_{f11} = 1$, $w_{f12} = -1$, $w_{f23} = 1$, $w_{f24} = 1$, $w_{f35} = 1$, $w_{f46} = m_{22}/m_{11} = 1$, $w_{f57} = -m_{11}/m_{22} = -1$, $w_{f58} = -d_{22}/m_{22} = -2$; $\phi_g(\mathbf{x}) = [1, 0; 0, 1]$; and the non-null elements of matrix $W_g \in \mathbb{R}^{6 \times 2}$ are $w_{g41} = 1/m_{11} = 10$ and $w_{g62} = 1/m_{33} = 10$. 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 presented control scheme does not need any prior knowledge on the values of these system parameters.

If the considered output is chosen as $\mathbf{y} = [x_1, x_2]^T$, 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 presented control unfeasible. To avoid the singularity problem, the considered output of the surface vessel is defined as follows:

$$\mathbf{y} = \begin{bmatrix} y_1 \\ y_2 \end{bmatrix} = \begin{bmatrix} x_1 + l \cos x_3 \\ x_2 + l \sin x_3 \end{bmatrix}, \quad (3.41)$$

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 Definitions 3.2 and 3.3, the relative degree ρ of the surface vessel system is 2 and $L_f L_h(\mathbf{x}) = [\cos x_3/m_{11}, -l \sin x_3/m_{33}; \sin x_3/m_{11}, l \cos x_3/m_{33}]$. 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 (3.40) is expected to track a desired circular path defined by $\mathbf{y}_d = [10 \cos(0.1t), 10 \sin(0.1t) + 1]^T$ 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 adaptive near-optimal control law (3.29) is set to $T = 0.2$ s and each element of β is set to 5. To avoid singularity, the regulation term as shown in Remark 3.7 is added with $v = 10^{-6}$. The parameters of the corresponding auxiliary system (3.23) are set to $K_x = 20 \times \text{diag}([1, 1, 1, 1, 1, 1])$, $K_f = 5000 \times \text{diag}([1, 1, 1, 1, 1, 1, 1, 1])$, and $K_g = 5000 \times \text{diag}([1, 1])$. Initial state $\hat{\mathbf{x}}(0)$ of auxiliary system (3.23) is set to be the same as initial state $\mathbf{x}(0)$ of surface vessel system (3.40) with $\mathbf{x}(0) = [7, 0, 0, 0, 0, 0]^T$. 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 this situation is very difficult due to the lack of a good prior knowledge of the system parameters.

With the above setups, Figs. 3.3 and 3.4 show the time history of values of the non-null elements of \hat{W}_g and \hat{W}_f generated by nonlinear auxiliary system (3.23) 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 (3.40)

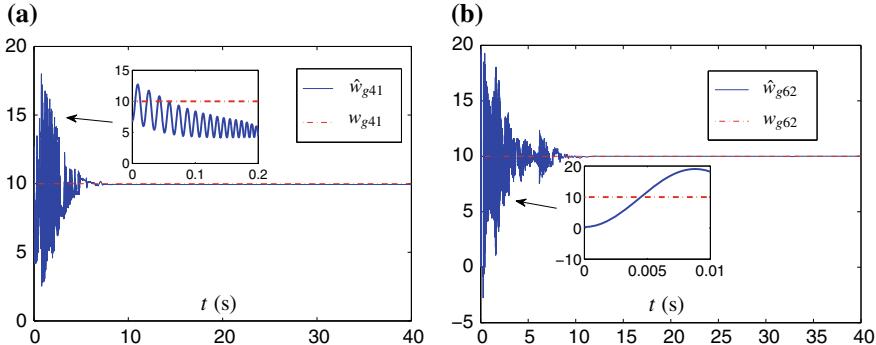


Fig. 3.3 Time history of the values of non-null elements of \hat{W}_g generated by auxiliary system (3.23) synthesized by saturated online adaptive near-optimal control law (3.29) in comparison with the corresponding elements of W_g of surface vessel system (3.40)

synthesized by saturated online adaptive near-optimal control law (3.29) is shown in Fig. 3.5. As seen from Fig. 3.5a, 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. Figure 3.5b, c show that the surface vessel successfully tracks the reference path and position errors $e_{y_1} = y_{d1} - y_1$ and $e_{y_2} = y_{d2} - y_2$ converge to zero with time. Figure 3.6 shows the time history of control input $\mathbf{u}_s(t)$ generated by saturated adaptive near-optimal control law (3.29). The above results substantiate the efficacy of the presented adaptive near-optimal control method under no measurement noises and verify the theoretical results.

3.6.2 With Measurement Noises

In this subsection, simulation results are presented to show the efficacy of the presented adaptive near-optimal control method under measurement noises.

Consider underactuated surface vessel (3.40) again but with measurement noises taken into account, i.e., in adaptive near-optimal control law (3.29), the state values and output values of underactuated surface vessel (3.40) 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 (3.41). The parameters of the corresponding auxiliary system (3.23) are set as $K_x = 5 \times \text{diag}([1, 1, 1, 1, 1, 1])$, $K_f = 50 \times \text{diag}([1, 1, 1, 1, 1, 1, 1])$, 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 Figs. 3.7 and 3.8. As seen from Fig. 3.7, under the measurement

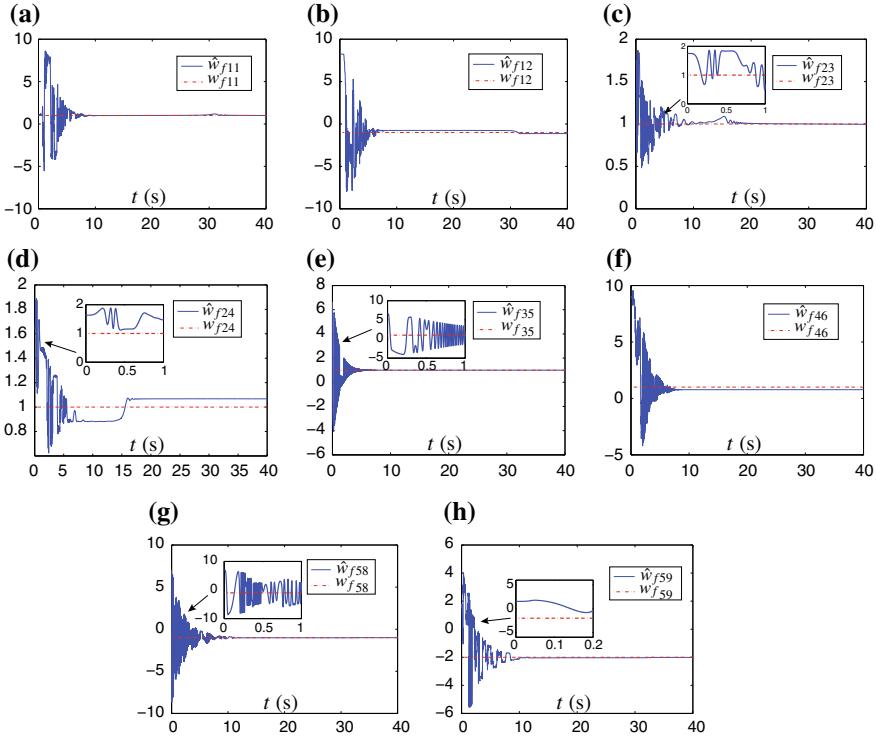


Fig. 3.4 Time history of values of the non-null elements of \hat{W}_f generated by auxiliary system (3.23) synthesized by saturated online adaptive near-optimal control law (3.29) in comparison with the corresponding elements of W_f of surface vessel system (3.40)

noises, the tracking performance of the surface vessel system synthesized by adaptive near-optimal control law (3.29) 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. 3.8. These results substantiate the efficacy of the presented adaptive near-optimal control method under measurement noises.

3.6.3 Capability for Real-Time Control

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

The results shown in Sect. 3.4.4 are employed to calculate the number of floating-point operations per sampling time instant for the presented 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 $\sin(x) \approx x - x^3/6 + x^5/120 - x^7/5040 +$

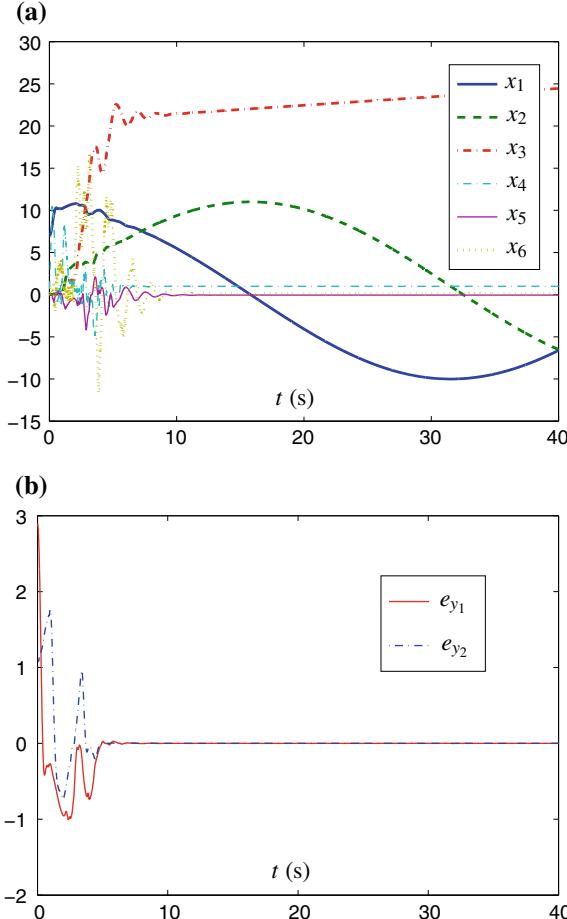


Fig. 3.5 Circular path tracking performance of surface vessel system (3.40) synthesized by saturated online adaptive near-optimal control law (3.29). **a** Time history of state variables. **b** Time history of position errors defined as $e_{y_1} = y_{d1} - y_1$ and $e_{y_2} = y_{d2} - y_2$

$x^9/362880 - x^{11}/39916800$, 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 Sect. 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 $\hat{Y}_n(t)$ is 64. Therefore, in this application, at each sampling time instant t , computing

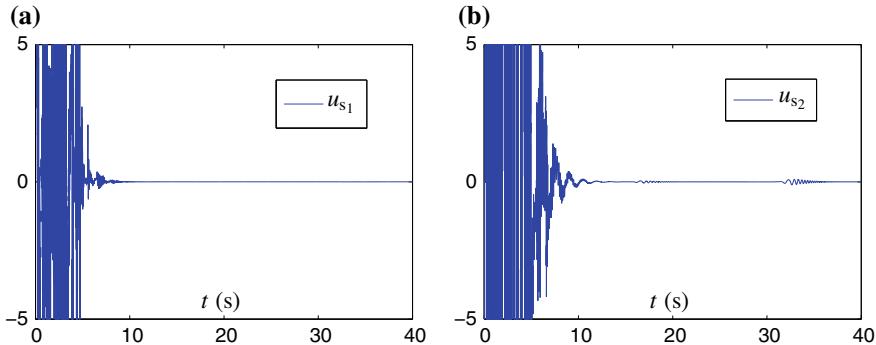


Fig. 3.6 Time history of input $u_s(t)$ generated by saturated adaptive near-optimal control law (3.29). **a** Time history of $u_{s1}(t)$. **b** Time history of $u_{s2}(t)$

Fig. 3.7 Circular path tracking performance of surface vessel system (3.40) synthesized by saturated online adaptive near-optimal control law (3.29) with independent zero-mean Gaussian measurement noises with the standard deviation being 0.01. **a** Time history of state variables. **b** Time history of position errors defined as
 $e_{y_1} = y_{d1} - y_1$ and
 $e_{y_2} = y_{d2} - y_2$

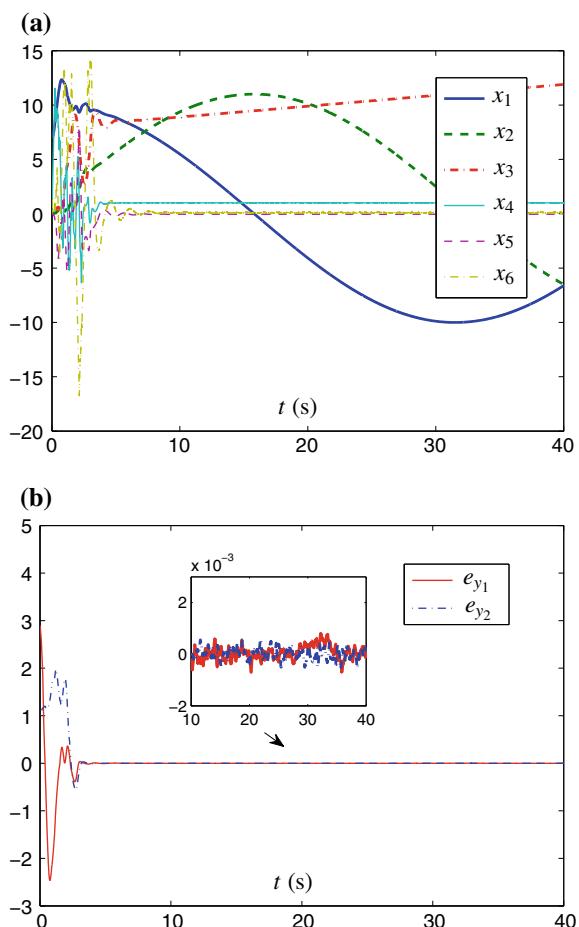
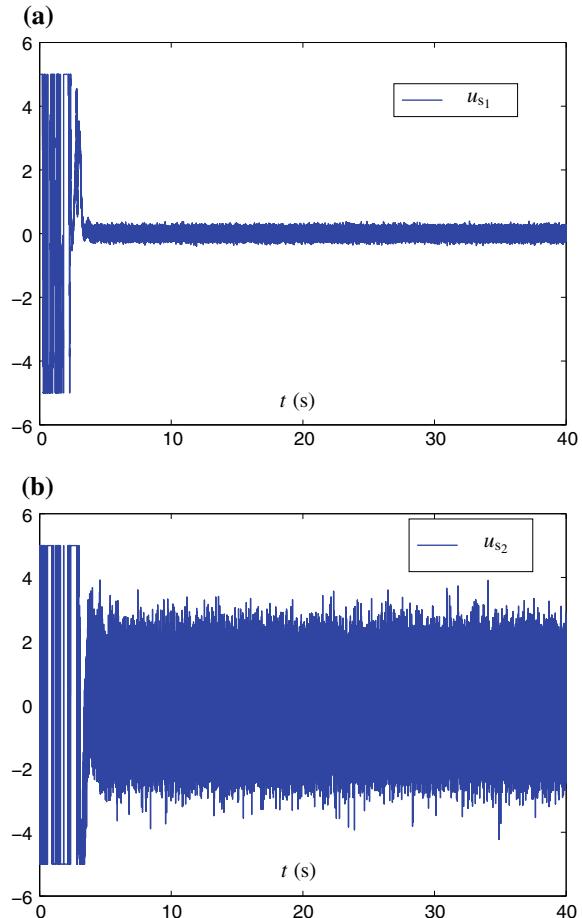


Fig. 3.8 Time history of input $\mathbf{u}_s(t)$ generated by saturated adaptive near-optimal control law (3.29) with independent zero-mean Gaussian measurement noises with the standard deviation being 0.01. **a** Time history of $u_{s1}(t)$. **b** Time history of $u_{s2}(t)$



$\mathbf{u}_a(t)$ 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 presented adaptive near-optimal control 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 presented adaptive near-optimal control method for the underactuated surface vessel is less than 2.194×10^5 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 3.75×10^8 floating-point operations per second [79]. Therefore, the presented adaptive near-optimal control method is feasible for the real-time control of underactuated surface vessels.

3.7 Questions and Answers

In these sections, some discussions about the content in this chapter are presented as follows in the form of questions and answers.

Question 3.1 “In the equation of $\dot{V}_1(t)$, the derivation of last two lines is not clear. Giving

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

how is the item $\text{tr}(\tilde{W}_g^T \tilde{\mathbf{x}}(t) \phi_g(\mathbf{x}) \mathbf{u}(t))$ cancelled?”

Answer: Recall

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

and nonlinear auxiliary system:

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

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

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

Define the following candidate Lyapunov function:

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

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

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

Then, one further obtains

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

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, $\tilde{\mathbf{x}}(t) = 0$ is asymptotically stable. Therefore, when $t \rightarrow \infty$, one has $\hat{\mathbf{x}} = \mathbf{x}$ and the auxiliary system becomes

$$\dot{\hat{\mathbf{x}}}(t) = \hat{W}_f\phi_f(\mathbf{x}) + \hat{W}_g\phi_g(\mathbf{x})\mathbf{u}(t), \quad (3.42)$$

which means that given the same input $\mathbf{u}(t)$, the state response of the auxiliary system is the same as the nonlinear system. In other words, the states and dynamics of nonlinear auxiliary system converges to those of the controlled nonlinear system with time.

Question 3.2 “In theorem 2, it simply mentioned that the linear system satisfies the bounded-input bounded-output property. However, $\delta(t)$ has control input $\mathbf{u}_a(t)$ and Lie derivative terms, which may not be bounded.”

Answer: As proved in Theorem 1, $\lim_{t \rightarrow +\infty} \delta_i(t) = 0$ for all $i = 1, 2, \dots, m$. By the definition of limitation, there always exist a $\varepsilon > 0$ and a $t' > 0$ such that for any $t > t'$, $|\delta_i(t)| < \varepsilon$, $\forall i = 1, 2, \dots, m$. Note that $\delta(0) = L_{\tilde{g}}L_{\tilde{f}}^{\rho-1}h(\mathbf{x}(0))\mathbf{u}_a(0) + \tilde{Y}_n(0)\mathbf{v}^T/\kappa$ is bounded, since $\mathbf{x}(0)$, $\mathbf{u}_a(0)$, and $\tilde{Y}_n(0)$ are bounded. In addition, $\delta(t)$ is continuous since all the related variables are continuous. It follows that $\delta(t)$ is bounded all the time.

Question 3.3 “Theorem 2 assumes weight matrix $R = 0$. What are the practical meanings to make this assumption? What about the non-zero weight matrix case?”

Answer: In performance index $J_n(t)$:

$$\begin{aligned} J_n(t) = & \int_0^T (\mathbf{y}_d(t + \tau) - \mathbf{y}(t + \tau))^T Q (\mathbf{y}_d(t + \tau) - \mathbf{y}(t + \tau)) d\tau \\ & + \int_0^T \mathbf{u}^T(t + \tau) R \mathbf{u}(t + \tau) d\tau, \end{aligned}$$

matrices Q and R are used to scale the trade-off between tracking accuracy described by $\mathbf{y}_d(t) - \mathbf{y}(t)$ and energy consumption described by the magnitude of control effort $\mathbf{u}(t)$. Practically, $R = 0$ means that tracking accuracy is far more important than energy consumption. When $R \neq 0$, it means that energy consumption is taken into account. In this situation, by the bounded-input bounded-output property, via our method, the tracking error $\mathbf{y}_d(t) - \mathbf{y}(t)$ can only be guaranteed bounded.

Question 3.4 “In Remark 4, it stated that ‘there always existed a time instant t_t ’ while Theorem 1 only proves asymptotical convergence. Please explain more about the existence of time instant t_t .”

Answer: Based on Theorem 1, \hat{W}_f and \hat{W}_g are always bounded. Together with Theorem 2, $\mathbf{u}_a(t)$ is bounded and continuous at each time instant t , when weight matrix $R = 0$, relative degree $\rho \in \{1, 2, 3, 4\}$, and matrix $L_g L_f^{\rho-1} h(\mathbf{x})$ is invertible. Since $\mathbf{u}_a(t)$ is bounded and continuous at each time instant t , there always exist a time instant t_t and a constant saturation vector β such that, for any $t > t_t$, $-\beta_j \leq u_{aj}(t) \leq \beta_j$ for all $j = 1, 2, \dots, m$. It follows that $\text{sat}_\beta(\mathbf{u}_a(t)) = \mathbf{u}_a(t)$ for all $t > t_t$. It is worth pointing out that the existence of t_t discussed in this remark depends on the boundedness of $\mathbf{u}_a(t)$.

Question 3.5 “Please give more explanations on contribution (2). What are the existing results compared with? Why is the presented control framework simpler and has a lower computational burden?”

Answer: The proposed adaptive near-optimal control is compared with adaptive dynamic programming (ADP) as follows:

- (1) For the adaptive near-optimal control, ADP generally needs actor and critic networks to guarantee near-optimality and identifier networks to handle system uncertainty. In our approach, there is only one network associated with the auxiliary system to handle system uncertainty. In this sense, compared with ADP, the proposed adaptive near-optimal control has a simpler structure.
- (2) Corresponding to the control structure, ADP needs time (and sufficient data) 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 adaptive near-optimal control does not need the training of the weights of actor and critic networks. Besides, the proposed control law is presented in an analytical form, of which the parameters are directly obtained from the auxiliary system. In this sense, the proposed adaptive near-optimal control has a lower computational burden.

Question 3.6 “Essentially, the nonlinear system that can be reconstructed by neural networks is considered. The question comes to how the presented control framework differs from the existing neural network-based near-optimal control.”

Answer: The authors sincerely thank the reviewer for pointing out the comment. Most of the existing neural network based near-optimal control methods are variants of adaptive dynamic programming (ADP). A general comparison between our approach and ADP is described as follows:

- (1) In ADP, near-optimal control law $\mathbf{u}(t)$ 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 our approach, we relax the performance index to a quadratic performance index, by which we obtain an analytical near-optimal control law.
- (2) ADP needs actor and critic networks to guarantee near-optimality and identifier networks to handle system uncertainty. In our approach, neural networks are only used to handle the uncertainty of the controlled system and thus less neurons are used in our approach.
- (3) ADP generally requires a stable initial policy to guarantee system stability [R2], which is not required in our method.

Question 3.7 “Why assume $\mathbf{u}(t + \tau) \approx \mathbf{u}(t)$ within a predictive period? Typically, for finite-horizon optimal control, it assumes a trajectory of $\mathbf{u}(t)$ instead of a constant input.”

Answer: To approximate performance indices, Taylor expansion is adopted for $\mathbf{u}(t + \tau)$, $\mathbf{y}_d(t + \tau)$, and $\mathbf{y}(t + \tau)$. Given that $\tau > 0$ is small, by Taylor expansion, one has

$$\mathbf{u}(t + \tau) = \mathbf{u}(t) + \tau \dot{\mathbf{u}}(t) + \frac{\tau^2}{2} \ddot{\mathbf{u}}(t) + \dots$$

In other words, $\mathbf{u}(t + \tau) \approx \mathbf{u}(t)$ is obtained by 0-order Taylor expansion.

Question 3.8 “The statement ‘If there exist a time instant $t_s > 0$ such that, for all $t > t_s$, $\hat{\mathbf{x}}(t) = \mathbf{x}(t)$ contradicts the asymptotic convergence in Theorem 1, since Theorem 1 doesn’t prove finite-time convergence.’”

Answer: We agree with the comment that the statement is not true, and have revised the corresponding paragraph. The revision is repeated as follows:

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

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

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 relative degree ρ of the controlled nonlinear system with parameter uncertainty is a prior knowledge, we thus have

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

Let $\hat{Y}_{\text{n}}(t) = [\mathbf{y}(t), L_{\hat{f}} h(\mathbf{x}), \dots, L_{\hat{f}}^\rho h(\mathbf{x})]$. Then, following similar steps in the linear system case, an online adaptive near-optimal control law is obtained as follows for the controlled nonlinear system with parameter uncertainty:

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

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

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

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

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

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 (Eq.x).

Question 3.9 “The derivation of the linear, nominal design case in Sect. III. A and B seem quite standard. I would like the authors to clarify if these are original results and how they differ from existing results.”

Answer: It is worth pointing out that our motivation to successively consider linear systems and nonlinear systems for both nominal and adaptive cases is to enhance the readability. We hope that such a manner can make the content more beneficial to readers.

In Sect. III-A, we present the problem formulation of the finite-horizon optimal control of linear systems. As we stated in Sect. III-A, such a formulation is generally adopted. We agree with the reviewer that the problem formulation for finite-horizon optimal control for linear system is standard. The problem can also be formulated as a linear quadratic regulation tracking:

$$\begin{aligned} & \text{minimize } J_l = \int_t^{t+T} ((\mathbf{y}_d(\tau) - \mathbf{y}(\tau))^T Q (\mathbf{y}_d(\tau) - \mathbf{y}(\tau)) + \mathbf{u}^T(\tau) R \mathbf{u}(\tau) d) \tau, \\ & \text{subject to } \dot{\mathbf{x}}(t) = A \mathbf{x}(t) + B \mathbf{u}(t), \\ & \mathbf{y}(t) = C \mathbf{x}(t), \end{aligned} \quad (\text{Eq.3.1})$$

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.

Standard Solution: The standard optimal control law for problem (Eq.3.1) is given by

$$\mathbf{u}(t) = -R^{-1}(t)B^T(P(t)\mathbf{x}(t) + \mathbf{s}(t)), \quad (\text{Eq.3.2})$$

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

$$-\dot{P} = P(t)A + A^T P(t) - P(t)BR^{-1}B^T P(t) + Q', \quad (\text{Eq.3.3})$$

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

$$-\dot{\mathbf{s}} = (A - BR^{-1}B^T P)^T \mathbf{s}(t) + Q' C^T (C C^T) \mathbf{x}_d(t) \quad (\text{Eq.3.4})$$

with $C \mathbf{x}_d(t) = \mathbf{y}_d(t)$ and $\mathbf{s}(t+T) = 0$.

Our Solution: In Sect. III-B, we present the nominal near-optimal design to solve the finite-horizon optimal control problem of linear systems formulated in Sect. III-A, which can also be formulated as (Eq.3.1). By utilizing Taylor expansion, in the manuscript, the problem is relaxed to a quadratic program, from which the following near-optimal control law is derived:

$$\mathbf{u}(t) = (TR + \kappa(CA^{\rho-1}B)^T Q CA^{\rho-1}B)^{-1} (QCA^{\rho-1}B)^T (Y_d(t) - Y_l(t)) \mathbf{v}^T, \quad (\text{Eq.3.5})$$

where $Y_d(t) = [\mathbf{y}_d(t), \dots, \mathbf{y}_d^{[\rho-1]}, \mathbf{y}_d^{[\rho]}(t)]$, $Y_l(t) = [\mathbf{y}(t), CA\mathbf{x}(t), \dots, CA^\rho\mathbf{x}(t)]$,

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

and

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

Evidently, near-optimal control law (Eq.3.5) is different from the standard solution (Eq.3.2). In addition, while the standard solution requires solving two differential equations, i.e., (Eq.3.2) and (Eq.3.3), near-optimal control law (Eq.3.5) does not need to solve any differential equation.

Question 3.10 “In both Sects. III and IV, it is assumed that the Θ and Θ_n matrices are positive definite. Later in Sect. V, R is assumed to be 0. A natural question is why we need to have a non-zero R and what if R is non-zero?”

Answer: The authors sincerely thank the reviewer for pointing out the comment. In performance index $J_n(t)$:

$$\begin{aligned} J_n(t) &= \int_0^T (\mathbf{y}_d(t+\tau) - \mathbf{y}(t+\tau))^T Q (\mathbf{y}_d(t+\tau) - \mathbf{y}(t+\tau)) d\tau \\ &\quad + \int_0^T \mathbf{u}^T(t+\tau) R \mathbf{u}(t+\tau) d\tau, \end{aligned}$$

matrices Q and R are used to scale the trade-off between tracking accuracy described by $\mathbf{y}_d(t) - \mathbf{y}(t)$ and energy consumption described by the magnitude of control effort $\mathbf{u}(t)$. Practically, $R = 0$ means that tracking accuracy is far more important than energy consumption. When $R \neq 0$, it means that energy consumption is taken into account. In this situation, by the bounded-input bounded-output property, via our method, the tracking error $\mathbf{y}_d(t) - \mathbf{y}(t)$ can only be guaranteed bounded.

Question 3.11 “It is assumed in Theorems 2 and 3 that this matrix with Lie derivatives of h is invertible. Does the example in Sect. VI satisfy this assumption?”

Answer: For the uncertain underactuated surface vessel consider in Sect. VI, one readily has

$$\dot{\mathbf{y}} = \begin{bmatrix} x_4 \cos x_3 - x_5 \sin x_3 - l x_6 \sin x_3 \\ x_4 \sin x_3 + x_5 \cos x_3 + l x_6 \cos x_3 \end{bmatrix} = L_f h(\mathbf{x}),$$

$$\begin{aligned} y_1^{[2]} &= \frac{m_{22}}{m_{11}} x_5 x_6 \cos x_3 - \frac{d_{11}}{m_{11}} x_4 \cos x_3 - x_4 x_6 \sin x_3 \\ &\quad + \frac{m_{11}}{m_{22}} x_4 x_6 \sin x_3 + \frac{d_{22}}{m_{22}} x_5 \sin x_3 - x_5 x_6 \cos x_3 \\ &\quad - \frac{m_{11} - m_{22}}{m_{33}} l x_4 x_5 \sin x_3 + \frac{d_{33}}{m_{33}} l x_6 \sin x_3 - l x_6^2 \cos x_3 \\ &\quad + \frac{1}{m_{11}} \cos x_3 u_1 - \frac{1}{m_{33}} l \sin x_3 u_2, \end{aligned}$$

and

$$\begin{aligned} y_2^{[2]} = & \frac{m_{22}}{m_{11}}x_5x_6\sin x_3 - \frac{d_{11}}{m_{11}}x_4\sin x_3 + x_4x_6\cos x_3 - \frac{m_{11}}{m_{22}}x_4x_6\cos x_3 \\ & - \frac{d_{22}}{m_{22}}x_5\cos x_3 - x_5x_6\sin x_3 \\ & + \frac{m_{11}-m_{22}}{m_{33}}lx_4x_5\cos x_3 - \frac{d_{33}}{m_{33}}lx_6\cos x_3 - lx_6^2\sin x_3 \\ & + \frac{1}{m_{11}}\sin x_3u_1 + \frac{1}{m_{33}}l\cos x_3u_2, \end{aligned}$$

which indicate that the underactuated surface vessel system has a relative degree of 2, i.e., $\rho = 2 < 4$. Besides, from the above derivation, one readily has

$$L_f L_g h(\mathbf{x}) = \begin{bmatrix} \cos x_3/m_{11} & -l\sin x_3/m_{33} \\ \sin x_3/m_{11} & l\cos x_3/m_{33} \end{bmatrix}$$

Since $l = 0.1$ and $m_{11} = m_{33} = 0.1$, the determinant of $L_f L_g h(\mathbf{x})$ is

$$\frac{l}{m_{11}m_{33}}\cos^2 x_3 + \frac{l}{m_{11}m_{33}}\sin^2 x_3 = \frac{l}{m_{11}m_{33}} = 10 > 0. \quad (3.43)$$

It follows that $L_f L_g h(\mathbf{x})$ is invertible.

Question 3.12 “What is the singularity problem?”

Answer: A necessary assumption for our method is that $L^g L_f^{\rho-1} h(\mathbf{x})$ is invertible. If the considered output of the surface vehicle is chosen as $\mathbf{y} = [x_1, x_2]^T$, then $\rho = 2$ and one has

$$L_g L_f h(\mathbf{x}) = \begin{bmatrix} \frac{\cos x_3}{m_{11}} & 0 \\ \frac{\sin x_3}{m_{11}} & 0 \end{bmatrix}.$$

Evidently, in this case, the determinant of $L_g L_f h(\mathbf{x})$ is zero, i.e., $L_g L_f h(\mathbf{x})$ is singular. In other words, $L_g L_f h(\mathbf{x})$ is not invertible in this case, which makes the proposed control unfeasible. This is referred to as the singularity problem.

Question 3.13 “How does convergence of $y(t)$ to $y_d(t)$ implies convergence of the derivatives? Note that this is generally not true. Consider the function $y(t) = 1/((t+1)(\sin t^2 + 2))$. We have $y(t)$ converges to 0, but $y'(t)$ does not. Unless additional assumptions are in place and used, this cannot be true.”

Answer: From Theorem 2, the closed-loop system consisting of the nonlinear system and the proposed adaptive near-optimal control law 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

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

where $v_i(t) = [0, \dots, 0, \delta_i(t)]^T \in \mathbb{R}^\rho$, $\mathbf{s}_i(t) = [e_i(t), \dot{e}_i(t), \dots, e_i^{[\rho-1]}(t)]^T \in \mathbb{R}^\rho$ with $i = 1, 2, \dots, m$, and

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

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 $\lim_{t \rightarrow +\infty} \mathbf{s}_i(t) = 0$ for all $i = 1, 2, \dots, m$. From Theorem 2, $\lim_{t \rightarrow +\infty} \delta(t) = 0$, i.e., $\lim_{t \rightarrow +\infty} v_i(t) = 0$ for all $i = 1, 2, \dots, m$. By the bounded-input bounded-output property of linear systems, the equilibrium $\mathbf{s}_i(t) = 0$ of linear system $\dot{\mathbf{s}}_i(t) = W\mathbf{s}_i(t) + v_i$ is thus asymptotically stable, i.e., $\lim_{t \rightarrow +\infty} \mathbf{s}_i(t) = 0$. Together with $\dot{\mathbf{s}}_i(t) = W\mathbf{s}_i(t) + v_i(t)$, it follows that $\lim_{t \rightarrow +\infty} \dot{\mathbf{s}}_i(t) = 0$. Recalling that $\mathbf{s}_i(t) = [e_i(t), \dot{e}_i(t), \dots, e_i^{[\rho-1]}(t)]^T$, one further has $\lim_{t \rightarrow +\infty} e_i^{[\rho]}(t) = 0$ for all $i = 1, 2, \dots, m$. It follows that $\lim_{t \rightarrow +\infty} \mathbf{e}^{[\rho]}(t) = 0$.

Question 3.14 “On the same page of Question 3.13, “It is obtained in the proof of Theorem 2 that...”. It is not clear how this displayed equation is proved in the proof of Theorem 2. Please be more specific and add more details accordingly.”

Answer: This equation is used to prove that $\lim_{t \rightarrow +\infty} \hat{J}_n(t) = 0$ when the proposed adaptive near-optimal control law is employed. From the proof of Theorem 2, one has $\lim_{t \rightarrow +\infty} \tilde{W}_f(t) = 0$ and $\lim_{t \rightarrow +\infty} \tilde{W}_g(t) = 0$, given that $\phi_f(\mathbf{x})$ and $\phi_g(\mathbf{x})\mathbf{u}_a(t)$ satisfy the persistent excitation condition. It follows that $\lim_{t \rightarrow +\infty} (\mathbf{u}(t) - \mathbf{u}_a(t)) = 0$, where $\mathbf{u}(t)$ is the nominal control law and minimizes the convex quadratic performance index $\hat{J}_n(t)$ with the optimal value being 0. It follows that $\lim_{t \rightarrow +\infty} \hat{J}_n(t) = 0$ when the adaptive near-optimal control law is employed.

Question 3.15 “What does asymptotic stability mean in Theorem 2? Does it mean attraction + Lyapunov stability in the usual sense? It would be helpful if the authors can clarify what has actually been proved.”

Answer: The authors sincerely thank the reviewer for pointing out the comment. The asymptotic stability in Theorem 2 means attraction + Lyapunov stability in

the usual sense. Specifically, the equilibrium $\mathbf{e}(t) = 0$ (with $\mathbf{e}(t) = \mathbf{y}_d(t) - \mathbf{y}(t)$) is asymptotically stable.

Question 3.16 “Finally, the authors termed their adaptive design ‘online’. I think this needs to be clarified. What does it mean by online? Does the computation achieve real-time performance? The authors also seem to suggest in the introduction that their methods are less computationally expensive, but there are no discussions of computational time later or in the example.”

Answer: The proposed adaptive design is termed ‘online’ since the parameters of the proposed adaptive near-optimal control laws are updated in a real-time manner. In other words, there is no need to do off-line training for the parameters of the proposed adaptive near-optimal control laws.

Before analyzing the computational complexity of the proposed adaptive near-optimal control method, we define a floating-point operation as one addition, subtraction, multiplication, or division of two floating-point numbers and recall the following facts.

- (1) The multiplication of a scalar and a vector in 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 in size $s_1 \times s_1$ requires s_1^3 floating-point operations.

Since nonlinear systems are more complicated than linear systems, we analyze the computational complexity of the proposed adaptive near-optimal control method when it is applied to nonlinear systems. In terms of the linear system case, the analysis on the computational complexity can be analyzed in a similar manner.

Computational Complexity of Nonlinear Auxiliary System: Consider the proposed nonlinear auxiliary system:

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

where $\hat{W}_f \in \mathbb{R}^{n \times N_f}$, $\phi_f(\mathbf{x}) \in \mathbb{R}^{N_f}$, $\hat{W}_g \in \mathbb{R}^{n \times N_g}$, $\phi_g(\mathbf{x}) \in \mathbb{R}^{N_g \times m}$, $\hat{\mathbf{x}} \in \mathbb{R}^n$, $\mathbf{x} \in \mathbb{R}^n$, $K_f \in \mathbb{R}^{n \times n}$, $K_g \in \mathbb{R}^{n \times n}$, $K_{\mathbf{x}} \in \mathbb{R}^{n \times n}$, and $\mathbf{u}(t) \in \mathbb{R}^m$. The analysis on the computational complexity of the nonlinear auxiliary system can be done by firstly discretizing it. By using the Euler difference formula, the corresponding discrete-time nonlinear auxiliary system is obtained as follows:

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

where $\tau > 0 \in \mathbb{R}$ is the sampling gap and $k = 1, 2, \dots$ is the updating index. To utilize the proposed adaptive near-optimal control, 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 $n + n + n(2N_f - 1) + (2nN_g m - nm) + n(2m - 1) + n(2n - 1) + n + nc_{\max} + mN_g c_{\max} = mn(1 + 2N_g) + 2n(N_f + n) + c_{\max}(n + mN_g)$ floating-point operations; computing \hat{W}_f^{k+1} requires less than $nN_f + nN_f + n(2n - 1) + n + mn + N_f c_{\max} = mn + 2n^2 + N_f(2n + c_{\max})$ floating-point operations; computing \hat{W}_g^{k+1} requires less than $nN_g + nN_g + n(2n - 1) + n + mn + 2nmN_g - nN_g + mN_g c_{\max} = mn(1 + 2N_g) + 2n^2 + nN_g + mN_g c_{\max}$ floating-point operations. In total, the proposed 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.

Computational Complexity of Control Law: Consider the proposed adaptive near-optimal control law for nonlinear systems:

$$\mathbf{u}_a(t) = (TR + \kappa(L_{\hat{g}} L_{\hat{f}}^{\rho-1} h(\mathbf{x}))^T Q L_{\hat{g}} L_{\hat{f}}^{\rho-1} h(\mathbf{x}))^{-1} (Q L_{\hat{g}} L_{\hat{f}}^{\rho-1} h(\mathbf{x}))^T (Y_d(t) - \hat{Y}_n(t)) \mathbf{v}^T,$$

where $TR \in \mathbb{R}^{m \times m}$, $\mathbf{v}^T \in \mathbb{R}^{\rho+1}$, $Q \in \mathbb{R}^{m \times m}$, $\kappa \in \mathbb{R}$ are constant and do not need to be calculated at each sampling time instant; $Y_d(t) \in \mathbb{R}^{m \times (\rho+1)}$ are calculated beforehand; $L_{\hat{g}} L_{\hat{f}}^{\rho-1} h(\mathbf{x}) \in \mathbb{R}^{m \times m}$ and $\hat{Y}_n(t) \in \mathbb{R}^{m \times (\rho+1)}$ 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 $L_{\hat{g}} L_{\hat{f}}^{\rho-1} h(\mathbf{x}) \in \mathbb{R}^{m \times m}$ or $\hat{Y}_n(t) \in \mathbb{R}^{m \times (\rho+1)}$.

Therefore, computing $\mathbf{u}_a(t)$ for one time requires less than $m^2 + m^2 + m^2 + 2m^3 - m^2 + 2m^3 - m^2 + m^3 + 2m^3 - m^2 + 2m^3 - m^2 + 2m^2(\rho + 1) - m(\rho + 1) + m(\rho + 1) + m(2(\rho + 1) - 1) + c'_{\max}(m^2 + m(\rho + 1)) = 9m^3 + (1 + 2\rho)m^2 + 2m\rho + m + c'_{\max}(m^2 + m\rho + m)$ floating-point operations.

Total Computational Complexity: The computational complexity of the proposed adaptive near-optimal control method consist of two parts: updating the auxiliary system and computing the adaptive near-optimal control law based on the parameter matrices generated by the auxiliary system. At each sampling time instant, both the control law and the auxiliary system need updating. Therefore, the total computational complexity of the proposed adaptive near-optimal control method

for nonlinear systems is the sum of the two parts. It follows that the proposed adaptive near-optimal control method for nonlinear systems 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.

Floating-Point Operations Needed for Application to Uncertain Underactuated Surface Vessel: For the application to the uncertain underactuated surface vessel, which is shown in Sect. VI of the manuscript, we have $m = 2$, $n = 6$, $N_f = 8$, and $N_g = 2$. Practically, 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.

In this application, $\phi_f(\mathbf{x}) \in \mathbb{R}^8$ with $\phi_{f1}(\mathbf{x}) = x_4 \cos x_3$, $\phi_{f2}(\mathbf{x}) = x_5 \sin x_3$, $\phi_{f3}(\mathbf{x}) = x_4 \sin x_3$, $\phi_{f4}(\mathbf{x}) = x_5 \cos x_3$, $\phi_{f5}(\mathbf{x}) = x_6$, $\phi_{f6}(\mathbf{x}) = x_5 x_6$, $\phi_{f7}(\mathbf{x}) = x_4 x_6$, $\phi_{f8}(\mathbf{x}) = x_5$, and $\phi_g(\mathbf{x}) = [1, 0; 0, 1]$. 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 $\phi_g(\mathbf{x}^k)$ is 41, i.e., $c_{\max} = 41$. Therefore, in this application, at each sampling time instant the nonlinear auxiliary system requires less than $mn(3 + 4N_g) + 4nN_f + 6n^2 + c_{\max}(n + 2mN_g + N_f) + nN_g = 2 \times 6 \times (3 + 4 \times 2) + 4 \times 6 \times 8 + 6 \times 6^2 + 41 \times (6 + 2 \times 2 \times 2 + 8) + 6 \times 2 = 1454$ floating-point operations.

In this application, $\hat{Y}_n(t) = [\mathbf{y}(t), L_{\hat{f}}h(\mathbf{x}), L_{\hat{f}}^2h(\mathbf{x})] \in \mathbb{R}^{2 \times 3}$, with $\mathbf{y}(t) = [y_1(t), y_2(t)]^T$,

$$\begin{aligned} L_{\hat{f}}h(\mathbf{x}) &= \begin{bmatrix} w_{f11}x_4 \cos x_3 + (w_{f12}x_5 - w_{f35}l x_6) \sin x_3 \\ w_{f23}x_4 \sin x_3 + (w_{f24}x_5 + w_{f35}l x_6) \cos x_3 \end{bmatrix}, \\ L_{\hat{f}}^2h(\mathbf{x})_1 &= (w_{f11}w_{f46}x_5x_6 + w_{f12}w_{f35}x_5x_6 - w_{f35}w_{f35}lx_6^2) \cos x_3 + (w_{f12}w_{f57}x_4x_6 \\ &\quad - w_{f11}w_{f35}x_4x_6 + w_{f12}w_{f58}x_5), \\ L_{\hat{f}}^2h(\mathbf{x})_2 &= (w_{f23}w_{f35}x_4x_6 + w_{f24}w_{f57}x_4x_6 + w_{f24}w_{f58}x_5 \\ &\quad + (w_{f23}w_{f46}x_5x_6 - w_{f35}w_{f35}lx_6^2 - w_{f24}w_{f35}x_5x_6) \sin x_3). \end{aligned}$$

$L_{\hat{g}}L_{\hat{f}}^{\rho-1}h(\mathbf{x}) = [\hat{w}_{f11}\hat{w}_{g41} \cos(x_3), -\hat{w}_{f35}\hat{w}_{g62}l \sin(x_3); \hat{w}_{f23}\hat{w}_{g41} \sin(x_3), \hat{w}_{f35}\hat{w}_{g62}l \cos(x_3)]$. It follows that 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 $\hat{Y}_n(t)$ depends on $L_{\hat{f}}^2h(\mathbf{x})_1$, which costs 64 floating-point operations, i.e., $c'_{\max} = 64$. Therefore, in this application, at each sampling time instant t , computing $\mathbf{u}_a(t)$ requires less than $9m^3 + (1 + 2\rho)m^2 + 2m\rho + m + c'_{\max}(m^2 + m\rho + m) = 9 \times 2^3 + (1 + 2 \times 2) \times 2^2 + 2 \times 2 \times 2 + 2 + 64 \times (2^2 + 2 \times 2 + 2) = 742$ floating-point operations.

Therefore, in the application to the underactuated surface vessel, the proposed adaptive near-optimal control 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 gap is selected as 0.01 s, then the computational

complexity of the proposed adaptive near-optimal control method for the underactuated surface vessel is less than 2.194×10^5 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 3.75×10^8 floating-point operations per second. In other words, the proposed adaptive near-optimal control method can be implemented in a real-time manner. Note that the number of required floating-point operations can be further reduced by improving the implementation of the proposed control method. For example, we can introduce a variable to store $\hat{\mathbf{x}}^k - \mathbf{x}^k$ 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} . One can also introduce a variable to store the values of $\cos x_3$ and $\sin x_3$ so as to avoid redundant computation of the values at the same update.

Question 3.17 “*I think that the assumption that all state variables are available is very restrictive and not very realistic when considering real applications.*”

Answer: The assumption that all state variables are available is generally made in related researches. For mechanical systems, such as robot manipulators, sensors can be used to measure all the state variables. When state variables cannot be directly measured, state observers may be developed, which would be investigated in the future.

Question 3.18 “*How are the gain matrices of the auxiliary system chosen?*”

Answer: The gain matrices of auxiliary systems are used to adjust the learning of the auxiliary system for the input-output properties of the controlled system. For the linear auxiliary system:

$$\begin{cases} \dot{\hat{\mathbf{x}}}(t) = \hat{A}\mathbf{x}(t) + \hat{B}\mathbf{u}(t) - K_x(\hat{\mathbf{x}}(t) - \mathbf{x}(t)), \\ \dot{\hat{A}} = -K_A(\hat{\mathbf{x}}(t) - \mathbf{x}(t))\mathbf{x}^T(t), \\ \dot{\hat{B}} = -K_B(\hat{\mathbf{x}}(t) - \mathbf{x}(t))\mathbf{u}^T(t). \end{cases}$$

The values of the diagonal elements in positive-definite matrix

$$K_x = \text{diag}([k_{x_1}, k_{x_2}, \dots, k_{x_n}]) \in \mathbb{R}^{n \times n}$$

are at least 10 times smaller than those in positive-definite matrices

$$K_A = \text{diag}([k_{A_1}, k_{A_2}, \dots, k_{A_n}]) \in \mathbb{R}^{n \times n}$$

and $K_B = \text{diag}([k_{B_1}, k_{B_2}, \dots, k_{B_n}]) \in \mathbb{R}^{n \times n}$. This setting can guarantee that parameter matrices \hat{A} and \hat{B} take a dominant role in learning the dynamics of the controlled system. Under the above condition, with larger values in the gain matrices, the convergence of $\hat{\mathbf{x}}(t)$ to $\mathbf{x}(t)$ is faster. Note that too large values in the gain matrices may lead to high overshooting. In this sense, they cannot be too large. The above manner also holds for choosing gain matrices K_x , K_f and K_g in the nonlinear auxiliary system.

3.8 Summary

In this chapter, a unified online adaptive near-optimal control framework has been presented for linear and nonlinear systems with parameter uncertainty. Based on this framework, online adaptive near-optimal control laws with a simple structure and no requirement for knowing system parameter values have been presented. Theoretical analysis has shown that closed-loop systems based on the presented control are asymptotically stable. It has also been proved that the presented adaptive near-optimal control laws asymptotically converge to the optimal. The application of the presented 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 presented framework can deal with nonlinear systems with large parameter uncertainty and enjoys model simplicity, online computation, adaptivity, and near-optimality. Further work of this chapter would be the adaptive near-optimal control of nonlinear systems with state observers or measurement noises considered. For example, robust control methodologies might be incorporated into the control design so as to enhance the efficacy of the presented control laws.

References

1. Lewis, F.L., Vrabie, D., Syrmos, V.: *Optimal Control*, 3rd edn. Wiley, New York (2012)
2. Kim, E.K., Mwasilu, F., Choi, H.H., Jung, J.W.: An observer-based optimal voltage control scheme for three-phase UPS systems. *IEEE Trans. Ind. Electron.* **62**(4), 2073–2081 (2015)
3. Moghadasi, S., Kamalasadan, S.: Optimal fast control and scheduling of power distribution system using integrated receding horizon control and convex conic programming. *IEEE Trans. Ind. Appl.* **52**(3), 2596–2606 (2016)
4. Dierks, T., Brenner, B., Jagannathan, S.: Neural network-based optimal control of mobile robot formations with reduced information exchange. *IEEE Trans. Control Syst. Technol.* **21**(4), 1407–1415 (2013)
5. Trelat, E.: Optimal control and applications to aerospace: some results and challenges. *J. Optim. Theory Appl.* **254**(3), 713–758 (2012)
6. Chen, W.H., Ballance, D.J., Gawthrop, P.J.: Optimal control of nonlinear systems: a predictive control approach. *Automatica* **39**(4), 633–641 (2003)
7. Abu-Khalaf, M., Lewis, F.L.: Nearly optimal control laws for nonlinear systems with saturating actuators using a neural network HJB approach. *Automatica* **41**(5), 779–791 (2005)
8. Wang, Z., Liu, X., Liu, K., Li, S., Wang, H.: Backstepping-based Lyapunov function construction using approximate dynamic programming and sum of square techniques. *IEEE Trans. Cybern.*, in press
9. Sassano, M., Astolfi, A.: Dynamic approximate solutions of the HJ inequality and of the HJB equation for input-affine nonlinear systems. *IEEE Trans. Autom. Control* **57**(10), 2490–2503 (2012)
10. Vamvoudakis, K.G., Lewis, F.L.: Online actor-critic algorithm to solve the continuous-time infinite horizon optimal control problem. *Automatica* **46**(5), 878–888 (2010)
11. Tao, G.: Multivariable adaptive control: a survey. *Automatica* **50**(11), 2737–2764 (2014)
12. Na, J., Chen, Q., Ren, X., Guo, Y.: Adaptive prescribed performance motion control of servo mechanisms with friction compensation. *IEEE Trans. Control Syst. Technol.* **61**(1), 486–494 (2014)

13. Li, G., Na, J., Stoten, D.P., Ren, X.: Adaptive neural network feedforward control for dynamically substructured systems. *IEEE Trans. Control Syst. Technol.* **22**(3), 944–954 (2014)
14. Modares, H., Lewis, F.L., Naghibi-Sistani, M.B.: Adaptive optimal control of unknown constrained-input systems using policy iteration and neural networks. *IEEE Trans. Neural Netw. Learn. Syst.* **24**(10), 1513–1525 (2013)
15. Wang, F.Y., Zhang, H., Liu, D.: Adaptive dynamic programming: an introduction. *IEEE Control Syst. Mag.* **4**(2), 39–47 (2009)
16. Liu, D., Wei, Q.: Policy iteration adaptive dynamic programming algorithm for discrete-time nonlinear systems. *IEEE Trans. Neural Netw. Learn. Syst.* **25**(3), 621–633 (2014)
17. Heydari, A., Balakrishnan, S.N.: Finite-horizon control-constrained nonlinear optimal control using single network adaptive critics. *IEEE Trans. Neural Netw. Learn. Syst.* **24**(1), 145–157 (2013)
18. Liu, Y.J., Tang, L., Tong, S., Chen, C.L.P., Li, D.J.: Reinforcement learning design-based adaptive tracking control with less learning parameters for nonlinear discrete-time MIMO systems. *IEEE Trans. Neural Netw. Learn. Syst.* **26**(1), 165–176 (2015)
19. Wang, D., Liu, D., Li, H.: Policy iteration algorithm for online design of robust control for a class of continuous-time nonlinear systems. *IEEE Trans. Autom. Sci. Eng.* **11**(2), 627–632 (2014)
20. Zhang, H., Wei, Q., Luo, Y.: 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.* **38**(4), 937–942 (2008)
21. Lv, Y., Na, J., Yang, Q., Wu, X., Guo, Y.: Online adaptive optimal control for continuous-time nonlinear systems with completely unknown dynamics. *Int. J. Control.* **89**(1), 99–112 (2016)
22. Mayne, D.Q., Michalska, H.: Receding horizon control of nonlinear systems. *IEEE Trans. Autom. Control* **35**(7), 814–824 (1990)
23. Li, S., Chen, S., Liu, B., Li, Y., Liang, Y.: Decentralized kinematic control of a class of collaborative redundant manipulators via recurrent neural networks. *Neurocomputing* **91**, 1–10 (2012)
24. Li, S., Cui, H., Li, Y., Liu, B., Lou, Y.: Decentralized control of collaborative redundant manipulators with partial command coverage via locally connected recurrent neural networks. *Neural Comput. Appl.* **23**(3), 1051–1060 (2013)
25. Jin, L., Zhang, Y., Li, S., Zhang, Y.: Modified ZNN for time-varying quadratic programming with inherent tolerance to noises and its application to kinematic redundancy resolution of robot manipulators. *IEEE Trans. Ind. Electron.* **63**(11), 6978–6988 (2016)
26. Li, S., He, J., Li, Y., Rafique, M.U.: Distributed recurrent neural networks for cooperative control of manipulators: a game-theoretic perspective. *IEEE Trans. Neural Netw. Learn. Syst.* **28**(2), 415–426 (2017)
27. Jin, L., Li, S., La, H.M., Luo, X.: Manipulability optimization of redundant manipulators using dynamic neural networks. *IEEE Trans. Ind. Electron.* **64**(6), 4710–4720 (2017)
28. Li, Y., Li, S., Hannaford, B.: A novel recurrent neural network for improving redundant manipulator motion planning completeness. In: ICRA, pp. 2956–2961 (2018)
29. Zhang, Y., Li, S.: A neural controller for image-based visual servoing of manipulators with physical constraints. *IEEE Trans. Neural Netw. Learn. Syst.* **29**(11), 5419–5429 (2018)
30. Li, S., Zhou, M., Luo, X.: Modified primal-dual neural networks for motion control of redundant manipulators with dynamic rejection of harmonic noises. *IEEE Trans. Neural Netw. Learn. Syst.* **29**(10), 4791–4801 (2018)
31. Li, S., Wang, H., Rafique, M.U.: A novel recurrent neural network for manipulator control with improved noise tolerance. *IEEE Trans. Neural Netw. Learn. Syst.* **29**(5), 1908–1918 (2018)
32. Jin, L., Li, S., Luo, X., Li, Y., Qin, B.: Neural dynamics for cooperative control of redundant robot manipulators. *IEEE Trans. Ind. Inform.* **14**(9), 3812–3821 (2018)
33. Li, J., Zhang, Y., Li, S., Mao, M.: New discretization-formula-based zeroing dynamics for real-time tracking control of serial and parallel manipulators. *IEEE Trans. Ind. Inform.* **14**(8), 3416–3425 (2018)

34. Chen, D., Zhang, Y., Li, S.: Tracking control of robot manipulators with unknown models: a Jacobian-matrix-adaption method. *IEEE Trans. Ind. Inform.* **14**(7), 3044–3053 (2018)
35. Zhang, Y., Li, S., Gui, J., Luo, X.: Velocity-level control with compliance to acceleration-level constraints: a novel scheme for manipulator redundancy resolution. *IEEE Trans. Ind. Inform.* **14**(3), 921–930 (2018)
36. Xiao, L., Liao, B., Li, S., Zhang, Z., Ding, L., Jin, L.: Design and analysis of FTZNN applied to the real-time solution of a nonstationary Lyapunov equation and tracking control of a wheeled mobile manipulator. *IEEE Trans. Ind. Inform.* **14**(1), 98–105 (2018)
37. Zhang, Y., Chen, S., Li, S., Zhang, Z.: Adaptive projection neural network for kinematic control of redundant manipulators with unknown physical parameters. *IEEE Trans. Ind. Electron.* **65**(6), 4909–4920 (2018)
38. Zhang, Z., Lin, Y., Li, S., Li, Y., Yu, Z., Luo, Y.: Tricriteria optimization-coordination motion of dual-redundant-robot manipulators for complex path planning. *IEEE Trans. Control Syst. Technol.* **26**(4), 1345–1357 (2018)
39. Jin, L., Li, S., Hu, B., Yi, C.: Dynamic neural networks aided distributed cooperative control of manipulators capable of different performance indices. *Neurocomputing* **291**, 50–58 (2018)
40. Jin, L., Li, S., Yu, J., He, J.: Robot manipulator control using neural networks: a survey. *Neurocomputing* **285**, 23–34 (2018)
41. Chen, D., Zhang, Y., Li, S.: Zeroing neural-dynamics approach and its robust and rapid solution for parallel robot manipulators against superposition of multiple disturbances. *Neurocomputing* **275**, 845–858 (2018)
42. Li, S., Shao, Z., Guan, Y.: A dynamic neural network approach for efficient control of manipulators. *IEEE Trans. Syst. Man Cybern. Syst.* **49**(5), 932–941 (2019)
43. Zhang, Y., Li, S., Zhou, X.: Recurrent-neural-network-based velocity-level redundancy resolution for manipulators subject to a joint acceleration limit. *IEEE Trans. Ind. Electron.* **66**(5), 3573–3582 (2019)
44. Zhang, Z., Chen, S., Li, S.: Compatible convex-nonconvex constrained QP-based dual neural networks for motion planning of redundant robot manipulators. *IEEE Trans. Control Syst. Technol.* **27**(3), 1250–1258 (2019)
45. Xu, Z., Li, S., Zhou, X., Yan, W., Cheng, T., Huang, D.: Dynamic neural networks based kinematic control for redundant manipulators with model uncertainties. *Neurocomputing* **329**, 255–266 (2019)
46. Li, S., Zhang, Y., Jin, L.: Kinematic control of redundant manipulators using neural networks. *IEEE Trans. Neural Netw. Learn. Syst.*, in press
47. Zhang, Y., Wang, J., Xia, Y.: A dual neural network for redundancy resolution of kinematically redundant manipulators subject to joint limits and joint velocity limits. *IEEE Trans. Neural Netw.* **14**(3), 658–667 (2003)
48. Zhang, Z., Li, Z., Zhang, Y., Luo, Y., Li, Y.: Neural-dynamic-method-based dual-arm CMG scheme with time-varying constraints applied to humanoid robots. *IEEE Trans. Neural Netw. Learn. Syst.* **26**(12), 3251–3262 (2015)
49. Zhang, Y., Li, W., Zhang, Z.: Physical-limits-constrained minimum velocity norm coordinating scheme for wheeled mobile redundant manipulators. *Robotica* **33**(6), 1325–1350 (2015)
50. Mohammed, A.M., Li, S.: Dynamic neural networks for kinematic redundancy resolution of parallel Stewart platforms. *IEEE Trans. Cybern.* **46**(7), 1538–1550 (2016)
51. Jin, L., Zhang, Y.: G2-type SRMPC scheme for synchronous manipulation of two redundant robot arms. *IEEE Trans. Cybern.* **45**(2), 153–164 (2015)
52. Jin, L., Li, S., Xiao, L., Lu, R., Liao, B.: Cooperative motion generation in a distributed network of redundant robot manipulators with noises. *IEEE Trans. Syst. Man Cybern. Syst.* **48**(10), 1715–1724 (2018)
53. Jin, L., Li, S.: Distributed task allocation of multiple robots: a control perspective. *IEEE Trans. Syst. Man Cybern. Syst.* **48**(5), 693–701 (2018)
54. Liu, Y.K., Zhang, Y.M.: Model-based predictive control of weld penetration in gas tungsten arc welding. *IEEE Trans. Control Syst. Technol.* **22**(3), 955–966 (2014)

55. Akter, M.P., Mekhilef, S., Tan, N.M.L., Akagi, H.: Modified model predictive control of a bidirectional AC-DC converter based on Lyapunov function for energy storage systems. *IEEE Trans. Ind. Electron.* **63**(2), 704–715 (2016)
56. Mayne, D.Q.: Model predictive control: recent developments and future promise. *Automatica* **50**(12), 2967–2986 (2014)
57. Pradhan, S.K., Subudhi, B.: Nonlinear adaptive model predictive controller for a flexible manipulator: an experimental study. *IEEE Trans. Control Syst. Technol.* **22**(5), 1754–1768 (2014)
58. Zhang, Y., Li, S.: Predictive suboptimal consensus of multiagent systems with nonlinear dynamics. *IEEE Trans. Syst. Man Cybern. Syst.* **47**(7), 1701–1711 (2017)
59. Jiang, Z.: Global tracking control of underactuated ships by Lyapunov's direct method. *Automatica* **38**(2), 301–309 (2002)
60. Behal, A., Dawson, D.M., Dixon, W.E., Fang, Y.: Tracking and regulation control of an underactuated surface vessel with nonintegrable dynamics. *IEEE Trans. Autom. Control* **47**(3), 495–500 (2002)
61. Yan, Z., Wang, J.: Model predictive control for tracking of underactuated vessels based on recurrent neural networks. *IEEE J. Ocean. Eng.* **37**(4), 717–726 (2012)
62. Yu, R., Zhu, Q., Xia, G., Liu, Z.: Sliding mode tracking control of an underactuated surface vessel. *IET Control Theory Appl.* **6**(3), 461–466 (2012)
63. Ashrafiuon, H., Muske, K.R., McNinch, L.C., Soltan, R.A.: Sliding-mode tracking control of surface vessels. *IEEE Trans. Ind. Electron.* **55**(11), 4004–4012 (2008)
64. Elmokadem, T., Zribi, M., Youcef-Toumi, K.: Trajectory tracking sliding mode control of underactuated AUVs. *Nonlinear Dyn.* **84**(2), 1079–1091 (2016)
65. Hu, C., Wang, R., Yan, F., Chen, N.: Robust composite nonlinear feedback path-following control for underactuated surface vessels with desired-heading amendment. *IEEE Trans. Ind. Electron.* **63**(10), 6386–6394 (2016)
66. Pan, C.-Z., Lai, X.-Z., Yang, S.X., Wu, M.: A biologically inspired approach to tracking control of underactuated surface vessels subject to unknown dynamics. *Expert Syst. Appl.* **42**(4), 2153–2161 (2015)
67. Zhang, Y., Chen, D., Jin, L., Zhang, Y., Yin, Y.: GD-aided IOL (input-output linearisation) controller for handling affine-form nonlinear system with loose condition on relative degree. *Int. J. Control.* **89**(4), 757–769 (2016)
68. Isidori, A.: *Nonlinear Control Systems: An Introduction*, 3rd edn. Springer, New York (1995)
69. Sun, W., Tang, S., Gao, H., Zhao, J.: Two time-scale tracking control of nonholonomic wheeled mobile robots. *IEEE Trans. Control Syst. Technol.* **24**(6), 2059–2069 (2016)
70. Mehrabian, A.R., Khorasani, K.: Distributed formation recovery control of heterogeneous multiagent Euler-Lagrange systems subject to network switching and diagnostic imperfections. *IEEE Trans. Control Syst. Technol.* **24**(6), 2158–2166 (2016)
71. Chen, C.T.: *Linear System Theory and Design*. Oxford University Press, New York (1999)
72. Yang, C., Li, Z., Cui, R., Xu, B.: Neural network-based motion control of underactuated wheeled inverted pendulum models. *IEEE Trans. Neural Netw. Learn. Syst.* **25**(11), 2004–2016 (2014)
73. Boyd, S., Vandenberghe, L.: *Convex Optimization*. Cambridge University Press, Cambridge (2004)
74. Liao, B., Zhang, Y., Jin, L.: Taylor $O(h^3)$ discretization of ZNN models for dynamic equality-constrained quadratic programming with application to manipulators. *IEEE Trans. Neural Netw. Learn. Syst.* **27**(2), 225–237 (2016)
75. Khalil, H.K.: *Nonlinear Systems*, 3rd edn. Prentice-Hall, Upper Saddle River (2002)
76. Kamalapurkar, R., Reish, B., Chowdhary, G., Dixon, W.E.: Concurrent learning for parameter estimation using dynamic state-derivative estimators. *IEEE Trans. Autom. Control* **62**(7), 3594–3601 (2017)
77. Kersting, S., Buss, M.: Direct and indirect model reference adaptive control for multivariable piecewise affine systems. *IEEE Trans. Autom. Control* **62**(11), 5634–5649 (2017)
78. Royden, H.L.: *Real Analysis*. Macmillan, New York (1988)
79. Liu, Z., Li, C., Xu, W.: Hybrid control of biped robots in the double-support phase H_∞ approach and fuzzy neural networks. *IEE Proc. Control Theory Appl.* **150**(4), 347–354 (2003)

Chapter 4

Adaptive Near-Optimal Control Using Sliding Mode



Abstract In this chapter, an adaptive near-optimal controller, 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 presented adaptive 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 presented controller guarantees asymptotic stability of the closed-system and asymptotic optimality of the performance index with time. Two illustrative examples and an application of the presented method to a van der Pol oscillator are presented to validate the efficacy of the presented adaptive 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 presented method.

4.1 Introduction

In recent decades, adaptive control of uncertain nonlinear systems has received much attention due to practical requirements. For example, an adaptive control technique based on barrier Lyapunov functions was developed in [1] for a class of nonlinear systems with full state constraints. The adaptive asymptotic tracking control of uncertain nonlinear system with input quantization was investigated in [2] on the basis of a novel decomposition of hysteresis quantizer. There are also adaptive control methods based on fuzzy systems, e.g., [3, 4]. However, the above reported results as well as traditional adaptive control cannot be directly used in situations with optimality requirements.

As an important research domain, optimal control aims at driving a controlled system to the desired state in an optimal way by optimizing a given cost function or performance index. A special case is the optimal kinematic control of redundant manipulators [5–32]. In terms of linear systems, Vamvoudakis [33] proposed a novel model-free infinite-horizon optimal control method for continuous-time

linear systems by using Q-learning. For nonlinear systems, traditional optimal control methods require solving a Hamilton–Jacobi–Bellman (HJB) equation [34]. When it comes to receding-horizon optimal control [35], the corresponding HJB equation is time-varying, which is more difficult to solve analytically. Johansen [36] conducted an in-depth survey of recent advancements of dependable embedded model predictive control methods, which are an alternative for dealing with the problem. Recently, Chakrabarty et al. [37] showed an interesting application of support vector machine in the design of an explicit nonlinear model predictive controller for the stabilization of nonlinear systems to improve computational efficiency. In [38], Taylor expansion was introduced to solve the receding-horizon optimal control problem of fully known nonlinear systems and an explicit controller is derived, which significantly reduces the computational effort. In [39], the method in [38] was extended to the consensus of second-order nonlinear multi-agent systems with fully known system dynamics. In [40], the method was further extended to the control of an underactuated system with fully known system dynamics and an input constraint. However, in [35, 38–40], system parameters are required to be exactly known, which may be restricted. Specifically, in practical applications, to implement these controllers, we need to first conduct parameter identification to obtain exact parameter values so as to guarantee the performance of the controller. Although there are versatile methodologies available in the literature for parameter identification, such as those reported in [41–43], it is still more favorable if an adaptive optimal controller can be employed which can adapt to the variation of system parameters. It is remarkable that Pin et al. [44] proposed a novel robust receding-horizon control method for a constrained discrete-time nonlinear systems with model uncertainty.

In this chapter, an adaptive near-optimal controller is presented 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. The results in this chapter can be viewed as an extension of those in [38] to nonlinear systems with unknown parameters. The presented adaptive near-optimal control is also inspired by the concept of sliding mode control [45, 46]. 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 [47] and the formation control problem [48]. Instead of directly using sliding mode control concept to design controllers, 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.

The remainder of the chapter is organized as follows. In Sect. 4.2, the problem formulation and preliminary are provided. In Sect. 4.3, we present the nominal near-optimal design for fully known nonlinear systems. In Sect. 4.4, we present the adaptive near-optimal control for nonlinear systems with fully unknown parameters. In Sect. 4.5, two illustrative examples are presented to demonstrate the efficacy of the presented adaptive near-optimal control. In Sect. 4.6, an application of the presented

control method to a van der Pol oscillator is presented to further show the efficacy and practical significance of the presented method. Then, experimental results are presented and compared in Sect. 4.7. Finally, our conclusions are given in Sect. 4.9.

4.2 Problem Formulation and Preliminary

Consider the following nonlinear system:

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

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; $f(\cdot) : \mathbb{R}^n \rightarrow \mathbb{R}^n$, $g(\cdot) : \mathbb{R}^n \rightarrow \mathbb{R}^{n \times m}$, and $h(\cdot) : \mathbb{R}^n \rightarrow \mathbb{R}^m$ are smooth functions. We consider the case that the system satisfies the following conditions: (1) The zero dynamics of system (4.1) are stable [38, 49]; (2) system (4.1) has a known relative degree ρ and $L_g L_f^{\rho-1} h(\mathbf{x})$ is invertible [38, 49]; (3) State variables and the output of system (4.1) as well as its output derivatives up to $\rho - 1$ order are measurable or observable [38, 50, 51].

The receding-horizon optimal control problem about system (4.1) is formulated as

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

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

$$\begin{aligned} J(t) = & \int_0^T (\mathbf{y}_d(t + \tau) - \mathbf{y}(t + \tau))^T Q (\mathbf{y}_d(t + \tau) \\ & - \mathbf{y}(t + \tau)) d\tau, \end{aligned} \quad (4.3)$$

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 $\mathbf{y}_d(t)$ denoting the desired output is a continuously differentiable function.

The following definitions about system (4.1) are used in this chapter.

Definition 4.1 ([49]) 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 $i = 0$, $L_f^0 h(\mathbf{x}) = h(\mathbf{x})$; and, for $i > 0$, $L_f^i h(\mathbf{x})$ is defined by $L_f^i h(\mathbf{x}) = \partial L_f^{i-1} h(\mathbf{x}) / \partial \mathbf{x} f(\mathbf{x})$. Similarly, $L_g L_f^i h(\mathbf{x})$ is defined by $L_g L_f^i h(\mathbf{x}) = \partial L_f^i h(\mathbf{x}) / \partial \mathbf{x} g(\mathbf{x})$.

Definition 4.2 ([49]) System (4.1) is said to have a relative degree of ρ if the following two properties holds true: (1) $\forall \mathbf{x} \in \mathbb{R}^n$, $L_g L_f^i h(\mathbf{x}) = 0$, for $0 \leq i < \rho - 1$; (2) $\forall \mathbf{x} \in \mathbb{R}^n$, $L_g L_f^{\rho-1} h(\mathbf{x}) \neq 0$.

4.3 Nominal Near-Optimal Design

In this section, the design process of near-optimal controllers for nonlinear system (4.1) is presented under the assumption that all the parameters of nonlinear system (4.1) are known [38]. For nonlinear system (4.1) with relative degree ρ , based on Definitions 4.1 and 4.2, one has

$$\begin{cases} \dot{\mathbf{y}}(t) = L_f h(\mathbf{x}(t)), \\ \vdots \\ \mathbf{y}^{[\rho-1]}(t) = L_f^{\rho-1} h(\mathbf{x}(t)), \\ \mathbf{y}^{[\rho]}(t) = L_f^\rho h(\mathbf{x}(t)) + L_g L_f^{\rho-1} h(\mathbf{x}(t)) \mathbf{u}(t), \end{cases} \quad (4.4)$$

where $\mathbf{y}^{[i]}(t)$ denotes the i th-order derivative of $\mathbf{y}(t)$ with respect to time instant t , for $i = 1, 2, \dots, \rho$. Let $Y(t) = [\mathbf{y}(t), L_f h(\mathbf{x}(t)), \dots, L_f^{\rho-1} h(\mathbf{x}(t)), L_f^\rho h(\mathbf{x}(t))]$. Then, by using Taylor expansion, $\mathbf{y}(t + \tau)$ of nonlinear system (4.1) can be approximated as

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

where $\mathbf{w}(\tau) = [1, \tau, \dots, \tau^{\rho-1}/(\rho-1)!, \tau^\rho/\rho!]^\top$. Similarly, $\mathbf{y}_d(t + \tau) \approx Y_d(t)\mathbf{w}(\tau)$ and $\mathbf{u}(t + \tau) \approx \mathbf{u}(t)$, where $Y_d(t) = [\mathbf{y}_d(t), \dot{\mathbf{y}}_d(t), \dots, \mathbf{y}_d^{[\rho]}(t)]$.

Performance index $J(t)$ in Eq. (4.3) is thus approximated as

$$\begin{aligned} J(t) &\approx \hat{J}(t) \\ &= \int_0^T \left(E(t)\mathbf{w}(\tau) - \frac{\tau^\rho}{\rho!} L_g L_f^{\rho-1} h(\mathbf{x}(t)) \mathbf{u}(t) \right)^\top Q \left(E(t)\mathbf{w}(\tau) \right. \\ &\quad \left. - \frac{\tau^\rho}{\rho!} L_g L_f^{\rho-1} h(\mathbf{x}(t)) \mathbf{u}(t) \right) d\tau \\ &= \int_0^T \mathbf{w}^\top(\tau) E^\top(t) Q E(t) \mathbf{w}(\tau) d\tau - 2 \int_0^T \frac{\tau^\rho}{\rho!} \mathbf{w}^\top(\tau) d\tau E^\top(t) Q L_g L_f^{\rho-1} h(\mathbf{x}(t)) \mathbf{u}(t) \\ &\quad + \int_0^T \frac{\tau^{2\rho}}{(\rho!)^2} d\tau \mathbf{u}^\top(t) (L_g L_f^{\rho-1} h(\mathbf{x}(t)))^\top Q L_g L_f^{\rho-1} h(\mathbf{x}(t)) \mathbf{u}(t), \end{aligned} \quad (4.5)$$

where $E = Y_d(t) - Y(t)$. Let

$$\begin{aligned}\mathbf{v} &= \int_0^T \frac{\tau^\rho}{\rho!} \mathbf{w}^T(\tau) d\tau \\ &= \left[\frac{T^{\rho+1}}{(\rho+1)\rho!}, \frac{T^{\rho+2}}{(\rho+2)\rho!1!}, \dots, \frac{T^{2\rho+1}}{(2\rho+1)(\rho!)^2} \right],\end{aligned}$$

and

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

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

$$\Psi(t) = \mathbf{u}^T(t) \Theta \mathbf{u}(t) + \mathbf{p}^T \mathbf{u}(t), \quad (4.6)$$

where $\Theta = \kappa(L_g L_f^{\rho-1} h(\mathbf{x}(t)))^T Q L_g L_f^{\rho-1} h(\mathbf{x}(t))$ and $\mathbf{p} = -2(L_g L_f^{\rho-1} h(\mathbf{x}(t)))^T Q^T E \mathbf{v}^T$. Given that Θ is positive definite, performance index Ψ shown in (4.6) is convex and the optimal solution can thus be obtained by solving for $\mathbf{u}(t)$ from $\partial\Psi(t)/\partial\mathbf{u} = 0$, which gives the following nominal near-optimal controller for system (4.1) [38]:

$$\mathbf{u}(t) = (L_g L_f^{\rho-1} h(\mathbf{x}(t)))^{-1} \frac{1}{\kappa} (Y_d(t) - Y(t)) \mathbf{v}^T, \quad (4.7)$$

which can only be used when all the parameters of system (4.1) are known.

The value of T that can be set when the near-optimal controller is used depends on the controlled system. For example, if the system works on a low frequency, i.e., the value of the term of interest needs to change slowly, the value of T can be large. On the other hand, if the system works on a high frequency, i.e., the value of the term of interest needs to change quickly, the value of T should be small. This is natural as the Taylor expansion is adopted to predict the behavior of the system output.

4.4 Adaptive Near-Optimal Design

In this section, we consider the case that the parameters of nonlinear system (4.1) are unknown.

Based on Definitions 4.1 and 4.2, for nonlinear system (4.1) with relative degree ρ , one has

$$\mathbf{y}^{[\rho]}(t) = L_f^\rho h(\mathbf{x}(t)) + L_g L_f^{\rho-1} h(\mathbf{x}(t)) \mathbf{u}(t). \quad (4.8)$$

It is assumed that (4.8) can be parameterized as follows:

$$\mathbf{y}^{[\rho]}(t) = W_1 \phi_1(\mathbf{x}(t)) + W_2 \phi_2(\mathbf{x}(t)) \mathbf{u}(t) \quad (4.9)$$

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

The following auxiliary system is designed to reconstruct the dynamics of system (4.1):

$$\begin{cases} \dot{\hat{\mathbf{y}}}^{[\rho]}(t) = \hat{W}_1(t)\phi_1(\mathbf{x}(t)) + \hat{W}_2(t)\phi_2(\mathbf{x}(t))\mathbf{u}(t) - \dot{\mathbf{s}}(t) - \lambda\mathbf{s}(t) + \tilde{\mathbf{y}}^{[\rho]}(t), \\ \dot{\hat{W}}_1(t) = -K_1\mathbf{s}(t)\phi_1^T(\mathbf{x}(t)), \\ \dot{\hat{W}}_2(t) = -K_2\mathbf{s}(t)\mathbf{u}^T(t)\phi_2^T(\mathbf{x}(t)), \end{cases} \quad (4.10)$$

where $s(t)$ is defined as

$$\mathbf{s}(t) = \sum_{j=0}^{\rho-1} \alpha_j \tilde{\mathbf{y}}^{[j]}(t). \quad (4.11)$$

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 presented method, we only need to guarantee that $\lim_{t \rightarrow \infty} s(t) = 0$ instead of reaching it in finite time (please refer to the proof of Theorem 4.1 regarding $\lim_{t \rightarrow \infty} s(t) = 0$), which is enough for establishing the stability results concluded in this chapter. Let $\alpha_{\rho-1} = 1$. Then auxiliary system (4.10) becomes

$$\begin{cases} \dot{\hat{\mathbf{y}}}^{[\rho]}(t) = \hat{W}_1(t)\phi_1(\mathbf{x}(t)) + \hat{W}_2(t)\phi_2(\mathbf{x}(t))\mathbf{u}(t) - \sum_{j=0}^{\rho-2} \alpha_j \tilde{\mathbf{y}}^{[j+1]}(t) - \lambda\mathbf{s}(t), \\ \dot{\hat{W}}_1(t) = -K_1\mathbf{s}(t)\phi_1^T(\mathbf{x}(t)), \\ \dot{\hat{W}}_2(t) = -K_2\mathbf{s}(t)\mathbf{u}^T(t)\phi_2^T(\mathbf{x}(t)). \end{cases} \quad (4.12)$$

Via properly choosing parameters α_j for $j = 0, 1, 2, \dots, \rho - 2$, it can be guaranteed that, on the sliding surface $\mathbf{s}(t) = 0$, $\tilde{\mathbf{y}}(t) = 0$ is exponentially stable [55].

Theorem 4.1 *The dynamics of sliding-mode auxiliary system (4.12) asymptotically converges to that of nonlinear system (4.1) of fully unknown parameters and satisfies*

$$\mathbf{y}^{[\rho]}(t) = \hat{\mathbf{y}}^{[\rho]}(t) = \hat{W}_1(t)\phi_1(\mathbf{x}(t)) + \hat{W}_2(t)\phi_2(\mathbf{x}(t))\mathbf{u}(t), \quad (4.13)$$

when $t \rightarrow +\infty$.

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:

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

Subtracting Eq. (4.9) from the first equation of (4.12) yields

$$\dot{\mathbf{s}}(t) = -\lambda \mathbf{s}(t) + \tilde{W}_1(t)\phi_1(\mathbf{x}(t)) + \tilde{W}_2(t)\phi_2(\mathbf{x}(t))\mathbf{u}(t). \quad (4.14)$$

Let $\alpha_{\rho-1} = 1$ for $s(t)$ defined in (4.11). Based on Eqs. (4.14) and (4.12), the following result is obtained:

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

By the property of trace, we have $\text{tr}(\mathbf{s}^T(t)\tilde{W}_1(t)\phi_1(\mathbf{x}(t))) = \text{tr}(\phi_1(\mathbf{x}(t))\mathbf{s}^T(t)\tilde{W}_1(t)) = \text{tr}(\tilde{W}_1^T(t)\mathbf{s}(t)\phi_1^T(\mathbf{x}(t)))$. In this manner, we also have $\text{tr}(\mathbf{s}^T(t)\tilde{W}_2(t)\phi_2(\mathbf{x}(t))\mathbf{u}(t)) = \text{tr}(\tilde{W}_2^T(t)\mathbf{s}(t)\mathbf{u}^T(t)\phi_2^T(\mathbf{x}(t)))$, which yield $\dot{V}_1(t) = -\lambda \|\mathbf{s}(t)\|_2^2$. Evidently, $\dot{V}_1(t) \leq 0$. Let set $S = \{\tilde{\mathbf{y}}(t) \in \mathbb{R}^m \mid \dot{V}_1(t) = 0\}$. From $\dot{V}_1(t) = 0$, one has $S = \{\tilde{\mathbf{y}}(t) \in \mathbb{R}^m \mid \mathbf{s}(t) = 0\}$. Recall $\mathbf{s}(t) = \sum_{j=0}^{\rho-1} \alpha_j \tilde{\mathbf{y}}^{[j]}(t)$. Given that the coefficients α_j for $j = 0, 1, 2, \dots, \rho - 1$ with $\alpha_{\rho-1} = 1$ satisfy the Routh stability criterion [55], one has $\lim_{t \rightarrow +\infty} \tilde{\mathbf{y}}(t) = 0$. 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 [56], equilibrium point $\tilde{\mathbf{y}}(t) = 0$ is asymptotic stable. It follows that, when $t \rightarrow +\infty$,

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

In other words, the dynamics of sliding-mode auxiliary system (4.12) asymptotically converges to that of nonlinear system (4.1) of fully unknown parameters. \square

Based on Theorem 4.1, an adaptive near-optimal controller for system (4.1) can be designed based on system (4.13). Given that the outputs of system (4.1) and their time derivatives up to order $\rho - 1$ are measurable. Let $\hat{E}(t) = Y_d(t) - \hat{Y}(t)$ with $\hat{Y}(t) = [\mathbf{y}(t), \dot{\mathbf{y}}(t), \dots, \hat{W}_1(t)\phi_1(\mathbf{x}(t))]$. Similar to the design procedure in the nominal case, a near-optimal controller for system (4.13) can be designed by minimizing the following performance index:

$$\begin{aligned} \bar{J}(t) &= \int_0^T \left(\hat{E}(t)\mathbf{w}(\tau) - \frac{\tau^\rho}{\rho!} \hat{W}_2(t)\phi_2(\mathbf{x}(\tau))\mathbf{u}(\tau) \right)^T Q \\ &\quad \times \left(\hat{E}(t)\mathbf{w}(\tau) - \frac{\tau^\rho}{\rho!} \hat{W}_2(t)\phi_2(\mathbf{x}(\tau))\mathbf{u}(\tau) \right) d\tau. \end{aligned} \quad (4.15)$$

Evidently,

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

where $E = Y_d(t) - Y(t)$. Recall that

$$\begin{aligned}\mathbf{v} &= \int_0^T \frac{\tau^\rho}{\rho!} \mathbf{w}^T(\tau) d\tau \\ &= \left[\frac{T^{\rho+1}}{(\rho+1)\rho!}, \frac{T^{\rho+2}}{(\rho+2)\rho!1!}, \dots, \frac{T^{2\rho+1}}{(2\rho+1)(\rho!)^2} \right],\end{aligned}$$

and

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

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

$$\Psi(t) = \mathbf{u}^T(t) \Theta \mathbf{u}(t) + \mathbf{p}^T \mathbf{u}(t), \quad (4.16)$$

where $\Theta = \kappa (\hat{W}_2(t) \phi_2(\mathbf{x}(t)))^T Q \hat{W}_2(t) \phi_2(\mathbf{x}(t))$ and $\mathbf{p} = -2(\hat{W}_2(t) \phi_2(\mathbf{x}(t)))^T Q^T E \mathbf{v}^T$. Given that Θ is positive definite, performance index Ψ shown in (4.16) 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 controller is presented:

$$\mathbf{u}(t) = \frac{1}{\kappa} (\hat{W}_2(t) \phi_2(\mathbf{x}(t)))^{-1} (Y_d(t) - \hat{Y}(t)) \mathbf{v}^T, \quad (4.17)$$

which is an adaptive near-optimal controller for system (4.1) with fully unknown parameters. The block diagram of nonlinear system (4.1) with fully unknown parameters synthesized by adaptive near-optimal controller (4.17) and sliding-mode auxiliary system (4.12) is shown in Fig. 4.1.

Remark 4.1 For adaptive near-optimal controller (4.17), when $\hat{W}_2(t) \phi_2(\mathbf{x}(t))$ is singular, a regulation term can be used. Specifically, one may use $v I_l + \hat{W}_2(t) \phi_2(\mathbf{x}(t))$ to replace $\hat{W}_2(t) \phi_2(\mathbf{x}(t))$, where $v > 0 \in \mathbb{R}$ is a small parameter, e.g., 10^{-6} , with I_l being an identity matrix of suitable dimension. Given that v is small enough, the control performance is still satisfactory due to the bounded-input bounded-output property [55].

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

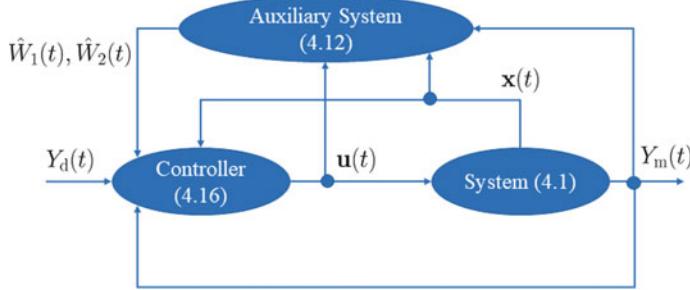
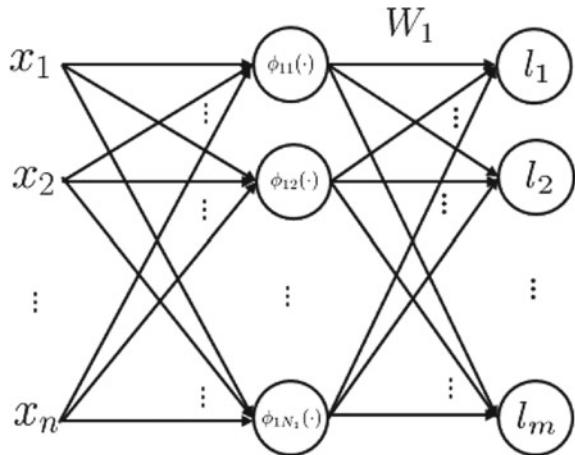


Fig. 4.1 Block diagram of nonlinear system (4.1) of fully unknown parameters synthesized by adaptive near-optimal controller (4.17) and sliding-mode auxiliary system (4.12) with the desired output matrix being $Y_d(t) = [y_d(t), \dot{y}_d(t), \dots, y_d^{[\rho]}(t)]$ and the measured output matrix being $Y_m(t) = [y(t), \dot{y}(t), \dots, y^{[\rho-1]}(t)]$

Fig. 4.2 Architecture of the sub neural network associated with W_1 with l_i denoting the i th element of $L_f^\rho h(\mathbf{x})$



Remark 4.2 System (4.9) 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. 4.2. The auxiliary system (4.12) 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 (4.9). Then, the adaptive controller is designed based on $\hat{W}_1(t)$ and $\hat{W}_2(t)$, which is expected to drive the system to the desired state.

Theorem 4.2 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 (4.1) of fully unknown parameters and adaptive near-optimal controller (4.17) is asymptotically stable.

Proof Substituting adaptive near-optimal controller (4.17) into Eq.(4.9) yields $\mathbf{y}^{[\rho]}(t) = W_1\phi_1(\mathbf{x}(t)) + W_2\phi_2(\mathbf{x}(t))(\hat{W}_2(t)\phi_2(\mathbf{x}(t)))^{-1}(Y_d(t) - \hat{Y}(t))\mathbf{v}^T/\kappa$. 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

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

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 $\mathbf{e}^{[j]} = \mathbf{y}_d^{[j]}(t) - \mathbf{y}^{[j]}(t)$. It follows that the closed-loop system consisting of nonlinear system (4.1) of fully unknown parameters and adaptive near-optimal controller (4.17) is

$$\kappa \sum_{j=0}^{\rho} \frac{T^{\rho+1+j}}{(\rho+1+j)\rho!j!} \mathbf{e}^{[j]}(t) = \delta(t). \quad (4.18)$$

By Routh stability criterion [55], it can be readily proved that when $\delta(t) = 0$ and $\rho \in \{1, 2, 3, 4\}$, system (4.18) is exponentially stable. Besides, according to Theorem 1, one has $\lim_{t \rightarrow +\infty} \delta(t) = \lim_{t \rightarrow +\infty} (\tilde{W}_1(t)\phi_1(\mathbf{x}(t)) + \tilde{W}_2(t)\phi_2(\mathbf{x}(t))\mathbf{u}(t)) = 0$. According to bounded-input bounded-output stability theory [55], equilibrium point $\mathbf{e}(t) = 0$ of closed-loop system (4.18) is asymptotically stable. \square

Theorem 4.3 *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 (4.1) with fully unknown parameters synthesized by adaptive near-optimal controller (4.17) is bounded and asymptotically converges to be optimal.*

Proof With Eq.(4.9) taken into account, based on Taylor expansion and in view of Eqs.(4.3) and (4.5) about $J(t)$ and $\hat{J}(t)$, one has

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

where $\Delta_1(t) = \tau^\rho / \rho! (\mathbf{y}_d^{[\rho]}(t + \varkappa\tau) - \mathbf{y}^{[\rho]}(t + \varkappa\tau) - (\mathbf{y}_d^{[\rho]}(t) - \mathbf{y}^{[\rho]}(t)))$ with $\varkappa \in (0, 1)$ and $\Delta_2(t) = \tau^\rho / \rho! (\tilde{W}_1(t)\phi_1(\mathbf{x}(t)) + \tilde{W}_2(t)\phi_2(\mathbf{x}(t))\mathbf{u}(t)) / \rho!$. Recall the definition of performance index $J(t)$ in (4.15). Based on triangle inequality, one further has

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

According to Theorem 4.1,

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

It follows that $\lim_{t \rightarrow +\infty} \|\Delta_2(t)\|_2^2 \|Q\|_2 = 0$. Besides, according to Theorem 4.2, when $\rho \in \{1, 2, 3, 4\}$ and $\hat{W}_2(t)\phi_2(\mathbf{x}(t))$ is invertible, nonlinear system (4.1) of fully unknown parameters synthesized by adaptive near-optimal controller (4.17) satisfies $\lim_{t \rightarrow +\infty} (\mathbf{y}^{[\rho]}(t) - \mathbf{y}_d^{[\rho]}(t)) = 0$. It follows that $\lim_{t \rightarrow +\infty} \sup_{0 < \varkappa < 1} \|\Delta_1(t)\|_2^2 \|Q\|_2 = 0$. Note that $J(t) \geq 0$. In addition, from the design procedure of near-optimal controller (4.17), it is known that controller (4.17) is optimal in terms of convex quadratic performance index $\bar{J}(t)$ defined in Eq. (4.15), which guarantees $\bar{J}(t) = 0$. Then, by the pinching theorem, $\lim_{t \rightarrow \infty} J(t) = 0$. \square

Remark 4.3 The performances of the presented adaptive near-optimal controller (4.17) is theoretically guaranteed for nonlinear system (4.1) of fully unknown parameters when relative degree $\rho \in \{1, 2, 3, 4\}$. It is worth pointing out that relative degree ρ of many mechanical systems is lower than 4 [49].

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

Remark 4.4 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 presented one does not need to solve it. Compared with the backstepping methods, e.g., [3], the presented method does not require tedious choices of virtual controllers. Compared with the adaptive sliding mode control method [59], the presented method inherently does not introduce the chattering phenomenon. Compared with the fuzzy neural network-based approach [60], the computational burden of the presented method is even more simpler owing to the lack of many Gaussian functions to be calculated.

4.5 Illustrative Examples

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

4.5.1 Example 1

Consider the following nonlinear system:

$$\begin{cases} \dot{x}_1(t) = p_1 x_2(t), \\ \dot{x}_2(t) = p_2 x_1(t)x_2(t) + p_3(\sin(x_2(t))) + 1.1u(t), \\ y(t) = x_1(t), \end{cases} \quad (4.19)$$

The system parameters are $p_1 = 2$, $p_2 = 4$ and $p_3 = 7$. Relative degree ρ of system (4.19) 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 (4.3) are set as $T = 0.6$ and $Q = 2$. The desired output is given as $y_d(t) = 0.5 \cos(0.5t) + 3.5$. With $\phi_1(\mathbf{x}) = x_1 x_2$, $\phi_2(\mathbf{x}) = \sin(x_2) + 1.1$, $\lambda = 3$, $\alpha_0 = 2$, $\alpha_1 = 1$, $K_1 = 4$, $K_2 = 9$, $x_1(0) = x_2(0) = \hat{y}(0) = \dot{\hat{y}}(0) = 0$, and $\hat{W}_1(0)$ and $\hat{W}_2(0)$ randomly generated at interval $(0, 10)$, the simulation results of nonlinear system (4.19) synthesized by adaptive near-optimal controller (4.17) are shown in Figs. 4.3, 4.4, and 4.5. From Fig. 4.3, it is observed that performance index $J_n(t)$ rapidly converges to near zero (i.e., optimal), and the output $y(t)$ of nonlinear system (4.19) quickly tracks the desired output $y_d(t)$ with tracking error $e(t) = y_d(t) - y(t)$ converging to zero. This validates Theorems 4.2 and 4.3. Besides, a comparison between the time histories of $\dot{y}(t)$ of nonlinear system (4.19) and $\dot{\hat{y}}(t)$ of sliding-mode auxiliary system (4.12) during the control process is shown in Fig. 4.4c, which validates Theorem 4.1. The evolutions of $\hat{W}_1(t)$ and $\hat{W}_2(t)$ are shown in Fig. 4.5. The above results validate the efficacy of the presented adaptive near-optimal control and the theoretical results.

The performance of adaptive near-optimal controller (4.17) for nonlinear system (4.19) under parameter uncertainty is compared with that of nominal near-optimal controller (4.7) adopted from [38]. The values of $\hat{W}_1(0)$ and $\hat{W}_2(0)$ for auxiliary system (4.12) are set as $\hat{W}_1(0) = \hat{W}_2(0) = 21$. The values of W_1 and W_2 in nominal near-optimal controller (4.7) are set as different values for comparison. The initial

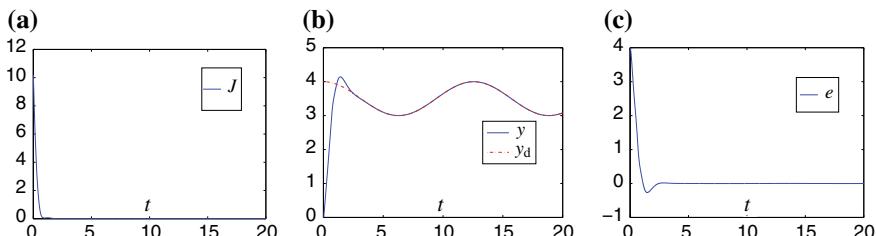


Fig. 4.3 Control performance of uncertain nonlinear system (4.19) synthesized by adaptive near-optimal controller (4.17) and sliding-mode auxiliary system (4.12) with $y_d(t) = 0.5 \cos(0.5t) + 3.5$, $T = 0.6$, and $Q = 2$. **a** Time history of performance index $J_n(t)$ defined in Eq. (4.3). **b** Time histories of output $y(t)$ and desired output $y_d(t)$. **c** Time history of tracking error $e(t) = y_d(t) - y(t)$

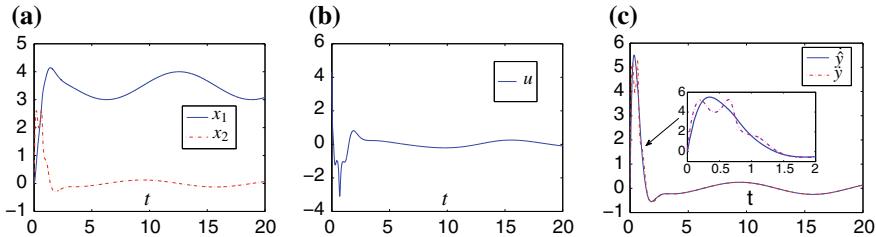


Fig. 4.4 Time histories of state variables $x_1(t)$ and $x_2(t)$, control input $u(t)$, $\dot{\hat{y}}(t)$ of sliding-mode auxiliary system (4.12), and $\hat{y}(t)$ of nonlinear system (4.19) during the control process. **a** Time histories of state variables $x_1(t)$ and $x_2(t)$. **b** Time history of control input $u(t)$. **c** Time histories of $\hat{y}(t)$ of sliding-mode auxiliary system (4.12) and $\dot{\hat{y}}(t)$ of nonlinear system (4.19)

Fig. 4.5 Time histories of parameters $\hat{W}_1(t)$ and $\hat{W}_2(t)$ of sliding-mode auxiliary system (4.12) during the control process of nonlinear system (4.19) by adaptive near-optimal controller (4.17). **a** Time history of $\hat{W}_1(t)$. **b** Time history of $\hat{W}_2(t)$

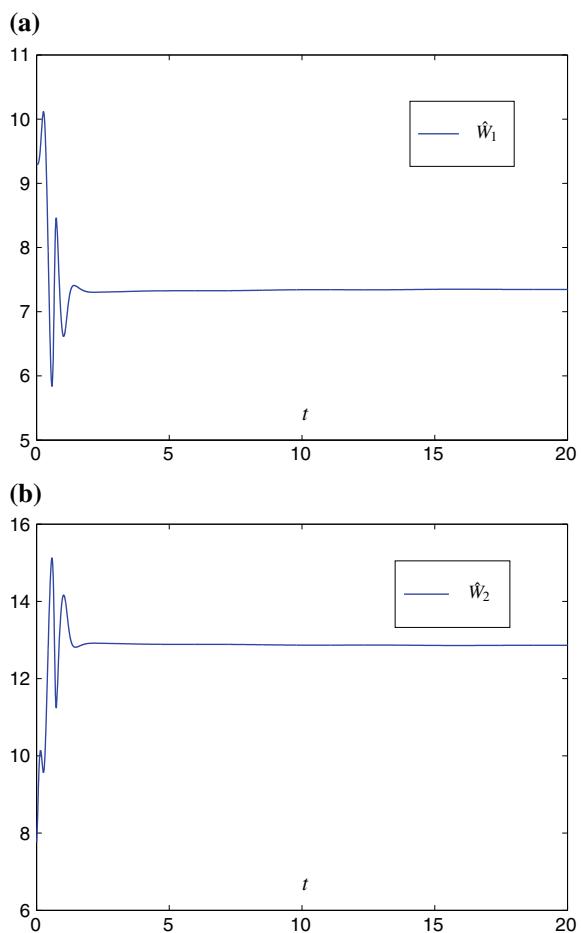
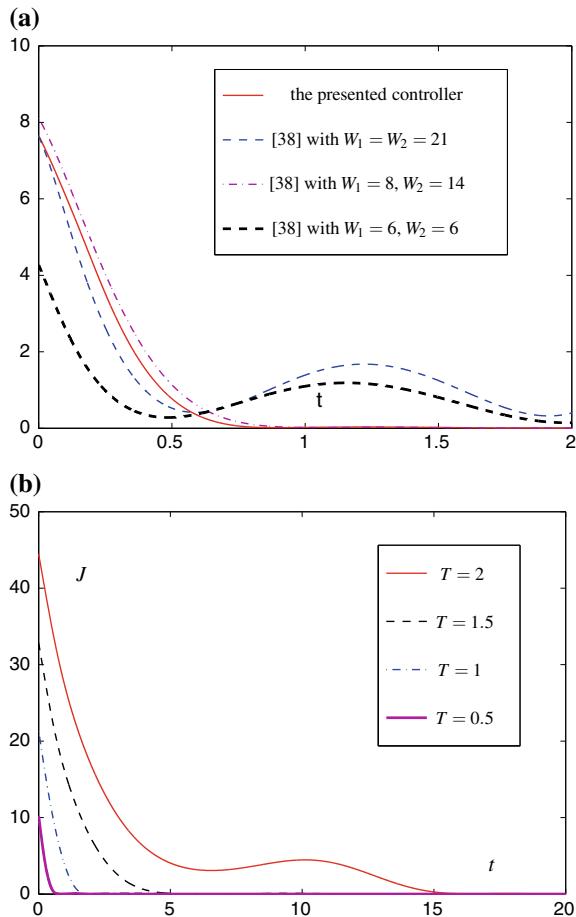


Fig. 4.6 Time histories of performance index J_n . **a** Time histories of performance index J_n (4.3) of nonlinear system (4.19) synthesized by the presented adaptive near-optimal controller (4.17) and by the existing nominal near-optimal controller (4.7) [38], where the actual system parameter values are $W_1 = 8$ and $W_2 = 14$. **b** Time histories of $J_n(t)$ of system (4.19) synthesized by adaptive near-optimal controller (4.17) under different values of T



states of nonlinear system (4.19) 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 (4.12). Other parameter values are set the same as the previous ones. Under these setups, time histories of performance index J_n (4.3) of nonlinear system (4.19) synthesized by the two controllers, respectively, are shown in Fig. 4.6a. From this figure, it can be observed that adaptive near-optimal controller (4.17) is superior to nominal near-optimal controller (4.7) 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 $J_n(t)$ (4.3) of system (4.19) synthesized by adaptive near-optimal controller (4.17) under different values of T are shown in Fig. 4.6b. This figure shows that under different values of T , performance index $J_n(t)$ of system (4.19) is optimized by the present adaptive near-optimal controller (4.17). 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.

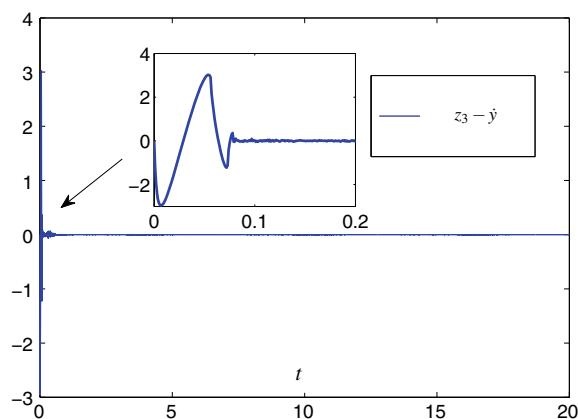
4.5.2 Example 2

A major difference of the presented 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 [52–54] can be used to construct the required derivatives for the output measurement. For example, to obtain the first-order derivative of a measurable function $w(t) \in \mathbb{R}$, the following first-order robust exact tracking differentiator [53] can be used:

$$\begin{cases} \dot{z}_1(t) = z_2(t) - \sqrt{C}|z_1(t) - w(t)|^{1/2}\text{sign}(z_1(t) - w(t)), \\ \dot{z}_2(t) = -1.1C\text{sign}(z_1(t) - w(t)), \\ z_3(t) = z_2(t) - \sqrt{C}|z_1(t) - w(t)|^{1/2}\text{sign}(z_1(t) - w(t)), \end{cases} \quad (4.20)$$

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 [54]. To see the performance of the presented adaptive near-optimal controller under the situation that the time derivatives are not directly available, simulations are also conducted based on Example 1. Specifically, the output of system (4.19) is used as the input of tracking differentiator (4.20), i.e., we let $w(t) = y(t)$. Besides, the output of tracking differentiator (4.20), i.e., z_3 is used in adaptive controller (4.17) 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 [54], 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 Figs. 4.7 and 4.8. These results substantiate the efficacy of the presented adaptive controller in the case that output derivatives are not directly available.

Fig. 4.7 Time history of $z_3(t) - \dot{y}$ where $z_3(t)$ is the output of tracking differentiator (4.20) and \dot{y} is the output derivative of system (4.19) during the control process



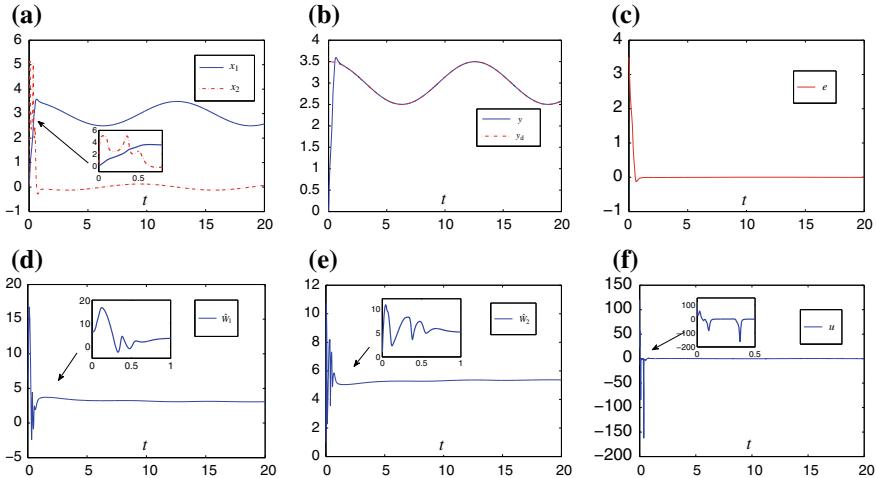


Fig. 4.8 Time histories of variables during the control process of system (4.19) via adaptive near-optimal controller (4.17) aided with auxiliary system (4.12) and tracking differentiator (4.20). **a** Time histories of state variables $x_1(t)$ and $x_2(t)$. **b** Time histories of system output $y(t)$ and desired output $y_d(t)$. **c** Time history of tracking error $e(t) = y_d(t) - y(t)$. **d** Time history of parameter $\hat{W}_1(t)$. **e** Time history of parameter $\hat{W}_2(t)$. **f** Time history of input $u(t)$

4.6 Application to van der Pol Oscillator

In this section, we further present the application of the presented control method to a van der Pol oscillator [57], 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:

$$\begin{cases} \dot{x}_1 = x_2, \\ \dot{x}_2 = -\varepsilon(x_1^2 - 1)x_2 - x_1 + u, \\ y = x_1, \end{cases} \quad (4.21)$$

where $\varepsilon > 0$ is a constant parameter of the system which depends on the corresponding circuit. In this application, only y is directly measurable, $\varepsilon = 0.3$, and the desired signal $y_d(t) = \sin(0.5t) + 0.2$. The parameters of performance index (4.3) are set as $T = 0.3$ and $Q = 2$. The presented method is used to address the problem without using the exact value of ε . As in Example 2, we use tracking differentiator (4.20) to obtain the derivative of y . The results are shown in Fig. 4.9 for the setup that $\phi_1(\mathbf{x}) = [-(x_1^2 - 1)x_2, -x_1]^T$, $\phi_2 = 1$, $W_1 = [\hat{\varepsilon}, 1]^T$, $W_2 = 1$, $\lambda = 3$, $\alpha_0 = 2$, $\alpha_1 = 1$, $K_1 = 7$, $x_1(0) = x_2(0) = \hat{y}(0) = \dot{\hat{y}}(0) = 0$, and $\hat{\varepsilon}$ is randomly generated at interval $(0, 1)$. These results further verify the efficacy and practical significance of the presented method.

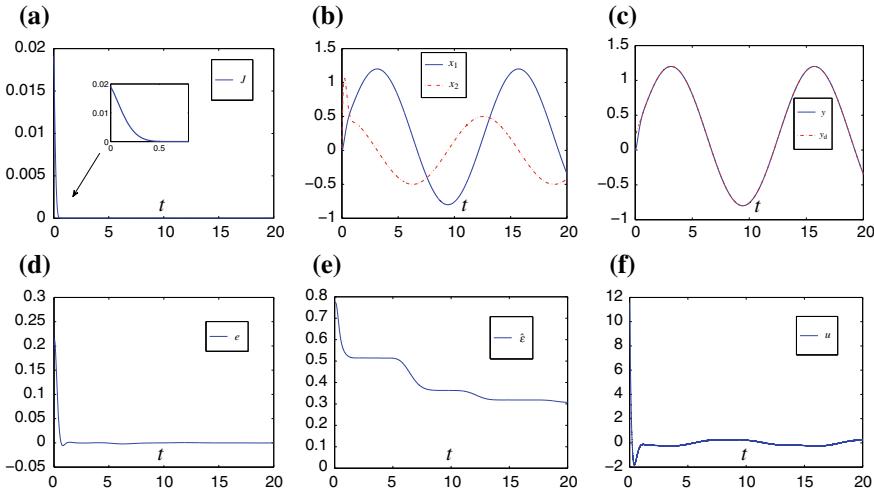


Fig. 4.9 Time histories of variables during the control process of van der Pol oscillator (4.21) via adaptive near-optimal controller (4.17) aided with auxiliary system (4.12) and tracking differentiator (4.20). **a** Time histories of performance index $J(t)$. **b** Time histories of state variables $x_1(t)$ and $x_2(t)$. **c** Time histories of system output $y(t)$ and desired output $y_d(t)$. **d** Time history of tracking error $e(t) = y_d(t) - y(t)$. **e** Time history of parameter $\hat{\epsilon}$. **f** Time history of input $u(t)$

4.7 Experimental Validation

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

The prototype of a motor control system shown in Fig. 4.10 is used in the experiment. The system power is provided by a 9-V 1-A AC/DC adaptor. The presented control method is implemented in the Arduino UNO board, and the control input computed by the presented method is converted into PWM signal which is fed into the 12 V motor drive by using an Arduino command *analogWrite(pin,value)*. The microcontroller of the Arduino UNO board used in the experiments is ATmega328P [64], which is a low-power CMOS 8-bit microcontroller and the clock speed is 16 MHz. In Arduino UNO, the *value* can be 0–255, which corresponds to 0–100% PWM. The conversion rule is thus as follows:

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

where u denotes the input generated by the presented 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-source Arduino software Arduino IDE.

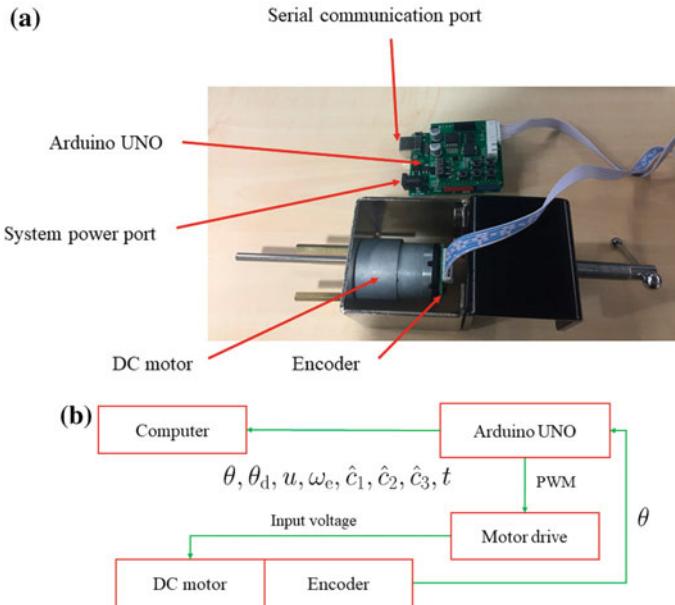


Fig. 4.10 Experimental platform for the DC motor control system. **a** A picture of the DC motor control system. **b** The overall system diagram in terms of signal transmissions

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 effects accurately, we use a term c_3 in (4.22) to represent the sum of the nonlinear terms about the dead-zone and friction effects [65], including other unmodeled disturbances. In the experiment, the motor system dynamics is modeled as follows:

$$\begin{cases} \dot{\theta} = \omega, \\ \dot{\omega} = -c_1\omega + c_2u + c_3, \end{cases} \quad (4.22)$$

where θ and ω denote 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 unmodeled terms such as frictions; u denotes the input voltage of the motor. The system model is based on the Newton's law. Multiplying both sides of the second sub-equation of (4.22) with the mass yields the result obtained via the Newton's law. Note that, for the presented method, the focus is not on identifying the parameter values of the system, but on realizing the adaptive near-optimal control. Evidently, the system has a well-defined relative degree $\rho = 2$. The model can be viewed as an extended version of the one in [58] by adding a disturbance term c_3 . In the experiment, the term ω , i.e., $\dot{\theta}$, is obtained via the tracking differentiator (4.20) in the manuscript with θ being the input. The obtained value of ω is denoted as ω_e . In the experiment, the motor angle θ is expected to track a desired trajectory $\theta_d = 2 \sin(4\pi t) + 1$ 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 (4.10) is set to $\lambda = 0.1$, $K_1 = \text{diag}([0.2, 0.4]^T)$, and $K_2 = 0.1$. As the auxiliary system is directly constructed based on ω_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 $u_s = \text{sat}(u)$ where u is calculated by Eq.(15) in the manuscript and the saturation function $\text{sat}(\cdot)$ is defined as follows:

$$\text{sat}(u) = \begin{cases} 7, & \text{if } u > 7; \\ u, & \text{if } -7 < u < 7; \\ -7, & \text{if } u < -7. \end{cases} \quad (4.23)$$

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. 4.11. The maximum tracking error is $\max_{t \in [0, 8]}(|\theta(t) - \theta_d(t)|) = 1.57$ rad, which is the initial tracking error. The minimum tracking error is $\min_{t \in [0, 8]}(|\theta(t) - \theta_d(t)|) \approx 0$ rad. The average tracking error is 0.0108 rad. The standard deviation of the tracking error is 0.1963 rad. The measures are listed in Table 4.1. The results verify the efficacy of the presented 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 presented 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:

$$u_s(t) = \text{sat}\left(K_p e(t) + K_i \int_0^t e(\tau) d\tau + K_d \dot{e}(t)\right),$$

where $e(t) = \theta_d(t) - \theta(t)$. Under the same settings, with $K_p = 45$, $K_i = 0.01$, and $K_d = 7$, which are tuned via the trial-and-error method [61–63], the experimental results are shown in Fig. 4.12. 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. 4.12c, it is observed that the performance index is much better when the presented adaptive near-optimal controller is used compared with the case with the PID controller.

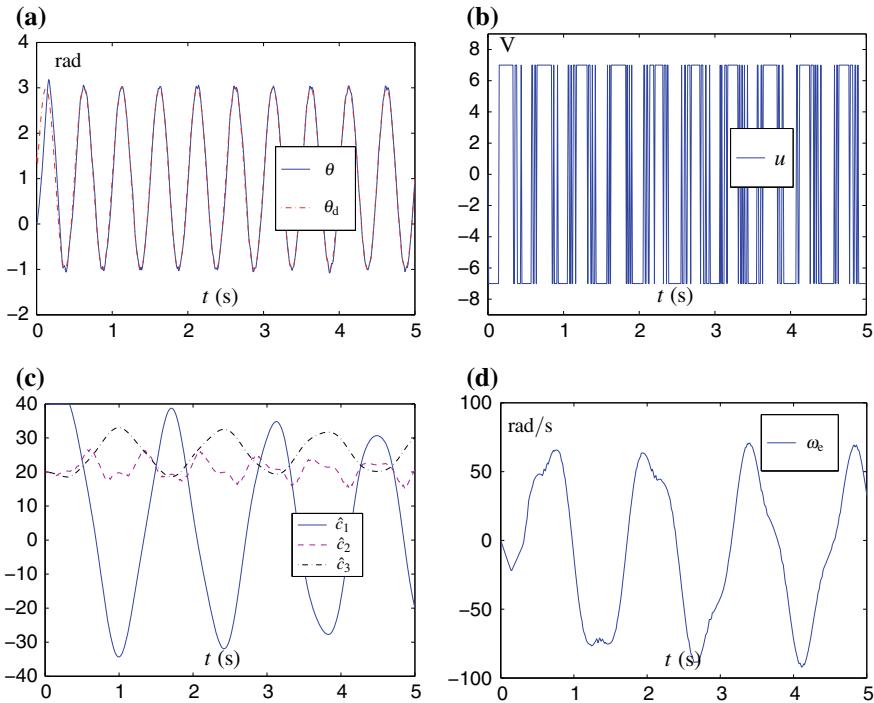


Fig. 4.11 Data profiles during the experiment of the tracking control of the DC motor by the presented adaptive near-optimal control method with saturation (4.23) based on the model (4.22). **a** Time history of motor angle $\theta(t)$ and desired angle $\theta_d(t)$. **b** Time histories of input u generated by the presented controller (4.17) with saturation specified in (4.23). **c** Time histories of parameters estimated by the auxiliary system. **d** Time history of ω estimated via the tracking differentiator (4.20) with the input being the joint angles measured by the encoder, which is denoted by ω_e

Table 4.1 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

We also conduct experiments based on the motor system for the case with load disturbance. The load disturbance is generated using a pen to introduce more frictions to the motor shaft (as shown in Fig. 4.13). With the same parameter settings stated above, the experimental results when the presented control method is used are shown in Fig. 4.14. As seen from the figure, the tracking performance is still satisfactory.

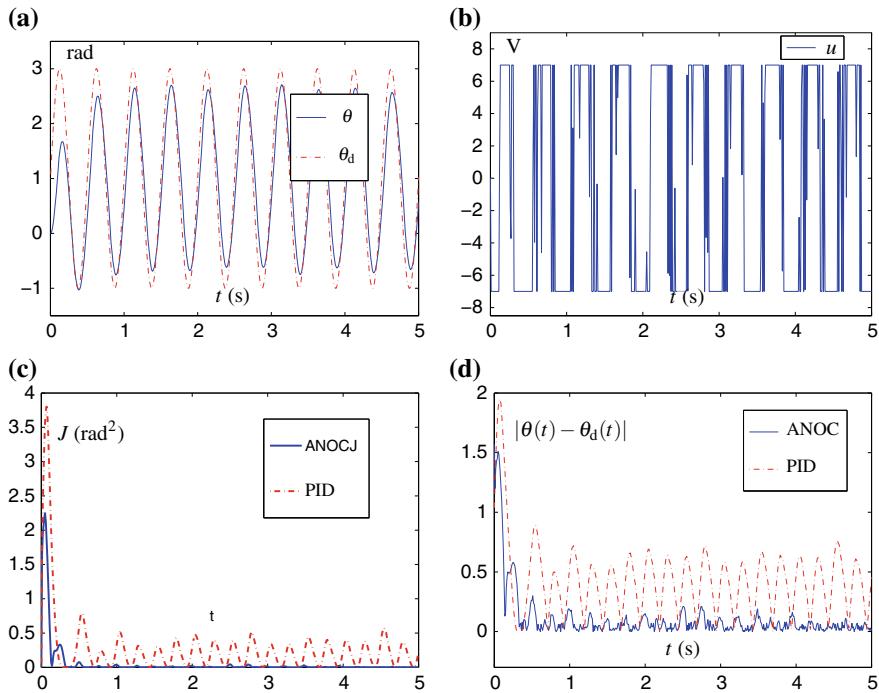
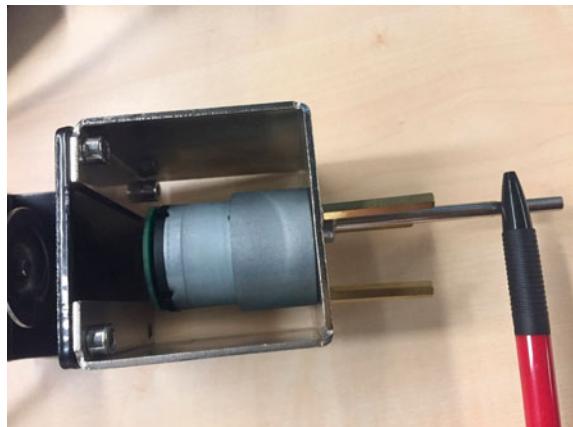


Fig. 4.12 Data profiles during the experiment of the tracking control of the DC motor by the PID controller and its performance comparison with the presented adaptive near-optimal controller (ANOC) with saturation (4.23). **a** Time history of motor angle $\theta(t)$ and desired angle $\theta_d(t)$. **b** Time histories of input u generated by the PID controller with saturation. **c** Comparison of performance index J defined in (4.3) when different controllers are used. **d** Comparison of tracking error $|\theta(t) - \theta_d(t)|$

Fig. 4.13 The experiment setup regarding the generation of load disturbance



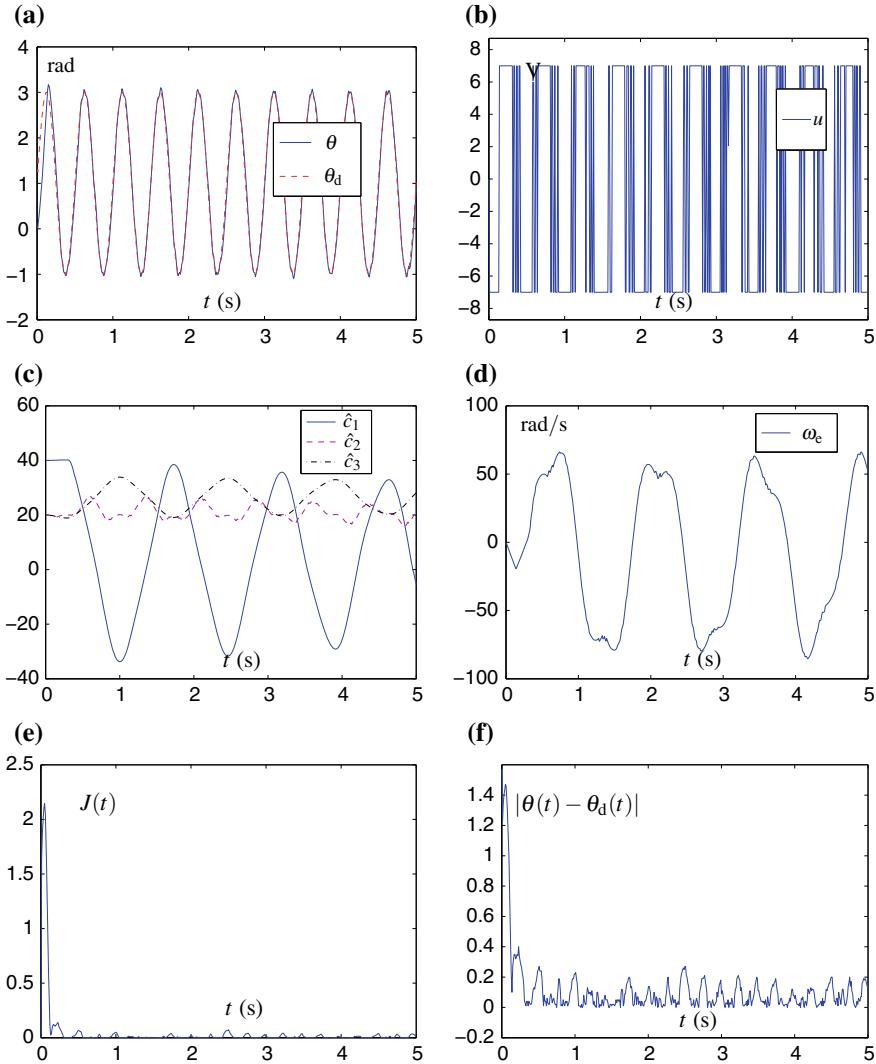


Fig. 4.14 Data profiles during the experiment of the tracking control of the DC motor by the presented adaptive near-optimal control method with saturation (4.23) based on the model (4.22) with artificially added load disturbance as shown in Fig. 4.13. **a** Time history of motor angle $\theta(t)$ and desired angle $\theta_d(t)$. **b** Time histories of input u generated by the presented controller (4.17) with saturation specified in (4.23). **c** Time histories of parameters estimated by the auxiliary system. **d** Time history of ω estimated via the the tracking differentiator. **e** Time history of performance index $J(t)$. **f** Time history of tracking error $|\theta(t) - \theta_d(t)|$

Table 4.2 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

The experimental data show that the maximum tracking error is $\max_{t \in [0,8]}(|\theta(t) - \theta_d(t)|) = 1.57$ rad, which is the initial tracking error. The minimum tracking error is $\min_{t \in [0,8]}(|\theta(t) - \theta_d(t)|) \approx 0$ 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 4.2. 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 presented method.

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

Remark 4.5 First, the performance of the presented approach is theoretically guaranteed. Second, we use vast simulation and experimental results to verify the theoretical results and the performance of the presented method. Meanwhile, PID control is widely used in DC motor control. The comparison with the PID controller shows the superiority of the presented 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.

4.8 Questions and Answers

In this section, some discussions about the content in this chapter are presented in the form of questions and answers.

Question 4.1 “I think it lacks the realization. A nice simulation does not imply a better realization.”

Answer:

We have performed experimental validation of the proposed method based on a DC motor. 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 an Arduino command *analogwrite(pin,value)*. In Arduino UNO, the *value* can be 0–255, which corresponds to 0–100% PWM. The conversion rule is thus as 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. All the computations are conducted by the Arduino UNO.

In the experiment, the motor system dynamics is modeled as follows:

$$\begin{cases} \dot{\theta} = \omega, \\ \dot{\omega} = -c_1\omega + c_2u + c_3, \end{cases}$$

where θ and ω 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 unmodeled terms such as frictions; u denotes the input voltage of the motor. Evidently, the system has a well-defined relative degree $\rho = 2$. The model can be viewed as an extended version of the one in [R1] by adding a disturbance term c_3 . In the experiment, the term ω , i.e., $\dot{\theta}$, is obtained via the tracking differentiator shown in (17) in the manuscript with θ being the input. The obtained value of ω is denoted as ω_e . In the experiment, the motor angle θ is expected to track a desired trajectory $\theta_d = 2 \sin(4\pi t) + 1$ 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 (12) is set to $\lambda = 0.1$, $K_1 = \text{diag}([0.2, 0.4]^T)$, and $K_2 = 0.1$. As the auxiliary system is directly constructed based on ω_e in the experiment, $\alpha_i = 0$. As the control method is implemented on the Arduino UNO, the auxiliary system and the tracking differentiator are 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 $u_s = \text{sat}(u)$ where u is calculated by Eq. (15) in the manuscript and the saturation function $\text{sat}(\cdot)$ is defined as follows:

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

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

$$u_s(t) = \text{sat} \left(K_p e(t) + K_i \int_0^t e(\tau) d\tau + K_d \dot{e}(t) \right),$$

where $e(t) = \theta_d(t) - \theta(t)$. Under the same settings, we set $K_p = 45$, $K_i = 0.01$, and $K_d = 7$. It is observed that the performance index is much better when the proposed adaptive near-optimal controller is used compared with the case with the PID controller. These experimental results validate the efficacy, superiority, realizability, and online control capability of the proposed method.

Question 4.2 “*Sliding mode control is well known for being of limited use in noisy (especially unmatched noise) scenarios. How does your method perform when $y = x_1 + \text{gaussian } (0, 1)$ noise? Again, the practicality of the method is not clear, and warrants more experimentation.*”

Answer: In the experiment about the DC motor whose control commands are sent by the Arduino, the obtained output measurement through the encoder is evidently noise-polluted. Noise always exists in sensor outputs especially for low-cost sensors. Besides, the encoder only have a measurement accuracy of $360/1040 \approx 2.94^\circ$. In addition, the actual input to the motor is implemented via the PWM of the Arduino UNO and the 12 V motor drive, which thus only has an accuracy of $12/255 = 0.05$ V. However, as the experimental results suggested, our method works, and the performance is satisfactory. In addition, the term that cannot be directly measured, i.e., the motor angle velocity, is obtained via the tracking differentiator with the joint angle being its input. Thus, the proposed method is feasible in practice. The feasibility is also indicated by the model we used, where all the parameters are unknown, which means that we only need a relatively small amount of knowledge of the controlled system.

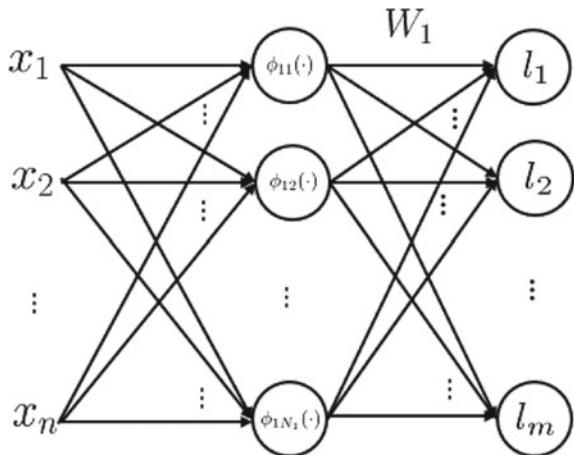
Question 4.3 “*I like the current example, but at least one example that this works on something practical (for example, noisy + larger state dimension) would be appreciated.*

Answer: It should be emphasized that the motor and encoder used in the physical experiment are very low-cost, which means that the dynamic performance of the motor is intrinsically poor and the measurement accuracy is also poor. This scenario is very noisy.

Question 4.4 “*In the arguments presented, you use Taylor expansions so τ should be very small; I’m curious if that suggests that T has to be quite small? That would, in turn, imply that the performance is computed over a small-time range.*”

Answer: The value of T that can be set when the near-optimal controllers are used depends on the controlled system. For example, if the system works on a low frequency, i.e., the value of the term of interest needs to change slowly, the value of T can be large. On the other hand, if the system works on a high frequency, i.e., the value of the term of interest needs to change quickly, the value of T should be small. This is natural as the Taylor expansion is adopted to predict the behavior of the system output.

Fig. 4.15 Architecture of the sub neural network associated with W_1 with l_i denoting the i th element of $L_f^\rho h(\mathbf{x})$



Question 4.5 “The authors require to present more details about the design methodology, particularly about ANN structure.”

Answer: System (9) 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, where 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. 4.15. The auxiliary system 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 the control system. Then, the adaptive 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.

Question 4.6 “The authors necessitate clarifying the following: The effectiveness of the proposed approach is not supported by a comparison with other approaches. Since the proposed system is rather complex the authors should show that it permits to gain in terms of performance with respect to a standard approach.”

Answer: First, the performance of the proposed approach is theoretically guaranteed. Second, 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.

Question 4.7 “Authors should include a comparative analysis of the proposed methodology against existing ones.”

Answer: 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 backstepping methods, the proposed method does not need tedious choices of virtual controllers. Compared with the adaptive sliding mode control method, the proposed method inherently does not introduce the chattering phenomenon. Compared with the fuzzy neural network-based approach, the computational burden of the proposed method is even more simpler owing to the lack of many Gaussian functions to be calculated.

Question 4.8 “*If in fact sliding mode controllers can reach sliding surface in finite time, ANN cannot achieve a weight estimation error equal to zero, then it is not possible to ensure an asymptotic convergence for the whole system.*”

Answer: First, we would like to clarify that the proposed controller is not a sliding mode controller. As indicated in the Introduction section, the idea in sliding mode control is only used to design an auxiliary system for the sake of system dynamics reconstruction. Second, as shown in Theorem 1, the dynamics of the proposed sliding-mode auxiliary system converges to that of the controlled nonlinear system in an asymptotic manner but not in finite time. For details, please refer to the theoretical analysis.

Question 4.9 “*The authors require to present clear details about the design procedure for the adaptive near-optimal control.*”

Answer: The authors sincerely thank the reviewer for pointing out the comment. From Eq.(14) in the original version, we have

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

where $E = Y_d(t) - Y(t)$. Recall that

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

and

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

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

$$\Psi(t) = \mathbf{u}^T(t) \Theta \mathbf{u}(t) + \mathbf{p}^T \mathbf{u}(t), \quad (4.24)$$

where $\Theta = \kappa (\hat{W}_2(t) \phi_2(\mathbf{x}(t)))^T Q \hat{W}_2(t) \phi_2(\mathbf{x}(t))$ and $\mathbf{p} = -2 (\hat{W}_2(t) \phi_2(\mathbf{x}(t)))^T Q^T E \mathbf{v}^T$. Given that Θ is positive definite, performance index Ψ 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 controller is obtained:

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

which is the proposed adaptive near-optimal controller.

Question 4.10 “The authors must study the effect of parameter variations on the dynamic position response for the DC motor.”

Answer: In the experiments, we do not have any prior knowledge about the values of the parameters of the DC motor system. The proposed method does not need any prior knowledge about the values of the parameters of the control system.

Question 4.11 “The authors want to give the novelty and contribution of this work.”

Answer: The major novelties and contributions of this work are listed as follows: (1) The proposed adaptive near-optimal controller simultaneously guarantees asymptotic stability of the closed-loop system and asymptotic optimality of the performance index; (2) It is shown in this paper how an auxiliary system can be designed by the sliding mode concept to reconstruct the dynamics of the controlled nonlinear system with unknown parameters, which may inspire the development of other adaptive control techniques; (3) The proposed control scheme has a low computational cost, which can be implemented on low-cost microcontrollers and is validated via experiments.

Question 4.12 “It is written: ... with the sliding mode surface being $s(t) = 0$,... One of the basic conditions of the well-known sliding mode control is to guarantee the reachability condition can be satisfied, i.e., the sliding mode surface attained in a finite time. How does the presented approach realize the “being $s(t) = 0$ ”? This is an important problem should be seriously analyzed, which is closely related to the design and analysis of auxiliary system.”

Answer: The design of the auxiliary system is partially inspired by the sliding mode control. However, different from the traditional sliding mode control, in the proposed method, we only need to guarantee that $\lim_{t \rightarrow \infty} s(t) = 0$ instead of reaching it in finite time (please refer to the proof of Theorem 1 regarding $\lim_{t \rightarrow \infty} s(t) = 0$ in the manuscript), which is enough for establishing the concluded stability results.

Question 4.13 “The research object of this paper is nonlinear systems, while the experimental model written here is a second-order linear system. The contradictory should be tackled, or the feasibility of the experimental model to illustrate the method efficacy should be clarified.”

Answer: 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 effects accurately, we use the term c_3 to represent the sum of the nonlinear terms about the dead-zone and friction effects, including other unmodeled disturbances. As the experimental results indicated, the performance is satisfied, which validates the efficacy of the proposed method for nonlinear systems.

4.9 Summary

In this chapter, an adaptive near-optimal controller based on a sliding-mode auxiliary system has been designed and presented 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 presented adaptive near-optimal controller 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 der Pol oscillator have validated the efficacy of the presented adaptive near-optimal control. The realizability, performance, and superiority of the presented method have also been validated through physical experiments based on a DC motor. In practice, the performance of the presented method may be improved via using the predictive smooth variable structure filters [66, 67], which can be further investigated. In addition, the extension of the presented method to distributed control systems, such as [32], can also be investigated.

References

1. Liu, Y., Tong, S.: Barrier Lyapunov functions-based adaptive control for a class of nonlinear pure-feedback systems with full state constraints. *Automatica* **64**, 70–75 (2016). Feb
2. Lai, G., Chen, C.L.P., Zhang, Y.: Adaptive asymptotic tracking control of uncertain nonlinear system with input quantization. *Syst. Control Lett.* **96**, 23–29 (2016). Oct
3. Liu, Y., Gao, Y., Tong, S., Li, Y.: Fuzzy approximation-based adaptive backstepping optimal control for a class of nonlinear discrete-time systems with dead-zone. *IEEE Trans. Fuzzy Syst.* **24**(1), 16–28 (2016). Feb
4. Liu, Y., Tong, S.: Adaptive fuzzy control for a class of unknown nonlinear dynamical systems. *Fuzzy Set. Syst.* **263**(15), 49–70 (2015). Mar
5. Li, S., Chen, S., Liu, B., Li, Y., Liang, Y.: Decentralized kinematic control of a class of collaborative redundant manipulators via recurrent neural networks. *Neurocomputing* **91**, 1–10 (2012)
6. Li, S., Cui, H., Li, Y., Liu, B., Lou, Y.: Decentralized control of collaborative redundant manipulators with partial command coverage via locally connected recurrent neural networks. *Neural Comput. Appl.* **23**(3), 1051–1060 (2013)
7. Jin, L., Zhang, Y., Li, S., Zhang, Y.: Modified ZNN for time-varying quadratic programming with inherent tolerance to noises and its application to kinematic redundancy resolution of robot manipulators. *IEEE Trans. Ind. Electron.* **63**(11), 6978–6988 (2016)
8. Li, S., He, J., Li, Y., Rafique, M.U.: Distributed recurrent neural networks for cooperative control of manipulators: a game-theoretic perspective. *IEEE Trans. Neural Netw. Learn. Syst.* **28**(2), 415–426 (2017)
9. Jin, L., Li, S., La, H.M., Luo, X.: Manipulability optimization of redundant manipulators using dynamic neural networks. *IEEE Trans. Ind. Electron.* **64**(6), 4710–4720 (2017)
10. Li, Y., Li, S., Hannaford, B.: A novel recurrent neural network for improving redundant manipulator motion planning completeness. In: ICRA, pp. 2956–2961 (2018)
11. Zhang, Y., Li, S.: A neural controller for image-based visual servoing of manipulators with physical constraints. *IEEE Trans. Neural Netw. Learn. Syst.* **29**(11), 5419–5429 (2018)

12. Li, S., Zhou, M., Luo, X.: Modified primal-dual neural networks for motion control of redundant manipulators with dynamic rejection of harmonic noises. *IEEE Trans. Neural Netw. Learn. Syst.* **29**(10), 4791–4801 (2018)
13. Li, S., Wang, H., Rafique, M.U.: A novel recurrent neural network for manipulator control with improved noise tolerance. *IEEE Trans. Neural Netw. Learn. Syst.* **29**(5), 1908–1918 (2018)
14. Jin, L., Li, S., Luo, X., Li, Y., Qin, B.: Neural dynamics for cooperative control of redundant robot manipulators. *IEEE Trans. Ind. Inform.* **14**(9), 3812–3821 (2018)
15. Li, J., Zhang, Y., Li, S., Mao, M.: New discretization-formula-based zeroing dynamics for real-time tracking control of serial and parallel manipulators. *IEEE Trans. Ind. Inf.* **14**(8), 3416–3425 (2018)
16. Chen, D., Zhang, Y., Li, S.: Tracking control of robot manipulators with unknown models: a Jacobian-matrix-adaption method. *IEEE Trans. Ind. Inf.* **14**(7), 3044–3053 (2018)
17. Zhang, Y., Li, S., Gui, J., Luo, X.: Velocity-level control with compliance to acceleration-level constraints: a novel scheme for manipulator redundancy resolution. *IEEE Trans. Ind. Inf.* **14**(3), 921–930 (2018)
18. Xiao, L., Liao, B., Li, S., Zhang, Z., Ding, L., Jin, L.: Design and analysis of FTZNN applied to the real-time solution of a nonstationary Lyapunov equation and tracking control of a wheeled mobile manipulator. *IEEE Trans. Ind. Inf.* **14**(1), 98–105 (2018)
19. Zhang, Y., Chen, S., Li, S., Zhang, Z.: Adaptive projection neural network for kinematic control of redundant manipulators with unknown physical parameters. *IEEE Trans. Ind. Electron.* **65**(6), 4909–4920 (2018)
20. Zhang, Z., Lin, Y., Li, S., Li, Y., Yu, Z., Luo, Y.: Tricriteria optimization-coordination motion of dual-redundant-robot manipulators for complex path planning. *IEEE Trans. Control Syst. Tech.* **26**(4), 1345–1357 (2018)
21. Jin, L., Li, S., Hu, B., Yi, C.: Dynamic neural networks aided distributed cooperative control of manipulators capable of different performance indices. *Neurocomputing* **291**, 50–58 (2018)
22. Jin, L., Li, S., Yu, J., He, J.: Robot manipulator control using neural networks: a survey. *Neurocomputing* **285**, 23–34 (2018)
23. Chen, D., Zhang, Y., Li, S.: Zeroing neural-dynamics approach and its robust and rapid solution for parallel robot manipulators against superposition of multiple disturbances. *Neurocomputing* **275**, 845–858 (2018)
24. Li, S., Shao, Z., Guan, Y.: A dynamic neural network approach for efficient control of manipulators. *IEEE Trans. Syst. Man Cybern. Syst.* **49**(5), 932–941 (2019)
25. Zhang, Y., Li, S., Zhou, X.: Recurrent-neural-network-based velocity-level redundancy resolution for manipulators subject to a joint acceleration limit. *IEEE Trans. Ind. Electron.* **66**(5), 3573–3582 (2019)
26. Zhang, Z., Chen, S., Li, S.: Compatible convex-nonconvex constrained QP-based dual neural networks for motion planning of redundant robot manipulators. *IEEE Trans. Control Syst. Tech.* **27**(3), 1250–1258 (2019)
27. Xu, Z., Li, S., Zhou, X., Yan, W., Cheng, T., Huang, D.: Dynamic neural networks based kinematic control for redundant manipulators with model uncertainties. *Neurocomputing* **329**, 255–266 (2019)
28. Li, S., Zhang, Y., Jin, L.: Kinematic control of redundant manipulators using neural networks. *IEEE Trans. Neural Netw. Learn. Syst.* (in press)
29. Mohammed, A.M., Li, S.: Dynamic neural networks for kinematic redundancy resolution of parallel stewart platforms. *IEEE Trans. Cybern.* **46**(7), 1538–1550 (2016). Jul
30. Jin, L., Zhang, Y.: G2-type SRMPC scheme for synchronous manipulation of two redundant robot Arms. *IEEE Trans. Cybern.* **45**(2), 153–164 (2015). Feb
31. Jin, L., Li, S., Xiao, L., Lu, R., Liao, B.: Cooperative motion generation in a distributed network of redundant robot manipulators with noises. *IEEE Trans. Syst. Man Cybern. Syst.* **48**(10), 1715–1724 (2018)
32. Jin, L., Li, S.: Distributed task allocation of multiple robots: a control perspective. *IEEE Trans. Syst. Man Cybern. Syst.* **48**(5), 693–701 (2018)

33. Vamvoudakis, K.G.: Q-learning for continuous-time linear systems: a model-free infinite horizon optimal control approach. *Syst. Control Lett.* **100**, 14–20 (2017). Feb
34. Lewis, F.L., Vrabie, D., Syrmos, V.L.: *Optimal Control*. Wiley, Hoboken (2012)
35. Mayne, D.Q., Michalska, H.: Receding horizon control of nonlinear systems. *IEEE Trans. Autom. Control* **35**(7), 814–824 (1990). Jul
36. Johansen, T.A.: Toward dependable embedded model predictive control. *IEEE Syst. J.* **11**(2), 1208–1219 (2017). Jun
37. Chakrabarty, A., Dinh, V., Corless, M.J., Rundell, A.E., Zack, S.H., Buzzard, G.T.: Support vector machine informed explicit nonlinear model predictive control using low-discrepancy sequences. *IEEE Trans. Autom. Control* **62**(1), 135–148 (2017). Jan
38. Chen, W.-H., Ballance, D.J., Gawthrop, P.J.: Optimal control of nonlinear systems: a predictive control approach. *Automatica* **39**(4), 633–641 (2003). Apr
39. Zhang, Y., Li, S.: Predictive suboptimal consensus of multiagent systems with nonlinear dynamics. *IEEE Trans. Syst. Man Cybern. Syst.* **47**(7), 1701–1711 (2017)
40. Zhang, Y., Li, S.: Time-scale expansion-based approximated optimal control for underactuated systems using projection neural networks. *IEEE Trans. Syst. Man, Cybern. Syst.* **48**(11), 1957–1967 (2018)
41. Yu, Z., Xiao, L., Li, H., Zhu, X., Huai, R.: Model parameter identification for lithium batteries using the coevolutionary particle swarm optimization method. *IEEE Trans. Ind. Electron.* **64**(7), 5690–5700 (2017). Jul
42. Lee, S.-H., Yoo, A., Lee, H.-J., Yoon, Y.-D., Han, B.-M.: Identification of induction motor parameters at standstill based on integral calculation. *IEEE Trans. Ind. Appl.* **53**(3), 2130–2139 (2017). May
43. Odhano, S.A., Bojoi, R., Armando, E., Homrich, G., Flores Filho, A.F., Popescu, M., Dorrell, D.G.: Identification of three-phase IPM machine parameters using torque tests. *IEEE Trans. Ind. Appl.* **53**(3), 1883–1891 (2017)
44. Pin, G., Magni, L., Parisini, T., Raimondo, D.M.: Robust receding - horizon control of nonlinear systems with state dependent uncertainties: an input-to-state stability approach. In: Proceedings of American Control Conference, pp. 1667–1672 (2008)
45. Zhao, X., Yang, H., Zong, G.: Adaptive neural hierarchical sliding mode control of nonstrict-feedback nonlinear systems and an application to electronic circuits. *IEEE Trans. Syst. Man Cybern. Syst.* **47**(7), 1394–1404 (2017)
46. Yin, S., Yang, H., Kaynak, O.: Sliding mode observer-based FTC for markovian jump systems with actuator and sensor faults. *IEEE Trans. Ind. Electron.* **62**(7), 3551–3558 (2017). Jul
47. Cao, L., Li, X., Chen, X., Zhao, Y.: Minimum sliding mode error feedback control for fault tolerant small satellite attitude control. *Adv. Space Res.* **53**(2), 309–324 (2014). Jan
48. Cao, L., Chen, X., Misra, A.K.: Minimum sliding mode error feedback control for fault tolerant reconfigurable satellite formations with J2 perturbations. *Acta Astronaut.* **96**, 201–216 (2014). Mar
49. Isidori, A.: *Nonlinear Control Systems: An Introduction*, 3rd edn. Springer, New York (1995)
50. Sun, W., Tang, S., Gao, H., Zhao, J.: Two time-scale tracking control of nonholonomic wheeled mobile robots. *IEEE Trans. Control Syst. Technol.* **24**(6), 2059–2069 (2016). Nov
51. Baek, J., Jin, M., Han, S.: A new adaptive sliding-mode control scheme for application to robot manipulators. *IEEE Trans. Ind. Electron.* **63**(5), 3632–3637 (2016). Jun
52. Han, J.: From PID to active disturbance rejection control. *IEEE Trans. Ind. Electron.* **56**(3), 900–906 (2009). Mar
53. Levant, A.: Robust exact differentiation via sliding mode technique. *Automatica* **34**(3), 379–384 (1998). Mar
54. Davila, J.: Exact tracking using backstepping control design and high-order sliding modes. *IEEE Trans. Autom. Control* **58**(8), 2077–2081 (2013). Aug
55. Chen, C.T.: *Linear System Theory and Design*. Oxford University Press, New York (1999)
56. Khalil, H.K.: *Nonlinear Systems*, 3rd edn. Prentice-Hall, Upper Saddle River (2002)
57. Atay, F.M.: Van der Pol's oscillator under delayed feedback. *J. Sound Vib.* **218**(2), 333–339 (1998)

58. Chuang, C.A., Lee, T.T., Tien, C.C., Hsu, C.F.: Adaptive wavelet neural network control for DC motors via second-order sliding-mode approach. In: Proceedings of International Conference on Machine Learning and Cybernetics, pp. 1174–1179 (2011)
59. Baek, J., Jin, M., Han, S.: A new adaptive sliding-mode control scheme for application to robot manipulators. *IEEE Trans. Ind. Electron.* **63**(6), 3628–3637 (2016). Jun
60. Lin, F.-J., Chou, P.-H.: Adaptive control of two-axis motion control system using interval type-2 fuzzy neural network. *IEEE Trans. Ind. Electron.* **56**(1), 178–193 (2009). Jan
61. Singh, R.K., Yadav, S.: Optimized PI controller for an interacting spherical tank system. In: Proceedings of 1st Conference Electronics Material Engineering Nano-Technology, pp. 1–6 (2017)
62. Shauri, R.L.A., Salleh, N.M., Hadi, A.K.A.: PID position control of 7-DOF three-fingered robotic hand for grasping task. In: Proceedings of IEEE International Conference on Control System Computing Engineering, pp. 70–74 (2014)
63. Anantachaisilp, P., Lin, Z.: An experimental study on PID tuning methods for active magnetic bearing systems. *Int. J. Adv. Mechatron. Syst.* **5**(2), 146–154 (2013)
64. Datasheet. http://www.atmel.com/Images/Atmel-42735-8-bit-AVR-Microcontroller-ATmega328-328P_Datasheet.pdf
65. Hughes, A., Drury, B.: Electric Motors and Drives: Fundamentals, Types and Applications. Newnes, Oxford (2013)
66. Cao, L., Li, H.: Unscented predictive variable structure filter for satellite attitude estimation with model errors when using low precision sensors. *Acta Astronaut.* **127**, 505–513 (2016). Oct
67. Cao, L., Chen, Y., Zhang, Z., Li, H.: Predictive smooth variable structure filter for attitude synchronization estimation during satellite formation flying. *IEEE Trans. Aerosp. Electron. Syst.* **53**(3), 1375–1383 (2017). Jun

Chapter 5

Model-Free Adaptive Near-Optimal Tracking Control



Abstract In this chapter, the receding-horizon near-optimal tracking control problem about a class of continuous-time nonlinear systems with fully unknown dynamics are considered. The main challenges of this problem lie in two aspects: (1) Most existing systems only restrict their considerations to the state feedback part while the input channel parameters are assumed to be known. This chapter considers fully unknown system dynamics in both the state feedback channel and the input channel. (2) The optimal control of nonlinear systems requires the solution of nonlinear Hamilton–Jacobi–Bellman equations. Up to today, there are no systematic approaches in the existing literature to solve it accurately. A novel model-free adaptive near-optimal control method is presented to solve this problem by utilizing the Taylor expansion based problem relaxation, the universal approximation property of sigmoid neural networks, and the concept of sliding mode control. By making an 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 adaptive near-optimal control 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 asymptotically converges to optimal. An illustrative example and experimental results are presented, which substantiate the efficacy of the presented method and verify the theoretical results.

5.1 Introduction

Uncertainties and nonlinearities are generally unavoidable when dealing with practical dynamical systems, which may degrade the performance of control laws based on exact system dynamics and make the control design difficult. For this reason, in recent decades, lots of effort has been paid on the studies of adaptive control methods for uncertain nonlinear systems. By introducing barrier Lyapunov functions, Liu et al. [1] proposed an adaptive control method via the backstepping technique for a class

of nonlinear systems whose uncertainties are linear in the parameters with explicitly defined regressors. A more general case is that, together with system parameters, the regressors are also unknown, for which traditional adaptive control approaches may not work [2].

Neural network-based methods have received more and more research interests in recent decades [2–52]. For example, based on the backstepping technique, Wang et al. [2] 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. [3] proposed a novel approximation-based adaptive tracking control method for strict-feedback nonlinear systems with input saturation. The method proposed in [3] was later extended to a class of nonstrict-feedback systems [4]. Considering that backstepping based adaptive control methods may encounter the explosion of complexity and circular issues, Na et al. [46] proposed an adaptive control method for nonlinear pure-feedback systems without using the backstepping technique. The result in [46] 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 [12, 53, 54]. In [55], 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 [56], a neural network-based control method was proposed for a piezoelectric-actuated stick-slip device. It is observed that most of the existing adaptive control methods do not consider optimality.

It is worth pointing out that optimality is one of the main concerns arising in control system designs. Chen et al. [57] investigated the receding-horizon optimal tracking control problem of a class of nonlinear systems with fully known dynamics, and proposed an explicit control law, which significantly reduces the computational burden compared with traditional methods. The results were further extended to various applications [58–61]. However, the results in [58–61] still require knowing partial dynamics of the control systems.

There are a few results reported on adaptive optimal control methods for nonlinear systems [62–68]. In [62], optimality is introduced in the adaptive backstepping control design for discrete-time nonlinear systems with unknown dynamics via a utility function, and the long-term system performance is approximated via neural networks. In terms of continuous-time nonlinear systems with unknown dynamics, some results have been reported in [66–68]. For example, the fuzzy adaptive optimal control design for strict-feedback nonlinear systems is considered in [66]. The method in [67] requires knowing the input gain, while the stability of the method in [68] was not stated.

Motivated by the above observations, 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 [66] 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 are thus derived. Based on the sliding mode control concept [46–48], 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 adaptive near-optimal control law is derived from the linear algebraic equation.

The rest of this chapter is organized in the following manner. In Sect. 5.2, the investigated problem is presented and some preliminaries are provided. In Sect. 5.3, the design process of the presented model-free adaptive near-optimal control law is illustrated. In Sect. 5.4, theoretical results are provided to guarantee the performance of the presented method. In Sect. 5.5, an illustrative examples is shown and discussed to substantiate the efficacy of the presented method and to verify the theoretical results. In Sect. 5.6, the experimental validation for the performance of the presented method is shown. In Sect. 5.8, conclusions for this chapter are provided.

5.2 Problem Description and Preliminary

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

5.2.1 Problem Description

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

$$\begin{cases} \dot{\mathbf{x}}(t) = f(\mathbf{x}(t)) + g(\mathbf{x}(t))u(t), \\ y(t) = h(\mathbf{x}(t)), \end{cases} \quad (5.1)$$

where $\mathbf{x}(t) = [x_1(t), x_2(t), \dots, x_n(t)]^T \in \mathbb{R}^n$ denotes the state vector, $y(t) \in \mathbb{R}$ denotes the system output, and $u(t) \in \mathbb{R}$ denotes the system input; $f(\cdot) : \mathbb{R}^n \rightarrow \mathbb{R}^n$, $g(\cdot) : \mathbb{R}^n \rightarrow \mathbb{R}^n$, and $h(\cdot) : \mathbb{R}^n \rightarrow \mathbb{R}$ 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 ρ [2, 57, 72, 76]. Note that examples of this class of systems with known relative degrees independent from unknown functions include many mechanical systems [47] and strict-feedback nonlinear systems [49]. Throughout this chapter, the standard Lie derivative notation and the definition of relative degree are utilized [76]. 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 (5.1) such that the tracking error defined by $e(t) = y_r(t) - y(t)$ between system output $y(t)$ and the reference output $y_r(t)$ asymptotically converges to zero under the assumption that the k th-order time derivatives $y_r^{[k]}(t)$, with $k = 0, 1, \dots, \rho$, and $y_r^{[0]}(t) = y_r(t)$ 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 [58], the control objective is formulated as the following receding-horizon optimal tracking control problem [57–61]:

$$\begin{aligned} & \text{minimize} && J(t) \\ & \text{subject to} && \dot{\mathbf{x}}(t) = f(\mathbf{x}(t)) + g(\mathbf{x}(t))u(t), \\ & && y(t) = h(\mathbf{x}(t)), \end{aligned} \quad (5.2)$$

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

$$J(t) = \int_0^T (y_r(t + \tau) - y(t + \tau))^2 d\tau, \quad (5.3)$$

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

Remark 5.1 The main challenges for the problem considered in this chapter lie in two aspects.

- (1) Receding-horizon optimal tracking control problem (5.2) 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 [57]. Up to today, there are no systematic approaches in existing literature to solve it accurately. Besides, numerically solving the nonlinear optimization problem (5.2) at each time instant is computationally intensive [57, 59].
- (2) Different from the cases considered in [57–61], 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 5.2 A performance index with the control input taken into account was provided in [77], i.e., $J(t) = v(y_r(t + \tau) - y(t + \tau)) + a \int_0^T (y_r(t + \tau) - y(t + \tau))^2 d\tau + b \int_0^T (u_r(t + \tau) - u(t + \tau))^2 d\tau$, where $a > 0 \in \mathbb{R}$, $b > 0 \in \mathbb{R}$, and $v(\cdot)$ needs to be a continuous and differentiable function with $v(0) = 0$ and $v(x) > 0$, $\forall x \neq 0$; $u_r(t)$ is the input function with which the output of the system $\dot{\mathbf{x}} = f(\mathbf{x}) + g(\mathbf{x})u_r(t)$, $y(t) = h(\mathbf{x}(t))$ satisfies $y(t) = y_r(t)$. Evidently, the physical meaning of such a performance index is not clear and it requires the knowledge of $u_r(t)$. Actually, under

system uncertainty, it is more difficult to obtain $u_r(t)$. In practice, using the performance index (5.3) means that the control accuracy is far more important than energy consumption.

5.2.2 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 $F_u(\mathbf{z}) : \mathbb{R}^n \rightarrow \mathbb{R}$ to facilitate the model-free adaptive near-optimal tracking control of system (5.1). The considered sigmoid neural network takes the following form:

$$F_{\text{nn}}(\mathbf{z}) = \sum_{j=1}^l w_{oj} \sigma_j(\mathbf{w}_{ij}^T \mathbf{z}), \quad (5.4)$$

where $\mathbf{z} = [z_1, z_2, \dots, z_n]^T \in \mathbb{R}^n$ denotes the input vector of the sigmoid neural network; $\mathbf{w}_o = [w_{o1}, w_{o2}, \dots, w_{on}]^T \in \mathbb{R}^n$ is the weight vector of the output-layer; $\mathbf{w}_{ij} = [w_{ij1}, w_{ij2}, \dots, w_{ijn}]^T \in \mathbb{R}^l$ is the input-layer weight vector with respect to the j th input neuron with $j = 1, 2, \dots, n$; $\sigma_j(\cdot)$ is the sigmoid activation function of the j th hidden neuron, which is defined as

$$\sigma_j(x) = \frac{1}{1 + \exp(-c_j(x - b_j))}, \quad (5.5)$$

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.

Lemma 5.1 (Universal Approximation [51, 78, 79]) *Consider sigmoid neural network (5.4) with sufficient number of neurons. If parameters c_j and b_j of sigmoid activation function (5.5), and \mathbf{w}_{ij} are randomly chosen, then for any continuous function $F : \mathbb{R}^n \rightarrow \mathbb{R}$, for any $\varepsilon_0 > 0$, there exist a positive integer l and an optimal weight vector \mathbf{w}_o^* such that*

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

Remark 5.3 Lemma 5.1 is also referred to as the universal approximation property. It indicates that, if the number of neurons is sufficiently large, sigmoid neural network (5.4) 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 5.1 also indicates that when utilizing sigmoid neural networks to approximate

continuous functions, constant weight vectors \mathbf{w}_{ij} , and constant parameters c_j and b_j of sigmoid activation function (5.5) can be randomly chosen.

5.2.3 Problem Reformulation

For system (5.1) with relative degree ρ , based on the definitions of relative degree and Lie derivatives [76], 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 5.1) of sigmoid neural networks, ignoring the approximation error ε_0 [73], if the number of neurons is sufficiently large, then there exist ideal unknown weight vectors \mathbf{w}_o^* and \mathbf{w}'_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 $L_f^\rho h(\mathbf{x}(t)) = \sum_{j=1}^l w_{oj}^* \sigma_j(\mathbf{w}_{ij}^T \mathbf{x}(t))$ and $L_g L_f^{\rho-1} h(\mathbf{x}(t)) = \sum_{j=1}^l w'_{oj}^* \sigma_j(\mathbf{w}_{ij}^T \mathbf{x}(t))$, respectively. It follows that the input-to-output relationship of nonlinear system (5.1) can be reformulated as follows:

$$y^{[\rho]}(t) = \sum_{j=1}^l w_{oj}^* \sigma_j(\mathbf{w}_{ij}^T \mathbf{x}(t)) + \sum_{j=1}^l w'_{oj}^* \sigma_j(\mathbf{w}_{ij}^T \mathbf{x}(t))u(t). \quad (5.6)$$

Then, problem (5.2) is reformulated as

$$\begin{aligned} & \text{minimize} && J(t) \\ & \text{subject to} && y^{[\rho]}(t) = \sum_{j=1}^l w_{oj}^* \sigma_j(\mathbf{w}_{ij}^T \mathbf{x}(t)) + \sum_{j=1}^l w'_{oj}^* \cdot \sigma_j(\mathbf{w}_{ij}^T \mathbf{x}(t))u(t), \end{aligned} \quad (5.7)$$

where \mathbf{w}_o^* and \mathbf{w}'_o^* are unknown.

5.3 Control Design

In this section, the control design process is illustrated. Since the parameters of the system (5.6) 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 presented. For the convenience of illustration and readability, the corresponding theoretical analysis is presented in the next section.

5.3.1 Problem Relaxation

In this subsection, the receding-horizon optimal tracking control problem (5.7) 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 a time instant t , i.e.,

$$y(t + \tau) \approx y(t) + \tau \dot{y}(t) + \frac{\tau^2}{2!} y^{[2]}(t) + \cdots + \frac{\tau^\rho}{\rho!} y^{[\rho]}(t), \quad (5.8)$$

where $y^{[i]}(t)$ denotes the i th order time derivative of $y(t)$ with $i = 0, 1, 2, \dots, \rho$ and $y^{[0]}(t) = y(t)$. Substituting Eq. (5.6) into Eq. (5.8) yields

$$y(t + \tau) \approx \rho^T(\tau) \Upsilon(t) + \frac{\tau^\rho}{\rho!} \sum_{j=1}^l w_{oj}^* \sigma_j (\mathbf{w}_{ij}^T \mathbf{x}(t)) u(t), \quad (5.9)$$

where

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

and

$$\rho(\tau) = [1, \tau, \tau^2/2, \dots, \tau^\rho/\rho!]^T.$$

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

$$y_r(t + \tau) \approx \rho^T(\tau) \Upsilon_r(t), \quad (5.10)$$

where $\Upsilon_r(t) = [y_r(t), \dot{y}_r(t), \dots, y_r^{[\rho]}(t)]^T$.

Let $\check{\Upsilon}(t) = \Upsilon_r(t) - \Upsilon(t)$. Based on Eqs. (5.9) and (5.10), performance index (5.3) is relaxed in the following manner:

$$\begin{aligned} J(t) &\approx \hat{J}(t) = \frac{1}{2} \int_0^T (\rho^T(\tau) \check{\Upsilon}(t) - \frac{\tau^\rho}{\rho!} \sum_{j=1}^l w_{oj}^* \sigma_j (\mathbf{w}_{ij}^T \mathbf{x}(t)) u(t))^2 d\tau \\ &= \frac{1}{2} \int_0^T \left((\rho^T(\tau) \check{\Upsilon}(t))^2 - \frac{2\tau^\rho}{\rho!} \rho^T(\tau) \check{\Upsilon}(t) \sum_{j=1}^l w_{oj}^* \sigma_j (\mathbf{w}_{ij}^T \mathbf{x}(t)) u(t) \right. \\ &\quad \left. + \frac{\tau^{2\rho}}{(\rho!)^2} \left(\sum_{j=1}^l w_{oj}^* \sigma_j (\mathbf{w}_{ij}^T \mathbf{x}(t)) \right)^2 u^2(t) \right) d\tau \end{aligned}$$

$$\begin{aligned}
&= \frac{1}{2} \int_0^T (\rho^T(\tau) \check{Y}(t))^2 d\tau - \rho_{\text{int}} \check{Y}(t) \sum_{j=1}^l w_{0j}'^* \sigma_j(\mathbf{w}_{ij}^T \mathbf{x}(t)) u(t) \\
&\quad + \frac{T^{2\rho+1}}{2(2\rho+1)(\rho!)^2} \left(\sum_{j=1}^l w_{0j}'^* \sigma_j(\mathbf{w}_{ij}^T \mathbf{x}(t)) \right)^2 u^2(t),
\end{aligned}$$

where constant vector ρ_{int} is calculated by

$$\begin{aligned}
\rho_{\text{int}} &= \int_0^T \tau^\rho \rho^T(\tau) / \rho! d\tau \\
&= \left[\frac{T^{\rho+1}}{(\rho+1)\rho!}, \frac{T^{\rho+2}}{(\rho+2)\rho!1!}, \dots, \frac{T^{2\rho+1}}{(2\rho+1)(\rho!)^2} \right]. \tag{5.11}
\end{aligned}$$

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 (5.7) is relaxed to the following unconstrained quadratic program:

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

where

$$p(t) = \frac{T^{2\rho+1} (\sum_{j=1}^l w_{0j}'^* \sigma_j(\mathbf{w}_{ij}^T \mathbf{x}(t)))^2}{(2\rho+1)(\rho!)^2},$$

$$q(t) = \rho_{\text{int}} \check{Y}(t) \sum_{j=1}^l w_{0j}'^* \sigma_j(\mathbf{w}_{ij}^T \mathbf{x}(t)),$$

and

$$c(t) = \frac{1}{2} \int_0^T (\rho^T(\tau) \check{Y}(t))^2 d\tau.$$

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 [80]. 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.,

$$-\rho_{\text{int}} \check{Y}(t) + \frac{T^{2\rho+1} \sum_{j=1}^l w_{0j}'^* \sigma_j(\mathbf{w}_{ij}^T \mathbf{x}(t)) u(t)}{(2\rho+1)(\rho!)^2} = 0,$$

from which the following equation is derived:

$$\begin{aligned}
& - \sum_{i=0}^{\rho-1} \frac{T^i}{(\rho+1+i)i!} (y_r^{[i]}(t) - y^{[i]}(t)) - \frac{T^\rho}{(2\rho+1)\rho!} y_r^{[\rho]}(t) \\
& + \frac{T^\rho}{(2\rho+1)\rho!} \left(\sum_{j=1}^l w_{oj}^* \sigma_j(\mathbf{w}_{ij}^T \mathbf{x}(t)) \right) + \sum_{j=1}^l w_{oj}'^* \sigma_j(\mathbf{w}_{ij}^T \mathbf{x}(t)) u(t) = 0. \quad (5.12)
\end{aligned}$$

By the above steps, receding-horizon optimal tracking control problem (5.7) for nonlinear system (5.6) is relaxed to an equation with unknown terms \mathbf{w}_o^* and $\mathbf{w}_o'^*$.

Remark 5.4 The Eq. (5.12) is a result of approximating performance index (5.3) by considering all the constraints in the receding-horizon optimal tracking control problem. 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 $t + \tau$.

5.3.2 Reconstruction of Input-to-Output Dynamics

Form the previous subsection, the solution to Eq. (5.12) depends on the fully unknown terms \mathbf{w}_o^* and $\mathbf{w}_o'^*$, which are closely related to the input-to-output property of nonlinear system (5.6). In this subsection, an auxiliary system is designed and presented to reconstruct the input-to-output dynamics of system (5.6), which makes solving Eq. (5.12) feasible.

Evidently, the problem becomes finding a group of $\mathbf{w}_{oj}(t)$ and $\mathbf{w}_{oj}'(t)$ such that

$$\begin{aligned}
& \lim_{t \rightarrow +\infty} \left(\left(\sum_{j=1}^l w_{oj}(t) \sigma_j(\mathbf{w}_{ij}^T \mathbf{x}(t)) + \sum_{j=1}^l w_{oj}'(t) \sigma_j(\mathbf{w}_{ij}^T \mathbf{x}(t)) u(t) \right) \right. \\
& \left. - \left(\sum_{j=1}^l w_{oj}^* \sigma_j(\mathbf{w}_{ij}^T \mathbf{x}(t)) + \sum_{j=1}^l w_{oj}'^* \sigma_j(\mathbf{w}_{ij}^T \mathbf{x}(t)) u(t) \right) \right) = 0.
\end{aligned}$$

To this end, we design the following auxiliary system:

$$\begin{cases} \hat{y}^{[\rho]}(t) = \sum_{j=1}^l w_{oj}(t) \sigma_j(\mathbf{w}_{ij}^T \mathbf{x}(t)) + \sum_{j=1}^l w_{oj}'(t) \sigma_j(\mathbf{w}_{ij}^T \mathbf{x}(t)) u(t) - \sum_{j=0}^{\rho-2} \alpha_j \tilde{y}^{[j+1]}(t) \\ \quad - \lambda \phi(s(t)), \\ \dot{w}_{oj}(t) = -\gamma s(t) \sigma_j(\mathbf{w}_{ij}^T \mathbf{x}(t)), \quad j = 1, 2, \dots, l, \\ \dot{w}_{oj}'(t) = -\gamma s(t) \sigma_j(\mathbf{w}_{ij}^T \mathbf{x}(t)) u(t) - p_j(t), \quad j = 1, 2, \dots, l, \end{cases} \quad (5.13)$$

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 $j = 0, 1, \dots, \rho-2$; $\phi(\cdot) : \mathbb{R} \rightarrow \mathbb{R}$ 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 (5.6); $\gamma > 0 \in \mathbb{R}$ is a parameter used to scale the parameter response to the output difference between auxiliary system (5.13) and system (5.6). In addition, $p_j(t)$ is the j th element of $\mathbf{p}(\mathbf{w}'_o(t))$ and

$$\mathbf{p}(\mathbf{w}'_o(t)) = \begin{cases} \mathbf{w}'_o(t), & \text{if } \mathbf{w}'_o(t) \in \Omega, \\ \|\gamma s(t)\sigma_{\text{vec}}(t)u(t)\|_2 \left(\frac{(\mathbf{w}'_o(t))^T \sigma_{\text{vec}}(t)}{\|\sigma_{\text{vec}}(t)\|_2} \right. \\ \left. - \frac{d_0}{\|\sigma_{\text{vec}}(t)\|_2} \right) \frac{\sigma_{\text{vec}}(t)}{\|\sigma_{\text{vec}}(t)\|_2}, & \text{if } \mathbf{w}'_o(t) \notin \Omega, \end{cases} \quad (5.14)$$

where

$$\Omega = \left\{ \mathbf{w}'_o \in \mathbb{R}^l \mid \sum_{j=1}^l w'_{oj}(t) \sigma_j(\mathbf{w}_{ij}^T \mathbf{x}(t)) \geq d_0 \right\},$$

$$\sigma_{\text{vec}}(t) = [\sigma_1(\mathbf{w}_{i1}^T \mathbf{x}(t)), \sigma_2(\mathbf{w}_{i2}^T \mathbf{x}(t)), \dots, \sigma_l(\mathbf{w}_{il}^T \mathbf{x}(t))]^T,$$

and $\|\cdot\|_2$ denotes the 2-norm of a vector. By the definition of the sigmoid function shown in Eq. (5.5), each element of $\sigma_{\text{vec}}(t)$ is strictly bigger than zero for any time instant t . It follows that $\|\sigma_{\text{vec}}(t)\|_2 > 0$ for any t , i.e., $\mathbf{p}(\mathbf{w}'_o(t))$ is well-defined.

We offer the following remark concerning the underlying intuitions in the design of auxiliary system (5.13).

Remark 5.5 The design of auxiliary system (5.13) is inspired by the progresses of sliding mode control [46–48, 81] and dynamical neural networks [29, 44]. In auxiliary system (5.13), the sliding mode surface is defined by $s(t) = 0$. The parameter evolution rules in (5.13) 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 α_j for $j = 0, 1, \dots, \rho - 1$, 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 [82]. It is worth pointing out that unlike [46–48, 81], 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., [44]), which are viewed as a powerful tool to enhance convergence of dynamical systems, nonlinear activation function $\phi(\cdot)$ is incorporated in the presented 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, $\sum_{j=1}^l w'_{oj}(t) \sigma_j(\mathbf{w}_{ij}^T \mathbf{x}(t)) \geq d_0$, so as to avoid the so-called control singularity problem [70]. The detailed proof is latter provided in Theorem 5.1.

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

Remark 5.6 As seen from auxiliary system (5.13), together with the selection of constant parameters of sigmoid functions (which is discussed in Remark 5.1) and α_j as discussed in Remark 5.5, there are some other constant parameters to be selected, i.e., λ and γ . Since λ and γ 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 [83]. As a result, a larger value of λ and γ 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 (5.6) via the evolution of parameters $w_{oj}(t)$ and $w'_{oj}(t)$ (with $j = 1, 2, \dots, l$), the value of γ should not be smaller than that of λ . In terms of monotonically increasing odd activation function $\phi(\cdot)$, there are many alternatives, some of which can be seen from [44, 84, 85].

5.3.3 Adaptive Near-Optimal Control Law

In this subsection, based on auxiliary system (5.13) and algebraic equation (5.12), an adaptive near-optimal control law is derived.

From auxiliary system (5.13), on the sliding mode surface defined by $s(t) = 0$, it can be derived that $\sum_{j=1}^l w_{oj}(t)\sigma_j(\mathbf{w}_{ij}^T \mathbf{x}(t)) + \sum_{j=1}^l w'_{oj}(t)\sigma_j(\mathbf{w}_{ij}^T \mathbf{x}(t))u(t) = \sum_{j=1}^l w_{oj}^*\sigma_j(\mathbf{w}_{ij}^T \mathbf{x}(t)) + \sum_{j=1}^l w'^*_j\sigma_j(\mathbf{w}_{ij}^T \mathbf{x}(t))u(t)$, which is latter theoretically guaranteed by Theorem 5.2 shown in Sect. 5.4. Together with Eq. (5.12), one further has

$$\begin{aligned} & -\sum_{i=0}^{\rho-1} \frac{T^i}{(\rho+1+i)i!} (y_r^{[i]}(t) - y^{[i]}(t)) - \frac{T^\rho}{(2\rho+1)\rho!} y_r^{[\rho]}(t) \\ & + \frac{T^\rho}{(2\rho+1)\rho!} \left(\sum_{j=1}^l w_{oj}(t)\sigma_j(\mathbf{w}_{ij}^T \mathbf{x}(t)) + \sum_{j=1}^l w'_{oj}(t) \right. \\ & \left. \cdot \sigma_j(\mathbf{w}_{ij}^T \mathbf{x}(t))u(t) \right) = 0. \end{aligned} \quad (5.15)$$

Note that $\mathbf{w}_o(t)$ and $\mathbf{w}'_o(t)$ are generated by the designed auxiliary system and they are thus totally known. Then, by solving Eq. (5.15), given that

$$\sum_{j=1}^l w'_{oj}(t)\sigma_j(\mathbf{w}_{ij}^T \mathbf{x}(t)) \neq 0,$$

an adaptive near-optimal control law is obtained as follows:

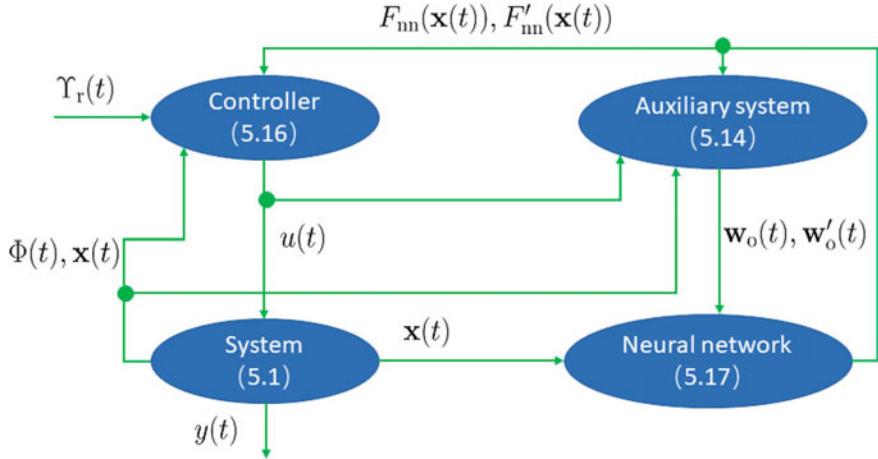


Fig. 5.1 Block diagram for the implementation of the presented adaptive near-optimal tracking control for nonlinear system (5.1) with fully unknown dynamics, where $\Upsilon_r(t) = [y_r(t), \dot{y}_r(t), \dots, y_r^{[\rho]}(t)]^T$ and $\Phi(t) = [y(t), \dot{y}(t), \dots, y^{[\rho-1]}(t)]^T$

$$u(t) = \frac{-1}{\sum_{j=1}^l w'_{oj}(t) \sigma_j(\mathbf{w}_{ij}^T \mathbf{x}(t))} \left(\sum_{i=0}^{\rho-1} \frac{T^i (2\rho+1)\rho!}{(\rho+1+i)i! T^\rho} (y_r^{[i]}(t) - y^{[i]}(t)) - y_r^{[\rho]}(t) - \sum_{j=1}^l w_{oj}(t) \sigma_j(\mathbf{w}_{ij}^T \mathbf{x}(t)) \right),$$

which can be rewritten in a more concise form:

$$u(t) = \frac{-1}{F'_{nn}(\mathbf{x}(t))} \left(\sum_{i=0}^{\rho-1} \frac{T^i (2\rho+1)\rho!}{(\rho+1+i)i! T^\rho} (y_r^{[i]}(t) - y^{[i]}(t)) - y_r^{[\rho]}(t) - F_{nn}(\mathbf{x}(t)) \right), \quad (5.16)$$

with

$$\begin{cases} F'_{nn}(\mathbf{x}(t)) = \sum_{j=1}^l w'_{oj}(t) \sigma_j(\mathbf{w}_{ij}^T \mathbf{x}(t)), \\ F_{nn}(\mathbf{x}(t)) = \sum_{j=1}^l w_{oj}(t) \sigma_j(\mathbf{w}_{ij}^T \mathbf{x}(t)). \end{cases} \quad (5.17)$$

Since the basis functions and input-layer weight vectors are the same for $F_{nn}(\mathbf{x}(t))$ and $F'_{nn}(\mathbf{x}(t))$, 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 5.7 A block diagram about the implementation of the presented adaptive near-optimal tracking control method is shown in Fig. 5.1. As seen from this figure, the presented adaptive near-optimal tracking control method for nonlinear system (5.1) with fully unknown dynamics consist of three parts, i.e., adaptive near-optimal control law (5.16), auxiliary system (5.13), and sigmoid neural network (5.17). Based on the output and state information of system (5.1) as well as system input $u(t)$, auxiliary system (5.13) adaptively generates better values of output-layer weight vectors $\mathbf{w}_o(t)$ and $\mathbf{w}'_o(t)$ for sigmoid neural network (5.17). As a result, the sigmoid neural network generates better values of $F_{nn}(\mathbf{x}(t))$ and $F'_{nn}(\mathbf{x}(t))$. Based on outputs $F_{nn}(\mathbf{x}(t))$ and $F'_{nn}(\mathbf{x}(t))$ of sigmoid neural network (5.17) and the output information $\Phi(t)$ of system (5.1), the adaptive near-optimal control law generates the input $u(t)$ to force output $y(t)$ of system (5.1) to track the desired output $y_r(t)$ while minimizing the performance index (5.3). By this type of feedback and information interaction, the presented adaptive near-optimal tracking control method is thus intuitively valid for system (5.1) with fully unknown dynamics.

Remark 5.8 The presented adaptive near-optimal tracking control 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 [2, 49, 63]. In our method, together with the system states and the output, the output derivatives $y^{[i]}$ with $i = 1, 2, \dots, \rho - 1$ need to be known. For practical systems, output derivatives may be measured by sensors [71]. Another alternative is tracking differentiators [74, 86, 87], 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 [74, 86, 87], some of which are finite-time convergent. The satisfactory performance of tracking differentiators has been substantiated by practical applications [75, 88].

Remark 5.9 The strict-feedback nonlinear system considered in [66] is a special case of the system considered in this chapter. In other words, compared with the backstepping approach, the presented method allows a more general system form. Besides, the control gain is assumed to be a known constant in [66], which is considered to be an 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 5.10 In practice, the input of a control system may be constrained and there are some existing results addressing the input saturation problem. For example, the vibration control of flexible marine riser systems with input saturation was investigated in [89]. The fuzzy tracking control problem for a class of nonlinear systems with input saturation was investigated in [90]. The attitude control of rigid spacecraft with actuator saturation was investigated in [91]. 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 $u_a(t)$ denotes the actual input given to the controlled system. Then, by setting

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

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

Regarding the relative degree of system (5.1) with unknown dynamics, we have the following remark.

Remark 5.11 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 prior knowledge is not available, we may use the controller by a trial and error method. Specifically, we may first 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 presented method cannot work for the system.

5.4 Theoretical Analysis

In this section, theoretical results are provided to guarantee the performance of the presented adaptive near-optimal tracking control method, including the convergence of the presented auxiliary system, the stability of the resultant closed-loop system, and the optimality of the performance index.

5.4.1 Confirmation of No Singularity Problems

As stated in Remark 5.5, a bias term $p_j(t)$ is incorporated into the design of auxiliary system (5.13) so as to avoid the singularity problem (i.e., $\sum_{j=1}^l w'_{oj}(t)\sigma_j(\mathbf{w}_{ij}^T \mathbf{x}(t)) = 0$) that could degrade the performance of adaptive near-optimal control law (5.16). In this subsection, we provide a theoretical result to prove that the presented method is singularity-free.

Theorem 5.1 *If $\sum_{j=1}^l w'_{oj}(0)\sigma_j(\mathbf{w}_{ij}^T \mathbf{x}(0)) \geq d_0 > 0$, then, for any time instant $t \geq 0$, $\mathbf{w}'_o(t)$ of auxiliary system (5.13) satisfies $\mathbf{w}'_o(t) \in \Omega$ with $\Omega = \{\mathbf{w}'_o \in \mathbb{R}^l | \sum_{j=1}^l w'_{oj}(t)\sigma_j(\mathbf{w}_{ij}^T \mathbf{x}(t)) \geq d_0\}$.*

Proof Define the following projection function:

$$P_{\Omega}(\mathbf{w}'_o(t)) = \begin{cases} \mathbf{w}'_o(t), & \text{if } \mathbf{w}'_o(t) \in \Omega, \\ \mathbf{w}'_o(t) - \frac{(\mathbf{w}'_o(t))^T \sigma_{\text{vec}}(t) - d_0}{\sigma_{\text{vec}}^T(t) \sigma_{\text{vec}}(t)} \sigma_{\text{vec}}(t), & \text{if } \mathbf{w}'_o(t) \notin \Omega, \end{cases}$$

where

$$\Omega = \{\mathbf{w}'_o \in \mathbb{R}^l \mid \sum_{j=1}^l w'_{oj}(t) \sigma_j(\mathbf{w}_{ij}^T \mathbf{x}(t)) \geq d_0\},$$

and

$$\sigma_{\text{vec}}(t) = [\sigma_1(\mathbf{w}_{11}^T \mathbf{x}(t)), \sigma_2(\mathbf{w}_{12}^T \mathbf{x}(t)), \dots, \sigma_l(\mathbf{w}_{il}^T \mathbf{x}(t))]^T.$$

Consider the Lyapunov function candidate $V_1(t) = \|\mathbf{w}'_o(t) - P_{\Omega}(\mathbf{w}'_o(t))\|_2^2/2$. Calculating the derivative of $V_1(t)$ yields

$$\dot{V}_1(t) = (\mathbf{w}'_o(t) - P_{\Omega}(\mathbf{w}'_o(t)))^T \dot{\mathbf{w}}'_o(t).$$

By auxiliary system (5.13),

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

It follows that

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

Note that, from $\dot{V}_1(t) = 0$, one has $\mathbf{w}'_o(t) = P_{\Omega}(\mathbf{w}'_o(t))$, i.e., $\mathbf{w}'_o(t) \in \Omega$, which is also the largest invariant set. Then, by LaSalle's invariance principle [100], if $\sum_{j=1}^l w'_{oj}(0) \sigma_j(\mathbf{w}_{ij}^T \mathbf{x}(0)) \geq d_0$, i.e., $\mathbf{w}'_o(0) \in \Omega$, then, for any time instant $t \geq 0$, $\mathbf{w}'_o(t)$ of auxiliary system (5.13) satisfies $\mathbf{w}'_o(t) \in \Omega$. The proof is complete. \square

Remark 5.12 According to Theorem 5.1, if $\mathbf{w}'_o(0)$ is properly set such that

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

then $\sum_{j=1}^l w'_{oj}(t) \sigma_j(\mathbf{w}_{ij}^T \mathbf{x}(t)) \geq d_0 > 0$ for any $t \geq 0$. It follows that there is no control singularity in adaptive near-optimal control law (5.16).

Corollary 5.1 If $\sum_{j=1}^l w'_{oj}(0)\sigma_j(\mathbf{w}_{ij}^T \mathbf{x}(0)) \geq d_0$, then auxiliary system (5.13) is equivalent to

$$\begin{cases} \hat{y}^{[\rho]}(t) = \sum_{j=1}^l w_{oj}(t)\sigma_j(\mathbf{w}_{ij}^T \mathbf{x}(t)) + \sum_{j=1}^l w'_{oj}(t)\sigma_j(\mathbf{w}_{ij}^T \\ \quad \cdot \mathbf{x}(t))u(t) - \sum_{j=0}^{\rho-2} \alpha_j \tilde{y}^{[j+1]}(t) - \lambda\phi(s(t)), \\ \dot{w}_{oj}(t) = -\gamma s(t)\sigma_j(\mathbf{w}_{ij}^T \mathbf{x}(t)), \quad j = 1, 2, \dots, l, \\ \dot{w}'_{oj}(t) = -\gamma s(t)\sigma_j(\mathbf{w}_{ij}^T \mathbf{x}(t))u(t), \quad j = 1, 2, \dots, l. \end{cases} \quad (5.18)$$

Proof From Theorem 5.1, if $\sum_{j=1}^l w'_{oj}(0)\sigma_j(\mathbf{w}_{ij}^T \mathbf{x}(0)) \geq d_0 > 0$, then for any time instant $t \geq 0$, $\mathbf{w}'_o(t)$ of auxiliary system (5.13) satisfies $\mathbf{w}'_o(t) \in \mathcal{Q}$. It follows that $\mathbf{p}(\mathbf{w}'_o(t))$ defined in Eq.(5.14) satisfies $\mathbf{p}(\mathbf{w}'_o(t)) = 0, \forall t \geq 0$, i.e., $p_j(t) \equiv 0, \forall j = 1, 2, \dots, l$. Therefore, auxiliary system (5.13) is equivalent to (5.18). The proof is complete. \square

5.4.2 Convergence of the Auxiliary System

One of the main components of the presented method is the auxiliary system (5.13), which is used to reconstruct the input-to-output property of the system (5.6). The following theorem guarantees the convergence of the auxiliary system.

Theorem 5.2 (Asymptotical Convergence of the Auxiliary System) *If*

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

and all the roots of characteristic equation $\sum_{j=0}^{\rho-1} \alpha_j v^j = 0$ are located on the left half-plane, then the input-to-output property of auxiliary system (5.13) asymptotically converges to that of nonlinear system (5.6) with $\lim_{t \rightarrow +\infty} (\sum_{j=1}^l \tilde{w}_{oj}(t)\sigma_j(\mathbf{w}_{ij}^T \mathbf{x}(t)) + \sum_{j=1}^l \tilde{w}'_{oj}(t)\sigma_j(\mathbf{w}_{ij}^T \mathbf{x}(t))u(t)) = 0$, where $\tilde{\mathbf{w}}_o(t) = \mathbf{w}_o(t) - \mathbf{w}_o^$ and $\tilde{\mathbf{w}}'_o(t) = \mathbf{w}'_o(t) - \mathbf{w}'_o^*$.*

Proof Let $\tilde{\mathbf{w}}_o(t) = \mathbf{w}_o(t) - \mathbf{w}_o^*$ and $\tilde{\mathbf{w}}'_o(t) = \mathbf{w}'_o(t) - \mathbf{w}'_o^*$. Subtracting Eq. (5.6) from the first equation of auxiliary system (5.13) yields

$$\begin{aligned} \tilde{y}^{[\rho]}(t) &= \sum_{j=1}^l \tilde{w}_{oj}(t)\sigma_j(\mathbf{w}_{ij}^T \mathbf{x}(t)) + \sum_{j=1}^l \tilde{w}'_{oj}(t)\sigma_j(\mathbf{w}_{ij}^T \mathbf{x}(t))u(t) \\ &\quad - \sum_{j=0}^{\rho-2} \alpha_j \tilde{y}^{[j+1]}(t) - \lambda\phi(s(t)), \end{aligned}$$

from which one has

$$\dot{s}(t) = \sum_{j=1}^l \tilde{w}_{0j}(t) \sigma_j(\mathbf{w}_{ij}^T \mathbf{x}(t)) + \sum_{j=1}^l \tilde{w}'_{0j}(t) \sigma_j(\mathbf{w}_{ij}^T \mathbf{x}(t)) u(t) - \lambda \phi(s(t)), \quad (5.19)$$

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 $j = 0, 1, \dots, \rho - 2$.

Consider the following Lyapunov function candidate:

$$V(t) = \frac{1}{2\gamma} \tilde{\mathbf{w}}_o^T(t) \tilde{\mathbf{w}}_o(t) + \frac{1}{2\gamma} \tilde{\mathbf{w}}'_o^T(t) \tilde{\mathbf{w}}'_o(t) + \frac{1}{2} s^2(t). \quad (5.20)$$

Calculating the derivative of $V(t)$ yields

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

From Corollary 5.1, if $\sum_{j=1}^l w'_{0j}(0) \sigma_j(\mathbf{w}_{ij}^T \mathbf{x}(0)) \geq d_0$, then auxiliary system (5.13) is equivalent to (5.18). Together with Eq. (5.19), one further has

$$\begin{aligned} \dot{V}(t) &= -s(t) \sum_{j=1}^l \tilde{w}_{0j}(t) \sigma_j(\mathbf{w}_{ij}^T \mathbf{x}(t)) - s(t) \sum_{j=1}^l \tilde{w}'_{0j}(t) \sigma_j(\mathbf{w}'^T_{ij} \mathbf{x}(t)) \\ &\quad + s(t) \left(\sum_{j=1}^l \tilde{w}_{0j}(t) \sigma_j(\mathbf{w}_{ij}^T \mathbf{x}(t)) + \sum_{j=1}^l \tilde{w}'_{0j}(t) \sigma_j(\mathbf{w}_{ij}^T \mathbf{x}(t)) u(t) - \lambda \phi(s(t)) \right) \\ &= -\lambda s(t) \phi(s(t)). \end{aligned}$$

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 α_j with $j = 0, 1, \dots, \rho - 1$ and initial values $y^{[j]}(0)$ with $j = 0, 1, \dots, \rho - 1$; $b_j(t)$ belong to set $\{b_1(t), b_2(t), \dots, b_{\rho-1}(t)\}$ which is a subset of the fundamental solution set $\{\exp(r_1 t), t \exp(r_1 t), \dots, t^{d_1-1} \exp(r_1 t), \exp(r_2 t), t \exp(r_2 t), \dots, t^{d_2-1} \exp(r_2 t), \dots\}$. r_i is a root repeated d_i times ($d_i \leq \rho - 1$) in the following characteristic equation [101]:

$$\sum_{j=0}^{\rho-1} \alpha_j v^j = 0. \quad (5.21)$$

By properly choosing the value of α_j with $j = 0, 1, \dots, \rho - 1$ such that all the roots of (5.21) are located in the left half-plane, one readily has $\lim_{t \rightarrow \infty} b_i^{[j]}(t) = 0, \forall i \in \{0, 1, \dots\}, \forall j \in \{0, 1, \dots\}$. It follows that, on the invariant set, $\lim_{t \rightarrow +\infty} \tilde{y}^{[j]}(t) = 0, \forall j \in \{0, 1, \dots\}$.

Based on the above results, by LaSalle's invariance principle [100], we have

$$\lim_{t \rightarrow +\infty} \tilde{y}^{[j]}(t) = 0, \forall j \in \{0, 1, \dots\}.$$

Then, it follows from Eq. (5.19) that

$$\lim_{t \rightarrow +\infty} \left(\sum_{j=1}^l \tilde{w}_{oj}(t) \sigma_j(\mathbf{w}_{ij}^T \mathbf{x}(t)) + \sum_{j=1}^l \tilde{w}'_{oj}(t) \sigma_j(\mathbf{w}_{ij}^T \mathbf{x}(t)) u(t) \right) = 0.$$

This completes the proof. \square

Based on Theorem 5.2 and the persistent excitation (PE) condition [92], 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 [50, 92, 93].

Corollary 5.2 (Exact Parameter Adaptation) *If the two vectors*

$$[\sigma_1(\mathbf{w}_{i1}^T \mathbf{x}(t)), \sigma_2(\mathbf{w}_{i2}^T \mathbf{x}(t)), \dots, \sigma_l(\mathbf{w}_{il}^T \mathbf{x}(t))]^T$$

and

$$[\sigma_1(\mathbf{w}_{i1}^T \mathbf{x}(t)) u(t), \sigma_2(\mathbf{w}_{i2}^T \mathbf{x}(t)) u(t), \dots, \sigma_l(\mathbf{w}_{il}^T \mathbf{x}(t)) u(t)]^T$$

simultaneously satisfy the PE condition, auxiliary system (5.13) satisfies

$$\lim_{t \rightarrow +\infty} \mathbf{w}_o(t) = \mathbf{w}_o^*$$

and

$$\lim_{t \rightarrow +\infty} \mathbf{w}'_o(t) = \mathbf{w}'_o^*.$$

Proof The proof can be generalized from [92] based on Theorem 5.2 and is thus omitted. \square

Note that the theorems in this chapter do not rely on Corollary 5.2, and thus the PE condition. This, as we consider, is also one of the differences of the presented 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 [94]. In practice, the PE condition could be guaranteed when using a sufficiently rich reference trajectory for the control system.

5.4.3 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 (5.6) and auxiliary system (5.13).

Theorem 5.3 (Asymptotical Stability of the Control System) *If*

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

all the roots of characteristic equation $\sum_{j=0}^{\rho-1} \alpha_j v^j = 0$ 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 (5.6) and adaptive near-optimal control law (5.16) is asymptotically stable.

Proof First, system (5.6) is rewritten as

$$y^{[\rho]}(t) = \sum_{j=1}^l w_{oj}(t)\sigma_j(\mathbf{w}_{ij}^T \mathbf{x}(t)) + \sum_{j=1}^l w'_{oj}(t)\sigma_j(\mathbf{w}_{ij}^T \mathbf{x}(t))u(t) + \eta(t), \quad (5.22)$$

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

$$\eta(t) = - \left(\sum_{j=1}^l \tilde{w}_{oj}\sigma_j(\mathbf{w}_{ij}^T \mathbf{x}(t)) \right) + \sum_{j=1}^l \tilde{w}'_{oj}\sigma_j(\mathbf{w}_{ij}^T \mathbf{x}(t))u(t))$$

with $\tilde{\mathbf{w}}_o(t) = \mathbf{w}_o(t) - \mathbf{w}_o^*$ and $\tilde{\mathbf{w}}'_o(t) = \mathbf{w}'_o(t) - \mathbf{w}'_o^*$. Recall adaptive near-optimal control law (5.16):

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

Substituting the expression of $u(t)$ into (5.22) yields

$$y^{[\rho]}(t) = \sum_{i=0}^{\rho-1} \frac{T^i (2\rho+1)\rho!(y_r^{[i]}(t) - y^{[i]}(t))}{(\rho+1+i)i!T^\rho} - y_r^{[\rho]}(t) + \eta(t). \quad (5.23)$$

Let $e(t) = y_r(t) - y(t)$. Then, from Eq.(5.23),

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

i.e., the closed-loop system consisting of nonlinear system (5.6) and adaptive near-optimal control law (5.16) can be described as follows:

$$\sum_{i=0}^{\rho} \frac{T^i}{(\rho+1+i)i!} e_i^{[i]}(t) = \eta(t), \quad (5.24)$$

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 [76], it is readily checked that the all the roots of the corresponding characteristic equation $\sum_{i=0}^{\rho} T^i v / ((\rho+1+i)i!) = 0$ 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 5.2, if all the roots of characteristic equation $\sum_{j=0}^{\rho-1} \alpha_j v^j = 0$ are located on the left half-plane, then

$$\lim_{t \rightarrow +\infty} \left(\sum_{j=1}^l \tilde{w}_{oj}(t) \sigma_j(\mathbf{w}_{ij}^T \mathbf{x}(t)) + \sum_{j=1}^l \tilde{w}'_{oj}(t) \sigma_j(\mathbf{w}_{ij}^T \mathbf{x}(t)) u(t) \right) = 0,$$

i.e., $\lim_{t \rightarrow +\infty} \eta(t) = 0$. Then, it follows by the bounded-input bounded-output property [76] of linear systems that equilibrium $e(t) = 0$ of closed-loop system (5.24) is asymptotically stable. The proof is complete. \square

Remark 5.13 As seen from Theorem 5.3, there are several conditions needed to guarantee the asymptotic stability of the closed-loop system. The condition that $\sum_{j=1}^l w'_{oj}(0) \sigma_j(\mathbf{w}_{ij}^T \mathbf{x}(0)) \geq d_0 > 0$ is very weak, which can be readily satisfied by choosing values of $w'_{oj}(0)$. The condition on the roots of characteristic equation $\sum_{j=0}^{\rho-1} \alpha_j v^j = 0$ is also very weak. Specifically, one can predefine desired roots r_i with $i = 1, 2, \dots, \rho - 1$. By letting $\sum_{j=0}^{\rho-1} \alpha_j v^j = \prod_{j=1}^{\rho-1} (v^j - r_j)$, one can readily find the corresponding α_j . For example, if $\rho = 2$ and the desired root is -5 . Then, from $\alpha_0 + \alpha_1 v = (v - (-5))$, one has $\alpha_0 + \alpha_1 v = v + 5$. 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 [70, 95, 96].

5.4.4 Asymptotic Optimality of Performance Index

As shown in the previous section, the motivation of the presented adaptive near-optimal control method is to achieve optimal tracking via minimizing the integral performance index (5.3) for nonlinear systems with fully unknown dynamics. Due to the problem relaxation, the presented adaptive near-optimal control law cannot be optimal all the time, which is also a reason why it is called a near-optimal one. Fortunately, theoretical analysis shows that the presented one is asymptotically optimal.

Theorem 5.4 (Asymptotical Optimality of the Performance Index) *If*

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

given all the roots of characteristic equation $\sum_{j=0}^{\rho-1} \alpha_j v^j = 0$ located on the left half-plane and relative degree $\rho \in \{1, 2, 3, 4\}$, then, synthesized by adaptive near-optimal control law (5.16), performance index (5.3) associated with nonlinear system (5.6) asymptotically converges to optimal. In other words, control law (5.16) is asymptotically optimal.

Proof By Taylor expansion, performance index (5.3) is rewritten as

$$J(t) = \frac{1}{2} \int_0^T (\rho^T(\tau) \ddot{Y}(t) - \frac{\tau^\rho}{\rho!} L_g L_f^{\rho-1} h(\mathbf{x}(t)) u(t) + \frac{\tau^\rho}{\rho!} \Delta(t))^2 d\tau, \quad (5.25)$$

where $\Delta(t) = (y_d^{[\rho]}(t + \varkappa\tau) - y^{[\rho]}(t + \varkappa\tau) - (y_d^{[\rho]}(t) - y^{[\rho]}(t)))$ and $0 < \varkappa < 1$. By Eq. (5.6), the performance index defined in Eq. (5.26) is rewritten as follows:

$$\begin{aligned} J(t) = & \frac{1}{2} \int_0^T \left(\sum_{i=0}^{\rho-1} \frac{\tau^i}{i!} (y_r^{[i]}(t) - y^{[i]}(t)) + \frac{\tau^\rho}{\rho!} y_r^\rho(t) - \frac{\tau^\rho}{\rho!} \sum_{j=1}^l w_{oj}(t) \sigma_j(\mathbf{w}_{ij}^T \mathbf{x}(t)) \right. \\ & \left. - \frac{\tau^\rho}{\rho!} \sum_{j=1}^l w'_{oj}(t) \sigma_j(\mathbf{w}_{ij}^T \mathbf{x}(t)) u(t) + \frac{\tau^\rho}{\rho!} \Delta(t) - \frac{\tau^\rho}{\rho!} \eta(t) \right)^2 d\tau, \end{aligned} \quad (5.26)$$

where $\eta(t)$ is defined as $\eta(t) = -(\sum_{j=1}^l \tilde{w}_{oj} \sigma_j(\mathbf{w}_{ij}^T \mathbf{x}(t)) + \sum_{j=1}^l \tilde{w}'_{oj} \sigma_j(\mathbf{w}_{ij}^T \mathbf{x}(t)) u(t))$. By the triangle inequality [101], from (5.26), one has

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

where

$$\begin{aligned} \bar{J}(t) &= \int_0^T \left(\sum_{i=0}^{\rho-1} \frac{\tau^i}{i!} (y_r^{[i]}(t) - y^{[i]}(t)) + \frac{\tau^\rho}{\rho!} y_r^\rho(t) - \frac{\tau^\rho}{\rho!} \sum_{j=1}^l w_{oj}(t) \sigma_j(\mathbf{w}_{ij}^T \mathbf{x}(t)) \right. \\ &\quad \left. - \frac{\tau^\rho}{\rho!} \sum_{j=1}^l w'_{oj}(t) \sigma_j(\mathbf{w}_{ij}^T \mathbf{x}(t)) u(t) \right)^2 d\tau \\ &= \int_0^T (\rho^T(\tau) \Gamma(t))^2 d\tau - 2\rho_{\text{int}} \Gamma(t) \sum_{j=1}^l w_{oj}(t) \sigma_j(\mathbf{w}_{ij}^T \mathbf{x}(t)) u(t) \\ &\quad + \frac{T^{2\rho+1}}{(2\rho+1)(\rho!)^2} \left(\sum_{j=1}^l w_{oj}(t) \sigma_j(\mathbf{w}_{ij}^T \mathbf{x}(t)) \right)^2 u^2(t) \geq 0 \end{aligned}$$

is a convex quadratic performance index with respect to decision variable $u(t)$; $\Gamma(t)$ is defined as $\Gamma(t) = [y_r(t) - y(t), \dot{y}_r(t) - \dot{y}(t), \dots, y_r^{[\rho-1]}(t) - y^{[\rho-1]}(t), y_r^{[\rho]}(t) - \sum_{j=1}^l w_{oj}(t) \sigma_j(\mathbf{w}_{ij}^T \mathbf{x}(t))]^T$; $\rho(\tau)$ is defined as $\rho(\tau) = [1, \tau, \tau^2/2, \dots, \tau^\rho/\rho!]^T$. In addition, ρ_{int} is defined in Eq.(5.11). 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.,

$$\rho_{\text{int}} \Gamma(t) \sigma_j(\mathbf{w}_{ij}^T \mathbf{x}(t)) + T^{2\rho+1} (\sigma_j(\mathbf{w}_{ij}^T \mathbf{x}(t)))^2 u(t) / ((2\rho+1)(\rho!)^2) = 0,$$

which yields

$$\begin{aligned} &- \sum_{i=0}^{\rho-1} \frac{T^i}{(\rho+1+i)i!} (y_r^{[i]}(t) - y^{[i]}(t)) - \frac{T^\rho}{(2\rho+1)\rho!} y_r^{[\rho]}(t) \\ &+ \frac{T^\rho}{(2\rho+1)\rho!} \left(\sum_{j=1}^l w_{oj}(t) \sigma_j(\mathbf{w}_{ij}^T \mathbf{x}(t)) + \sum_{j=1}^l w'_{oj}(t) \sigma_j(\mathbf{w}_{ij}^T \mathbf{x}(t)) u(t) \right) = 0. \quad (5.27) \end{aligned}$$

Evidently, adaptive near-optimal control law (5.16) is the solution of equation (5.27), i.e., control law (5.16) satisfies $\partial \bar{J}(t)/\partial u = 0$. It follows that $\bar{J}(t) = 0$ when control law (5.16) is adopted. Then, one has

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

From Theorem 5.2, if $\sum_{j=1}^l w'_{0j}(0)\sigma_j(\mathbf{w}_{ij}^T \mathbf{x}(0)) \geq d_0 > 0$ and all the roots of characteristic equation $\sum_{j=0}^{\rho-1} \alpha_j v^j = 0$ are located on the left half-plane, then $\lim_{t \rightarrow +\infty} \eta(t) = 0$. Together with the conditions in Theorem 5.2, with $\rho \in \{1, 2, 3, 4\}$, from Theorem 5.3, $\lim_{t \rightarrow +\infty} (y_r(t) - y(t)) = 0$. By the solution property of linear ordinary differential equations (see the proof of Theorem 5.2) and the extension of Barbalat's lemma [69], one further has $\lim_{t \rightarrow +\infty} (y_r^{[\rho]}(t) - y^{[\rho]}(t)) = 0$. It follows that $\lim_{t \rightarrow +\infty} \Delta(t) = 0$. Note that $J(t) \geq 0$. Then, by the pinching theorem [101], $\lim_{t \rightarrow +\infty} J(t) = 0$, i.e., performance index (5.3) asymptotically converges to optimal. In other words, control law (5.16) is asymptotically optimal. This completes the proof. \square

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 presented 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.

5.5 Illustrative Example

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

Consider the following nonlinear system:

$$\begin{cases} \dot{x}_1(t) = 2x_2(t) + x_1(t), \\ \dot{x}_2(t) = 4x_1^3(t)x_2(t) + \cos(x_2(t)) + \sin(x_1(t))x_2^2(t) + 7(\sin(x_2(t)) + 1.1)u(t), \\ y(t) = x_1(t), \end{cases} \quad (5.28)$$

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 Theorems 5.3 and 5.4. To implement the presented adaptive near-optimal tracking control 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 $y_d(t) = 0.5 \cos(0.6t) + 0.1$. 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 (5.13) are set to $\lambda = 2$, $\gamma = 5$, $\alpha_1 = 1$, and $\alpha_0 = 2$. Evidently, under this setup, the root of characteristic equation $\alpha_1 v + \alpha_0 = 0$ is $v = -2$, which satisfies the requirement stated in Theorem 5.2. 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 [44]. In terms of the setup of sigmoid neural network (5.17), c_j is set to $c_j = 0.0001$ for any $j = 1, 2, \dots, l$; each element of \mathbf{w}_{ij} 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 $\mathbf{x}(0) = [x_1(0), x_2(0)]^T$ of system (5.28) and $\hat{\mathbf{x}}(0) = [\hat{x}_1(0), \hat{x}_2(0)]^T$ of auxiliary system (5.13), without loss of generality, are set as $\mathbf{x}(0) = \hat{\mathbf{x}}(0) = [0, 0]^T$. In addition, each element of initial output-layer weight vectors $\mathbf{w}_o(0)$ and $\mathbf{w}'_o(0)$ of sigmoid neural network (5.17) is randomly set at interval $(0, 1)$. It is checked that $\mathbf{w}'_o(0)$ satisfies the condition stated in Theorem 5.1. Figure 5.2a shows the convergence of the input-to-output property of auxiliary system (5.13) to that of nonlinear system (5.28), which verifies Theorem 5.2 and also substantiates the efficacy of the presented sigmoid neural network based auxiliary system for dynamics reconstruction of unknown nonlinear systems. As seen from Fig. 5.2b and c, output $y(t)$ of system (5.28) successfully tracks reference output $y_r(t)$ with output error $e(t) = y(t) - y_r(t)$ and its derivatives converging to zero, which verifies Theorem 5.3. As seen from Fig. 5.2d, performance index $J(t)$ asymptotically converges to zero, which verifies Theorem 5.4. In addition, Fig. 5.2e and 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 (5.17) during the control process are shown in Fig. 5.3. 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. 5.4. As seen from Fig. 5.4a, with a smaller value of T , the performance index converges to zero faster. In addition, from Fig. 5.4b, 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 presented adaptive near-optimal tracking control method for nonlinear systems with fully unknown dynamics and verify the theoretical results.

5.6 Experimental Validation

The performance of the presented method is also tested via a low-cost DC motor system. DC motor systems are second-order and the output motor angle θ 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 θ is directly measured by the encoder. The angle velocity is obtained via the following tracking differentiator [86]:

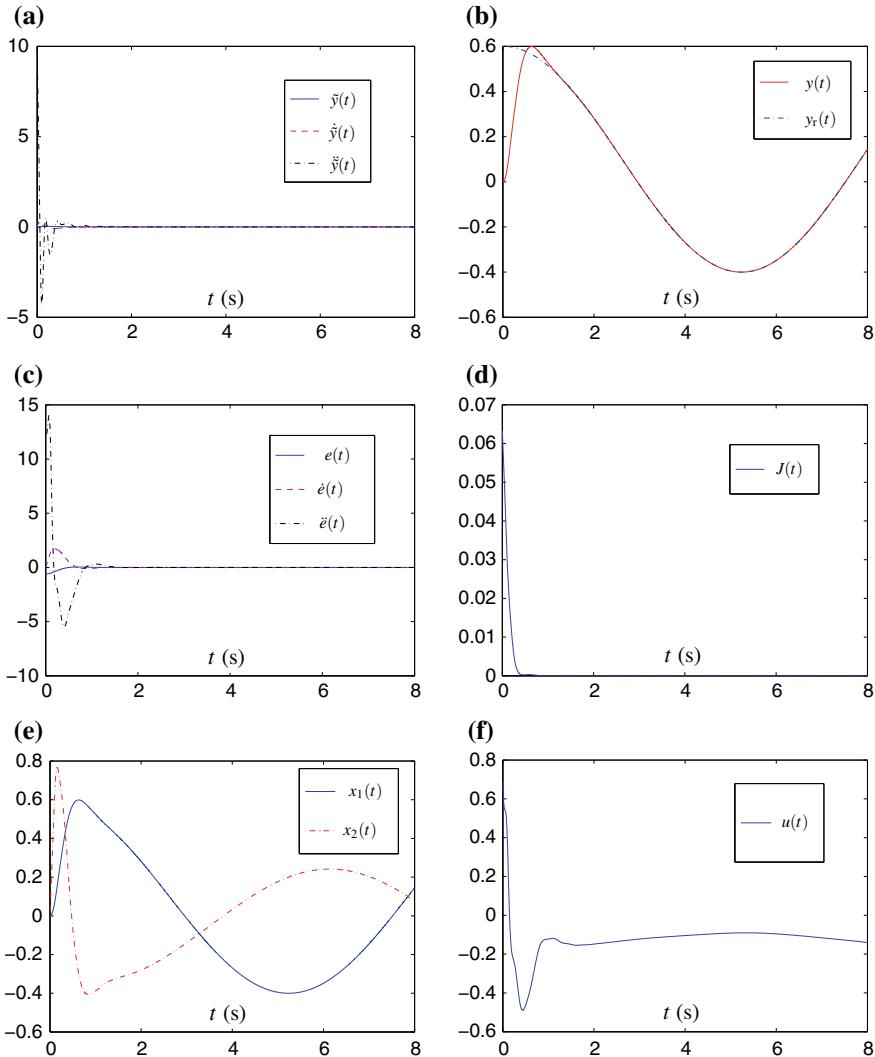
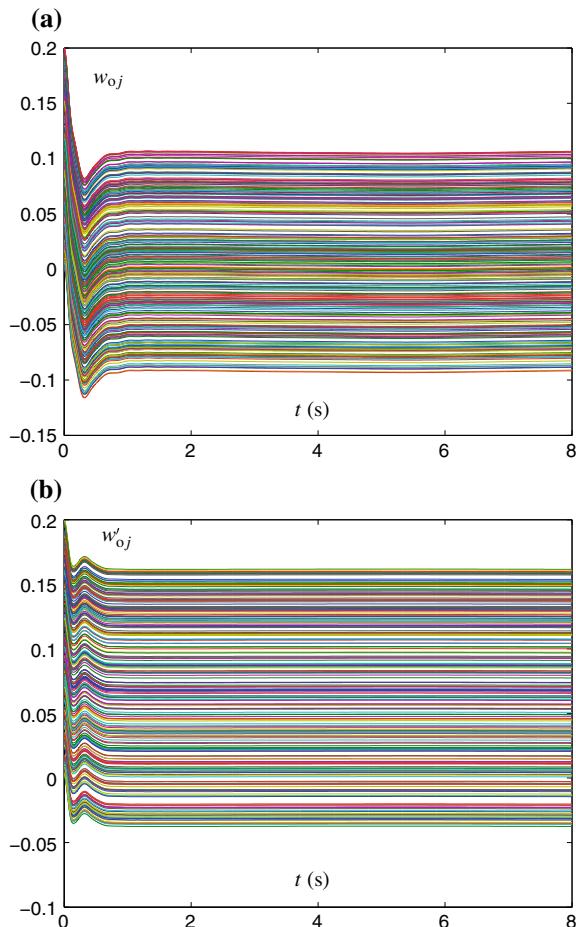


Fig. 5.2 Convergence of the input-to-output property of auxiliary system (5.13) to that of nonlinear system (5.28) and performance of nonlinear system (5.28) in tracking time-varying reference output $y_r(t)$ under the control of adaptive near-optimal control law (5.16). **a** Time profiles of $\tilde{y}(t)$, $\dot{\tilde{y}}(t)$, 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 (5.28) with $T = 0.3$ s. **e** Time profiles of state variables $x_1(t)$ and $x_2(t)$ of system (5.28). **f** Time profile of control input $u(t)$

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

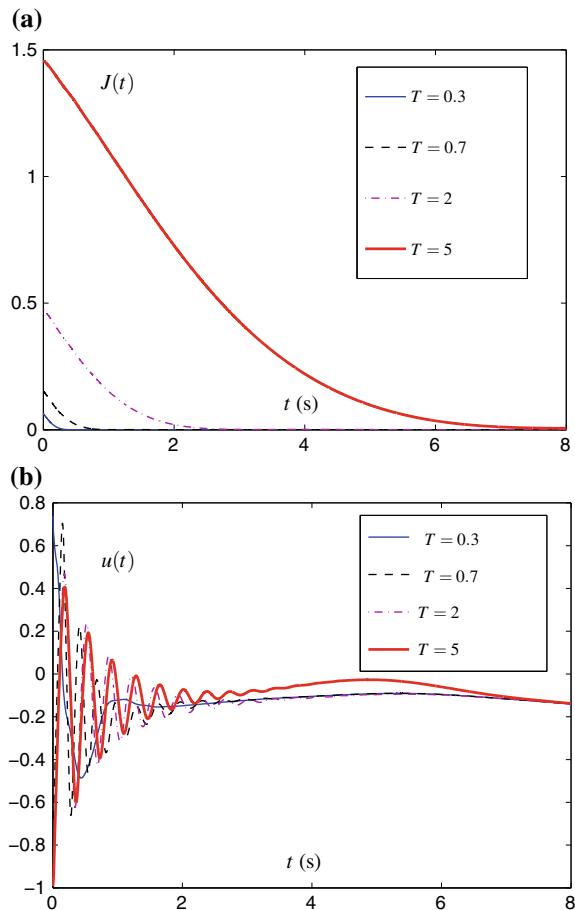
Fig. 5.3 Time profiles of neural network parameters during the control process of nonlinear system (5.28) via adaptive near-optimal control law (5.16). **a** Time profiles of $w_{oj}(t)$ with $j = 1, 2, \dots, l$. **b** Time profiles of $w'_{oj}(t)$ with $j = 1, 2, \dots, l$



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 $\dot{\theta}(t)$ infinite time given that design parameter $C > 0 \in \mathbb{R}$ is set to be large enough. The presented 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:

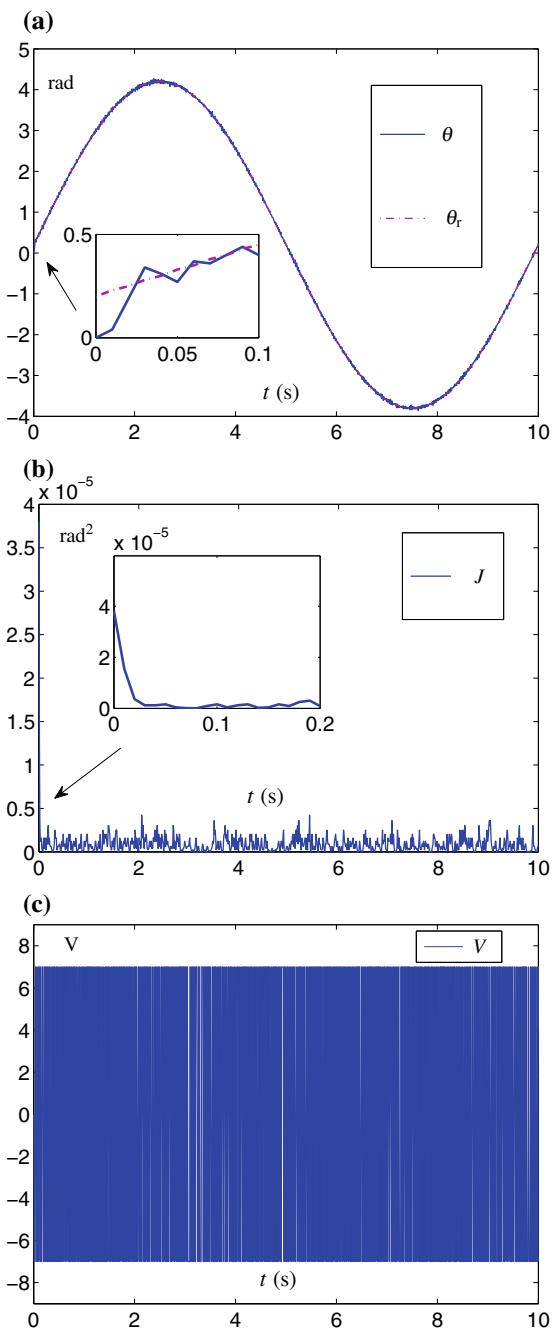
$$V(t) = \begin{cases} u^+, & \text{if } u(t) > u^+ \\ u(t), & \text{if } u^- \leq u(t) \leq u^+, \\ u^-, & \text{if } u(t) < u^-, \end{cases}$$

Fig. 5.4 Performance comparison of adaptive near-optimal control law (5.16) for the tracking control of nonlinear system (5.28) under different values of T in performance index $J(t)$ defined in Eq. (5.3).
a Time profiles of performance index $J(t)$.
b Time profiles of input $u(t)$ generated by the control law



where $u(t)$ is the input calculated by the presented controller (5.16) 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 $\theta_r = 4 \sin(\pi t/5) + 0.2$ rad. The other settings are the same as those in the simulative example. As seen from Fig. 5.5, by the presented method, the output of the DC motor successfully tracks the desired output and the value of the performance index is less than 5×10^{-6} at steady state. The experimental data shows that the average tracking error is 1.7110×10^{-4} rad.

Fig. 5.5 Data profiles for the motor control experiment by using the presented control method. **a** Time profiles of system output $\theta(t)$ and reference output $\theta_r(t)$. **b** Time profile of performance index $J(t)$. **c** Time profile of input voltage $V(t)$



In addition, the input given to the DC motor does not exceed the limit. The results further verify the efficiency and realizability of the presented method.

5.7 Questions and Answers

In this section, some discussions about earlier versions of the chapter are presented in the form of questions and answers.

Question 5.1 “*Comparing with an existing method, the main difference is that the unknown control gain has been taken into account. To clarify, it is suggested to add some remarks explain explicitly what difficulties the unknown control gain terms cause and how the proposed scheme overcomes these difficulties.*”

Answer: The strict-feedback nonlinear system considered in the existing method is a special case of the system considered in the manuscript. In other words, compared with the back stepping approach, the method used in the manuscript allows a more general system form. Besides, the control gain is assumed to be a known constant in the existing method, which is considered to be an unknown function of the state variables in this paper. 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 paper, a projection operator is used for the estimation of the control gain so as to avoid the problem.

Question 5.2 “*In Corollary 2, the accurate convergence of neural network parameters is closely related to the satisfaction of the persistent excitation (PE). Therefore, it is recommended to provide a definition of PE condition for ease of understanding the parameter convergence. Moreover, it is well known that the PE condition is very difficult to be satisfied, especially in nonlinear control. The referee has a doubt about how to make the terms with control inputs u satisfy the PE condition.*”

Answer: Firstly, we would like to point out that the theorems (main results) do not rely on Corollary 2, and thus the persistent excitation (PE) condition. This, as we consider, is also one of the differences of the proposed approach compared with some existing neural network-based adaptive control methods. We agree with the reviewer that it is difficult to verify the PE condition online. To some extent, the PE condition requires a signal to be sufficiently rich [R1]. In practice, the PE condition could be guaranteed when using a sufficiently rich reference trajectory for the control system.

Question 5.3 “*The theoretical proof of this work is correct. However, the authors are expected to point out the essential ideas behind this proof. What is the underlying principle of this work?*”

Answer: 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.

Question 5.4 “*This work aims to develop a receding-horizon near-optimal tracking control scheme for a class of continuous-time nonlinear systems with fully unknown dynamic. What is the reason that the control input has not been considered in the performance index $J(t)$? Furthermore, the constraints of the control input have also not been considered.*”

Answer: For the tracking control problem of the nonlinear system considered in the manuscript, when the following performance index is adopted:

$$J(t) = a \int_0^T (y_r(t + \tau) - y(t + \tau))^2 d\tau + b \int_0^T (u(t + \tau))^2 d\tau \quad (5.29)$$

with $a > 0 \in \mathbb{R}$ and $b > 0 \in \mathbb{R}$, the system stability cannot be guaranteed even when the system dynamics is fully known. A feasible performance index with the control input taken into account was also provided in the existing literature, which has the following form:

$$\begin{aligned} J(t) = & v(y_r(t + \tau) - y(t + \tau)) \\ & + a \int_0^T (y_r(t + \tau) - y(t + \tau))^2 d\tau + b \int_0^T (u_r(t + \tau) - u(t + \tau))^2 d\tau, \end{aligned} \quad (5.30)$$

where $v(\cdot)$ needs to be a continuous and differentiable function with $v(0) = 0$ and $v(x) > 0, \forall x \neq 0$; $u_r(t)$ is the input function with which the output of the system $\dot{\mathbf{x}} = f(\mathbf{x}) + g(\mathbf{x})u_r(t)$, $y(t) = h(\mathbf{x}(t))$ satisfies $y(t) = y_r(t)$. Evidently, the physical meaning of such a performance index is not clear and it requires the knowledge of $u_r(t)$. Actually, under system uncertainty, it is more difficult to obtain $u_r(t)$. In practice, using the performance index given in our manuscript means that the control accuracy is far more important than the energy consumption.

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 $u_a(t)$ denotes the actual input given to the controlled system. Then, by setting

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

with $u(t)$ being the input calculated based on the proposed control law, then the input constraint is always satisfied.

Question 5.5 “Although the fully unknown dynamic has been considered, the time-varying disturbance has not been considered in the developed method. At the same time, the time-varying disturbance cannot be tackled by using the neural network.”

Response: Regarding time-varying disturbance, disturbance observers could be adopted to estimate the disturbance, by which we may have a composite controller being the sum of the proposed controller and a compensate controller for the disturbance. This is the general intuition to enhance the disturbance rejection capability of a controller. Considering the page limit and the difficulty in the overall stability analysis, this enhancement of the proposed method will be our future work.

Question 5.6 “The fully unknown dynamic has been considered in this work. In such a case, how to decide the relative degree of the studied system?”

Answer: In practice, although the system dynamics are fully unknown, we generally have 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 prior knowledge is not available, we may use the controller by a trial and error method. Specifically, we may first 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.

Question 5.7 “From the viewpoint of control engineering, only experimental results can be used to verify the effectiveness of the proposed control scheme in practical applications. The authors should try to provide some practical results to verify the effectiveness.”

Answer: As suggested, we have tested the performance of the proposed method in a low-cost DC motor system. DC motor systems are second-order and the output motor angle θ have 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 θ is directly measured by the encoder. The angle velocity is obtained via the following tracking differentiator:

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

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 $\dot{\theta}(t)$ infinite 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 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 $\gamma = 3$, $\lambda = 3$, $\alpha_0 = 2$, $\alpha_1 = 1$. The parameters of the sigmoid basis, functions are set the same as those in the simulative example in the manuscript except that, owing to the computational capability of the Arduino UNO board, we only use ten neurons, 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 $\theta_r = 4 \sin(\pi t/5) + 0.2$ rad. The other settings are the same as those in the simulative example. 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 5×10^{-6} at the steady state. The experimental data shows that the average tracking error is 1.7110×10^{-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.

5.8 Summary

In this chapter, a novel model-free adaptive near-optimal tracking control method has been presented for a class of continuous-time nonlinear systems with fully unknown system dynamics. The presented 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 presented 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. Future research directions about this work include but are not limited to the extension to nonlinear systems with multiple inputs and multiple outputs, and the enhancement of the presented method by incorporating the disturbance observer technique [97–99].

References

1. Liu, Y.J., Tong, S.: Barrier Lyapunov functions-based adaptive control for a class of nonlinear pure-feedback systems with full state constraints. *Automatica* **64**, 70–75 (2016)
2. Wang, H., Shi, P., Li, H., Zhou, Q.: Adaptive neural tracking control for a class of nonlinear systems with dynamic uncertainties. *IEEE Trans. Cybern.* **47**(10), 3075–3087 (2017)
3. Zhou, Q., Shi, P., Tian, Y., Wang, M.: Approximation-based adaptive tracking control for MIMO nonlinear systems with input saturation. *IEEE Trans. Cybern.* **45**(10), 2119–2128 (2015)
4. Zhou, Q., Li, H., Wu, C., Wang, L., Ahn, C.K.: Adaptive fuzzy control of nonlinear systems with unmodeled dynamics and input saturation using small-gain approach **47**(8), 1979–1989 (2017)
5. Li, S., Chen, S., Liu, B., Li, Y., Liang, Y.: Decentralized kinematic control of a class of collaborative redundant manipulators via recurrent neural networks. *Neurocomputing* **91**, 1–10 (2012)
6. Li, S., Cui, H., Li, Y., Liu, B., Lou, Y.: Decentralized control of collaborative redundant manipulators with partial command coverage via locally connected recurrent neural networks. *Neural Comput. Appl.* **23**(3), 1051–1060 (2013)
7. Jin, L., Zhang, Y., Li, S., Zhang, Y.: Modified ZNN for time-varying quadratic programming with inherent tolerance to noises and its application to kinematic redundancy resolution of robot manipulators. *IEEE Trans. Ind. Electron.* **63**(11), 6978–6988 (2016)
8. Li, S., He, J., Li, Y., Rafique, M.U.: Distributed recurrent neural networks for cooperative control of manipulators: a game-theoretic perspective. *IEEE Trans. Neural Netw. Learn. Syst.* **28**(2), 415–426 (2017)
9. Jin, L., Li, S., La, H.M., Luo, X.: Manipulability optimization of redundant manipulators using dynamic neural networks. *IEEE Trans. Ind. Electron.* **64**(6), 4710–4720 (2017)
10. Li, Y., Li, S., Hannaford, B.: A novel recurrent neural network for improving redundant manipulator motion planning completeness. *ICRA*, 2956–2961 (2018)
11. Zhang, Y., Li, S.: A neural controller for image-based visual servoing of manipulators with physical constraints. *IEEE Trans. Neural Netw. Learn. Syst.* **29**(11), 5419–5429 (2018)
12. Li, S., Zhou, M., Luo, X.: Modified primal-dual neural networks for motion control of redundant manipulators with dynamic rejection of harmonic noises. *IEEE Trans. Neural Netw. Learn. Syst.* **29**(10), 4791–4801 (2018)
13. Li, S., Wang, H., Rafique, M.U.: A novel recurrent neural network for manipulator control with improved noise tolerance. *IEEE Trans. Neural Netw. Learn. Syst.* **29**(5), 1908–1918 (2018)
14. Jin, L., Li, S., Luo, X., Li, Y., Qin, B.: Neural dynamics for cooperative control of redundant robot manipulators. *IEEE Trans. Ind. Inform.* **14**(9), 3812–3821 (2018)
15. Li, J., Zhang, Y., Li, S., Mao, M.: New discretization-formula-based zeroing dynamics for real-time tracking control of serial and parallel manipulators. *IEEE Trans. Ind. Inform.* **14**(8), 3416–3425 (2018)
16. Chen, D., Zhang, Y., Li, S.: Tracking control of robot manipulators with unknown models: a jacobian-matrix-adaption method. *IEEE Trans. Ind. Inform.* **14**(7), 3044–3053 (2018)
17. Zhang, Y., Li, S., Gui, J., Luo, X.: Velocity-level control with compliance to acceleration-level constraints: a novel scheme for manipulator redundancy resolution. *IEEE Trans. Ind. Inform.* **14**(3), 921–930 (2018)
18. Xiao, L., Liao, B., Li, S., Zhang, Z., Ding, L., Jin, L.: Design and analysis of FTZNN applied to the real-time solution of a nonstationary Lyapunov equation and tracking control of a wheeled mobile manipulator. *IEEE Trans. Ind. Inform.* **14**(1), 98–105 (2018)
19. Zhang, Y., Chen, S., Li, S., Zhang, Z.: Adaptive projection neural network for kinematic control of redundant manipulators with unknown physical parameters. *IEEE Trans. Ind. Electron.* **65**(6), 4909–4920 (2018)
20. Zhang, Z., Lin, Y., Li, S., Li, Y., Yu, Z., Luo, Y.: Tricriteria optimization-coordination motion of dual-redundant-robot manipulators for complex path planning. *IEEE Trans. Contr. Sys. Techn.* **26**(4), 1345–1357 (2018)

21. Jin, L., Li, S., Hu, B., Yi, C.: Dynamic neural networks aided distributed cooperative control of manipulators capable of different performance indices. *Neurocomputing* **291**, 50–58 (2018)
22. Jin, L., Li, S., Yu, J., He, J.: Robot manipulator control using neural networks: a survey. *Neurocomputing* **285**, 23–34 (2018)
23. Chen, D., Zhang, Y., Li, S.: Zeroing neural-dynamics approach and its robust and rapid solution for parallel robot manipulators against superposition of multiple disturbances. *Neurocomputing* **275**, 845–858 (2018)
24. Li, S., Shao, Z., Guan, Y.: A dynamic neural network approach for efficient control of manipulators. *IEEE Trans. Syst., Man, Cybern., Syst.* **49**(5), 932–941 (2019)
25. Zhang, Y., Li, S., Zhou, X.: Recurrent-neural-network-based velocity-level redundancy resolution for manipulators subject to a joint acceleration limit. *IEEE Trans. Ind. Electron.* **66**(5), 3573–3582 (2019)
26. Zhang, Z., Chen, S., Li, S.: Compatible convex-nonconvex constrained QP-based dual neural networks for motion planning of redundant robot manipulators. *IEEE Trans. Contr. Sys. Techn.* **27**(3), 1250–1258 (2019)
27. Xu, Z., Li, S., Zhou, X., Yan, W., Cheng, T., Huang, D.: Dynamic neural networks based kinematic control for redundant manipulators with model uncertainties. *Neurocomputing* **329**, 255–266 (2019)
28. Li, S., Zhang, Y., Jin, L.: Kinematic control of redundant manipulators using neural networks. *IEEE Trans. Neural Netw. Learn. Syst.* in press
29. Mohammed, A.M., Li, S.: Dynamic neural networks for kinematic redundancy resolution of parallel stewart platforms. *IEEE Trans. Cybern.* **46**(7), 1538–1550 (2016)
30. Jin, L., Zhang, Y.: G2-type SRMPC scheme for synchronous manipulation of two redundant robot Arms. *IEEE Trans. Cybern.* **45**(2), 153–164 (2015)
31. Jin, L., Li, S., Xiao, L., Lu, R., Liao, B.: Cooperative motion generation in a distributed network of redundant robot manipulators with noises. *IEEE Trans. Syst., Man, Cybern., Syst.* **48**(10), 1715–1724 (2018)
32. Jin, L., Li, S.: Distributed task allocation of multiple robots: a control perspective. *IEEE Trans. Syst., Man, Cybern., Syst.* **48**(5), 693–701 (2018)
33. Jin, L., Li, S., Hu, B., Liu, M.: A survey on projection neural networks and their applications. *Appl. Soft Comput.* **76**, 533–544 (2019)
34. Xiao, L., Li, K., Tan, Z., Zhang, Z., Liao, B., Chen, K., Jin, L., Li, S.: Nonlinear gradient neural network for solving system of linear equations. *Inf. Process. Lett.* **142**, 35–40 (2019)
35. Xiang, Q., Liao, B., Xiao, L., Lin, L., Li, S.: Discrete-time noise-tolerant Zhang neural network for dynamic matrix pseudoinversion. *Soft Comput.* **23**(3), 755–766 (2019)
36. Xiao, L., Li, S., Yang, J., Zhang, Z.: A new recurrent neural network with noise-tolerance and finite-time convergence for dynamic quadratic minimization. *Neurocomputing* **285**, 125–132 (2018)
37. Xiao, L., Liao, B., Li, S., Chen, K.: Nonlinear recurrent neural networks for finite-time solution of general time-varying linear matrix equations. *Neural Netw.* **98**, 102–113 (2018)
38. Xiao, L., Zhang, Z., Zhang, Z., Li, W., Li, S.: Design, verification and robotic application of a novel recurrent neural network for computing dynamic Sylvester equation. *Neural Netw.* **105**, 185–196 (2018)
39. Zhang, Z., Lu, Y., Zheng, L., Li, S., Yu, Z., Li, Y.: A new varying-parameter convergent-differential neural-network for solving time-varying convex QP problem constrained by linear-equality. *IEEE Trans. Automat. Contr.* **63**(12), 4110–4125 (2018)
40. Jin, L., Li, S.: Nonconvex function activated zeroing neural network models for dynamic quadratic programming subject to equality and inequality constraints. *Neurocomputing* **267**, 107–113 (2017)
41. Jin, L., Li, S., Liao, B., Zhang, Z.: Zeroing neural networks: a survey. *Neurocomputing* **267**, 597–604 (2017)
42. Mao, M., Li, J., Jin, L., Li, S., Zhang, Y.: Enhanced discrete-time Zhang neural network for time-variant matrix inversion in the presence of bias noises. *Neurocomputing* **207**, 220–230 (2016)

43. Jin, L., Zhang, Y., Li, S.: Integration-enhanced Zhang neural network for real-time-varying matrix inversion in the presence of various kinds of noises. *IEEE Trans. Neural Netw. Learn. Syst.* **27**(12), 2615–2627 (2016)
44. Li, S., Li, Y.: Nonlinearly activated neural network for solving time-varying complex sylvester equation. *IEEE Trans. Cybern.* **44**(8), 1397–1407 (2014)
45. Li, S., Li, Y., Wang, Z.: A class of finite-time dual neural networks for solving quadratic programming problems and its k-winners-take-all application. *Neural Netw.* **39**, 27–39 (2013)
46. Na, J., Ren, X., Zheng, D.: Adaptive control for nonlinear pure-feedback systems with high-order sliding mode observer. *IEEE Trans. Neural Netw. Learn. Syst.* **24**(3), 370–382 (2013)
47. Kayacan, E., Kayacan, E., Ramon, H., Saeys, W.: Adaptive neuro-fuzzy control of a spherical rolling robot using sliding-mode-control-theory-based online learning algorithm. *IEEE Trans. Cybern.* **43**(1), 170–179 (2013)
48. Shen, Q., Shi, P., Shi, Y.: Distributed adaptive fuzzy control for nonlinear multiagent systems via sliding mode observers. *IEEE Trans. Cybern.* **46**(12), 3086–3097 (2016)
49. Liu, Y.J., Tong, S.C., Wang, D., Li, T.S., Chen, C.L.P.: Adaptive neural output feedback controller design with reduced-order observer for a class of uncertain nonlinear SISO systems. *IEEE Trans. Neural Netw.* **22**(8), 1328–1334 (2011)
50. Dai, S.L., Wang, M., Wang, C.: Neural learning control of marine surface vessels with guaranteed transient tracking performance. *IEEE Trans. Ind. Electron.* **63**(3), 1717–1727 (2016)
51. Kostarigka, A.K., Rovithakis, G.A.: Adaptive dynamic output feedback neural network control of uncertain MIMO nonlinear systems with prescribed performance. *IEEE Trans. Neural Netw. Learn. Syst.* **23**(1), 138–149 (2012)
52. El-Ferik, S., Qureshi, A., Lewis, F.L.: Neuro-adaptive cooperative tracking control of unknown higher-order affine nonlinear systems. *Automatica* **50**(3), 798–808 (2014)
53. Zhou, Q., Li, H., Shi, P.: Decentralized adaptive fuzzy tracking control for robot finger dynamics. *IEEE Trans. Fuzzy Syst.* **23**(3), 501–510 (2015)
54. Yang, C., Wang, X., Cheng, L., Ma, H.: Neural-learning-based telerobot control with guaranteed performance. *IEEE Trans. Cybern.* **47**(10), 3148–3159 (2017)
55. He, W., Yan, Z., Sun, C., Chen, Y.: Adaptive neural network control of a flapping wing micro aerial vehicle with disturbance observer. *IEEE Trans. Cybern.* **47**(10), 3452–3465 (2017)
56. Cheng, L., Liu, W., Yang, C., Huang, T., Hou, Z., Tan, M.: A neural-network-based controller for piezoelectric-actuated stick-slip devices. *IEEE Trans. Ind. Electron.* **65**(3), 2598–2607 (2018)
57. Chen, W.H., Ballance, D.J., Gawthrop, P.J.: Optimal control of nonlinear systems: a predictive control approach. *Automatica* **39**(4), 633–641 (2003)
58. Chen, W.H., Ballance, D.J., Gawthrop, P.J., Gribble, J.J., Reill, J.O.: Nonlinear PID predictive controller. *Control Theory Appl.* **146**(6), 603–611 (1999)
59. Liu, C., Chen, W.H., Andrews, J.: Tracking control of small-scale helicopters using explicit nonlinear MPC augmented with disturbance observers. *Control Eng. Pract.* **20**(3), 258–268 (2012)
60. Errouissi, R., Al-Durra, A., Muyeen, S.M.: Design and implementation of a nonlinear PI predictive controller for a grid-tied photovoltaic inverter. *IEEE Trans. Ind. Electron.* **64**(2), 1241–1250 (2017)
61. Errouissi, R., Ouhrouche, M., Chen, W.H., Trzynadlowski, A.M.: Robust nonlinear predictive controller for permanent-magnet synchronous motors with an optimized cost function. *IEEE Trans. Ind. Electron.* **59**(7), 2849–2858 (2011)
62. Liu, Y.J., Tong, S.: Optimal control-based adaptive NN design for a class of nonlinear discrete-time block-triangular systems. *IEEE Trans. Cybern.* **46**(11), 2670–2680 (2016)
63. Liu, Y.J., Gao, Y., Tong, S., Li, Y.: Fuzzy approximation-based adaptive backstepping optimal control for a class of nonlinear discrete-time systems with dead-zone. *IEEE Trans. Fuzzy Syst.* **24**(1), 16–28 (2016)
64. Sheu, J.W., Lin, W.S.: Adaptive optimal control for designing automatic train regulation for metro line. *IEEE Trans. Control Syst. Technol.* **20**(5), 1319–1327 (2012)
65. Heydari, A., Balakrishnan, S.N.: Finite-horizon control-constrained nonlinear optimal control using single network adaptive critics. *IEEE Trans. Neural Netw. Learn. Syst.* **24**(1), 145–157 (2013)

66. Sun, K., Li, Y., Tong, S.: Fuzzy adaptive output feedback optimal control design for strict-feedback nonlinear systems. *IEEE Trans. Syst. Man Cybern. Syst.* **47**(1), 33–44 (2017)
67. Huang, Y.: Neuro-observer based online finite-horizon optimal control for uncertain non-linear continuous-time systems. *IET Control Theory Appl.* **11**(3), 401–410 (2017)
68. Dutta, S., Patchaikani, P.K., Behera, L.: Near-optimal controller for nonlinear continuous-time systems with unknown dynamics using policy iteration. *IEEE Trans. Neural Netw. Learn. Syst.* **27**(7), 1537–1549 (2015)
69. Wang, Y., Miao, Z., Zhong, H., Pan, Q.: Simultaneous stabilization and tracking of nonholonomic mobile robots: a Lyapunov-based approach. *IEEE Trans. Control Syst. Technol.* **23**(4), 1440–1450 (2015)
70. Zhang, Y., Yu, X., Yin, Y., Peng, C., Fan, Z.: Singularity-conquering ZG controllers of z2g1 type for tracking control of the IPC system. *Int. J. Control* **87**(9), 1729–1746 (2014)
71. Baek, J., Jin, M., Han, S.: A new adaptive sliding-mode control scheme for application to robot manipulators. *IEEE Trans. Ind. Electron.* **63**(5), 3632–3637 (2016)
72. Zhang, Y., Chen, D., Jin, L., Zhang, Y., Yin, Y.: GD-aided IOL (input-output linearisation) controller for handling affine-form nonlinear system with loose condition on relative degree. *Int. J. Control* **89**(4), 757–769 (2016)
73. Zhang, Y., Tao, G., Chen, M.: Adaptive neural network based control of noncanonical nonlinear systems. *IEEE Trans. Neural Netw. Learn. Syst.* **27**(9), 1864–1877 (2016)
74. Davila, J.: Exact tracking using backstepping control design and high-order sliding modes. *IEEE Trans. Autom. Control* **58**(8), 2077–2081 (2013)
75. Castaneda, H., Plestan, F., Chriette, A., León-Morales, J.D.: Continuous differentiator based on adaptive second-order sliding-mode control for a 3-DOF helicopter. *IEEE Trans. Ind. Electron.* **63**(9), 5786–5793 (2016)
76. Isidori, A.: Nonlinear Control Systems: An Introduction, 3rd edn. Springer, New York (1995)
77. Gu, D., Hu, H.: Receding horizon tracking control of wheeled mobile robots. *IEEE Trans. Control Syst. Technol.* **14**(4), 743–749 (2016)
78. Cybenko, G.: Approximations by superpositions of a sigmoidal function. *Math. Control Signals Syst.* **2**(4), 303–314 (1989)
79. Huang, G.B., Zhu, Q.Y., Siew, C.K.: Extreme learning machine: theory and applications. *Neurocomputing* **70**, 489–501 (2006)
80. Zhang, Z., Zhang, Y.: Acceleration-level cyclic-motion generation of constrained redundant robots tracking different paths. *IEEE Trans. Syst., Man, Cybern. B, Cybern.* **42**(4), 1257–1269 (2012)
81. Hassine, I.M.B., Naouar, M.W., Mrabet-Bellaaj, N.: Model predictive-sliding mode control for three-phase grid-connected converters. *IEEE Trans. Ind. Electron.* **64**(2), 1341–1349 (2017)
82. Chen, C.T.: Linear System Theory And Design. Oxford University Press, New York (1995)
83. Jin, L., Zhang, Y., Li, S., Zhang, Y.: Noise-tolerant ZNN models for solving time-varying zero-finding problems: a control-theoretic approach. *IEEE Trans. Autom. Control* **62**(2), 992–997 (2017)
84. Joya, G., Atencia, M.A., Sandoval, F.: Hopfield neural networks for optimization: study of the different dynamics. *Neurocomputing* **43**, 219–237 (2002)
85. Cichocki, A., Unbehauen, R.: Robust neural networks with on-line learning for blind identification and blind separation of sources. *IEEE Trans. Circuits Syst. I, Fundam. Theory Appl.* **43**(11), 894–906 (1996)
86. Levant, A.: Robust exact differentiation via sliding mode technique. *Automatica* **34**(3), 379–384 (1998)
87. Guo, B.Z., Zhao, Z.L.: On convergence of tracking differentiator. *Int. J. Control* **84**(4), 693–701 (2011)
88. Tian, D., Shen, H., Dai, M.: Improving the rapidity of nonlinear tracking differentiator via feedforward. *IEEE Trans. Ind. Electron.* **61**(7), 3736–3743 (2014)
89. He, W., He, X., Ge, S.S.: Vibration control of flexible marine riser systems with input saturation. *IEEE/ASME Trans. Mechatron.* **21**(1), 254–265 (2016)

90. Yu, J., Zhao, L., Yu, H., Lin, C., Dong, W.: Fuzzy finite-time command filtered control of nonlinear systems with input saturation. *IEEE Trans. Cybern.* **48**(8), 2378–2387 (2018)
91. Jiang, B., Hu, Q., Friswell, M.I.: Fixed-time attitude control for rigid spacecraft with actuator saturation and faults. *IEEE Trans. Control Syst. Technol.* **24**(5), 1892–1898 (2016)
92. Boyd, S., Sastry, S.S.: Necessary and sufficient conditions for parameter convergence in adaptive control. *Automatica* **22**(6), 629–639 (1986)
93. Chen, W., Wen, C., Hua, S., Sun, C.: Distributed cooperative adaptive identification and control for a group of continuous-time systems with a cooperative PE condition via consensus. *IEEE Trans. Autom. Control* **59**(1), 91–106 (2014)
94. Dixon, W.E., Dawson, D.M., Zhang, F., Zergeroglu, E.: Global exponential tracking control of a mobile robot system via a PE condition. *IEEE Trans. Syst., Man, Cybern. B, Cybern.* **30**(1), 129–142 (2000)
95. Münz, U., Papachristodoulou, A., Allgöwer, F.: Robust consensus controller design for nonlinear relative degree two multi-agent systems with communication constraints. *IEEE Trans. Autom. Control* **56**(1), 145–151 (2011)
96. Loría, A.: Observers are unnecessary for output-feedback control of lagrangian systems. *IEEE Trans. Autom. Control* **61**(4), 905–920 (2016)
97. Chen, W., Ballance, D.J., Gawthrop, P.J., O'Reilly, J.: A nonlinear disturbance observer for robotic manipulators. *IEEE Trans. Ind. Electron.* **47**(4), 932–938 (2000)
98. Chen, W.: Disturbance observer based control for nonlinear systems. *IEEE/ASME Trans. Mechatron.* **9**(4), 706–710 (2004)
99. Yang, J., Li, S., Yu, X.: Sliding-mode control for systems with mismatched uncertainties via a disturbance observer. *IEEE Trans. Ind. Electron.* **60**(1), 160–169 (2013)
100. Khalil, H.: *Nonlinear Systems*. Prentice Hall, New Jersey (1996)
101. Nagle, R.K., Saff, E., Snider, A.: *Fundamentals of Differential Equations*, 7th edn. Pearson Addison Wesley, Boston (2008)

Chapter 6

Adaptive Kinematic Control of Redundant Manipulators



Abstract 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 adaptive control techniques for simultaneous parameter identification and neural control. However, the inherent nonlinearity and non-smoothness of the neural model prohibits direct applications of adaptive control to this model and there has been no existing result on adaptive control 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 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 presented 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 presented 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 presented neural network, and verify the theoretical results.

6.1 Introduction

Manipulators are said to be redundant if they have more degrees of freedom (DOF) than the required to achieve a given effector primary task [1]. 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 of achieving additional objectives, such as joint physical limit avoidance

[2], obstacle avoidance [3], and singularity avoidance [4]. 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 [5]. 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 [6]. Pseudoinverse-type methods [7–10] 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 [11] and the computational intensity in performing pseudoinversion [12].

To overcome the drawbacks of pseudoinverse-type methods, quadratic program (QP)-based methods have been developed and widely investigated [11–17]. Neural network-based methods are widely adopted to solve various scientific and engineering problems [18–30]. 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 [15], 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. [16] 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. [17] 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.

While extensive progress (e.g., [31–33] and the references therein) has been made to address the kinematic uncertainty of nonredundant 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 [5–17]. In the presence of uncertainty, these methods may result in errors or even unstable responses in the motion of end effectors. In [34], Cheah et al. adopted a unified framework to address the kinematic uncertainty of both redundant and nonredundant 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 [35], which is a one-to-many relationship [36], it is difficult to directly handle kinematic uncertainty via utilizing existing results in adaptive control, e.g., [37–40].

In this chapter, we present 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 [5–17], in this chapter, some parameters associated with the Jacobian matrix are assumed

to be unknown, i.e., the Jacobian matrix is uncertain. The presented method is capable of simultaneously identifying the Jacobian matrix and handling both parameters 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 are no existing methods for online simultaneous parameter learning and redundancy resolution, except our work presented in this chapter. In addition, the presented 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:

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

where $\mathbf{r}(t) = [r_1(t), r_2(t), \dots, r_m]^T \in \mathbb{R}^m$ and $\theta(t) = [\theta_1(t), \theta_2(t), \dots, \theta_n(t)]^T \in \mathbb{R}^n$ 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 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 [41].

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:*

$$J(\theta(t)) = W\phi(\theta(t)), \quad (6.2)$$

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 sides 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:

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

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 [41].

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 [5–17].

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:

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

$$s.t. \quad J(\theta(t))\mathbf{u}(t) = \dot{\mathbf{r}}_d(t) + \zeta(\mathbf{r}_d(t) - f(\theta(t))), \quad (6.5)$$

$$J(\theta(t)) = W\phi(\theta(t)), \quad (6.6)$$

$$\mathbf{u}(t) \in \Omega, \quad (6.7)$$

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; $\mathbf{r}_d(t) \in \mathbb{R}^m$ 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 (6.4), we offer the following remark.

Remark 6.1 The performance index shown in (6.4) 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 [1, 3, 15, 42, 43] 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 [5, 14, 17] 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 [5, 14, 17], the only requirement on constant parameter γ 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 $\gamma^2(\theta(t) - \theta(0))^T(\theta(t) - \theta(0))/2$ is directly removed from the performance index, which does not affect the optimality of the solution [5, 14,

[17]. When A equals to the inertial matrix of the manipulator and $\mathbf{b} = 0$, performance index (6.4) serves as the kinematic energy performance index investigated in [15].

About the derivation of the equality constraint shown in (6.5), we offer the following remark.

Remark 6.2 In the angle level, the equality constraint is

$$f(\theta(t)) = \mathbf{r}_d(t), \quad (6.8)$$

when the end effector is expected to track a smooth desired path defined by $\mathbf{r}_d(t)$. There are two approaches in the existing literature to deriving the relationship between the desired path $\mathbf{r}_d(t)$ and the joint angle vector $\theta(t)$ at the velocity level. The first one directly computes time derivatives on both sides of (6.8), which yields [1]

$$\dot{\mathbf{r}}_d(t) = J(\theta(t))\dot{\theta}(t). \quad (6.9)$$

Velocity-level redundancy resolution methods based on (6.9) 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 $\mathbf{e}(t) = \mathbf{r}(t) - \mathbf{r}_d(t)$ to zero and is thus less favorable [44]. 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 [2]. Let $\beta(t) = \mathbf{r}_d(t) - f(\theta(t))$. Then, by using the formula, the equality constraint (6.5) is derived.

About the set constraint shown in (6.7), we offer the following remark.

Remark 6.3 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 θ^- and θ^+ 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 [15], the joint angle limit can be incorporated into the set constraint by setting $\eta^- = \max\{\dot{\theta}_{\min}, -k(\theta - \theta_{\min})\}$ and $\eta^+ = \min\{\dot{\theta}_{\max}, -k(\theta - \theta_{\max})\}$ 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., [5–15]) with the aid of the Karush–Kuhn–Tucker condition [45].

The nominal design process can be divided into three steps. First, a Lagrange function is defined: $L(\theta, \lambda) = \mathbf{u}^T A \mathbf{u} / 2 + \mathbf{b}^T \mathbf{u} + \lambda^T (\dot{\mathbf{r}}_d - W\phi(\theta)\mathbf{u} + \zeta(\mathbf{r}_d - f(\theta)))$ where $\lambda \in \mathbb{R}^m$ is called the Lagrange multiplier. Second, the Karush–Kuhn–Tucker condition [45] about the optimal solution of problem (6.4) is written as follows:

$$\begin{aligned}\mathbf{u} &= P_{\Omega} \left(\mathbf{u} - \frac{\partial L}{\partial \mathbf{u}} \right), \\ W\phi(\theta)\mathbf{u} &= \dot{\mathbf{r}}_d + \zeta(\mathbf{r}_d - f(\theta)),\end{aligned}\tag{6.10}$$

where $P_{\Omega}(\cdot)$ is a projection function defined as $P_{\Omega}(\mathbf{u}) = \arg \min_{\mathbf{y} \in \Omega} \|\mathbf{u} - \mathbf{y}\|^2$. Note that $\partial L / \partial \theta = A\mathbf{u} + \mathbf{b} - \phi^T(\theta)W^T\lambda$. Then, a projection neural network (PNN) can be designed as follows to solve (6.4) [46]:

$$\begin{aligned}\varepsilon \dot{\mathbf{u}} &= -\mathbf{u} + P_{\Omega} \left(\mathbf{u} - \frac{\partial L}{\partial \mathbf{u}} \right) \\ &= -\mathbf{u} + P_{\Omega}(\mathbf{u} - (A\mathbf{u} + \mathbf{b} - \phi^T(\theta)W^T\lambda)), \\ \varepsilon \dot{\lambda} &= \dot{\mathbf{r}}_d - W\phi(\theta)\mathbf{u} + \zeta(\mathbf{r}_d - \mathbf{r}),\end{aligned}\tag{6.11}$$

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 [46] that the state trajectory of the PNN is exponentially convergent to the optimal solution of (6.4).

About extending PNN (6.11) to solve the problem considered in this chapter, we offer the following remark.

Remark 6.4 The design of PNN (6.11) is based on the ideal condition that the Jacobian matrix $J(\theta) = W\phi(\theta)$ is known. Extending PNN (6.11) 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.

6.4 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 presented adaptive PNN for solving redundancy resolution problem (6.4) with unknown constant matrix W is described as follows:

$$\varepsilon \dot{\check{\mathbf{u}}} = -\check{\mathbf{u}} + P_{\Omega}(\check{\mathbf{u}} - (A\check{\mathbf{u}} + \mathbf{b} - \phi^T(\theta)\hat{W}^T\lambda)), \quad (6.12)$$

$$\varepsilon \dot{\lambda} = \dot{\mathbf{r}}_d - \hat{W}\phi(\theta)\check{\mathbf{u}} + \zeta(\mathbf{r}_d - \mathbf{r}), \quad (6.13)$$

$$\dot{\hat{W}} = -\nu(\hat{W}\phi(\theta)\mathbf{u} - \dot{\mathbf{r}})\mathbf{u}^T\phi^T(\theta), \quad (6.14)$$

$$\mathbf{u} = \check{\mathbf{u}} + \rho, \quad (6.15)$$

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 σ deviation, where $\|\rho\|_2 \leq \rho_0$ with $\rho_0 > 0 \in \mathbb{R}$ denoting the bound.

As seen from (6.12), to address the uncertainty of constant matrix W , an evolution rule (6.14) is designed. From (6.3), $\dot{\mathbf{r}} = W\phi(\theta)\dot{\theta} = W\phi(\theta)\mathbf{u}$. It follows that (6.14) can be rewritten as

$$\begin{aligned} \dot{\hat{W}} &= -\nu(\hat{W}\phi(\theta)\mathbf{u} - W\phi(\theta)\mathbf{u})\mathbf{u}^T\phi^T(\theta) \\ &= -\nu(\hat{W} - W)\phi(\theta)\mathbf{u}\mathbf{u}^T\phi^T(\theta). \end{aligned} \quad (6.16)$$

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 $\dot{\hat{W}}\phi(\theta)$ converges to $W\phi(\theta)$, i.e., $\dot{\hat{W}}\phi(\theta) - J(\theta) = 0$, it is evident that $\dot{\hat{W}} = 0$, which means that the dynamical system described by (6.16), i.e., (6.14), achieves its equilibrium. Besides, in the presented PNN, the output \mathbf{u} is the sum of the state vector $\check{\mathbf{u}}$ and additive random noise ρ . The reason why noise is introduced into the output of the presented 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 6.5 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 is 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 3×10^{-4} m [47], is becoming more and more popular in the robotics and control community as a reliable tool for position measurement [48–50]. The OptiTrack have been widely used in robotics [51–53]. 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 of 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 [54]. In terms of the end-effector velocity $\dot{\mathbf{r}}$, it can be obtained from the position measurement via using tracking differentiators [55, 56]. Since the end-effector acceleration is always

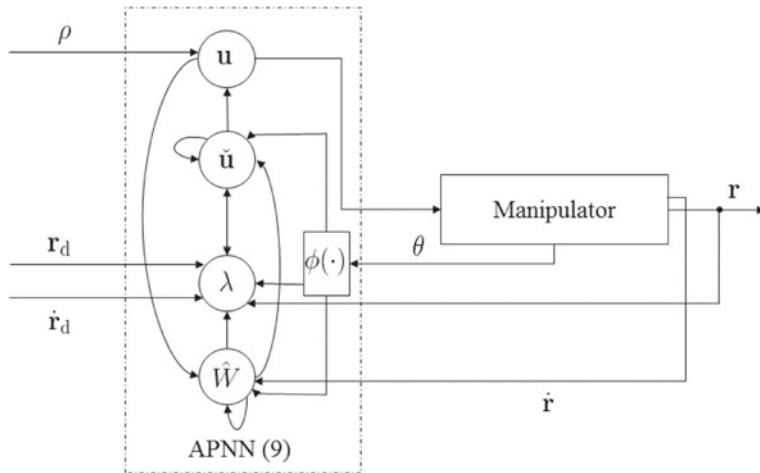


Fig. 6.1 The block diagram about the implementation of the presented adaptive projection neural network (APNN) described in (6.12) for real-time redundancy resolution of manipulators with an uncertain constant matrix W in kinematics

bounded, according to Theorem 6.1 in [56], 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 presented adaptive PNN, we offer the following remark.

Remark 6.6 Figure 6.1 shows a block diagram about the implementation of the presented 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., θ , \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}_d and $\dot{\mathbf{r}}_d$, and artificially added additive noise ρ , 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 presented method, we provide the following remark.

Remark 6.7 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

Table 6.1 D-H parameters of the PUMA 560 manipulator

Joint	a_i (m)	α_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

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 [57–59]. 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 presented adaptive PNN are presented.

Theorem 6.1 *The state variable \hat{W} of the adaptive projection neural network (6.12) satisfies $\hat{W}\phi(\theta) = J(\theta)$ when $t \rightarrow +\infty$ and the state variable \mathbf{u} of (6.12) converges to the optimal solution to the resolution problem shown in (6.4) with an error bounded by ρ_0 .*

Proof Let $\tilde{W}(t) = \hat{W}(t) - W$. Consider the following function: $V_1 = \frac{1}{2} \|\tilde{W}\|_{\text{F}}^2 = \frac{1}{2} \text{trace}(\tilde{W}^T \tilde{W})$, where $\|\cdot\|_{\text{F}}$ 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 (6.12), calculating the time derivative of V_1 gives

$$\begin{aligned}
 \dot{V}_1 &= \text{trace}(\tilde{W}^T \dot{\tilde{W}}) \\
 &= -v \text{trace}(\tilde{W}^T (\hat{W}\phi(\theta)\mathbf{u} - \dot{\mathbf{r}})\mathbf{u}^T \phi^T(\theta)) \\
 &= -v \text{trace}(\tilde{W}^T (\hat{W}\phi(\theta)\mathbf{u} - W\phi(\theta)\mathbf{u})\mathbf{u}^T \phi^T(\theta)) \\
 &= -v \text{trace}(\tilde{W}^T \tilde{W}\phi(\theta)\mathbf{u}\mathbf{u}^T \phi^T(\theta)).
 \end{aligned} \tag{6.17}$$

Note that $\text{trace}(XY) = \text{trac}(X^T Y^T)$ with X and Y being two compatible matrices [60]. Let $X = \tilde{W}^T$ and $Y = \tilde{W}\phi(\theta)\mathbf{u}\mathbf{u}^T\phi^T(\theta)$. Then, it follows from (6.17) and (6.12) that

$$\begin{aligned}\dot{V}_1 &= -\nu \text{trace}(\tilde{W}\phi(\theta)\mathbf{u}\mathbf{u}^T\phi^T(\theta)\tilde{W}^T) \\ &= -\nu \text{trace}((\tilde{W}\phi(\theta)\mathbf{u})(\tilde{W}\phi(\theta)\mathbf{u})^T) \\ &= -\nu \|\tilde{W}\phi(\theta)\mathbf{u}\|_F^2 \\ &= -\nu \|\tilde{W}\phi(\theta)(\check{\mathbf{u}} + \rho)\|_F^2 \\ &\leq 0.\end{aligned}$$

Then, together with (6.17) and (6.12), the LaSalle's invariance principle [61] is employed, which gives

$$\text{trace}(\tilde{W}^T\tilde{W}\phi(\theta)(\check{\mathbf{u}} + \rho)(\check{\mathbf{u}} + \rho)^T\phi^T(\theta)) = 0, \quad (6.18)$$

when $t \rightarrow +\infty$. Calculating expected values on both sides of (6.18) yields

$$E(\text{trace}(\tilde{W}^T\tilde{W}\phi(\theta)(\check{\mathbf{u}} + \rho)(\check{\mathbf{u}} + \rho)^T\phi^T(\theta))) = 0, \quad (6.19)$$

when $t \rightarrow +\infty$. Recalling that ρ is i.i.d. zero-mean random noise of deviation σ , one has

$$\begin{aligned}E(\text{trace}(\tilde{W}^T\tilde{W}\phi(\theta)(\check{\mathbf{u}} + \rho)(\check{\mathbf{u}} + \rho)^T\phi^T(\theta))) \\ &= E(\text{trace}(\tilde{W}^T\tilde{W}\phi(\theta)\check{\mathbf{u}}\check{\mathbf{u}}^T\phi^T(\theta))) + E(\text{trace}(\tilde{W}^T\tilde{W}\phi(\theta)\check{\mathbf{u}}\rho^T\phi^T(\theta))) \\ &\quad + E(\text{trace}(\tilde{W}^T\tilde{W}\phi(\theta)\rho\check{\mathbf{u}}^T\phi^T(\theta))) + E(\text{trace}(\tilde{W}^T\tilde{W}\phi(\theta)\rho\rho^T\phi^T(\theta))) \\ &= E(\text{trace}(\tilde{W}\phi(\theta)\check{\mathbf{u}}\check{\mathbf{u}}^T\phi^T(\theta)\tilde{W}^T)) + E(\text{trace}(\tilde{W}\phi(\theta)\rho\check{\mathbf{u}}^T\phi^T(\theta)\tilde{W}^T)) \\ &\quad + E(\text{trace}(\phi^T(\theta)\tilde{W}^T\tilde{W}\phi(\theta)\check{\mathbf{u}}\rho^T)) + E(\text{trace}(\phi^T\tilde{W}^T\tilde{W}\phi(\theta)\rho\rho^T)) \\ &= E(\text{trace}(\tilde{W}\phi(\theta)\check{\mathbf{u}}\check{\mathbf{u}}^T\phi^T(\theta)\tilde{W}^T)) + E(\text{trace}(\phi^T(\theta)\tilde{W}^T\tilde{W}\phi(\theta)\check{\mathbf{u}}\rho^T)) \\ &\quad + E(\text{trace}(\phi^T(\theta)\tilde{W}^T\tilde{W}\phi(\theta)\check{\mathbf{u}}\rho^T)) + E(\text{trace}(\phi^T\tilde{W}^T\tilde{W}\phi(\theta)\rho\rho^T)) \\ &= \text{trace}(E(\tilde{W}\phi(\theta)\check{\mathbf{u}}\check{\mathbf{u}}^T\phi^T(\theta)\tilde{W}^T)) + \text{trace}(E(\phi^T(\theta)\tilde{W}^T\tilde{W}\phi(\theta)\check{\mathbf{u}})E^T(\rho)) \\ &\quad + \text{trace}(E(\phi^T(\theta)\tilde{W}^T\tilde{W}\phi(\theta)\check{\mathbf{u}})E^T(\rho)) + \text{trace}(E(\phi^T\tilde{W}^T\tilde{W}\phi(\theta))E(\rho\rho^T)) \\ &= \text{trace}(E(\tilde{W}\phi(\theta)\check{\mathbf{u}}\check{\mathbf{u}}^T\phi^T(\theta)\tilde{W}^T)) + \sigma^2 \text{trace}(E(\phi^T(\theta)\tilde{W}^T\tilde{W}\phi(\theta))) \\ &= E(\|\tilde{W}\phi(\theta)\check{\mathbf{u}}\|_F^2) + \sigma^2 E(\|\tilde{W}\phi(\theta)\|_F^2),\end{aligned}$$

which indicates that $E(\|\tilde{W}\phi(\theta)\check{\mathbf{u}}\|_F^2) + \rho^2 E(\|\tilde{W}\phi(\theta)\|_F^2) = 0$ when $t \rightarrow +\infty$ with (6.16) taken into account. Since $E(\|\tilde{W}\phi(\theta)\check{\mathbf{u}}\|_F^2) \geq 0$ and $E(\|\tilde{W}\phi(\theta)\|_F^2) \geq 0$, it is further concluded that $E(\|\tilde{W}\phi(\theta)\|_F^2) = 0$ when $t \rightarrow +\infty$. It follows that $\tilde{W}\phi(\theta) = 0$, i.e., $\hat{W}\phi(\theta) = J(\theta)$, when $t \rightarrow +\infty$. In other words, the dynamics of the adaptive PNN described in (6.12) asymptotically converges to a invariant set, in which $\hat{W}\phi(\theta) = J(\theta)$. According to LaSalle's invariance principle [61], the following

analysis is conducted on the invariant set. Specifically, in the invariant set, based on (6.12), the dynamics of λ and $\check{\mathbf{u}}$ becomes

$$\begin{aligned}\varepsilon \dot{\check{\mathbf{u}}} &= -\check{\mathbf{u}} + P_{\Omega}(\check{\mathbf{u}} - (A\check{\mathbf{u}} + \mathbf{b} - J^T(\theta)\lambda)), \\ \varepsilon \dot{\lambda} &= \dot{\mathbf{r}}_d - J(\theta)\check{\mathbf{u}} + \zeta(\mathbf{r}_d - \mathbf{r}),\end{aligned}$$

which is further rewritten as

$$\varepsilon \mathbf{z} = -\mathbf{z} + P_{\bar{\Omega}}(\mathbf{z} - F(\mathbf{z})) \quad (6.20)$$

with $\mathbf{z} = [\check{\mathbf{u}}^T, \lambda^T]^T$, $\bar{\Omega} = \{(\check{\mathbf{u}}, \lambda) | \check{\mathbf{u}} \in \Omega, \lambda \in \mathbb{R}^m\}$ and

$$F(\mathbf{z}) = \begin{bmatrix} A\check{\mathbf{u}} + \mathbf{b} - J^T(\theta)\lambda \\ -\dot{\mathbf{r}}_d + J(\theta)\check{\mathbf{u}} - \zeta(\mathbf{r}_d - \mathbf{r}) \end{bmatrix}.$$

Consider a Lyapunov candidate function $V_2 = (\mathbf{z} - P_{\bar{\Omega}}(\mathbf{z}))^T(\mathbf{z} - P_{\bar{\Omega}}(\mathbf{z}))/2 \geq 0$. Calculating its time derivative along the dynamics (6.20) 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 [61], $\mathbf{z} - P_{\bar{\Omega}}$ asymptotically converges to zero, i.e., \mathbf{z} asymptotically converges to be within set $\bar{\Omega}$.

From the analytical expression of $F(\mathbf{z})$,

$$\nabla F = \frac{\partial F(\mathbf{z})}{\partial \mathbf{z}} = \begin{bmatrix} A & -J^T(\theta) \\ J(\theta) & 0 \end{bmatrix}.$$

Then,

$$\nabla F + \nabla^T F = \begin{bmatrix} 2A & 0 \\ 0 & 0 \end{bmatrix},$$

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 $(\mathbf{z}_1 - \mathbf{z}_2)^T(F(\mathbf{z}_1) - F(\mathbf{z}_2)) = (\mathbf{z}_1 - \mathbf{z}_2)^T \nabla F(\mathbf{z}_3)(\mathbf{z}_1 - \mathbf{z}_2) \geq 0$, indicating that $F(\cdot)$ is monotone. Based on Theorem 6.1 in [62], it is further concluded that dynamical system (6.20) is stable in the sense of Lyapunov and \mathbf{z} globally converges to $\mathbf{z}^* = [\check{\mathbf{u}}^{*T}, \lambda^{*T}]^T$, which satisfies the following inequality:

$$(\mathbf{z} - \mathbf{z}^*)^T F(\mathbf{z}^*) \geq 0, \quad \forall \mathbf{z} \in \bar{\Omega}.$$

In other words, $\forall \check{\mathbf{u}} \in \Omega$ and $\lambda \in \mathbb{R}^m$, the following inequality holds:

$$(\check{\mathbf{u}} - \check{\mathbf{u}}^*)^T(A\check{\mathbf{u}}^* + \mathbf{b} - J^T(\theta)\lambda^*) + (\lambda - \lambda^*)^T(-\dot{\mathbf{r}}_d + J(\theta)\check{\mathbf{u}}^* - \zeta(\mathbf{r}_d - \mathbf{r})) = 0.$$

It follows that one can always find a value of λ such that $(\lambda - \lambda^*)^T(-\dot{\mathbf{r}}_d + J(\theta)\check{\mathbf{u}}^* - \zeta(\mathbf{r}_d - \mathbf{r}))$ tends to infinity when $-\dot{\mathbf{r}}_d + J(\theta)\check{\mathbf{u}}^* - \zeta(\mathbf{r}_d - \mathbf{r}) \neq 0$. As a result,

$$\begin{cases} -\dot{\mathbf{r}}_d + J(\theta)\check{\mathbf{u}}^* - \zeta(\mathbf{r}_d - \mathbf{r}) = 0, \\ (\check{\mathbf{u}} - \check{\mathbf{u}}^*)^T(A\check{\mathbf{u}}^* + \mathbf{b} - J^T(\theta)\lambda^*) \geq 0, \quad \forall \check{\mathbf{u}} \in \Omega. \end{cases} \quad (6.21)$$

Evidently, (6.21) satisfies the solution of the saddle point problem described as follows:

$$\min_{\check{\mathbf{u}} \in \Omega} \max_{\lambda} L(\check{\mathbf{u}} \in \Omega, \lambda) = \check{\mathbf{u}}^T A \check{\mathbf{u}} / 2 + \mathbf{b}^T \check{\mathbf{u}} + \lambda^T (\dot{\mathbf{r}}_d - W\phi(\theta)\check{\mathbf{u}} + \zeta(\mathbf{r}_d - f(\theta))),$$

of which the solution is identical to the following constrained optimization problem:

$$\begin{aligned} \min_{\check{\mathbf{u}}(t)} \quad & \frac{1}{2} \check{\mathbf{u}}^T(t) A \check{\mathbf{u}}(t) + \mathbf{b}^T \check{\mathbf{u}}(t), \\ \text{s.t.} \quad & J(\theta(t))\check{\mathbf{u}}(t) = \dot{\mathbf{r}}_d(t) + \zeta(\mathbf{r}_d(t) - f(\theta(t))), \\ & \check{\mathbf{u}}(t) \in \Omega. \end{aligned}$$

Thus, $\check{\mathbf{u}}^*$ is the optimal solution of (6.22). Since the actual output of the adaptive PNN (6.12) 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 ρ bounded by ρ_0 . The proof is complete. \square

Theorem 6.1 shows the identification capability of the presented 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 6.8 From the proof of Theorem 6.1, 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 [63] is generally required to guarantee parameter convergence. To some extent, the PE condition requires a signal to be sufficiently rich [64]. To rigorously verify whether a reference trajectory satisfies the PE condition, one can refer to its definition [65, 66]. However, the PE condition is difficult to be checked online [67]. 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 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 6.1, the convergence analysis is not based on the PE condition. In terms of parameter convergence, it can be observed from (6.11) that ν is a gain parameter to adjust the convergence of \hat{W} . By using a larger value of ν , the parameter convergence can be faster.

6.5 Simulative Verifications and Comparisons

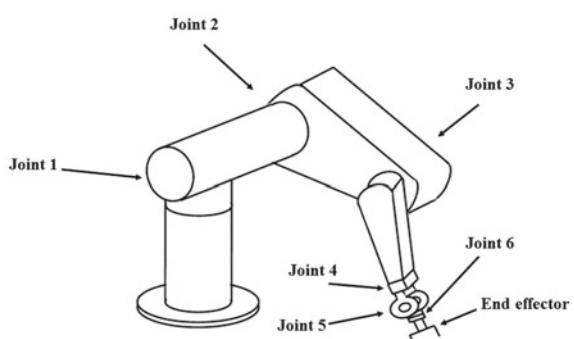
In this section, simulation results are presented and compared to verify the theoretical results, and substantiate the efficacy and superiority of the presented adaptive PNN.

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 6.9 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 an 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 presented 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 presented 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 presented adaptive PNN.

Fig. 6.2 The schematic of the physical structure of the PUMA 560 manipulator



6.5.2 Minimum-Velocity-Norm Redundancy Resolution

In this subsection, we present the simulation results when the presented adaptive PNN (6.12) 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 (6.11).

The minimum-velocity-norm redundancy resolution problem corresponds to the case that, in (6.4), A is an 3×3 identity matrix and $\mathbf{b} = 0$. The parameter in (6.4) is set to $\zeta = 0.1$. Besides, the joint velocity bounds are $\eta^+ = [1, 1, 1, 1, 1, 1]^T$ rad/s and $\eta^- = [-1, -1, -1, -1, -1, -1]^T$ rad/s. In the simulations, the end effector of the manipulator is expected to track the path described as follows:

$$\mathbf{r}_d(t) = \begin{bmatrix} \cos((\pi t)/5)/5 + 7/20 \\ -(\sqrt{3} \sin((\pi t)/5))/10 \\ \sin((\pi t)/5)/10 + 13/10 \end{bmatrix}, \quad (6.22)$$

which is cyclic with the period being 10 s.

In the simulation, the parameters of the adaptive PNN (6.12) are set as $\varepsilon = 0.0001$ and $\nu = 10000$. The deviation of noise ρ 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; $\theta(0) = [3.1649, 1.9548, 0.4584, 4.9330, 0.9870, 2.6527]^T$ 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. 6.3a and b, respectively. Besides, as seen from Fig. 6.3c and d, the end effector of the PUMA 560 manipulator successfully tracks the desired path defined in (6.22) with the position errors being less than 3×10^{-3} m (i.e., 3 mm) and the velocity errors being less than 2×10^{-4} m/s (i.e., 0.4 mm/s). As seen from Fig. 6.3e, the profiles of $\hat{\mathbf{u}}$ are smooth and remain in the bounds η^+ and η^- . Figure 6.3f shows the smooth evolution of the Lagrange multiplier λ . Besides, as seen from Fig. 6.3g, 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. 6.3h, from which it is observed that $\|\hat{W}\phi(\theta) - J(\theta)\|_2^2$ quickly converges to zero, substantiating Theorem 6.1. 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 \hat{W} . It is worth pointing out that, to show the merit of parameter convergence, when $t = 5$ s during the 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 presented 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 (6.11) 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

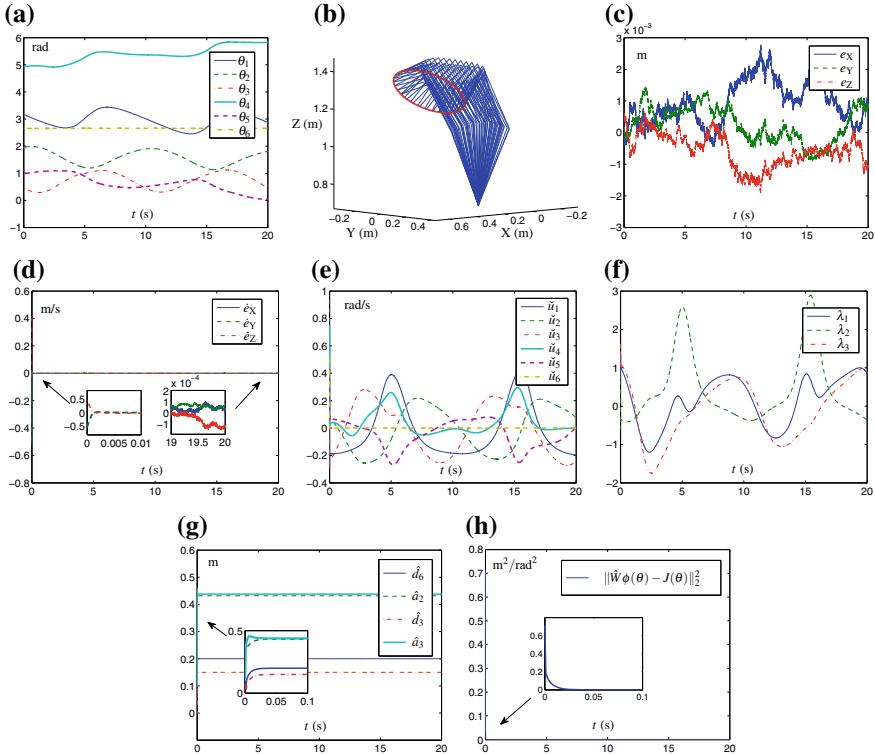
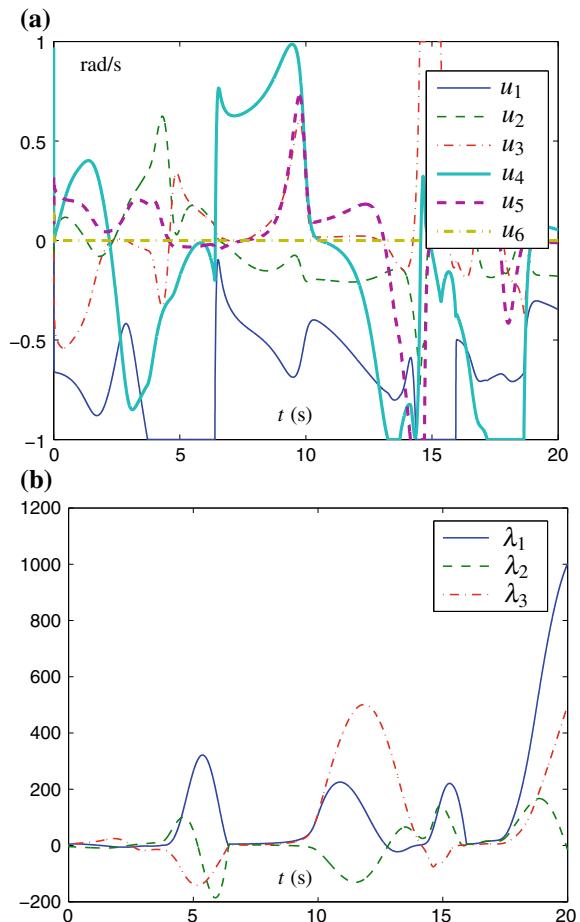


Fig. 6.3 Simulation results about the minimum-velocity-norm redundancy resolution of the PUMA 560 manipulator with unknown parameters a_2 , a_3 , d_3 , and d_6 via the presented adaptive PNN (6.12). **a** Joint angle profiles. **b** Motion trajectory of the manipulator, where the red line denotes the 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** $\hat{\mathbf{u}}$ profiles. **f** λ profiles. **g** \hat{W} profiles. **h** Estimation profiles. Note that each nonzero 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 Appendix of this chapter

in (6.11), which is static, is set to the same value of $\hat{W}(0)$ in the adaptive PNN. The state variables of the nominal PNN (6.11) are shown in Fig. 6.4. As seen from Fig. 6.4a, the magnitude of each element of \mathbf{u} generated by the nominal PNN (6.11) is larger than that in Fig. 6.3e. In addition, Fig. 6.4b shows a divergence of λ . 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 denoted by φ , which is calculated by

Fig. 6.4 Simulation results on profiles of state variables of the nominal PNN (6.11) 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** λ profiles



$$\varphi = \frac{\text{nominal parameter value} - \text{true parameter value}}{\text{true parameter value}}.$$

In the simulations, the values of elements in $\hat{W}(0)$ corresponds to the nominal values. Under the same setup, but with different levels of initial parameter errors, simulations are conducted, for which the error norms are shown in Fig. 6.5c and 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 4×10^{-3} m when the presented adaptive PNN is adopted. These results substantiate the superiority of the presented adaptive PNN over the existing one.

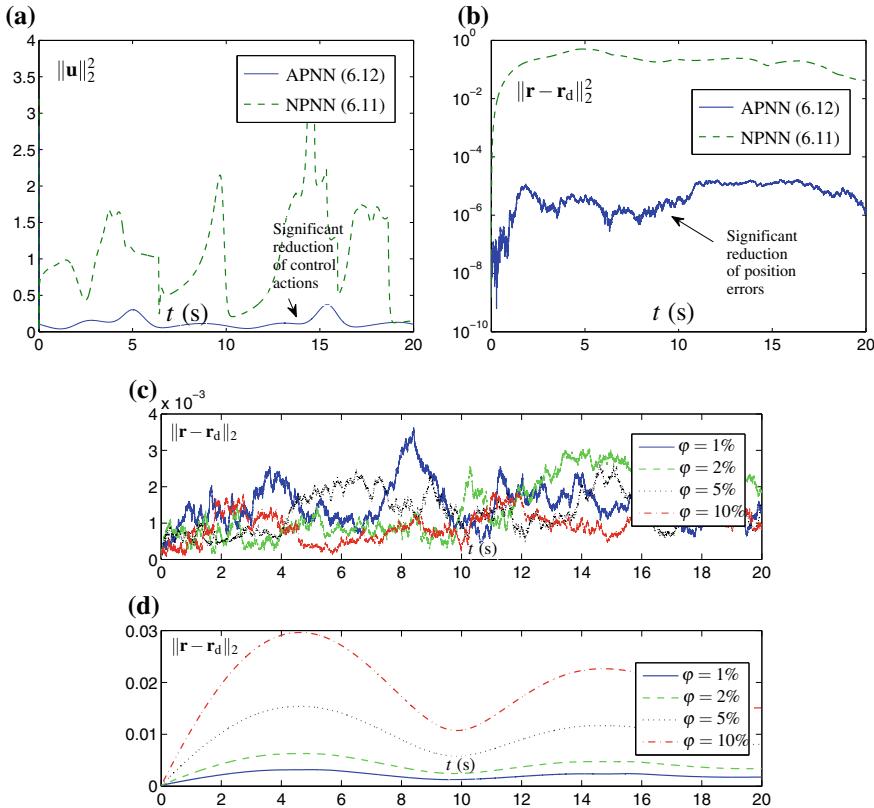


Fig. 6.5 Comparison of performances during the process of minimum-velocity-norm redundancy resolution of the PUMA 560 manipulator with unknown parameters a_2 , a_3 , d_3 , and d_6 via the presented adaptive projection neural network (APNN) (6.12) and the nominal projection neural network (NPNN) (6.11). **a** Control actions. **b** Position errors. **c** Profiles of position error norm when APNN (6.12) is adopted. **d** Profiles of position error norm when NPNN (6.11) is adopted

To further show the efficacy of the presented adaptive projection neural network (6.12), we have conducted simulations based on the PUMA560 manipulator used in this chapter for the two cases mentioned in Remark 6.7. 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 of the presented 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 the 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 Figs. 6.6 and 6.7, under both cases, the presented method can effectively handle the parameter uncertainty and guarantees the convergence of end-effector errors to a small neighbor of zero.

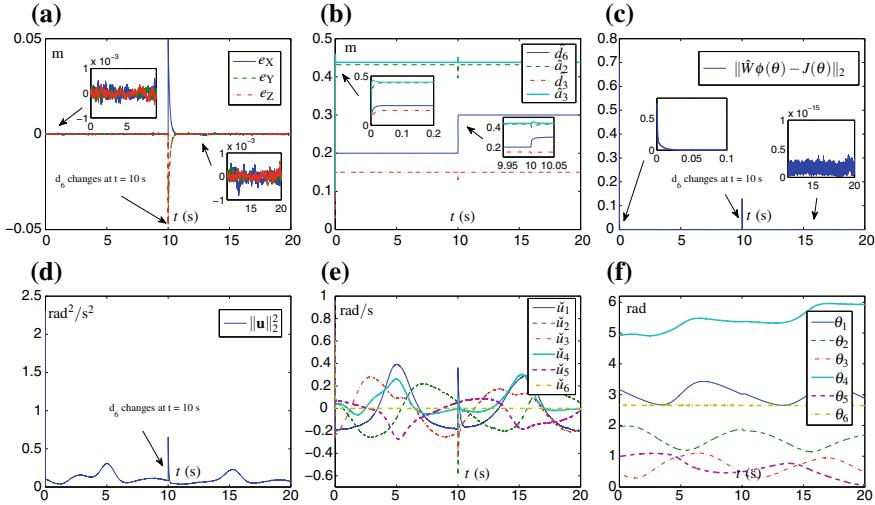


Fig. 6.6 Simulation results about the minimum-velocity-norm redundancy resolution of the PUMA 560 manipulator with unknown parameters a_2 , a_3 , d_3 , and d_6 via the presented adaptive PNN (6.12) for the case that d_6 suddenly changes from 0.2 to 0.3 m. **a** Position error profiles. **b** Parameter profiles. **c** Jacobian matrix error profiles. **d** Velocity norm profiles. **e** Joint velocity profiles. **f** Joint angle profiles

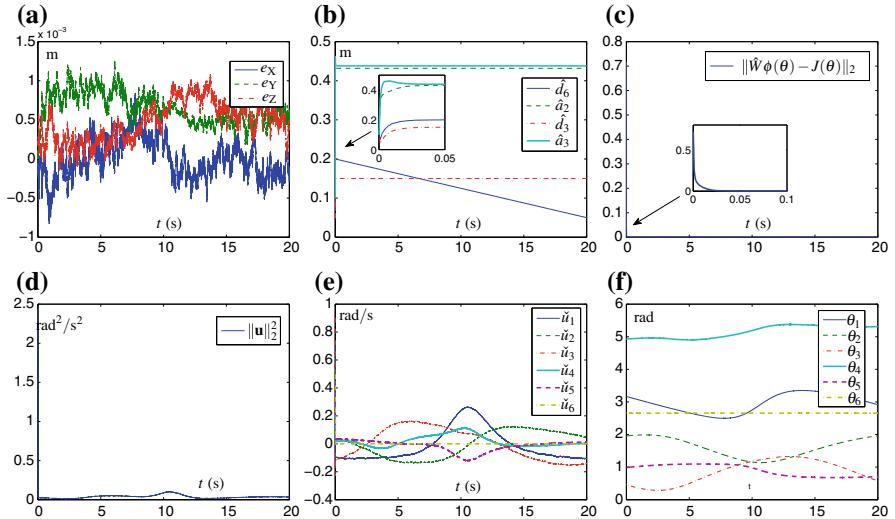


Fig. 6.7 Simulation results about the minimum-velocity-norm redundancy resolution of the PUMA 560 manipulator with unknown parameters a_2 , a_3 , d_3 , and d_6 via the presented adaptive PNN (6.12) for the case that d_6 keeps on decreasing with $d_6(t) = 0.2 - 0.0075t$ m. **a** Position error profiles. **b** Parameter profiles. **c** Jacobian matrix error profiles. **d** Velocity norm profiles. **e** Joint velocity profiles. **f** Joint angle profiles

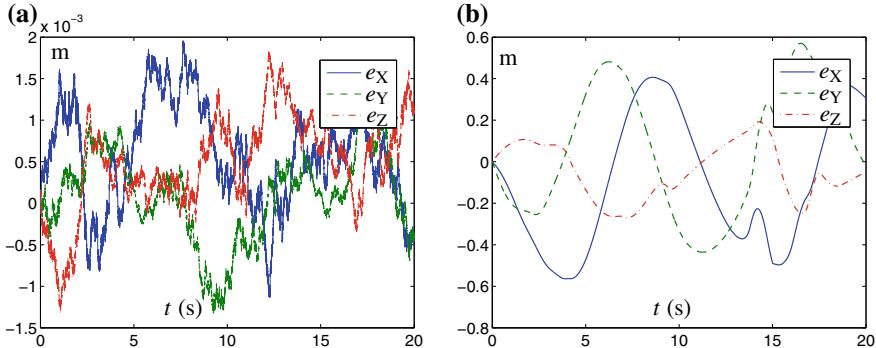


Fig. 6.8 Comparison of position errors during the process of repetitive-motion redundancy resolution of the PUMA 560 manipulator with unknown parameters a_2 , a_3 , d_3 , and d_6 . **a** Via adaptive projection neural network (6.12). **b** Via nominal projection neural network (6.11) with $W = \tilde{W}(0)$

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 (6.22) via different projection neural networks (PNNs)

Joint displacement	Adaptive PNN (6.12)	Nominal PNN (6.11)
$\Delta\theta_1$	1.5704×10^{-3}	9.1862
$\Delta\theta_2$	5.1877×10^{-4}	1.8226
$\Delta\theta_3$	2.1331×10^{-4}	5.3637×10^{-1}
$\Delta\theta_4$	1.4095×10^{-2}	3.9954
$\Delta\theta_5$	8.1973×10^{-4}	8.98897×10^{-1}
$\Delta\theta_6$	7.5146×10^{-6}	0

6.5.3 Repetitive-Motion Redundancy Resolution

In this subsection, we further show the efficacy and superiority of the presented adaptive PNN in repetitive-motion redundancy resolution. We consider the repetitive-motion redundancy resolution of the UMA560 manipulator, which corresponds to the case that A is an 6×6 identity matrix and $\mathbf{b} = \gamma(\theta(t) - \theta(0))$ in the problem formulation (6.4). In the simulations, γ is set to 0.1. The results are also compared with the nominal PNN (6.11). 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 presented 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 the nominal one is used. The above results further substantiate the efficacy and superiority of the presented adaptive PNN (6.12) in repetitive-motion redundancy resolution of manipulators with unknown parameters.

6.6 Experimental Verification

To further show the efficacy of the presented adaptive PNN, experiments based on a Kinova JACO² manipulator have been conducted for the minimum-velocity-norm redundancy resolution aided by the presented adaptive projection neural network (6.12).

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 ρ is set to 0.001. By D-H convention [41], the analytical expression of W for the Kinova JACO² 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, $\mathbf{r}_d(t)$ and $\dot{\mathbf{r}}_d(t)$ 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 $\eta^+ = [1, 1, 1, 1, 1, 1]^T$ rad/s and $\eta^- = [-1, -1, -1, -1, -1, -1]^T$ rad/s. Some snapshots during the experiment process are show in Fig. 6.9b, from which we can observe that the task is successfully

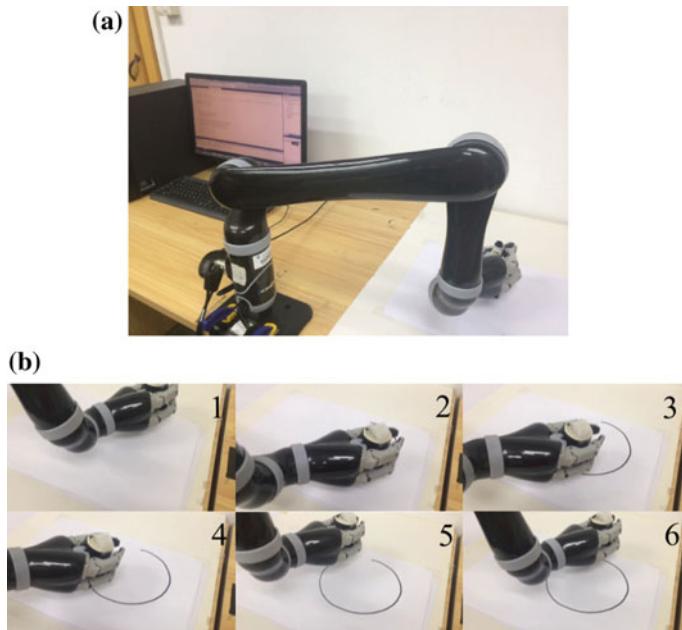
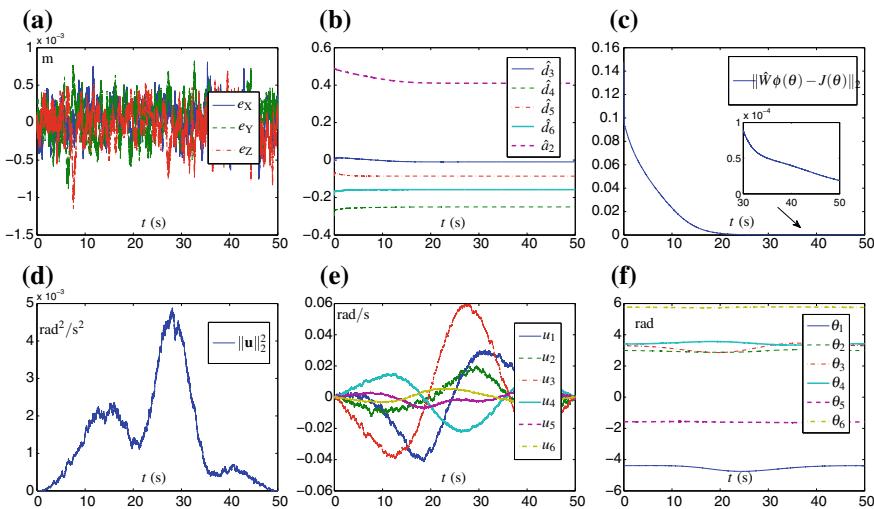


Fig. 6.9 The experiment platform, which includes a personal computer and a Kinova JACO² manipulator holding a pen and the snapshots during the experiment process. **a** Experiment platform. **b** Snapshots

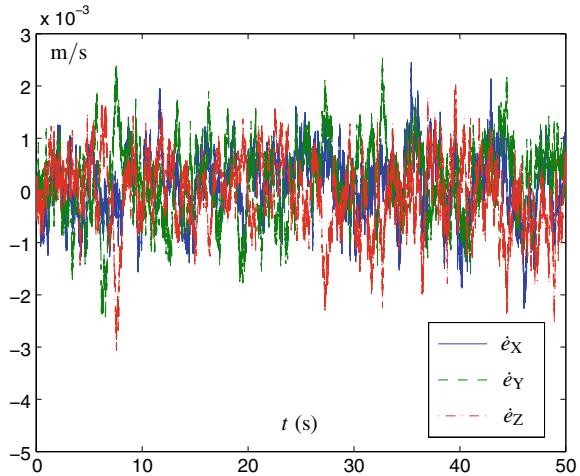
Table 6.3 D-H parameters of the Kinova JACO² manipulator

Joint	a_i (m)	α_i (rad)	d_i (m)
1	0	$\pi/2$	d_1
2	a_2	π	0
3	0	$\pi/2$	d_3
4	0	$\pi/3$	d_4
5	0	$\pi/3$	d_5
6	0	π	d_6

**Fig. 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

completed. The related profiles are shown in Fig. 6.10. As seen from Fig. 6.10a, the end-effector errors are less than 1 mm. From Fig. 6.10b, it can be observed that the parameters are convergent, which, together with Fig. 6.10c, 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. 6.10d and e, respectively. Evidently, the joint angle velocity does not exceed the given bounds. In addition, the joint angle profiles are shown in Fig. 6.10f, and the end-effector velocity-error profiles are shown in Fig. 6.11. The experimental results further show the validity and efficacy of the presented adaptive PNN in the redundancy resolution of the Kinova JACO² physical manipulator.

Fig. 6.11 End-effector velocity error profiles during the experiment process



6.7 Questions and Answers

In this section, some discussions related to earlier versions of this chapter are presented in the form of questions and answers.

Question 6.1 “*The formulated problem does not contain the constraints, among which at least the joint limits should be considered for any kinematic control problem of manipulator.*”

Answer: The authors sincerely thank the reviewer for pointing out the comment. In the problem formulation in the previous version, the set constraint corresponds to joint limits. The set constraint can also be rewritten as follows:

$$\eta^- \leq \dot{\theta} \leq \eta^+, \quad (6.23)$$

which means that the joint velocity $\dot{\theta}$ is limited with η^- being the lower bound and η^+ being the upper bound. It should be noted that the joint angle limit can be converted into the velocity limit. Suppose that the physical joint angle limit is $\theta^- \leq \theta \leq \theta^+$, where θ^- and θ^+ 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 the literature, the joint angle limit can be incorporated into the set constraint by setting $\eta^- = \max\{\dot{\theta}_{\min}, -k(\theta - \theta_{\min})\}$ and $\eta^+ = \min\{\dot{\theta}_{\max}, -k(\theta - \theta_{\max})\}$ with $k > 0 \in \mathbb{R}$ being a design parameter to scale the strength of negative feedback to comply with the joint angle limit.

Question 6.2 “In the proposed method, the derivative $\dot{\mathbf{r}}$ is required, which in fact is more difficult to obtain exactly than the physical parameters. It is strange to use difficulty to solve easy.”

Answer: We agree that 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 3×10^{-4} m, is becoming more and more popular in the robotics and control community as a reliable tool for position measurement. In terms of the end-effector velocity $\dot{\mathbf{r}}$, it can be obtained from the position measurement via using tracking differentiators. Since the end-effector acceleration is always bounded, 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.

Question 6.3 “In proof of Theorem 1, the noise ρ plays a key role to show the convergence. This point seems incorrect. Critically, ρ in essence presents a vector not a scalar.”

Answer: The authors sincerely thank the reviewer for pointing out the comment. We agree with the reviewer 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 is generally required to guarantee parameter convergence. However, the PE condition is difficult to be checked online. In this work, 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 in our work is not based on the PE condition.

Question 6.4 “The middle part of Remark 1 is unclear about the repetitive-motion performance index. Where is the term $b^T b$ for such an index?”

Answer: Recall the general performance index shown in the problem formulation as follows:

$$\mathbf{u}^T(t) A \mathbf{u}(t) + \mathbf{b}^T \mathbf{u}(t),$$

where $\mathbf{u} = \dot{\theta}$. The repetitive-motion performance index $\|\dot{\theta}(t) + k(\theta(t) - \theta(0))\|_2^2/2$ aiming at handling the joint angle drift phenomenon corresponds to the case that case that A is an $n \times n$ identity matrix and $\mathbf{b} = k(\theta(t) - \theta(0))$, where $k > 0 \in \mathbb{R}$ is a constant parameter. Specifically, we have

$$\|\dot{\theta}(t) + k(\theta(t) - \theta(0))\|_2^2/2 = \dot{\theta}^T(t)\dot{\theta}(t)/2 + k\dot{\theta}^T(t)(\theta(t) - \theta(0)) + k^2(\theta(t) - \theta(0))^T(\theta(t) - \theta(0))/2.$$

Since the redundancy resolution problem is resolved at the velocity level, i.e., the decision variable is $\dot{\theta}$, the term $k^2(\theta(t) - \theta(0))^T(\theta(t) - \theta(0))/2$ can be directly removed from the performance index. This does not affect the optimality of the solution and has been widely adopted in related literature.

Question 6.5 “*It is advised to present some explanations for the very fast convergence of the estimation dynamics.*”

Answer: The convergence rate of the proposed adaptive projection neural network depends on the values of the design parameters. Recall the proposed adaptive PNN as follows:

$$\begin{aligned}\varepsilon\ddot{\mathbf{u}} &= -\ddot{\mathbf{u}} + P_{\Omega}(\ddot{\mathbf{u}} - (A\ddot{\mathbf{u}} + \mathbf{b} - \phi^T(\theta)\hat{W}^T\lambda)), \\ \varepsilon\dot{\lambda} &= \dot{\mathbf{r}}_d - \hat{W}\phi(\theta)\ddot{\mathbf{u}} + \zeta(\mathbf{r}_d - \mathbf{r}), \\ \dot{\hat{W}} &= -\nu(\hat{W}\phi(\theta)\mathbf{u} - \dot{\mathbf{r}})\mathbf{u}^T\phi^T(\theta), \\ \mathbf{u} &= \ddot{\mathbf{u}} + \rho.\end{aligned}$$

It can be readily observed that ν is a parameter associated with the feedback for $\dot{\hat{W}}$. In the simulation, we set $\nu = 10000$ (i.e., using a relatively high gain), which makes the convergence of the estimation dynamics fast.

Question 6.6 “*If the physical parameters are time-varying, can this method be valid as well?*”

Answer: First, it should be noted that our manuscript focuses on addressing the uncertainty of D-H parameters of manipulators. D-H parameters of manipulators are time-invariant physical parameters. The case that the unknown parameters are time-varying is out of the scope of our manuscript. In the future, we may extend our method to other systems with time-varying unknown parameters.

Question 6.7 “*In Theorem 1, the parameter convergence is based on a deliberated exogenous noise. However, the noises are usually with small amplitudes and therefore the convergence may be rather slow. Alternatively, the persistent excitation can also be fulfilled via the richness of reference trajectories. Please give some comments on such issues.*”

Answer: We agree with the reviewer that the parameter convergence is based on the noise. It is well understood in the field of control that the persistent excitation (PE) condition is generally required to guarantee parameter convergence. However, the PE condition is difficult to be checked online. In the manuscript, 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. Recall the parameter update dynamics as follows:

$$\dot{\hat{W}} = -\nu(\hat{W}\phi(\theta)\mathbf{u} - \dot{\mathbf{r}})\mathbf{u}^T\phi^T(\theta).$$

It can be observed that ν is a gain parameter to adjust the convergence of \hat{W} . By using a larger value of ν , the parameter convergence can be faster. This observation is also verified in the simulation results, where one can readily find that the parameter convergence is quick. On the other hand, we agree that the PE condition can be guaranteed via using a sufficiently rich reference trajectory. However, 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 is not satisfied. This is also a reason why we introduce noise in the work.

Question 6.8 “In simulation, the proposed method is compared with the nominal PNN. The parameter matrix W is set to the value of $\hat{W}(0)$ of the adaptive PNN. Such a comparison is not fair since the nominal PNN solves the resolution problem only when the physical parameters are known *a priori*, which can be identified beforehand.”

Answer: The comparisons correspond to practical applications. 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 an 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 our manuscript, $\hat{W}(0)$ corresponds to the nominal values. So, 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. To see the effect of the difference between nominal values and true values of the physical parameters on the performance of control of manipulators, we conduct simulations with different levels of initial parameter errors. The level of parameter error is denoted by φ , which is calculated by

$$\varphi = \frac{\text{nominal parameter value} - \text{true parameter value}}{\text{true parameter value}}.$$

Under the same setup, but with different levels of initial parameter errors, simulations are conducted. As seen from the simulation results, with the increase of parameter error, the position error norm increases dramatically when the nominal 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 4×10^{-3} m when the proposed adaptive PNN is adopted.

Question 6.9 “*The reviewer appreciates the authors for the effort of adding a real experiment to illustrate their method. However, the significance of online estimating the physical parameters by using the position and speed error of end effector is still doubtful. In the experiment, the true position and speed is calculated by the exact physical parameters, instead of by a third-party capture system. The critical point is that the off-line calibration of physical parameters has been verified very successfully with high accuracy, it is not reasonable to make an online physical estimation according to the errors obtained by third-party capture systems with much lower accuracy than the off-line calibration.*”

Answer: 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 II of the paper. 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 is changed. 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. 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.

To further show the efficacy of the proposed method, we have conducted simulations based on the PUMA560 manipulator used in the manuscript for the two cases mentioned above. 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 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 in Section IV-B except that $\zeta = 8$. The simulation results are shown in Fig. 6.6. 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.

6.8 Summary

In this chapter, an adaptive PNN has been presented for the redundancy resolution of manipulators with unknown physical parameters. Theoretical results have been presented to guarantee the performance of the presented 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 presented PNN and verified the theoretical results. Future work of this chapter would be extending the results to acceleration-level redundancy resolution. As a final remark, it is worth mentioning that this is the first attempt on extending conventional model-based PNN to adaptive situation with online learning of inherently time-invariant physical parameters for the control of robotic manipulators. It should be noted that this chapter addresses the uncertainty of D-H parameters of manipulators. D-H parameters of manipulators are generally time-invariant physical parameters. In the future, we may extend our method to other systems with time-varying unknown parameters. In addition, a low-cost scheme for the external measurement of end-effector position and speed of a manipulator can be investigated.

Appendix

According to the D-H convention [41], based on Table 6.1, the forward kinematics of the PUMA 560 manipulator is described as (6.1), where $\mathbf{r} = [r_x, r_y, r_z]^T$, and $f(\theta) = [f_1(\theta), f_2(\theta), f_3(\theta)]^T$ with

$$\begin{aligned} f_1(\theta) &= (-((\cos \theta_1 \cos^2 \theta_2 - \cos \theta_1 \sin \theta_2 \sin \theta_3) \cos \theta_4 - \sin \theta_1 \sin \theta_4) \sin \theta_5 \\ &\quad - (\cos \theta_1 \cos \theta_2 \sin \theta_3 + \cos \theta_1 \sin \theta_2 \cos \theta_3) \cos \theta_5) d_6 + a_2 \cos \theta_1 \cos \theta_2 \cos \theta_3 \\ &\quad - a_2 \cos \theta_1 \sin \theta_2 \sin \theta_3 + d_3 \sin \theta_1 + a_3 \cos \theta_1 \cos \theta_2, \\ f_2(\theta) &= (-((\sin \theta_1 \cos^2 \theta_2 - \sin \theta_1 \sin \theta_2 \sin \theta_3) \cos \theta_4 + \cos \theta_1 \sin \theta_4) \sin \theta_5 \\ &\quad - (\sin \theta_1 \cos \theta_2 \sin \theta_3 + \sin \theta_1 \sin \theta_2 \cos \theta_3) \cos \theta_5) d_6 + a_2 \sin \theta_1 \cos \theta_2 \cos \theta_3 \\ &\quad - a_2 \sin \theta_1 \sin \theta_2 \sin \theta_3 - d_3 \cos \theta_1 + a_3 \sin \theta_1 \cos \theta_2, \\ f_3(\theta) &= d_1 + (-(\sin \theta_2 \cos \theta_2 + \cos \theta_2 \sin \theta_3) \cos \theta_4 \sin \theta_5 + (-\sin \theta_2 \sin \theta_3 \\ &\quad + \cos \theta_2 \cos \theta_3) \cos \theta_5) d_6 + a_2 \sin \theta_2 \cos \theta_3 + a_2 \cos \theta_2 \sin \theta_3 + a_3 \sin \theta_2. \end{aligned}$$

Then, the Jacobian matrix $J(\theta) = \partial f(\theta) / \partial \theta$ is derived, which is described as (6.2) with the nonzero elements of $W \in \mathbb{R}^{3 \times 11}$ being $w_{11} = d_6$, $w_{12} = a_2$, $w_{13} = d_3$, $w_{14} = a_3$, $w_{25} = d_6$, $w_{26} = a_2$, $w_{27} = d_3$, $w_{28} = a_3$, $w_{39} = d_6$, $w_{3,10} = a_2$, and $w_{3,11} = a_3$. The nonzero elements of the corresponding $\phi(\theta) \in \mathbb{R}^{11 \times 6}$ are

$$\begin{aligned}
\phi_{11}(\theta) &= \sin \theta_5 (\cos \theta_1 \sin \theta_4 + \cos^2 \theta_2 \cos \theta_4 \sin \theta_1 - \cos \theta_4 \sin \theta_1 \sin \theta_2 \sin \theta_3) \\
&\quad + \sin(\theta_2 + \theta_3) \cos \theta_5 \sin \theta_1, \\
\phi_{12}(\theta) &= \cos \theta_1 \cos \theta_2 \cos \theta_4 \sin \theta_5 (2 \sin \theta_2 + \sin \theta_3) - \cos(\theta_2 + \theta_3) \cos \theta_1 \cos \theta_5, \\
\phi_{13}(\theta) &= \cos \theta_5 (\cos \theta_1 \sin \theta_2 \sin \theta_3 - \cos \theta_1 \cos \theta_2 \cos \theta_3) + \cos \theta_1 \cos \theta_3 \\
&\quad \cdot \cos \theta_4 \sin \theta_2 \sin \theta_5, \\
\phi_{14}(\theta) &= \sin \theta_5 (\sin \theta_4 (\cos \theta_1 \cos^2 \theta_2 - \cos \theta_1 \sin \theta_2 \sin \theta_3) + \cos \theta_4 \sin \theta_1), \\
\phi_{15}(\theta) &= \sin \theta_5 (\cos \theta_1 \cos \theta_2 \sin \theta_3 + \cos \theta_1 \cos \theta_3 \sin \theta_2) - \cos \theta_5 (\cos \theta_4 \\
&\quad \cdot (\cos \theta_1 \cos^2 \theta_2 - \cos \theta_1 \sin \theta_2 \sin \theta_3) \sin \theta_1 \sin \theta_4), \\
\phi_{21}(\theta) &= -\cos(\theta_2 + \theta_3) \sin \theta_1, \\
\phi_{22}(\theta) &= \phi_{23}(\theta) = -\sin(\theta_2 + \theta_3) \cos \theta_1, \\
\phi_{31}(\theta) &= \cos \theta_1, \\
\phi_{41}(\theta) &= -\sin \theta_1 \cos \theta_2, \\
\phi_{42}(\theta) &= -\cos \theta_1 \sin \theta_2, \\
\phi_{51}(\theta) &= \sin \theta_5 (\sin \theta_1 \sin \theta_4 - \cos \theta_1 \cos^2 \theta_2 \cos \theta_4 + \cos \theta_1 \cos \theta_4 \sin \theta_2 \sin \theta_3) \\
&\quad - \sin(\theta_2 + \theta_3) \cos \theta_1 \cos \theta_5, \\
\phi_{52}(\theta) &= \cos \theta_2 \cos \theta_4 \sin \theta_1 \sin \theta_5 (2 \sin \theta_2 + \sin \theta_3) - \cos(\theta_2 + \theta_3) \cos \theta_5 \sin \theta_1, \\
\phi_{53}(\theta) &= \cos \theta_5 (\sin \theta_1 \sin \theta_2 \sin \theta_3 - \cos \theta_2 \cos \theta_3 \sin \theta_1) + \cos \theta_3 \cos \theta_4 \sin \theta_1 \\
&\quad \cdot \sin \theta_2 \sin \theta_5, \\
\phi_{54}(\theta) &= -\sin \theta_5 (\cos \theta_1 \cos \theta_4 - \sin \theta_4 (\cos^2 \theta_2 \sin \theta_1 - \sin \theta_1 \sin \theta_2 \sin \theta_3)), \\
\phi_{55}(\theta) &= \sin(\theta_2 + \theta_3) \sin \theta_1 \sin \theta_5 - \cos \theta_5 (\cos \theta_1 \sin \theta_4 + \cos^2 \theta_2 \cos \theta_4 \sin \theta_1 \\
&\quad - \cos \theta_4 \sin \theta_1 \sin \theta_2 \sin \theta_3), \\
\phi_{61}(\theta) &= \cos(\theta_2 + \theta_3) \cos \theta_1, \\
\phi_{62}(\theta) &= \phi_{63}(\theta) = -\sin(\theta_2 + \theta_3) \sin \theta_1, \\
\phi_{71}(\theta) &= \sin \theta_1, \\
\phi_{81}(\theta) &= \cos \theta_1 \cos \theta_2, \\
\phi_{82}(\theta) &= -\sin \theta_1 \sin \theta_2, \\
\phi_{92}(\theta) &= \cos \theta_4 \sin \theta_5 (\sin \theta_2 \sin \theta_3 + 2 \sin^2 \theta_2 - 1) - \sin(\theta_2 + \theta_3) \cos \theta_5, \\
\phi_{93}(\theta) &= -\sin(\theta_2 + \theta_3) \cos \theta_5 - \cos \theta_2 \cos \theta_3 \cos \theta_4 \sin \theta_5, \\
\phi_{94}(\theta) &= \cos \theta_2 \sin \theta_4 \sin \theta_5 (\sin \theta_2 + \sin \theta_3), \\
\phi_{95}(\theta) &= -\cos(\theta_2 + \theta_3) \sin \theta_5 - \cos \theta_2 \cos \theta_4 \cos \theta_5 (\sin \theta_2 + \sin \theta_3), \\
\phi_{10,2}(\theta) &= \phi_{10,3}(\theta) = \cos(\theta_2 + \theta_3), \\
\phi_{11,2}(\theta) &= \cos \theta_2.
\end{aligned}$$

References

1. Zhang, Z., Beck, A., Magnenat-Thalmann, N.: Human-like behavior generation based on head-arms model for robot tracking external targets and body parts. *IEEE Trans. Cybern.* **45**(8), 1390–1400 (2015)
2. Jin, L., Zhang, Y.: G2-Type SRMPC scheme for synchronous manipulation of two redundant robot arms. *IEEE Trans. Cybern.* **45**(2), 153–164 (2015)
3. Guo, D., Zhang, Y.: A new inequality-based obstacle-avoidance MVN scheme and its application to redundant robot manipulators. *IEEE Trans. Syst., Man, Cybern. C, Appl. Rev.* **42**(6), 1326–1340 (2012)
4. Xu, W., Zhang, J., Liang, B., Li, B.: Singularity analysis and avoidance for robot manipulators with nonspherical wrists. *IEEE Trans. Ind. Electron.* **63**(1), 277–290 (2016)
5. Zhang, Y., Wang, J., Xia, Y.: A Dual neural network for redundancy resolution of kinematically redundant manipulators subject to joint limits and joint velocity limits. *IEEE Trans. Neural Netw.* **14**(3), 658–667 (2003)
6. Zhang, Y., Wang, J., Xu, Y.: A dual neural network for bi-criteria kinematic control of redundant manipulators. *IEEE Trans. Robot. Autom.* **18**(6), 923–931 (2002)
7. Liao, B., Liu, W.: Pseudoinverse-type bi-criteria minimization scheme for redundancy resolution of robot manipulators. *Robotica* **33**(10), 2100–2113 (2015)
8. Ding, H., Tso, S.K.: Redundancy resolution of robotic manipulators with neural computation. *IEEE Trans. Ind. Electron.* **46**(1), 230–233 (1999)
9. Klein, C.A., Huang, C.H.: Review of pseudoinverse control for use with kinematically redundant manipulators. *IEEE Trans. Syst., Man, Cybern.* (2), 245–250 (1983)
10. Cherubini, A., Passama, R., Crosmier, A., Lasnier, A., Fraisse, P.: Collaborative manufacturing with physical human-robot interaction. *Robot. Cim-Int. Manuf.* **31**, 1–13 (2016)
11. Xiao, L., Zhang, Y.: Acceleration-level repetitive motion planning and its experimental verification on a six-link planar robot manipulator. *IEEE Trans. Control Syst. Techno.* **21**(3), 906–914 (2013)
12. Jin, L., Li, S., La, H.M., Luo, X.: Manipulability optimization of redundant manipulators using dynamic neural networks. *IEEE Trans. Ind. Electron.* **64**(6), 4710–4720 (2017)
13. Cheng, F.-T., Sheu, R.-J., Chen, T.-H.: The improved compact QP method for resolving manipulator redundancy. *IEEE Trans. Syst., Man, Cybern.* **25**, 1521–1530 (1995)
14. Zhang, Z., Li, Z., Zhang, Y., Luo, Y., Li, Y.: Neural-dynamic-method-based dual-arm CMG scheme with time-varying constraints applied to humanoid robots. *IEEE Trans. Neural Netw. Learn. Syst.* **26**(12), 3251–3262 (2015)
15. Zhang, Y., Ge, S.S., Lee, T.H.: A unified quadratic-programming-based dynamical system approach to joint torque optimization of physically constrained redundant manipulators. *IEEE Trans. Syst., Man, Cybern. B, Cybern.* **34**(5), 2126–2132 (2004)
16. Hou, Z.-G., Cheng, L., Tan, M.: Multicriteria optimization for coordination of redundant robots using a dual neural network. *IEEE Trans. Syst., Man, Cybern. B, Cybern.* **40**(4), 1075–1087 (2010)
17. Chen, D., Zhang, Y.: A hybrid multi-objective scheme applied to redundant robot manipulators. *IEEE Trans. Autom. Sci. Eng.* **14**(3), 1337–1350 (2017)
18. Jin, L., Li, S., Hu, B., Liu, M.: A survey on projection neural networks and their applications. *Appl. Soft Comput.* **76**, 533–544 (2019)
19. Xiao, L., Li, K., Tan, Z., Zhang, Z., Liao, B., Chen, K., Jin, L., Li, S.: Nonlinear gradient neural network for solving system of linear equations. *Inf. Process. Lett.* **142**, 35–40 (2019)
20. Xiang, Q., Liao, B., Xiao, L., Lin, L., Li, S.: Discrete-time noise-tolerant Zhang neural network for dynamic matrix pseudoinversion. *Soft Comput.* **23**(3), 755–766 (2019)
21. Xiao, L., Li, S., Yang, J., Zhang, Z.: A new recurrent neural network with noise-tolerance and finite-time convergence for dynamic quadratic minimization. *Neurocomputing* **285**, 125–132 (2018)
22. Xiao, L., Liao, B., Li, S., Chen, K.: Nonlinear recurrent neural networks for finite-time solution of general time-varying linear matrix equations. *Neural Netw.* **98**, 102–113 (2018)

23. Xiao, L., Zhang, Z., Zhang, Z., Li, W., Li, S.: Design, verification and robotic application of a novel recurrent neural network for computing dynamic Sylvester equation. *Neural Netw.* **105**, 185–196 (2018)
24. Zhang, Z., Lu, Y., Zheng, L., Li, S., Yu, Z., Li, Y.: A new varying-parameter convergent-differential neural-network for solving time-varying convex QP problem constrained by linear-equality. *IEEE Trans. Autom. Control* **63**(12), 4110–4125 (2018)
25. Jin, L., Li, S.: Nonconvex function activated zeroing neural network models for dynamic quadratic programming subject to equality and inequality constraints. *Neurocomputing* **267**, 107–113 (2017)
26. Jin, L., Li, S., Liao, B., Zhang, Z.: Zeroing neural networks: a survey. *Neurocomputing* **267**, 597–604 (2017)
27. Mao, M., Li, J., Jin, L., Li, S., Zhang, Y.: Enhanced discrete-time Zhang neural network for time-variant matrix inversion in the presence of bias noises. *Neurocomputing* **207**, 220–230 (2016)
28. Jin, L., Zhang, Y., Li, S.: Integration-enhanced Zhang neural network for real-time-varying matrix inversion in the presence of various kinds of noises. *IEEE Trans. Neural Netw. Learn. Syst.* **27**(12), 2615–2627 (2016)
29. Li, S., Li, Y.: Nonlinearly activated neural network for solving time-varying complex sylvester equation. *IEEE Trans. Cybern.* **44**(8), 1397–1407 (2014)
30. Li, S., Li, Y., Wang, Z.: A class of finite-time dual neural networks for solving quadratic programming problems and its k-winners-take-all application. *Neural Netw.* **39**, 27–39 (2013)
31. Li, M., Li, Y., Ge, S.S., Lee, T.H.: Adaptive control of robotic manipulators with unified motion constraints. *IEEE Trans. Syst., Man, Cybern. Syst.* **47**(1), 184–194 (2017)
32. Wang, H.: Adaptive control of robot manipulators with uncertain kinematics and dynamics. *IEEE Trans. Autom. Control* **62**(2), 948–954 (2017)
33. Aghili, F.: Adaptive control of manipulators forming closed kinematic chain with inaccurate kinematic model. *IEEE/ASME Trans. Mechatron.* **18**(5), 1544–1554 (2013)
34. Cheah, C.C., Hirano, M., Kawamura, S., Arimoto, S.: Approximate Jacobian control for robots with uncertain kinematics and dynamics. *IEEE Trans. Robot. Autom.* **19**(4), 192–702 (2003)
35. Shimizu, M., Kakuya, H., Yoon, W.-K., Kitagaki, K., Kosuge, K.: Analytical inverse kinematic computation for 7-DOF redundant manipulators with joint limits and its application to redundancy resolution. *IEEE Trans. Robot.* **24**(5), 1131–1141 (2008)
36. Patchaikani, P.K., Behera, L., Prasad, G.: A single network adaptive critic-based redundancy resolution scheme for robot manipulators. *IEEE Trans. Ind. Electron.* **59**(8), 3241–3253 (2012)
37. Wang, H., Shi, P., Li, H., Zhou, Q.: Adaptive neural tracking control for a class of nonlinear systems with dynamic uncertainties. *IEEE Trans. Cybern.* **47**(10), 3075–3087 (2017)
38. Na, J., Chen, Q., Ren, X., Guo, Y.: Adaptive prescribed performance motion control of servo mechanisms with friction compensation. *IEEE Trans. Ind. Electron.* **61**(1), 486–494 (2014)
39. Li, Z., Huang, Z., He, W., Su, C.-Y.: Adaptive impedance control for an upper limb robotic exoskeleton using biological signals. *IEEE Trans. Ind. Electron.* **64**(2), 1664–1674 (2017)
40. Jaramillo-Lopez, F., Kenne, G., Lamnabhi-Lagarrigue, F.: Adaptive control for a class of uncertain nonlinear systems: application to photovoltaic control systems. *IEEE Trans. Autom. Control* **62**(1), 393–398 (2017)
41. Spong, M.W., Hutchinson, S., Vidyasagar, M.: *Robot Modeling and Control*. Wiley, New York (2006)
42. Fang, J., Zhao, J., Mei, T., Chen, J.: Online optimization scheme with dual-mode controller for redundancy-resolution with torque constraints. *Robot. Cim-Int. Manuf.* **40**, 44–54 (2016)
43. Zhang, Y., Li, S., Gui, J., Luo, X.: Velocity-level control with compliance to acceleration-level constraints: a novel scheme for manipulator redundancy resolution. *IEEE Trans. Ind. Inform.* **14**(3), 921–930 (2018)
44. Li, S., Zhang, Y., Jin, L.: Kinematic control of redundant manipulators using neural networks. *IEEE Trans. Neur. Netw. Learn. Syst.* **28**(10), 2243–2254 (2017)
45. Boyd, S., Vandenberghe, L.: *Convex Optimization*. Cambridge University Press, England (2004)

46. Xia, Y., Feng, G.: On convergence rate of projection neural networks. *IEEE Trans. Autom. Control* **49**(1), 91–96 (2004)
47. OptiTrack Motion Capture Systems (2017). Available: <http://optitrack.com/> [Online]
48. Salehian, S.S.M., Khoramshahi, M., Billard, A.: A dynamical system approach for softly catching a flying object: theory and experiment. *IEEE Trans. Robot.* **32**(2), 462–471 (2016)
49. Davis, E., Pounds, P.E.I.: Direct sensing of thrust and velocity for a quadrotor rotor array. *IEEE Robot. Autom. Lett.* **2**(3), 1360–1366 (2017)
50. Wang, A., Mu, B., Shi, Y.: Consensus control for a multi-agent system with integral-type event-triggering condition and asynchronous periodic detection. *IEEE Trans. Ind. Electron.* **64**(7), 5629–5639 (2017)
51. Bartelds, T., Capra, A., Hamaza, S., Stramigioli, S., Fumagalli, M.: Compliant aerial manipulators: toward a new generation of aerial robotic workers. *IEEE Robot. Autom. Lett.* **1**(1), 477–483 (2016)
52. Ajoudani, A., Tsagarakis, N.G., Bicchim, A.: Tele-impedance: towards transferring human impedance regulation skills to robots. In: Proceedings of IEEE International Conference Robotics Automation, pp. 382–388 (2012)
53. Papini, G.P.R., Fontana, M., Bergamasco, M.: Desktop haptic interface for simulation of hand-tremor. *IEEE Trans. Haptics* **9**(1), 33–42 (2016)
54. Du, G., Zhang, P.: Online serial manipulator calibration based on multisensory process via extended kalman and particle filters. *IEEE Trans. Ind. Electron.* **61**(12), 6852–6859 (2014)
55. Han, J.: From PID to active disturbance rejection control. *IEEE Trans. Ind. Electron.* **56**(3), 900–906 (2009)
56. Levant, A.: Robust exact differentiation via sliding mode technique. *Automatica* **34**(3), 379–384 (1998)
57. Dixon, W.E.: Adaptive regulation of amplitude limited robot manipulators with uncertain kinematics and dynamics. *IEEE Trans. Autom. Control* **52**(3), 488–493 (2007)
58. Cheah, C.C., Liu, C., Slotine, J.J.E.: Adaptive jacobian tracking control of robots with uncertainties in kinematic, dynamic and actuator models. *IEEE Trans. Autom. Control* **51**(6), 1024–1029 (2006)
59. Wang, H., Xie, Y.: Passivity based adaptive Jacobian tracking for free-floating space manipulators without using spacecraft acceleration. *Automatica* **45**, 1510–1517 (2009)
60. Bellman, R.: Introduction to Matrix Analysis. Society for Industrial and Applied Mathematics, Philadelphia (1997)
61. Khalil, H.K.: Nonlinear Systems. Prentice-Hall, New Jersey (2002)
62. Gao, X.: Exponential stability of globally projected dynamic systems. *IEEE Trans. Neural Netw.* **14**(2), 426–431 (2003)
63. Adetola, V., Guay, M.: Finite-time parameter estimation in adaptive control of nonlinear systems. *IEEE Trans. Autom. Control* **53**(3), 807–811 (2008)
64. Mishkov, R., Darmonsk, S.: Nonlinear adaptive control system design with asymptotically stable parameter estimation error. *Int. J. Control* **91**(1), 181–203 (2018)
65. Bai, E.W., Sastry, S.S.: Persistency of excitation, sufficient richness and parameter convergence in discrete time adaptive control. *Syst. Control Lett.* **6**, 153–163 (1985)
66. Dixon, W.E., Dawson, D.M., Zhang, F., Zergeroglu, E.: Global exponential tracking control of a mobile robot system via a PE condition. *IEEE Trans. Syst., Man, Cybern. B, Cybern.* **30**(1), 129–142 (2000)
67. Modares, H., Lewis, F.L., Naghibi-Sistani, M.B.: Integral reinforcement learning and experience replay for adaptive optimal control of partially-unknown constrained-input continuous-time systems. *Automatica* **50**(1), 193–202 (2014)

Chapter 7

Redundancy Resolution with Periodic Input Disturbance



Abstract Input disturbances and physical constraints are important issues in the kinematic control of redundant manipulators. In this chapter, we present a novel recurrent neural network to simultaneously address the periodic input disturbance, joint angle constraint, and joint velocity constraint, and optimize a general quadratic performance index. The presented recurrent neural network applies to both regulation and tracking tasks. Theoretical analysis shows that, with the presented neural network, the end-effector tracking and regulation errors asymptotically converge to zero in the presence of both input disturbance and the two constraints. Simulation examples and comparisons with an existing controller are also presented to validate the effectiveness and superiority of the presented controller.

7.1 Introduction

Manipulators are widely used in industry to complete labor-intensive and high-accuracy operations, such as welding, painting, and assembly. Redundant manipulators, which have more degrees-of-freedom than that required to achieve a primary task defined on its end-effector, are receiving a significant amount of attention. The existence of redundancy makes it possible for manipulators to simultaneously complete secondary and primary tasks. Specifically, due to redundancy, redundant manipulators have more than one control solutions to an achievable task, from which we may choose the best one for a secondary task [1].

In the past decades, lots of effort have been devoted to the kinematic control of redundant manipulators. Pseudoinverse-type methods, which generally do not consider joint constraints, are investigated in [2–5]. Generally, due to the physical structure of a manipulator, the joint angles are limited. As each joint is often driven by a motor, the joint velocities are also limited. If the joint constraints are violated, saturations may occur and the tracking error may significantly increase. Meanwhile, physical damages to the manipulator may also happen [6]. As the name indicates, pseudoinverse-type methods require the online calculation of the pseudoinverse of the Jacobian matrix associated with the forward kinematics of the manipulator. It was found that, for pseudoinverse-type methods, the drift in joint space can be

generated when a cyclic task is performed [5]. To deal with joint constraints, Huang et al. [7] proposed a clamping weighted least-norm method, which can guarantee the avoidance of the joint angle limit. Cheng et al. [8] proposed to formulate the tracking control problem of redundant manipulators as a constrained quadratic program. Related works that formulate the kinematic control problem as an optimization problem and solve it via numerical algorithms were also reported in [9]. It should be noted that the solution to the optimization problems such as the quadratic program in the kinematic control of redundant manipulators is essentially time dependent for tracking tasks, for which traditional methods intrinsically designed for solving static optimization problems are not efficient.

Neural networks have been widely applied to intelligent computing and control [10–27]. For example, Jin et al. proposed an algorithm based on recurrent neural networks to solve time-varying linear equations with consideration of noises. The seminar work of Hopfield and Tank [28] opened a door for the research on using recurrent neural networks to solve optimization problems in a highly efficient manner. Based on the work of Cheng et al. [8] and the development of recurrent neural networks, neural network-based kinematic control of redundant manipulators was developed [6, 29–40]. The advantages of this type of methods include but are not limited to the real-time calculation of the control input and a general framework for dealing various problems in the kinematic control of redundant manipulators. For example, the joint-drift phenomenon, which is undesirable for long-term cyclic motion, can be addressed via properly setting the performance index [37]. Under the framework, Jin et al. [34] proposed a recurrent neural network-based approach to maximize the manipulability of a redundant manipulator during the tracking control process so as to avoid kinematic singularity. Zhang et al. [35] proposed a velocity-level kinematic control method which can simultaneously guarantee the compliance with the joint acceleration constraint and the convergence of end-effector tracking error to zero. The extensions to the cooperative control of multiple redundant manipulators were reported in [41–45]. Recently, a neural network-based approach was also proposed to deal with the image-based visual servoing of manipulators [46]. The ideas in the neural network design were also utilized to the task allocation of multiple redundant manipulators [47].

However, most of the existing results on the kinematic control of manipulators do not consider input disturbances, which are inevitable in practice [48]. While many progresses have been made in the field of control theory and engineering about disturbance rejection, such as [49–52] and the references therein, they are generally not directly applicable to the kinematic control of redundant manipulators due to the existence of redundancy, constraints, and the objective to optimize certain performance index of interest. Based on the earlier works on primal-dual neural networks, in [53], a modified primal-dual neural network is proposed for the kinematic control of redundant manipulators with dynamic rejection of harmonic disturbances in the input channel, which are a type of periodic disturbances. However, the method in [53] does not apply to the case with general periodic disturbances. In addition, the controller in [53] can only deal with tracking tasks and does not work for regulation tasks.

To tackle the difficulties in the about discussion, in this chapter, we focus on the kinematic control of redundant manipulators by designing a neural network controller that is simultaneously capable of (1) complying with the joint angle constraint and joint velocity constraint; (2) optimizing a general quadratic performance index; (3) completely learning and compensating periodic input disturbances.

7.2 Preliminary and Problem Description

In this section, the kinematic control problem investigated in this chapter is described.

7.2.1 Manipulator Kinematics Model

Consider an n -degrees-of-freedom manipulator, the mapping from its joint space to the Cartesian coordinate $\mathbf{r} \in \mathbb{R}^m$ of its end-effector is described by a nonlinear function as follows:

$$\mathbf{r} = f(\theta), \quad (7.1)$$

where $\theta \in \mathbb{R}^n$ is the joint angle vector. For serial manipulators, the nonlinear function $f(\cdot)$ can be derived via a systematic method called the Denavit–Hartenberg (D–H) convention [54]. For redundant manipulators, $n > m$. Directly calculating the time derivatives on both sides of (7.1) yields

$$\dot{\mathbf{r}} = J(\theta)\omega, \quad (7.2)$$

where $\dot{\mathbf{r}} = d\mathbf{r}/dt$, $J(\theta) = \partial f(\theta)/\partial\theta$, and $\omega = \dot{\theta} = d\theta/dt$.

Generally, the joint angle θ and joint velocity ω of a manipulator are physically constrained. Let the joint constraints be described as follows:

$$\begin{aligned} \theta^- &\leq \theta \leq \theta^+, \\ \omega^- &\leq \omega \leq \omega^+, \end{aligned} \quad (7.3)$$

where the operator \leq works in a element-wise manner with each element of $\theta^- \in \mathbb{R}^n$ and $\theta^+ \in \mathbb{R}^n$ denotes the lower bound and upper bound of the corresponding element in the joint angle vector θ ; each element of $\omega^- \in \mathbb{R}^n$ and $\omega^+ \in \mathbb{R}^n$ denotes the lower bound and upper bound of the corresponding element in the joint velocity vector ω .

7.2.2 Problem Description

In this chapter, we are interested in the kinematic control of redundant manipulators with forward kinematics (7.1) and physical constraints (7.3). It is expected to

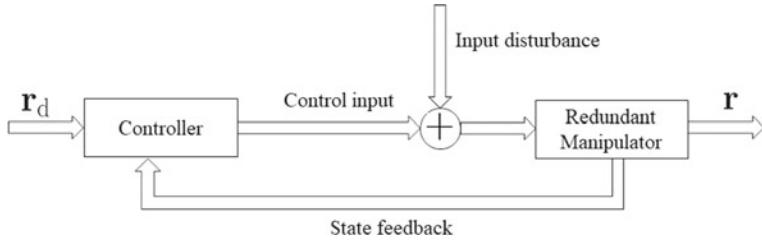


Fig. 7.1 A diagram for the kinematic control of redundant manipulators

find an optimal controller at the acceleration level such that the coordinate error $\mathbf{e}(t) = \mathbf{r}_d(t) - \mathbf{r}(t)$ asymptotically converges to zero and a user defined quadratic performance index $\omega^T A \omega / 2 + \mathbf{b}^T \omega$ is minimized, where $\mathbf{r}_d(t)$ denotes the desired smooth end-effector coordinate with t denoting the time instant. Note that $A \in \mathbb{R}^{n \times n}$ is a positive definite symmetric matrix and $\mathbf{b} \in \mathbb{R}^n$ is a vector. The relationship between the joint velocity and the input $\mathbf{u}(t) \in \mathbb{R}^n$ is described as follows:

$$\dot{\omega} = \mathbf{u}(t) + \eta(t), \quad (7.4)$$

where $\eta(t)$ is a periodic input disturbance with period T , for which

$$\eta(t) = \eta(t - T). \quad (7.5)$$

Note that $\mathbf{u}(t)$ denotes the given end-effector joint acceleration command. This type of control is referred to as acceleration-level kinematic control. The problem can be briefly described by Fig. 7.1, where our focus is to design the controller as shown in the diagram such that the given requirements are satisfied.

Regarding the performance index $\omega^T A \omega / 2 + \mathbf{b}^T \omega$, we offer the following remark.

Remark 7.1 There are several types of objectives that can be described by the performance index. For example, to minimize kinematic energy consumption, we can set the parameters of the performance index to $A = I$ and $\mathbf{b} = 0$ with I being a n -by- n identity matrix [39, 41, 55]. Previous studies showed that the joint-drift problem can be solved via minimizing the performance index by setting $A = I$ and $\mathbf{b} = \nu(\theta(t) - \theta(0))$ with $\nu > 0 \in \mathbb{R}$ [56–58].

Examples of periodic input disturbances in practice are discussed in the following remark.

Remark 7.2 Because the actuator of each joint of a manipulator is often a motor, the periodic input disturbance can be caused by the cogging torque, which is a result of the interaction between the permanent magnets and the stator slots [59]. This kind of disturbance is of the sinusoidal form. When the manipulator is executing periodic tasks, the difference between the actual joint acceleration and the given one can also be viewed as a periodic input disturbance. In this case, the period of the disturbance is the period of the periodic task.

The following definition about projection operator will be adopted in the latter discussions.

Definition 7.1 ((*Projection Operator*) [34]) The projection operator for a set $\mathbb{S} \subset \mathbb{R}^n$ and a vector $\mathbf{y} \in \mathbb{R}^n$ is defined by

$$P_{\mathbb{S}}(\mathbf{y}) = \arg \min_{\mathbf{z} \in \mathbb{S}} \|\mathbf{z} - \mathbf{y}\|_2,$$

where $\|\cdot\|_2$ denotes the Euclidean norm of a vector.

7.2.3 Problem Reformulation as Quadratic Program

In this subsection, we reformulate the problem as a quadratic program. Evidently, the equality constraint for the problem is

$$f(\theta(t)) = \mathbf{r}_d(t). \quad (7.6)$$

Since the decision variable is ω according to the performance index $\omega^T A\omega + \mathbf{b}^T \omega$, the equality constraint (7.6) must be converted from the joint angle level to the joint velocity level. Recall that

$$\mathbf{e}(t) = \mathbf{r}_d(t) - \mathbf{r}(t).$$

According to Zhang et al.'s method [60], by setting

$$\dot{\mathbf{e}}(t) = -\lambda \mathbf{e}(t), \quad (7.7)$$

$\mathbf{e}(t)$ exponentially converges to zero with $\lambda > 0 \in \mathbb{R}$ being a positive parameter to scale the convergence rate. It follows that, (7.6) is asymptotically equivalent to the following equation:

$$J(\theta(t))\omega(t) - \mathbf{r}_d(t) = -\lambda (\mathbf{r}(t) - \mathbf{r}_d(t)), \quad (7.8)$$

which is obtained by substituting (7.1), (7.2), and (7.6) into (7.7). According to [6], the two joint constraints (7.3) can be unified into the following one:

$$\varpi^- \leq \omega \leq \varpi^+, \quad (7.9)$$

with $\varpi^-(t) = \max\{\omega^-, \alpha(\theta^- - \theta(t))\}$ and $\varpi^+(t) = \min\{\omega^+, \alpha(\theta^+ - \theta(t))\}$, where $\alpha > 0 \in \mathbb{R}$. For the sake of latter discussion, we define the following set:

$$\Omega = \{\omega \in \mathbb{R}^n | \varpi^- \leq \omega \leq \varpi^+\}. \quad (7.10)$$

Evidently, Ω is convex, and $0 \in \Omega$ since the joint velocity can be zero. It follows that the unified joint constraint (7.9) can be further described as

$$\omega \in \Omega.$$

Thus, together with the performance index, the quadratic program for the kinematic control problem is formulated as follows:

$$\min_{\omega(t)} \frac{1}{2} \omega^T(t) A \omega(t) + \mathbf{b}^T \omega(t), \quad (7.11a)$$

$$s.t. \quad J(\theta) \omega(t) = \dot{\mathbf{r}}_d(t) + \lambda(\mathbf{r}_d(t) - f(\theta)), \quad (7.11b)$$

$$\omega(t) \in \Omega. \quad (7.11c)$$

Now, the problem becomes finding a control law $\mathbf{u}(t)$ for which $\dot{\omega}(t) = \mathbf{u}(t) + \eta(t)$ such that $\omega(t)$ solves the quadratic program (7.11).

For the sake of presentation, in the rest of this chapter, the argument t is omitted somewhere.

7.3 Neural Network Design

In this section, we design a neural network controller to solve the problem described in the previous section. The design is divided into two steps. In the first step, a nominal neural network controller is designed, which guarantees the performance for the case that $\eta(t) \equiv 0$ (i.e., without disturbance). Then, an improved one is designed based on the nominal one and guarantees the performance for the case with the periodic input disturbance.

7.3.1 Step 1: Nominal Design Without Disturbance

For the nominal case, a neural network controller is designed by the following steps. First, consider the following Lagrange function:

$$\begin{aligned} L(\omega \in \Omega, \gamma) = & \frac{1}{2} \omega^T A \omega + \mathbf{b}^T \omega + \frac{1}{2} (J(\theta) \omega - \dot{\mathbf{r}}_d \\ & - \lambda(\mathbf{r}_d - f(\theta)))^T C (J(\theta) \omega - \dot{\mathbf{r}}_d - \lambda(\mathbf{r}_d - f(\theta))) \\ & + \gamma^T J(\theta) \omega - \dot{\mathbf{r}}_d - \lambda(\mathbf{r}_d - f(\theta)) \end{aligned}$$

with $C \in \mathbb{R}^{m \times m}$ being a positive definite symmetric matrix, where the augmented term associated with C is incorporated to enhance the convergence of the resultant neural network. In addition, $\gamma \in \mathbb{R}^m$ is referred to as the Lagrangian multiplier. The

optimal solution to (7.11), according to the Karush–Kuhn–Tucker condition [61], is equivalent to the solution of the following equation set:

$$\begin{aligned}\omega &= P_{\Omega}(\omega - \frac{\partial L}{\partial \omega}), \\ 0 &= \frac{\partial L}{\partial \lambda},\end{aligned}$$

which gives

$$\begin{aligned}\omega &= P_{\Omega} \left(\omega - \frac{\partial L}{\partial \omega} \right) \\ &= P_{\Omega}(\omega - A\omega - J^T(\theta)B(J(\theta)\omega - \dot{\mathbf{r}}_d - \lambda(\mathbf{r}_d - f(\theta))) - \mathbf{b} - J^T(\theta)\gamma), \\ 0 &= \frac{\partial L}{\partial \gamma} = J(\theta)\omega - \dot{\mathbf{r}}_d - \lambda(\mathbf{r}_d - f(\theta)).\end{aligned}\tag{7.12}$$

As the theoretical solution of (7.12) cannot be obtained, a recurrent neural network is designed as follows to solve the problem:

$$\begin{aligned}\varepsilon \dot{\omega} &= -\omega + P_{\Omega}(\omega - A\omega - J^T(\theta)B(J(\theta)\omega - \dot{\mathbf{r}}_d - \lambda(\mathbf{r}_d - f(\theta))) - \mathbf{b} - J^T(\theta)\gamma), \\ \varepsilon \dot{\gamma} &= J(\theta)\omega - \dot{\mathbf{r}}_d - \lambda(\mathbf{r}_d - f(\theta)),\end{aligned}\tag{7.13}$$

where $\varepsilon > 0 \in \mathbb{R}$ is a parameter used to scale the convergence of the neural network. Evidently, the equilibrium of (7.13) is identical to the solution of (7.12). According to (7.4), when $\eta(t) \equiv 0$, we have $\dot{\omega} = \mathbf{u}$. Thus, the neural controller is obtained as follows for the case without input disturbance:

$$\begin{aligned}\mathbf{u} &= (-\omega + P_{\Omega}(\omega - A\omega - J^T(\theta)B(J(\theta)\omega - \dot{\mathbf{r}}_d - \lambda(\mathbf{r}_d - f(\theta))) - \mathbf{b} - J^T(\theta)\gamma))/\varepsilon, \\ \varepsilon \dot{\gamma} &= J(\theta)\omega - \dot{\mathbf{r}}_d - \lambda(\mathbf{r}_d - f(\theta)).\end{aligned}\tag{7.14}$$

About the performance of the neural network controller (7.14), we have the following theorem.

Lemma 7.1 *When neural network controller (7.14) is adopted, the optimization problem (7.11) is asymptotically solved for the case without input disturbance, i.e., $\eta(t) \equiv 0$.*

Proof Substituting (7.14) into (7.4) with $\eta(t) \equiv 0$ yields the closed-loop dynamics as (7.13). Let $\chi = [\omega^T, \gamma^T]^T$, then (7.13) can be rewritten as

$$\varepsilon \dot{\chi} = -\chi + P_{\Gamma}(\chi - F(\chi)),\tag{7.15}$$

where $F(\chi) = [(A\omega + J^T(\theta)B(J(\theta)\omega - \dot{\mathbf{r}}_d - \lambda(\mathbf{r}_d - f(\theta))) + \mathbf{b} + J^T(\theta)\gamma)^T, (-J(\theta)\omega + \dot{\mathbf{r}}_d + \lambda(\mathbf{r}_d - f(\theta)))^T]^T$, and $\Gamma = \{\chi = [\omega^T, \gamma^T]^T | \omega \in \Omega \subset \mathbb{R}^n, \gamma \in$

$\mathbb{R}^m\}$. The gradient of $F(\chi)$ is

$$\nabla F(\chi) = \begin{bmatrix} A + J^T(\theta)BJ(\theta) & J^T(\theta) \\ -J^T(\theta) & 0 \end{bmatrix}$$

It follows that

$$\nabla F(\chi) + \nabla^T F(\chi) = \begin{bmatrix} 2A + 2J^T(\theta)BJ(\theta) & 0 \\ 0 & 0 \end{bmatrix},$$

which is positive semidefinite by recalling that A and B are symmetric and positive definite. Thus, $F(\chi)$ is monotone with $(\chi_1 - \chi_2)^T(F(\chi_1) - F(\chi_2)) \geq 0, \forall \chi_1 \in \mathbb{R}^{m+n}, \forall \chi_2 \in \mathbb{R}^{m+n}$. With the above properties, according to Theorem 1 in [62], (7.15) is Lyapunov stable and converges to $\chi^* = [\omega^{*\top}, \gamma^{*\top}]^\top$, for which,

$$(\chi - \chi^*)^T F(\chi^*) \geq 0, \forall \chi \in \Gamma. \quad (7.16)$$

It follows that, $\forall \chi \in \Gamma$,

$$\|\chi^* - \chi - F(\chi^*)\|_2^2 - \|F(\chi^*)\|_2^2 = \|\chi - \chi^*\|_2^2 + 2(\chi^* - \chi)^T F(\chi^*) \geq 0.$$

Thus, according to Definition 7.1, (7.16) can be written as

$$\chi^* = P_\Gamma(\chi^* - F(\chi^*)), \quad (7.17)$$

which is actually the equilibrium of (7.15). Expanding (7.17) with the definition of χ^* yields (7.12) by choosing $\omega = \omega^*$ and $\gamma = \gamma^*$. Together with the equivalence between (7.12) and (7.11), it is concluded that ω^* is the optimal solution to (7.11). The proof is complete. \square

7.3.2 Step 2: Modified Controller with Disturbance Rejection

In this subsection, we further design a modified controller based on the previous one to deal with the case with general periodic input disturbance (7.5).

In the previous subsection, a neural controller is designed, which guarantees the optimal solution to the optimal kinematic control problem for the case without input disturbance. To deal with the case with input disturbance, an intuitive way is to provide a mechanism to learn the disturbance and then compensate for it. Based on this intuition, we design the following neural network controller:

$$\begin{aligned} \mathbf{u} = & (-\omega + P_\Omega(\omega - A\omega - J^T(\theta)B(J(\theta)\omega - \dot{\mathbf{r}}_d - \lambda(\mathbf{r}_d \\ & - f(\theta))) - \mathbf{b} - J^T(\theta)\gamma))/\varepsilon - k\delta/\varepsilon - \hat{\eta}, \end{aligned}$$

$$\begin{aligned}
\varepsilon \dot{\gamma} &= J(\theta)\omega - \dot{\mathbf{r}}_d - \lambda(\mathbf{r}_d - f(\theta)), \\
\delta &= \epsilon\omega - \int_0^t (-\omega + P_\Omega(\omega - A\omega - J^T(\theta)B(J(\theta)\omega \\
&\quad - \dot{\mathbf{r}}_d - \lambda(\mathbf{r}_d - f(\theta))) - \mathbf{b} - J^T(\theta)\gamma))), \\
\hat{\eta}(t) &= \hat{\eta}(t - T) + \beta\delta(t),
\end{aligned} \tag{7.18}$$

where $k > 0 \in \mathbb{R}$ and $\beta > 0 \in \mathbb{R}$ are design parameters; $\hat{\eta}$ is the estimation of the periodic input disturbance η with $\hat{\eta}(t) = 0$, $\forall t < 0$.

About the difference between controller (7.18) and controller (7.14), we offer the following remark.

Remark 7.3 Compared with controller (7.14), a disturbance learning component and a disturbance compensation component are added in controller (7.18). Specifically, two terms are added in the expression of \mathbf{u} , where the term $-\hat{\eta}$ is used to compensate for the periodic input disturbance by its estimation and the term $-k\delta/\varepsilon$ is added to guarantee system stability. The disturbance estimation component is described via the formula $\hat{\eta} = \hat{\eta}(t - T) + \beta\delta$, which has two properties: (1) The period information of the input disturbance is used; (2) The update of the disturbance estimation is based on the integral of the difference between the actual joint velocity and the one that guarantees the stability of the closed dynamics and optimality of the kinematic control solution.

About the performance of the presented controller (7.18), we have the following theorem.

Theorem 7.1 *When neural network controller (7.18) is adopted, the optimization problem (7.11) is asymptotically solved in the presence of periodic input disturbance (7.5).*

Proof Let

$$\tilde{\eta} = \hat{\eta} - \eta. \tag{7.19}$$

Substituting (7.18) into (7.4) yields

$$\begin{aligned}
\varepsilon \dot{\omega} &= -\omega + P_\Omega(\omega - A\omega - J^T(\theta)B(J(\theta)\omega - \dot{\mathbf{r}}_d - \lambda(\mathbf{r}_d \\
&\quad - f(\theta))) - \mathbf{b} - J^T(\theta)\gamma) - k\delta - \varepsilon\tilde{\eta}, \\
\varepsilon \dot{\gamma} &= J(\theta)\omega - \dot{\mathbf{r}}_d - \lambda(\mathbf{r}_d - f(\theta)).
\end{aligned} \tag{7.20}$$

In view of (7.19), (7.5), and $\hat{\eta} = \hat{\eta}(t - T) + \beta\delta(t)$ in (7.18), we have

$$\tilde{\eta}(t) = \tilde{\eta}(t - T) + \beta\delta(t) \tag{7.21}$$

From (7.18), we have

$$\dot{\delta} = -\epsilon \dot{\omega} + \omega - P_Q(\omega - A\omega - J^T(\theta)B(J(\theta)\omega - \dot{\mathbf{r}}_d - \lambda(\mathbf{r}_d - f(\theta))) - \mathbf{b} - J^T(\theta)\gamma),$$

which, together with the first equation of (7.20), gives

$$\dot{\delta}(t) = -k\delta(t) - \tilde{\eta}(t). \quad (7.22)$$

By Laplacian transformation [63], for the i th subsystem of (7.21) and (7.22), we have

$$\tilde{\eta}_i(s) = \tilde{\eta}_i(s) \exp(-sT) + \beta \delta_i(s), \quad (7.23)$$

$$s\delta_i(s) - \delta_i(0) = -k\delta_i(s) - \tilde{\eta}_i(s). \quad (7.24)$$

Substituting (7.23) into (7.24) yields

$$\delta_i(s) = \frac{\delta_i(0)}{s + k + \frac{\beta}{1 - \exp(-sT)}}, \quad (7.25)$$

where the denominator of the transfer function is

$$s + k + \frac{\beta}{1 - \exp(-sT)}.$$

To check the stability of the system (7.25), we only need to consider the roots of the following equation:

$$s + k + \frac{\beta}{1 - \exp(-sT)} = 0. \quad (7.26)$$

Let the root of (7.26) be denoted by $s = a + bj$ with $a \in \mathbb{R}$ and $b \in \mathbb{R}$. Then, substituting $s = a + bj$ into (7.26) yields

$$\begin{aligned} \exp(-aT) \exp(-bTj) &= 1 + \frac{\beta}{k + a + bj} \\ &= \frac{(k + a)^2 + \beta(k + a) + b^2 - \beta bj}{(k + a)^2 + b^2}. \end{aligned} \quad (7.27)$$

Since $\exp(-bTj) = \cos(-bT) + j \sin(-bT)$, for the real part of (7.27), we have

$$\exp(-aT) \cos(-bT) = \frac{(k + a)^2 + \beta(k + a) + b^2}{(k + a)^2 + b^2}. \quad (7.28)$$

Assume that $a \geq 0$. Then, for the left-hand of Eq.(7.28), we have

$$\exp(-aT) \cos(-bT) \leq 1 \quad (7.29)$$

because $a \geq 0$, $T > 0$, and $\cos(-bT) \leq 1$. However, for the right-hand of Eq. (7.28), we have

$$\frac{(k+a)^2 + \beta(k+a) + b^2}{(k+a)^2 + b^2} = 1 + \frac{\beta(k+a)}{(k+a)^2 + b^2} > 1,$$

because $k > 0$, $a > 0$, and $\beta > 0$. This, together with (7.29) yields

$$\exp(-aT) \exp(-bT) < \frac{(k+a)^2 + \beta(k+a) + b^2}{(k+a)^2 + b^2},$$

which contradicts with (7.28). Thus, we conclude that the solution of Eq. (7.26), i.e., the poles of (7.25) are located in the left complex plane. It follows that the final-value theorem [63] applies to (7.25) and we have

$$\begin{aligned} \lim_{t \rightarrow +\infty} \delta_i(t) &= \lim_{s \rightarrow 0} s \delta_i(s) \\ &= \lim_{s \rightarrow 0} \frac{s \delta_i(0)}{s + k + \frac{\beta}{1 - \exp(-sT)}} \\ &= 0. \end{aligned}$$

Besides, from (7.24) and (7.25), we have

$$\tilde{\eta}_i(s) = \frac{\frac{\delta_i(0)}{s + k + \frac{\beta}{1 - \exp(-sT)}}}{-(k-s) + s + k + \frac{\beta}{1 - \exp(-sT)}}.$$

Then, by following the steps for the analysis of $\delta_i(s)$, it is concluded that $\lim_{t \rightarrow +\infty} \tilde{\eta}_i(t) = 0$. Note that $\lim_{t \rightarrow +\infty} \tilde{\eta}(t) = 0$ and $\lim_{t \rightarrow +\infty} \delta(t) = 0$ mean that (7.20) ultimately reduces to (7.13). Then, according to the theoretical analysis in Lemma 7.1, we further conclude that when neural network controller (7.18) is adopted, the optimization problem (7.11) is asymptotically solved. The proof is complete. \square

According to Theorem 7.1, by the presented method, the optimization problem is asymptotically solved, by which the performance index asymptotically converges to be optimal, and the constraints are asymptotically satisfied. As a result, the position error asymptotically converges to zero.

7.4 Simulation Examples and Comparisons

In this section, simulation examples and comparisons are presented to validate the theoretical results, the efficacy, and the superiority of the presented method for the kinematic control of redundant manipulators in both regulation tasks and tracking tasks in the presence of periodic input disturbances.

Table 7.1 D–H Parameters of the PUMA 560 Manipulator

Link	a_i (m)	α_i (rad)	d_i (m)
1	0	$\pi/2$	0.67
2	0.4318	0	0
3	0.438	$-\pi/2$	0.15005
4	0	$\pi/2$	0
5	0	$-\pi/2$	0
6	0	0	0.2

7.4.1 Simulation Setup

The simulations are conducted on MATLAB based on a six degrees-of-freedom PUMA 560 manipulator, which has the physical structure that is widely adopted in industrial manipulators. In the simulation, we are concerned about the regulation or tracking of the end-effector position $\mathbf{r} \in \mathbb{R}^3$. Thus, the manipulator can be viewed as a redundant one. The D–H parameters of the manipulator is listed in Table 7.1. Based on the D–H convention [54] and the table, the forward kinematics of the manipulator is derived. The bounds of the joint limits of the manipulator are

$$\begin{aligned}\theta^- &= [-2.7925, -4.2761, -0.7854, -1.9199, -1.7453, -4.6426]^T \text{rad}, \\ \theta^+ &= [2.7925, 0.7854, 3.9270, 2.9671, 1.7453, 4.6426]^T \text{rad}, \\ \omega^- &= [-0.5, -0.5, -0.5, -0.5, -0.5, -0.5]^T \text{rad/s}, \\ \omega^+ &= [0.5, 0.5, 0.5, 0.5, 0.5, 0.5]^T \text{rad/s}.\end{aligned}$$

Note that the joint constraints are set for the purpose of validating the performance of the controller in avoiding joint constraint violation. In the simulations, we set $\lambda = 5$, $\varepsilon = 0.01$, $\beta = 100,000$, $\alpha = 20$, and $B = 10I \in \mathbb{R}^3$.

7.4.2 End-Effector Regulation

When the desired end-effector coordinate $\mathbf{r}_d(t)$ becomes a constant vector, i.e., $\dot{\mathbf{r}}_d(t) \equiv 0$, the task is called a regulation task. In this subsection, the simulation results for the end-effector regulation of the manipulator are presented and compared. Specifically, the desired end-effector coordinate is given by $\mathbf{r}_d = [0.2, 0.2, 1.5]^T \text{ m}$. The parameters of the performance index in (7.11) is set to $A = I \in \mathbb{R}^{6 \times 6}$ and $\mathbf{b} = 0$, which requires the angle velocity norm to be minimized.

We first consider the case that the input disturbance is a sinusoidal signal $\eta(t) = [\eta_1(t), \eta_2(t), \dots, \eta_6(t)]^T$ with $\eta_i(t) = 100 \sin(20\pi t)$, $\forall i = 1, 2, \dots, 6$, which is a periodic disturbance with the period being $T = 0.1 \text{ s}$. As seen from Figs. 7.2 to 7.3, when the presented controller (7.18) is adopted, both the joint angle limit and joint

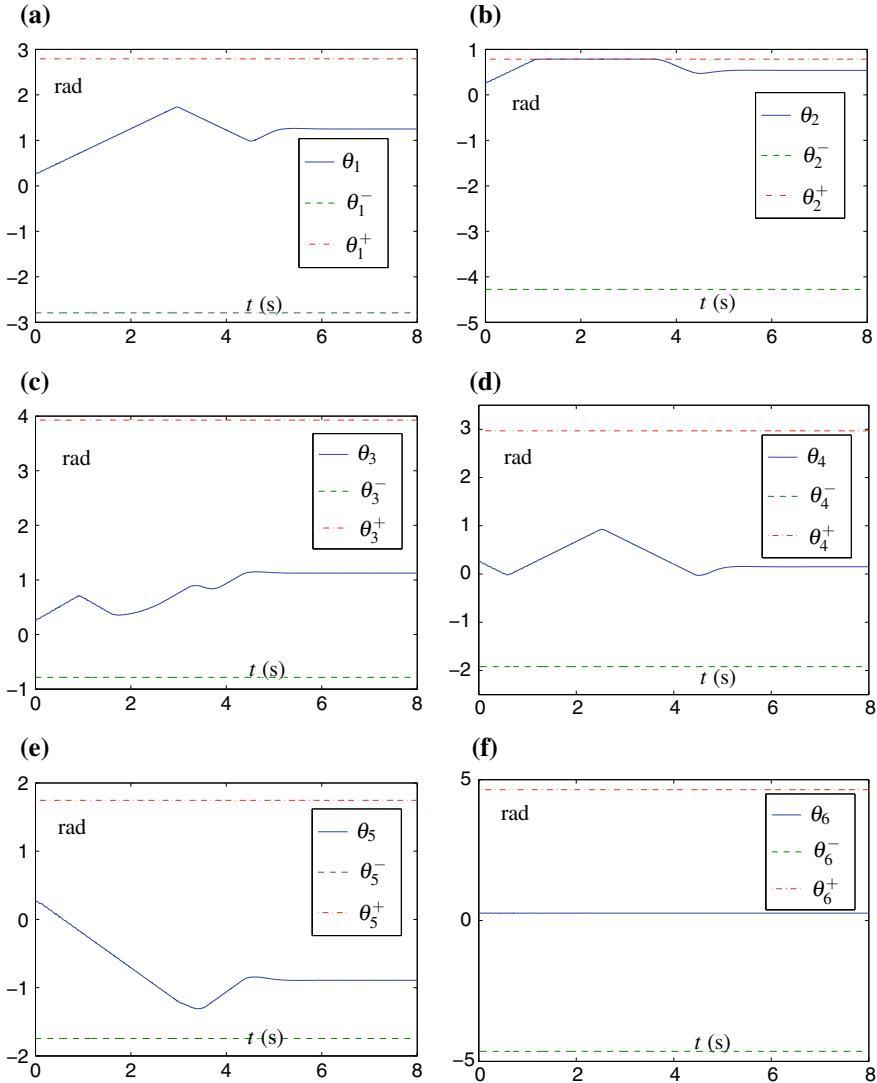


Fig. 7.2 Profiles of joint angles during the end-effector regulation process of the PUMA 560 manipulator via the presented controller (7.18) in the presence of input disturbance $\eta(t) = [\eta_1(t), \eta_2(t), \dots, \eta_6(t)]^T$ with $\eta_i(t) = 100 \sin(20\pi t)$, $\forall i = 1, 2, \dots, 6$, for which the period $T = 0.1$ s. **a** Profile of θ_1 . **b** Profile of θ_2 . **c** Profile of θ_3 . **d** Profile of θ_4 . **e** Profile of θ_5 . **f** Profile of θ_6

velocity limit are satisfied after several seconds. As seen from Fig. 7.3a, the frequency of the oscillation of the joint velocity is about 10Hz, which is consistent with the fact that the disturbance is of the sinusoidal form with the period being 0.1 s. This is normal, because the input disturbance is added on the acceleration of the joint

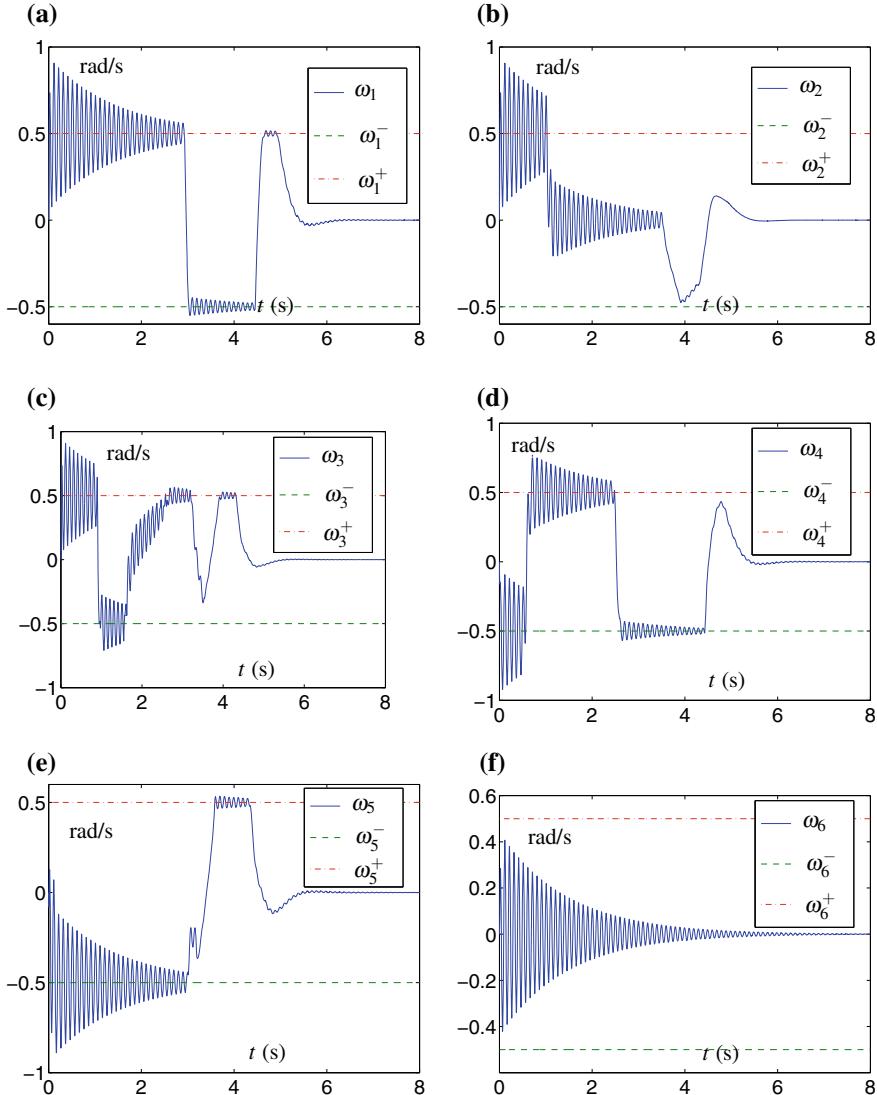


Fig. 7.3 Profiles of joint velocities during the end-effector regulation process of the PUMA 560 manipulator via the presented controller (7.18) in the presence of input disturbance $\eta(t) = [\eta_1(t), \eta_2(t), \dots, \eta_6(t)]^T$ with $\eta_i(t) = 100 \sin(20\pi t)$, $\forall i = 1, 2, \dots, 6$, for which the period $T = 0.1$ s. **a** Profile of ω_1 . **b** Profile of ω_2 . **c** Profile of ω_3 . **d** Profile of ω_4 . **e** Profile of ω_5 . **f** Profile of ω_6

angle and the magnitude of the disturbance is large enough. As seen from Fig. 7.4a, $\|\delta\|_2$ asymptotically converges to zero, which indicates that $\lim_{t \rightarrow +\infty} \delta(t) = 0$. The joint accelerations are shown in Fig. 7.5, which are bounded. Besides, Fig. 7.4b shows

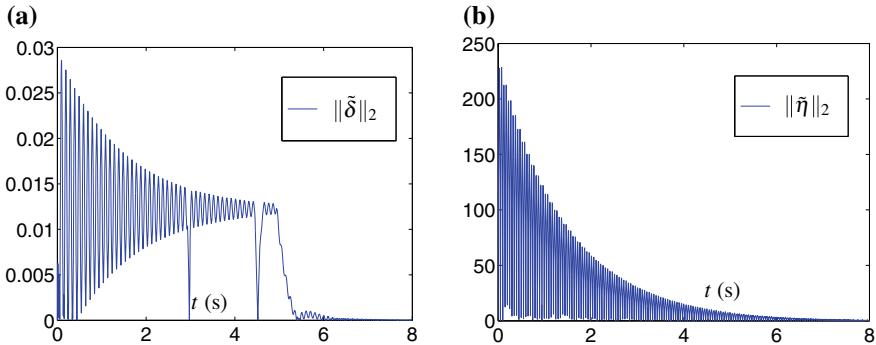
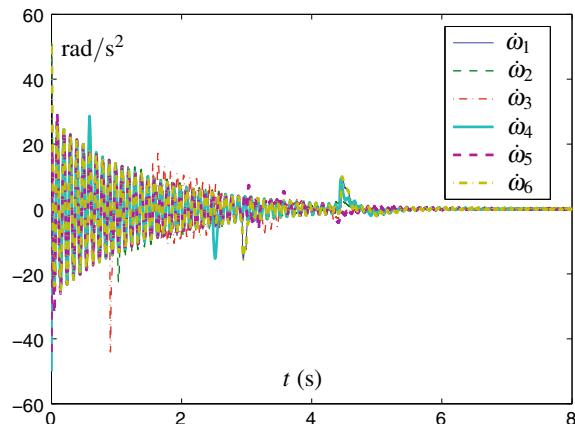


Fig. 7.4 Profiles of the norm of δ in the controller (7.18), i.e., $\|\delta\|_2$, and the norm of disturbance estimation error, i.e., $\|\tilde{\eta}\|_2$ with $\tilde{\eta}(t) = \hat{\eta}(t) - \eta(t)$, during the end-effector regulation process of the PUMA 560 manipulator via the presented controller (7.18) in the presence of input disturbance $\eta(t) = [\eta_1(t), \eta_2(t), \dots, \eta_6(t)]^T$ with $\eta_i(t) = 100 \sin(20\pi t)$, $\forall i = 1, 2, \dots, 6$, for which the period $T = 0.1$ s. **a** Profile of $\|\delta\|_2$. **b** Profile of $\|\tilde{\eta}\|_2$

Fig. 7.5 Profiles of joint accelerations during the end-effector regulation process of the PUMA 560 manipulator via the presented controller (7.18) in the presence of input disturbance $\eta(t) = [\eta_1(t), \eta_2(t), \dots, \eta_6(t)]^T$ with $\eta_i(t) = 100 \sin(20\pi t)$, $\forall i = 1, 2, \dots, 6$, for which the period $T = 0.1$ s



that the norm of disturbance estimation error $\|\tilde{\eta}\|_2$ converges to zero, which indicates that $\lim_{t \rightarrow +\infty} \tilde{\eta}(t) = 0$. In addition, as seen from Fig. 7.6, the regulation error e also converges to zero with the end-effector coordinate $r = [r_X, r_Y, r_Z]$ converging to $r_d = [0.2, 0.2, 1.5]^T$ m. The corresponding profile of the performance index during the end-effector regulation process is shown in Fig. 7.7.

Under the same setups, simulations are also conducted for the case with other types of periodic input disturbances to further show the performance of the presented controller (7.18) for the end-effector regulation of the PUMA 560 manipulator. Specifically, sawtooth and square disturbances are considered. In the simulations, the periods are set to 0.02 s and 0.01 s. As shown in Fig. 7.8, in the presence of the two types of periodic input disturbances, the regulation errors also converge to zero. It is also found that the joint constraints are satisfied in an asymptotic manner, which

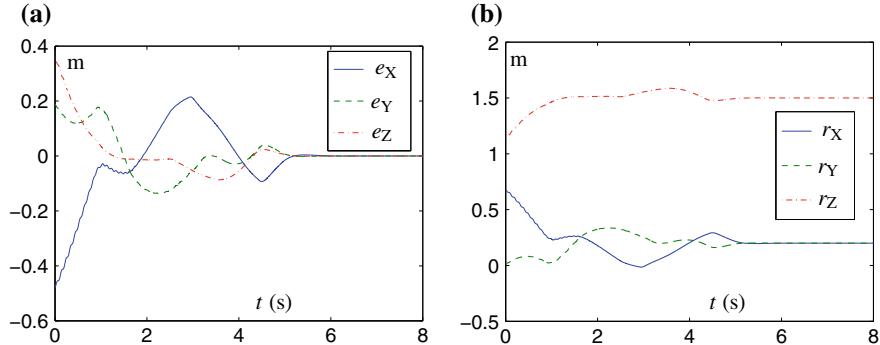
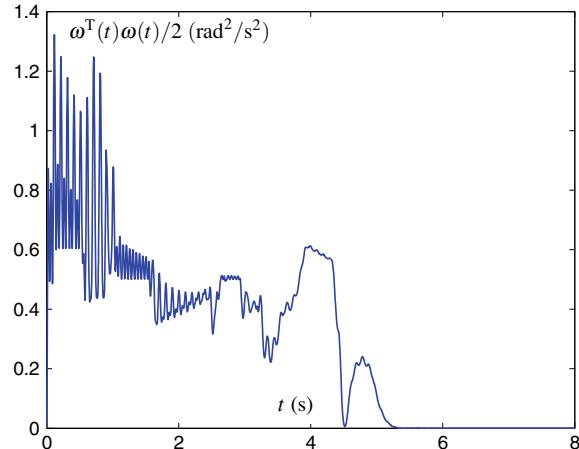


Fig. 7.6 Profiles of regulation error $\mathbf{e}(t) = [e_X(t), e_Y(t), e_Z(t)]^T = \mathbf{r}_d - \mathbf{r}(t)$ and end-effector coordinate $\mathbf{r}(t)$ during the end-effector regulation process of the PUMA 560 manipulator via the presented controller (7.18) in the presence of input disturbance $\eta(t) = [\eta_1(t), \eta_2(t), \dots, \eta_6(t)]^T$ with $\eta_i(t) = 100 \sin(20\pi t)$, $\forall i = 1, 2, \dots, 6$, for which the period $T = 0.1$ s. **a** Profiles of $\mathbf{e}(t)$. **b** Profiles of $\mathbf{r}(t)$

Fig. 7.7 Profiles of performance index $\omega^T(t)\omega(t)/2$ during the end-effector regulation process of the manipulator via the presented controller (7.18)



are omitted due to that fact that they are quite similar to Figs. 7.2 and 7.3. All the results are consistent with Theorem 7.1, and show that the presented controller (7.18) is effective for the end-effector regulation of redundant manipulators subject to joint angle and velocity constraints in the presence of different types of periodic input disturbances. Before ending this subsection, we would like to point out that, to the best of our knowledge, the kinematic control of redundant manipulators for regulation tasks with complete input disturbance learning and compensation has not been studied in the existing literature.

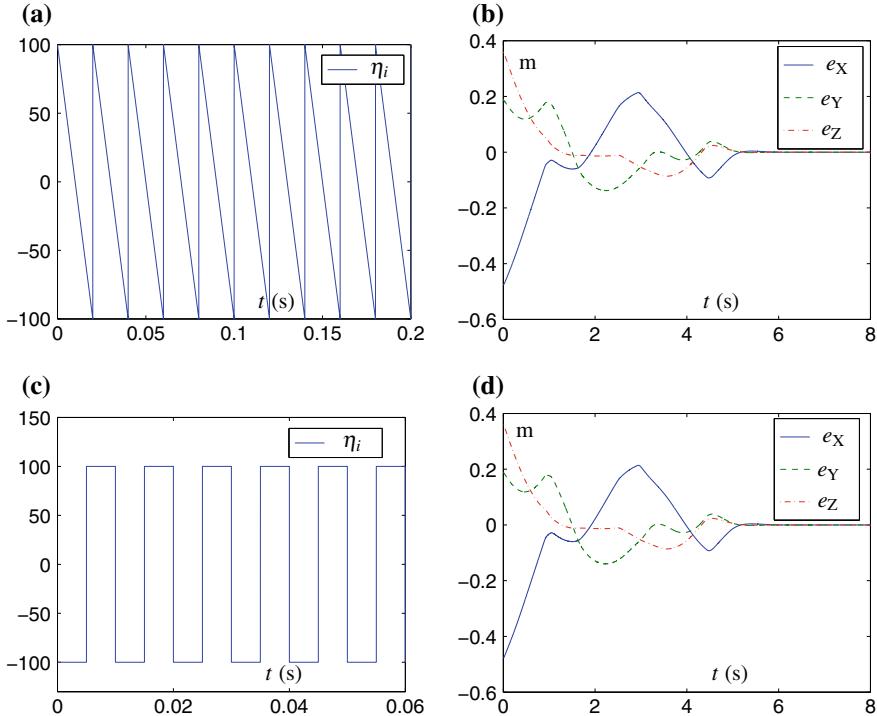


Fig. 7.8 Profiles of periodic input disturbances $\eta(t) = [\eta_1(t), \eta_2(t), \dots, \eta_6(t)]^T$ with $\eta_1(t) = \eta_2(t) = \dots = \eta_6(t)$ and the corresponding regulation errors $\mathbf{e}(t) = [e_X(t), e_Y(t), e_Z(t)]^T = \mathbf{r}_d - \mathbf{r}(t)$ during the end-effector regulation process of the PUMA 560 manipulator via the presented controller (7.18). **a** The profiles of each element of the sawtooth disturbance at $t \in [0, 0.2]$ s. **b** The regulation error corresponding to the case with the sawtooth disturbance. **c** The profiles of each element of the square disturbance at $t \in [0, 0.06]$ s. **d** The regulation error corresponding to the case with the square disturbance

7.4.3 End-Effector Tracking

In this subsection, the simulation results for the end-effector tracking of the manipulator are presented and compared. All the parameter settings are the same as those described in the previous subsection.

In the simulations, the desired end-effector trajectory is described as follows:

$$\mathbf{r}(t) = \begin{bmatrix} r_X = R \cos\left(2\pi \sin^2\left(\frac{\pi t}{2T}\right)\right) + r_X(0) - R \\ R \sin\left(2\pi \sin^2\left(\frac{\pi t}{2T}\right)\right) + r_Y(0) \\ r_Z(0) \end{bmatrix},$$

with $R = 0.1$ m, $T = 10$ s, and $\mathbf{r}(0) = [r_X(0), r_Y(0), r_Z(0)]^T$. We first consider sinusoidal input disturbance. As seen from Fig. 7.9a, when the presented controller is

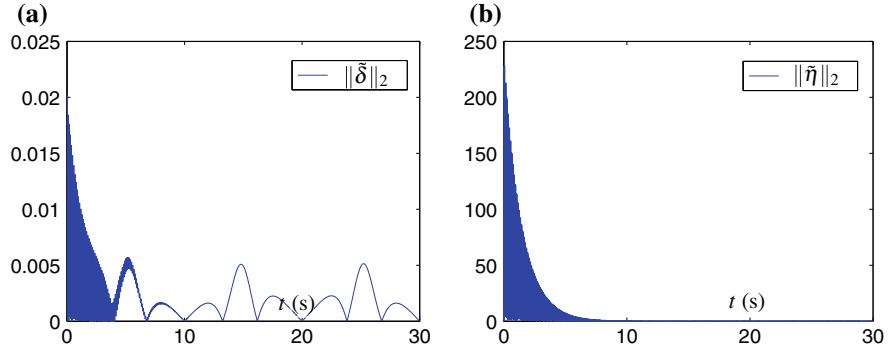


Fig. 7.9 Profiles of the norm of δ in the controller (7.18), i.e., $\|\delta\|_2$, and the norm of disturbance estimation error, i.e., $\|\tilde{\eta}\|_2$ with $\tilde{\eta}(t) = \hat{\eta}(t) - \eta(t)$, during the end-effector tracking process of the PUMA 560 manipulator via the presented controller (7.18) in the presence of input disturbance $\eta(t) = [\eta_1(t), \eta_2(t), \dots, \eta_6(t)]^T$ with $\eta_i(t) = 100 \sin(20\pi t)$, $\forall i = 1, 2, \dots, 6$, for which the period $T = 0.1$ s. **a** Profile of $\|\delta\|_2$. **b** Profile of $\|\tilde{\eta}\|_2$

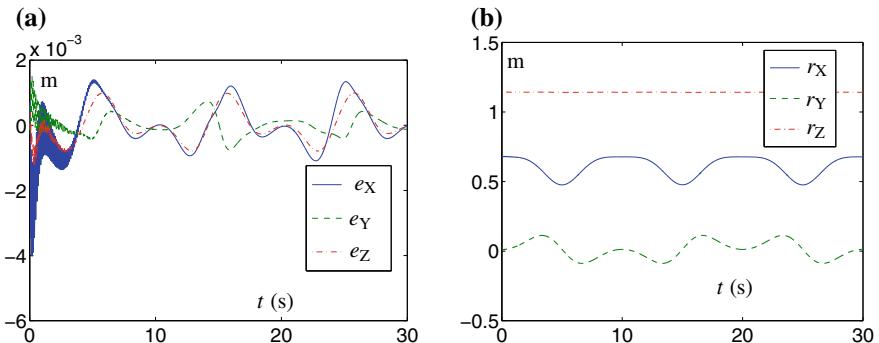


Fig. 7.10 Profiles of regulation error $e(t) = [e_X(t), e_Y(t), e_Z(t)]^T = \mathbf{r}_d - \mathbf{r}(t)$ and the end-effector coordinate $\mathbf{r}(t)$ during the end-effector tracking process of the PUMA 560 manipulator via the presented controller (7.18) in the presence of input disturbance $\eta(t) = [\eta_1(t), \eta_2(t), \dots, \eta_6(t)]^T$ with $\eta_i(t) = 100 \sin(20\pi t)$, $\forall i = 1, 2, \dots, 6$, for which the period $T = 0.1$ s. **a** Profiles of $e(t)$. **b** Profiles of $\mathbf{r}(t)$

adopted, the norm of $\delta(t)$, i.e., $\|\delta(t)\|_2$, is very small during the tracking process and is less than 5×10^{-3} for most of the time. As seen from Fig. 7.9b, the norm of $\tilde{\eta}(t)$, i.e., $\|\tilde{\eta}(t)\|_2$, quickly vanishes to zero with time. Besides, Fig. 7.10 shows that the tracking error is very small, the magnitude of which is less than 2×10^{-3} m, i.e., 2 mm. Figures 7.11 and 7.12 show that the joint constraints are asymptotically satisfied. In addition, Fig. 7.13 shows that the joint accelerations are bounded during the whole process.

For comparison, the following controller proposed in [30] is also considered:

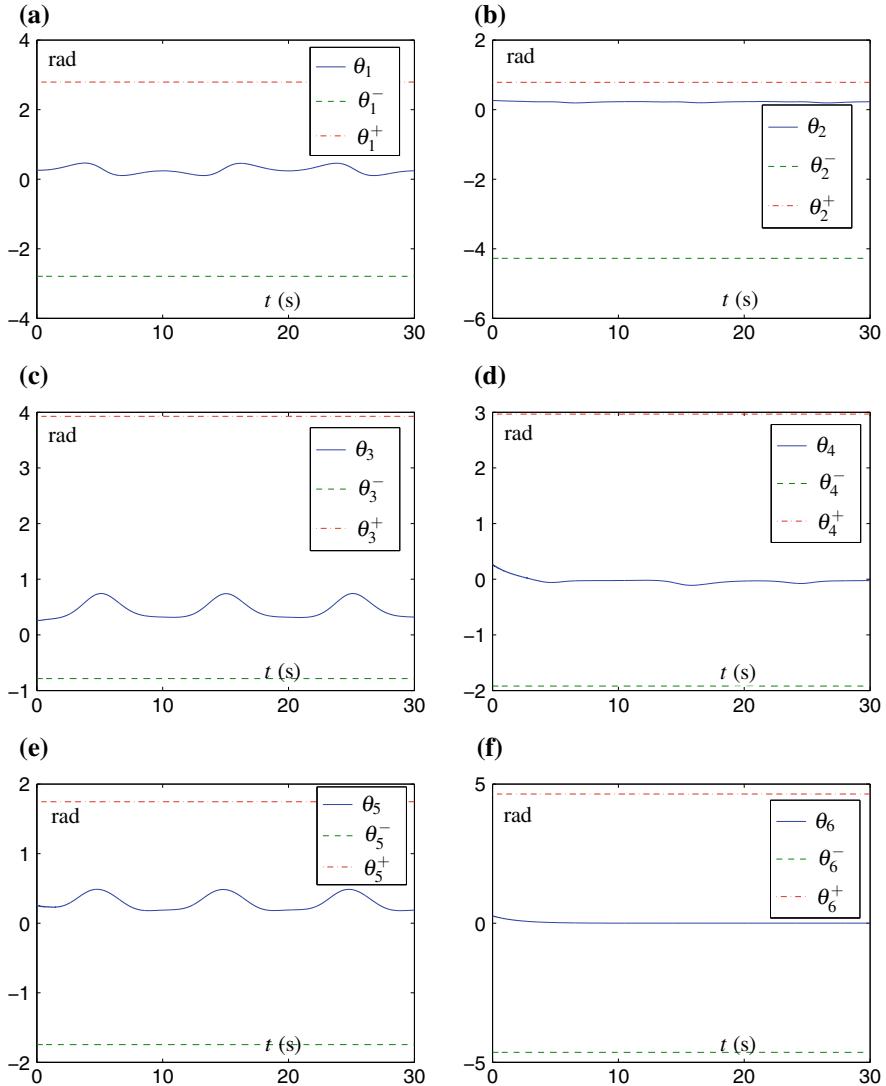


Fig. 7.11 Profiles of joint angles during the end-effector tracking process of the PUMA 560 manipulator via the presented controller (7.18) in the presence of input disturbance $\eta(t) = [\eta_1(t), \eta_2(t), \dots, \eta_6(t)]^\top$ with $\eta_i(t) = 100 \sin(20\pi t)$, $\forall i = 1, 2, \dots, 6$, for which the period $T = 0.1$ s. **a** Profile of θ_1 . **b** Profile of θ_2 . **c** Profile of θ_3 . **d** Profile of θ_4 . **e** Profile of θ_5 . **f** Profile of θ_6

$$\begin{aligned} \mathbf{u}(t) &= (-\omega(t) + P_\Omega(J^\top(\theta(t))\gamma(t)))/\varepsilon, \\ \varepsilon\dot{\gamma}(t) &= \dot{\mathbf{r}}_d(t) - J^\top(\theta(t))\omega(t), \end{aligned} \quad (7.30)$$

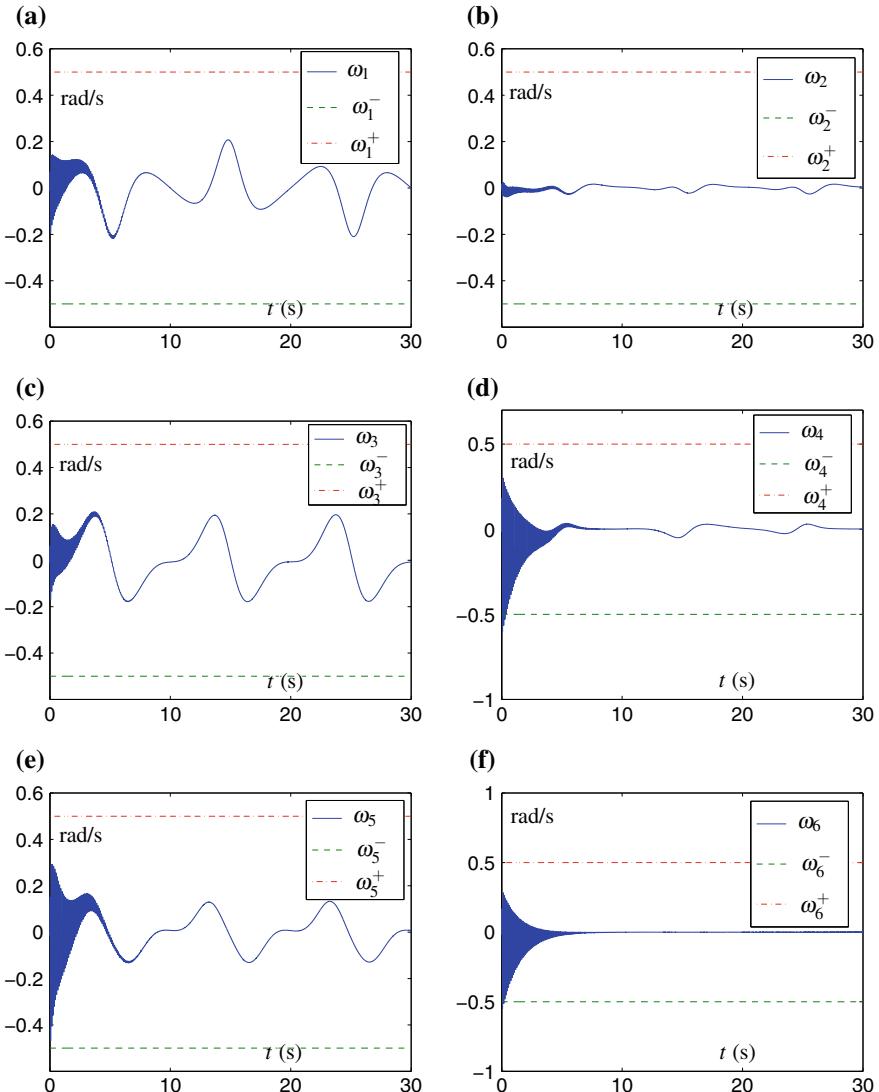
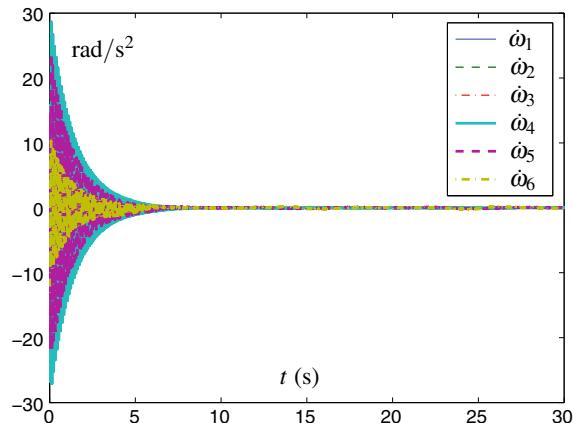


Fig. 7.12 Profiles of joint velocities during the end-effector tracking process of the PUMA 560 manipulator via the presented controller (7.18) in the presence of input disturbance $\eta(t) = [\eta_1(t), \eta_2(t), \dots, \eta_6(t)]^T$ with $\eta_i(t) = 100 \sin(20\pi t)$, $\forall i = 1, 2, \dots, 6$, for which the period $T = 0.1$ s. **a** Profile of ω_1 . **b** Profile of ω_2 . **c** Profile of ω_3 . **d** Profile of ω_4 . **e** Profile of ω_5 . **f** Profile of ω_6

where $\varepsilon > 0 \in \mathbb{R}$. With the same settings and initial states, but in the presence of different types of input disturbances, the comparisons regarding the end-effector tracking error norm when different controllers are performed, where the unbiased sinusoidal input disturbance, the sawtooth input disturbance, and the square input

Fig. 7.13 Profiles of joint accelerations during the end-effector tracking process of the PUMA 560 manipulator via the presented controller (7.18) in the presence of input disturbance $\eta(t) = [\eta_1(t), \eta_2(t), \dots, \eta_6(t)]^T$ with $\eta_i(t) = 100 \sin(20\pi t)$, $\forall i = 1, 2, \dots, 6$, for which the period $T = 0.1$ s



disturbance with different periods and magnitudes are considered. For fair comparison, all the setups for controller (7.30) are the same as those for controller (7.18). It is found that the presented method has a consistent performance for different types of periodic input disturbances regardless of their magnitudes and periods, where the mean of $\|\mathbf{e}(t)\|_2$, i.e., the tracking error norm, is of order 10^{-4} m, i.e., 0.1 mm for the case that the magnitude of the disturbance is $1/\varepsilon = 100$, and are of order 10^{-3} m, i.e., 1 mm for the case that the magnitude of the disturbance is $10/\varepsilon = 1000$. In addition, the changes in the period of the disturbance have very small effects on the performance regarding the tracking error norm of the presented controller. Comparatively, the presented controller (7.18) performs much better than controller (7.30) proposed in [30] in almost all the simulations. For example, for a sinusoidal input disturbance with period 0.1 s and magnitude $1/\varepsilon$, the maximal tracking error norm for the presented controller is 4.32×10^{-3} m which is much smaller than that for controller (7.30), i.e., 2.20×10^{-2} m. In this scenario, the performance difference is more significant in terms of the mean of the tracking error norm. Specifically, the mean of the tracking error norm for the presented controller is 7.84×10^{-4} m, which is 1.25×10^{-2} m for controller (7.30). Meanwhile, it can be found that controller (7.30) does not have a consistent performance for different types of input disturbances and is sensitive to the period of the input disturbance. Besides, as seen from Fig. 7.14, in the presence of different types of periodic input disturbances, the value of the performance index is much better when the presented controller is used compared with the case that the controller (7.30) presented in [30] is used. Intuitively, the method in [30] assumes that there is no disturbance. In contrast, the presented method provides a mechanism to learn and compensate for the disturbance. As a result, the method presented in this chapter performs much better than the one proposed in [30] in the presence of periodic input disturbances.

By the analysis on the above simulation results and comparisons, we conclude that the simulation results are consistent with Theorem 7.1 and validate the

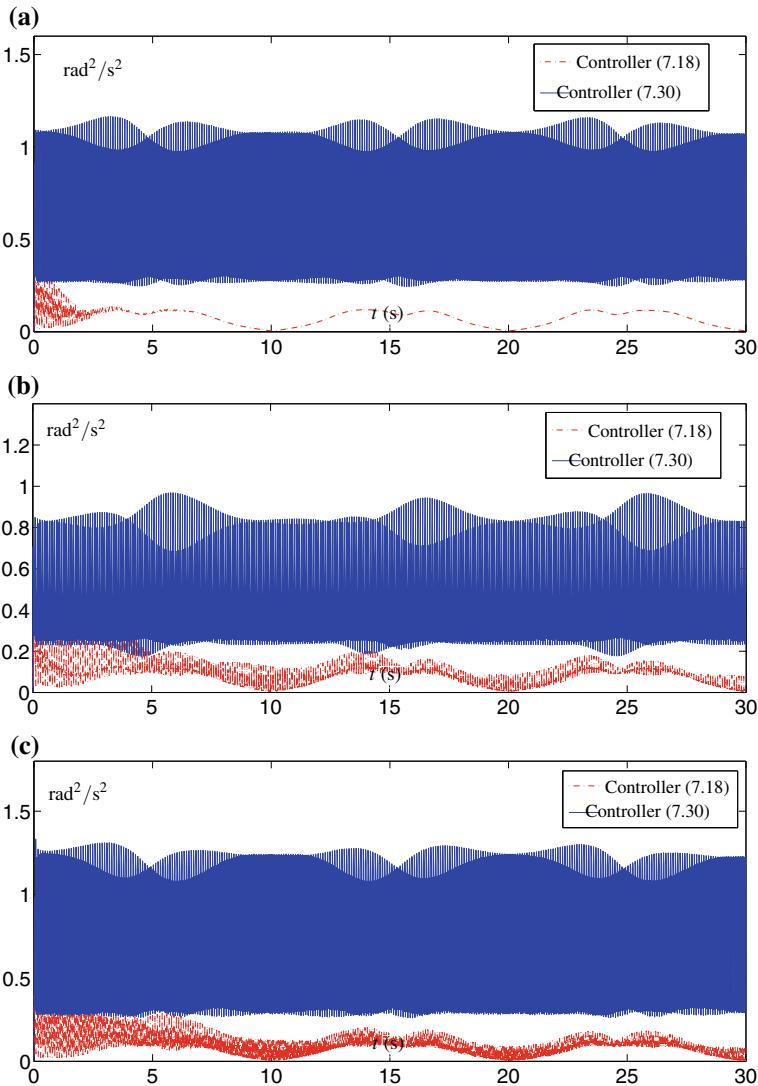


Fig. 7.14 Comparison of the profiles of performance index $\omega^T(t)\omega(t)/2$ during the end-effector tracking process when different controllers, i.e., controller (7.18) presented in this chapter and controller (7.30) presented in [30], are used in the presence of different types of periodic input disturbances. **a** Sinusoidal disturbance with period 0.01 s and magnitude $1/\varepsilon$. **b** Sawtooth disturbance with period 0.01 s and magnitude $1/\varepsilon$. **c** Square disturbance with period 0.01 s and magnitude $1/\varepsilon$

efficacy and superiority of the presented method for the kinematic control of redundant manipulators in the presence of periodic input disturbances.

7.5 Questions and Answers

In this section, some discussions about earlier versions of this chapter are presented in the form of questions and answers.

Question 7.1 “*To benefit practitioners, examples of periodical input disturbances in practice should be included and discussed.*”

Answer: Because the actuator of each joint of a manipulator is often a motor, the periodic input disturbance can be caused by the cogging torque, which is a result of the interaction between the permanent magnets and the stator slots. This kind of disturbance is of the sinusoidal form. When the manipulator is executing periodic tasks, the difference between the actual joint acceleration and the given one can also be viewed as a periodic input disturbance. In this case, the period of the disturbance is the period of the periodic task.

Question 7.2 “*It is expected to see theoretical analysis and discussion on possible reasons why the proposed controller performs much better than [16]. This may be instructive for future researchers in control design.*”

Answer: Intuitively, the method in [16] assumes that there is no disturbance. In contrast, the proposed method provides a mechanism to learn and compensate the disturbance. As a result, the proposed method performs much better than the one proposed. Since this work focuses on presenting the main results of the proposed method, we do not provide theoretical analysis on the robustness of the method proposed in [16].

Question 7.3 “*There are several parameters in the proposed neural network controller. It would be better if the authors can provide some remarks about how to set the values of those parameters in practice.*”

Answer: In the proposed neural network controller, there are three parameters, i.e., ε , k , and β . The requirement on the parameter setting is $\varepsilon > 0$, $k > 0$, and $\beta > 0$.

Question 7.4 “*Regulation tasks are suggested to explained clearly for the reader’ understanding.*”

Answer: When the desired end-effector coordinate $\mathbf{r}_d(t)$ becomes a constant vector, i.e., $\dot{\mathbf{r}}_d(t) \equiv 0$, the task is called a regulation task.

Question 7.5 “*For Eq. (4), the variable $u(t)$ is suggested to be explained clearly for the reader understanding.*”

Answer: The variable $\mathbf{u}(t)$ denotes the given end-effector joint acceleration command, which is the input of the kinematic control problem. This type of control is referred to as acceleration-level kinematic control.

Question 7.6 “*The oscillation appearing in joint velocities is more dense, and can this velocities be applied to practice?*”

Answer: As seen from the figures, the frequency of the oscillation appearing in joint velocities is almost the same as the frequency of the input disturbance. For example, as seen from Fig. 4a, the frequency of the oscillation of the velocity is about 10 Hz, which is consistent with the fact that the disturbance is of the sinusoidal form with the period being 0.1 s. This is normal, because the input disturbance is added on the acceleration of the joint angle and the magnitude of the disturbance is large enough. As a result, the velocities applied to practice.

7.6 Summary

In this chapter, a neural network-based kinematic control method has been presented for redundant manipulators in the presence of general periodic input disturbances. Both joint constraints and performance indices are considered in the chapter for both regulation and tracking tasks. Theoretical analysis and simulation examples have shown that, by using the presented controller, the formulated constrained optimization problems for the kinematic control problem can be asymptotically solved with the end-effector coordinate error asymptotically converging to zero, while the input disturbance can be asymptotically learned and rejected. The comparisons with an existing controller have further shown the superiority of the presented method. Before ending this chapter, it is worth noting that, to the best of the authors’ knowledge, this is the first work for neural network-based kinematic control for redundant manipulators subject to joint constraints and general periodic input disturbances with theoretically guaranteed performance. In addition, the extension of the presented method to the cooperative control of multiple redundant manipulators may be investigated in the future.

References

1. Li, S., Zhang, Y., Jin, L.: Kinematic control of redundant manipulators using neural networks. *IEEE Trans. Neural Netw. Learn. Syst.* **28**(10), 2243–2254 (2017)
2. Klein, C.A., Huang, C.-H.: Review of pseudoinverse control for use with kinematically redundant manipulators. *IEEE Trans. Syst. Man Cybern. SMC*-**13**(2), 245–250 (1983)
3. Liao, B., Liu, W.: Pseudoinverse-type bi-criteria minimization scheme for redundancy resolution of robot manipulators. *Robotica* **33**(10), 2100–2113 (2015)
4. Flacco, F., Luca, A.: Discrete-time redundancy resolution at the velocity level with acceleration/torque optimization properties. *Robot. Auton. Syst.* **70**, 191–201 (2015)
5. Klein, C.A., Kee, K.B.: The nature of drift in pseudoinverse control of kinematically redundant manipulators. *IEEE Trans. Robot. Autom.* **5**(2), 231–234 (1989)
6. Zhang, Y., Wang, J., Xia, Y.: A dual neural network for redundancy resolution of kinematically redundant manipulators subject to joint limits and joint velocity limits. *IEEE Trans. Neural Netw.* **14**(3), 658–667 (2003)

7. Huang, S., Peng, Y., Wei, W., Xiang, J.: Clamping weighted least-norm method for the manipulator kinematic control with constraints. *Int. J. Control.* **89**(11), 2240–2249 (2016)
8. Cheng, F.T., Chen, T.H., Sun, Y.Y.: Resolving manipulator redundancy under inequality constraints. *IEEE Trans. Robot. Autom.* **10**(1), 65–71 (1994)
9. Patchaikani, P.K., Behera, L., Prasad, G.: A single network adaptive critic-based redundancy resolution scheme for robot manipulators. *IEEE Trans. Ind. Electron.* **59**(8), 3241–3253 (2012)
10. He, W., Huang, B., Dong, Y., Li, Z., Su, C.: Adaptive neural network control for robotic manipulators with unknown deadzone. *IEEE Trans. Cybern.* **48**(9), 2670–2682 (2018)
11. Li, D., Liu, Y., Tong, S., Chen, C.L.P., Li, D.: Neural networks-based adaptive control for nonlinear state constrained systems with input delay. *IEEE Trans. Cybern.* **49**(4), 1249–1258 (2019)
12. Guo, D., Yan, L., Nie, Z.: Design, analysis, and representation of novel five-step dtzd algorithm for time-varying nonlinear optimization. *IEEE Trans. Neural Netw. Learn. Syst.* **29**(9), 4248–4260 (2018)
13. Yang, C., Li, Z., Cui, R., Xu, B.: Neural network-based motion control of underactuated wheeled inverted pendulum models. *IEEE Trans. Neural Netw. Learn. Syst.* **25**(11), 2004–2016 (2014)
14. Jin, L., Li, S., Hu, B., Liu, M., Yu, J.: Noise-suppressing neural algorithm for solving time-varying system of linear equations: a control-based approach. *IEEE Trans. Ind. Inform.* **15**(1), 236–246 (2019)
15. Jin, L., Li, S., Hu, B., Liu, M.: A survey on projection neural networks and their applications. *Appl. Soft Comput.* **76**, 533–544 (2019)
16. Xiao, L., Li, K., Tan, Z., Zhang, Z., Liao, B., Chen, K., Jin, L., Li, S.: Nonlinear gradient neural network for solving system of linear equations. *Inf. Process. Lett.* **142**, 35–40 (2019)
17. Xiang, Q., Liao, B., Xiao, L., Lin, L., Li, S.: Discrete-time noise-tolerant Zhang neural network for dynamic matrix pseudoinverson. *Soft Comput.* **23**(3), 755–766 (2019)
18. Xiao, L., Li, S., Yang, J., Zhang, Z.: A new recurrent neural network with noise-tolerance and finite-time convergence for dynamic quadratic minimization. *Neurocomputing* **285**, 125–132 (2018)
19. Xiao, L., Liao, B., Li, S., Chen, K.: Nonlinear recurrent neural networks for finite-time solution of general time-varying linear matrix equations. *Neural Netw.* **98**, 102–113 (2018)
20. Xiao, L., Zhang, Z., Zhang, Z., Li, W., Li, S.: Design, verification and robotic application of a novel recurrent neural network for computing dynamic Sylvester equation. *Neural Netw.* **105**, 185–196 (2018)
21. Zhang, Z., Lu, Y., Zheng, L., Li, S., Yu, Z., Li, Y.: A new varying-parameter convergent-differential neural-network for solving time-varying convex QP problem constrained by linear-equality. *IEEE Trans. Autom. Control* **63**(12), 4110–4125 (2018)
22. Jin, L., Li, S.: Nonconvex function activated zeroing neural network models for dynamic quadratic programming subject to equality and inequality constraints. *Neurocomputing* **267**, 107–113 (2017)
23. Jin, L., Li, S., Liao, B., Zhang, Z.: Zeroing neural networks: a survey. *Neurocomputing* **267**, 597–604 (2017)
24. Mao, M., Li, J., Jin, L., Li, S., Zhang, Y.: Enhanced discrete-time Zhang neural network for time-variant matrix inversion in the presence of bias noises. *Neurocomputing* **207**, 220–230 (2016)
25. Jin, L., Zhang, Y., Li, S.: Integration-enhanced Zhang neural network for real-time-varying matrix inversion in the presence of various kinds of noises. *IEEE Trans. Neural Netw. Learn. Syst.* **27**(12), 2615–2627 (2016)
26. Li, S., Li, Y.: Nonlinearly activated neural network for solving time-varying complex sylvester equation. *IEEE Trans. Cybern.* **44**(8), 1397–1407 (2014)
27. Li, S., Li, Y., Wang, Z.: A class of finite-time dual neural networks for solving quadratic programming problems and its k-winners-take-all application. *Neural Netw.* **39**, 27–39 (2013)
28. Hopfield, J.J., Tank, D.W.: Neural' computation of decisions in optimization problems. *Biol. Cybern.* **52**(3), 141–152 (1985)

29. Zhang, Y., Chen, S., Li, S., Zhang, Z.: Adaptive projection neural network for kinematic control of redundant manipulators with unknown physical parameters. *IEEE Trans. Ind. Electron.* **65**(6), 4909–4920 (2017)
30. Xia, Y., Feng, G., Wang, J.: A primal-dual neural network for online resolving constrained kinematic redundancy in robot motion control. *IEEE Trans. Syst. Man Cybern. B Cybern.* **35**(1), 54–64 (2005)
31. Zhang, Y., Wang, J., Xu, Y.: A dual neural network for bi-criteria kinematic control of redundant manipulators. *IEEE Trans. Robot. Autom.* **18**(6), 923–931 (2002)
32. Zhang, Y., Wang, J.: Obstacle avoidance for kinematically redundant manipulators using a dual neural network. *IEEE Trans. Syst. Man Cybern. B Cybern.* **34**(1), 752–759 (2004)
33. Zhang, Y., Ge, S.S., Lee, T.H.: A unified quadratic-programming-based dynamical system approach to joint torque optimization of physically constrained redundant manipulators. *IEEE Trans. Syst. Man Cybern. B Cybern.* **34**(5), 2126–2132 (2004)
34. Jin, L., Li, S., La, H.M., Luo, X.: Manipulability optimization of redundant manipulators using dynamic neural networks. *IEEE Trans. Ind. Electron.* **64**(6), 4710–4720 (2017)
35. Zhang, Y., Li, S., Gui, J., Luo, X.: Velocity-level control with compliance to acceleration-level constraints: a novel scheme for manipulator redundancy resolution. *IEEE Trans. Ind. Inform.* **14**(3), 921–930 (2018)
36. Guo, D., Zhang, Y.: Acceleration-level inequality-based MAN scheme for obstacle avoidance of redundant robot manipulators. *IEEE Trans. Ind. Electron.* **61**(12), 6903–6914 (2014)
37. Zhang, Z., et al.: A varying-parameter convergent-differential neural network for solving joint-angular-drift problems of redundant robot manipulators. *IEEE/ASME Trans. Mechatron.* **23**(2), 679–689 (2018)
38. Xiao, L., et al.: Design and analysis of FTZNN applied to the real-time solution of a nonstationary Lyapunov equation and tracking control of a wheeled mobile manipulator. *IEEE Trans. Ind. Inform.* **14**(1), 98–105 (2018)
39. Zhang, Z., Beck, A., Magnenat-Thalmann, N.: Human-like behavior generation based on head-arms model for robot tracking external targets and body parts. *IEEE Trans. Cybern.* **45**(8), 1390–1400 (2015)
40. Chen, D., Zhang, Y.: A hybrid multi-objective scheme applied to redundant robot manipulators. *IEEE Trans. Autom. Sci. Eng.* **14**(3), 1337–1350 (2017)
41. Li, S., Chen, S., Liu, B., Li, Y., Liang, Y.: Decentralized kinematic control of a class of collaborative redundant manipulators via recurrent neural networks. *Neurocomputing* **91**, 1–10 (2012)
42. Li, S., He, J., Li, Y., Rafique, U.: Distributed recurrent neural networks for cooperative control of manipulators: a game-theoretic perspective. *IEEE Trans. Neural Netw. Learn. Syst.* **28**(2), 415–426 (2017)
43. Jin, L., Zhang, Y.: G2-type SRMPC scheme for synchronous manipulation of two redundant robot arms. *IEEE Trans. Cybern.* **45**(2), 153–164 (2015)
44. Hou, Z.G., Cheng, L., Tan, M.: Multicriteria optimization for coordination of redundant robots using a dual neural network. *IEEE Trans. Syst. Man Cybern. B Cybern.* **40**(4) 1075–1087 (2010)
45. Jin, L., Li, S., Luo, X., Li, Y., Qin, B.: Neural dynamics for cooperative control of redundant robot manipulators. *IEEE Trans. Ind. Inform.* **14**(9), 3812–3821 (2018)
46. Zhang, Y., Li, S.: A neural controller for image-based visual servoing of manipulators with physical constraints. *IEEE Trans. Neural Netw. Learn. Syst.* **29**(11), 5419–5429 (2018)
47. Jin, L., Li, S.: Distributed task allocation of multiple robots: a control perspective. *IEEE Trans. Syst. Man Cybern. Syst.* **48**(5), 693–701 (2018)
48. Wang, Z., et al.: Neural network learning adaptive robust control of an industrial linear motor-driven stage with disturbance rejection ability. *IEEE Trans. Ind. Inform.* **13**(5), 2172–2183 (2017)
49. Chen, W., Yang, J., Guo, L., Li, S.: Disturbance-observer-based control and related methods—an overview. *IEEE Trans. Ind. Electron.* **63**(2), 1083–1095 (2016)

50. Liu, F., Li, Y., Cao, Y., She, J., Wu, M.: A two-layer active disturbance rejection controller design for load frequency control of interconnected power system. *IEEE Trans. Power Electorn.* **31**(4), 3320–3321 (2016)
51. Fedele, G., Ferrise, A.: On the uncertainty on the phase of a stable linear system in the periodic disturbance cancellation problem. *IEEE Trans. Autom. Control* **61**(9), 2720–2726 (2016)
52. Muramatsu, H., Katsura, S.: An Adaptive periodic-disturbance observer for periodic-disturbance suppression. *IEEE Trans. Ind. Inform.* **14**(10), 4446–4456 (2018)
53. Li, S., Zhou, M., Luo, X.: Modified primal-dual neural networks for motion control of redundant manipulators with dynamic rejection of harmonic noises. *IEEE Trans. Neural Netw. Learn. Syst.* **29**(10), 4791–4801 (2018)
54. Spong, M.W., Hutchinson, S., Vidyasagar, M.: *Robot Modeling and Control*. Wiley, New York (2006)
55. Guo, D., Zhang, Y.: A new inequality-based obstacle-avoidance MVN scheme and its application to redundant robot manipulators. *IEEE Trans. Syst. Man Cybern. C Appl. Rev.* **42**(6), 1326–1340 (2012)
56. Zhang, Z., Zheng, L., Yu, J., Li, Y., Yu, Z.: Three recurrent neural networks and three numerical methods for solving a repetitive motion planning scheme of redundant robot manipulators. *IEEE/ASME Trans. Mechatron.* **22**(3), 1423–1434 (2017)
57. Assal, S.F.M.: Learning from hint for the conservative motion of the constrained industrial redundant manipulators. *Neural Comput. App.* **23**(6), 1649–6660 (2013)
58. Kong, Y., Lu, H., Xue, Y., Xia, H.: Terminal neural computing: finite-time convergence and its applications. *Neurocomputing* **217**, 133–141 (2016)
59. Dosiek, L., Pillay, P.: Cogging torque reduction in permanent magnet machines. *IEEE Trans. Ind. Appl.* **43**(6), 1565–1571 (2007)
60. Zhang, Y., Ge, S.S.: Design and analysis of a general recurrent neural network model for time-varying matrix inversion. *IEEE Trans. Neural Netw.* **16**(6), 1477–1490 (2005)
61. Boyd, S., Vandenberghe, L.: *Convex Optimization*. Cambridge University Press, New York (2004)
62. Gao, X.B.: Exponential stability of globally projected dynamic systems. *IEEE Trans. Neural Netw.* **14**(2), 426–431 (2003)
63. Oppenheim, A.V., Willsky, A.S.: *Signals & Systems*. Prentice-Hall, Englewood Cliffs (1997)