Xuefeng Zhou · Zhihao Xu · Shuai Li · Hongmin Wu · Taobo Cheng · Xiaojing Lv

# AI based Robot Safe Learning and Control

OPEN

Springer

# AI based Robot Safe Learning and Control

Xuefeng Zhou · Zhihao Xu · Shuai Li ·
Hongmin Wu · Taobo Cheng · Xiaojing Lv

# AI based Robot Safe Learning and Control

OPEN

Springer

Xuefeng Zhou
Robotic Team
Guangdong Institute of Intelligent Manufacturing
Guangzhou, Guangdong, China

Shuai Li
School of Engineering
Swansea University
Swansea, UK

Taobo Cheng
Robotic Team
Guangdong Institute of Intelligent Manufacturing
Guangzhou, Guangdong, China

Zhihao Xu
Robotic Team
Guangdong Institute of Intelligent Manufacturing
Guangzhou, Guangdong, China

Hongmin Wu
Robotic Team
Guangdong Institute of Intelligent Manufacturing
Guangzhou, Guangdong, China

Xiaojing Lv
School of Aircraft Maintenance Engineering
Guangzhou Civil Aviation College
Guangzhou, Guangdong, China

ISBN 978-981-15-5502-2 ISBN 978-981-15-5503-9 (eBook)
https://doi.org/10.1007/978-981-15-5503-9

This Springer imprint is published by the registered company Springer Nature Singapore Pte Ltd.
The registered company address is: 152 Beach Road, #21-01/04 Gateway East, Singapore 189721, Singapore

*To our ancestors and parents, as always*

# Preface

Robot is hailed as “the pearl at the top of manufacturing crown”, and it is an important carrier of a new round of technological revolution and manufacturing integration innovation. With the continuous improvement of robot intelligence, human–robot integration has become the inevitable development direction of the new generation of robots. Human and robot play their own advantages, through various environments, operators and robot natural interaction, to complete complex work. Robots with the ability of human–computer integration will play an important role in intelligent manufacturing, family service, medical education, space exploration and other fields, with a very broad research and application prospects.

In order to achieve human–robot integration, the ability of human–computer cooperation and autonomous operation is required. In tasks such as human–computer interaction and flexible assembly, the robot interacts with external environment directly, which requires the ability to accurately control the interaction force, to ensure human–computer friendliness and task execution ability. On the other hand, due to the unstructured environment, the workspace of robot is affected by human beings and other objects. In order to ensure the safety of the system, the robot must adapt to the time-varying space constraints: for instance, when human beings enter the working space of robots, robots must be able to avoid human beings so as to avoid collision. In the more extreme cases, after the collision happened, the robot needs to be compliant with the collision. As to the robot itself, restricted by the mechanical structure, driving system and other factors, the robot must meet its own behavior constraints such as angle, angular speed limit, *etc.*; in addition, in order to improve the flexibility of the system, the new generation robots usually have redundant degrees of freedom, using their own structural characteristics to achieve the optimization of specific performance indicators. Therefore, the study of robot compliance control under complex space and behavior constraints is one of the key technologies to achieve human–robot integration.

Neural network can simulate the working mechanism of biological neural system, learn from the environment and realize information processing. Among them, the dynamic neural network has the characteristics of adaptability, nonlinearity, parallelism, distributed storage and so on. It can be used to deal with the complex problems which are difficult to be solved by traditional methods. At present, dynamic neural network has made a great progress in deep learning, estimation and prediction, image processing, complex system control and other fields.

In this book, mainly focusing on the safe control of robot manipulators, we design dynamic neural network based control schemes for robots with redundant Degrees of Freedom (DOFs). The control strategies include adaptive tracking control for robots with model uncertainties, compliance control in uncertain environments, obstacle avoidance in dynamic workspaces. The idea for this book on solving safe control of robot arms was conceived during the industrial applications and the research discussion in the laboratory. Most of the materials in this book are derived from the authors' papers published in journals, such as IEEE Transactions on Industrial Electronics. The robots considered in this book include SCARA and collaborative robots (such as Kinova JACO$_2$ and LBR iiwa). Therefore, the control methods developed in this book can be used in real applications after proper modification. 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 6 chapters.

Chapter 1 In this chapter, an adaptive tracking controller is designed for redundant manipulators. Model uncertainties and repeatability are considered. The control scheme requires neither joint accelerations nor cartesian velocity, which is more suitable in practical engineering. By using pseudo-inverse method, repeatability is optimized in the null space of the Jacobian, the continuity of joint speed is also guaranteed. Future studies will concentrate on the experimental validation of the proposed controller.

Chapter 2 An adaptive kinematic identifier is used to learn kinematic parameters online, and a dynamic neural network is presented to solve the redundancy resolution problem. The interplay of the adaptive online identifier and the neural controller makes it a coupled system with nonlinearity. Using the Lyapunov theory, the global converges of tracking error is theoretically verified. Numerical experiment results and comparisons based on a JACO$_2$ robot arm illustrate the effectiveness of the proposed algorithm and demonstrate advantages over existing ones. The Jacobian adaption strategy together with recurrent neural network (RNN) achieves task space tracking both in static and dynamic situations. Pseudo-inverse calculation of Jacobian matrix is avoided, so that the real-time performance of controller is guaranteed. The boundedness of joint speed can also protect the robot and enhance the safety performance. Before ending this chapter, it is worth pointing out that this is the first kinematic regression based dynamic neural model for self-adaptive redundant manipulator motion control, with provable convergence and guaranteed performance bounds.

Chapter 3 In this chapter, we propose an adaptive admittance control method for redundant manipulators based on RNN, in which model uncertainties of both interaction model and physical parameters are taken into consideration. Theoretical derivation using the Lyapunov technique shows the convergence of the proposed adaptive RNN, and numerical results on a 7-DOF robot iiwa demonstrate the effectiveness of the proposed control strategy. Compared with existing control methods, the proposed controller shows good performance not only in handling physical constraints, but also in eliminating the calculation of pseudo-inversion. At last, it is remarkable that this is the first time to extend RNN based method to the case of force control for redundant manipulators, especially the ones with model uncertainties. This study will be of great significance in industrial application such as grinding robots, assembling robots, *etc*.

Chapter 4 In this chapter, a novel obstacle avoidance strategy is proposed based on a deep recurrent neural network. The robots and obstacles are presented by sets of critical points, then the distance between the robot and obstacle can be approximately described as point-to-points distances. By understanding the nature escape velocity methods, a more general description of obstacle avoidance strategy is proposed. Using Minimum-Velocity-Norm (MVN) scheme, the obstacle avoidance together with path tracking problem is formulated as a Quadratic Planning (QP) problem, in which physical limits are also considered. By introducing model information, a deep RNN with a simple structure is established to solve the QP problem online. Simulation results show that the proposed method can realize the avoidance of static and dynamic obstacles.

Chapter 5 In this chapter, a novel collision-free compliance controller is constructed based on the idea of QP programming and neural networks. Different from existing methods, in this chapter, the control problem is described from an optimization perspective, and the compliance control and collision avoidance are formulated as equality or inequality constraints. The physical constraints such as limitations of joint angles and velocities are also taken into consideration. Before ending this chapter, it is worth pointing out that it is the first RNN based compliance control method, which considers collision avoidance problem in real time, and also shows great potential in handling physical limitations. In this chapter, simple numerical simulations in MATLAB are carried out to verify the efficiency of the proposed controller. In the future, we will check the control framework with different impedance models in physically realistic simulation environments, and then consider machine vision technology and system delay problem on physical experimental platforms.

Chapter 6 This paper focuses on motion–force control problem for redundant manipulators, while physical constraints and torque optimization are taken into consideration. Firstly, tracking error and contact force are modelled in orthogonal spaces, respectively, and then the control problem is turned into a QP problem, which is further rewritten in velocity level by rewriting objective function and constraints. To handle multiple physical constraints, a RNN based scheme is designed to solve the redundancy resolution online. Numerical experiment results show the validity of the proposed control scheme. Before ending this paper, it is noteworthy that this is the first paper to deal with motion–force control of redundant manipulators in the framework of RNNs and redundant manipulators with force sensitivity, e.g., grinding robots, can be readily controlled with the proposed RNN model but cannot with existing RNN models in this field.

At the end of this preface, it is worth pointing out that, in this book, some distributed methods for the cooperative control of multiple robot arms and their applications are provided. The ideas in this book may trigger more studies and researches in neural networks and robotics, especially neural network based cooperative control of multiple robot arms. There is no doubt that this book can be extended. Any comments or suggestions are welcome, and the authors can be contacted via e-mail: shuaili@ieee.org (Shuai Li).

Guangzhou, China Xuefeng Zhou
Guangzhou, China Zhihao Xu
Swansea, UK Shuai Li
Guangzhou, China Hongmin Wu
Guangzhou, China Taobo Cheng
Guangzhou, China Xiaojing Lv
Feb 2020

# 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 particularly allowed us to improve our book manuscript.

The continuous support of our research by the Nature Science Foundation of Guangdong Province (Grant NO. 2020A1515010631), Guangdong Province Key Areas R&D Program (Grant NO. 2019B090919002 and 2020B090925001), Foshan Key Technology Research Project (Grant NO. 1920001001148), Foshan Innovation and Entrepreneurship Team Project (Grant NO. 2018IT100173), GDAS' Project of Science and Technology Development 2017GDASCX-0115, Guangzhou Science Research Plan—Major Project (Grant No. 201804020095), Guangdong Province Science and Technology Major Projects (Grant No. 2017B010110010), Guangdong Innovative Talent Project of Young College (Grant No. 2016TQ03X463).

# Contents

# Acronyms

| | |
|---|---|
| DH | Denavit and Hartenberg |
| DOF | Degrees of freedom |
| JMPI | Jacobian matrix Pseudo inverse |
| KKT | Karash–Kuhn–Tucker |
| MVN | Minimum velocity norm |
| QP | Quadratic program |
| RNN | Recurrent neural networks |

# Chapter 1
# Adaptive Jacobian Based Trajectory Tracking for Redundant Manipulators with Model Uncertainties in Repetitive Tasks

**Abstract** Tracking control of manipulators, which is also called kinematic control, has always been a fundamental problem in robot control, especially for redundant robots with higher degrees of freedom. This problem would become more difficult for systems with model uncertainties. This chapter presents an adaptive tracking controller that considers uncertain physical parameters. Based on the realtime feedback of task-space coordinates, by updating the motion parameters online, a Jacobian adaptive control strategy that does not require cartesian velocity and joint acceleration is established, which makes the controller much simpler. Then the Jacobian pseudo-inverse method is used to obtain the optimal repetitive solution as a secondary task. Lyapunov theory is used to prove that the tracking error of the end effector could asymptotically converge to zero. Numerical simulations verify the effectiveness of the proposed method.

## 1.1 Introduction

Robots have been widely used in industrial, agricultural, aerospace and other fields. Therefore, research on robotics, especially robot control technology, has been a hot issue in recent decades [1–5]. In order to improve the operation accuracy of the robot, tracking control has always been a fundamental problem in robot control, which has attracted wide attention from researchers.

The tracking control of manipulators can be divided into two categories: joint space tracking and task space tracking. The target of joint space tracking is to design a controller to drive each joint of the robot to track a predetermined trajectory (see, for example [6, 7] and references therein). Another direction of tracking control is task space tracking, which is to establish the desired trajectory in cartesian space. Since the control commands do not match the target (control commands are sent to the actuators of each joint, and then the end effector is controlled to execute in Cartesian space, the mapping between the two spaces is highly nonlinear), task-space tracking is more difficult than joint space. Therefore, we should first solve the kinematic inverse problem, that is, obtain the required joint space position or speed, and realize task space tracking. This can be done off-line or online. The desired path

X. Zhou et al., *AI based Robot Safe Learning and Control*,
https://doi.org/10.1007/978-981-15-5503-9_1

in cartesian space is discretized into a set of key points, and the corresponding joint configurations are determined in turn, and the desired joint velocity and acceleration are obtained by interpolation [8]. Similar studies can be seen in [9, 10]. This method is currently widely used in industrial applications, but it will have a certain impact on the real-time performance of the system. For redundant manipulators, there is an infinite section configuration corresponding to a particular Cartesian description. Therefore, the second task can be accomplished by adjusting the joints, such as avoiding obstacles and optimizing energy consumption.

With full knowledge of physical parameters, a series of studies on real-time controllers can be found in [11–14]. In fact, robots usually have model uncertainties, including kinematic uncertainties, which may be caused by processing and measurement errors. On the other hand, robots may work with different tools, which can also lead to model uncertainties. The parametric drift may lead to inaccuracy in Jacobian, resulting in performance degradation or unpredictable response, which should be compensated. Before designing the controller, several calibration methods are proposed to determine the exact parameters [15, 16]. With the development of optical technology, researchers could measure the exact position and direction of the end-effector online. A series of real-time tracking controllers are proposed. Liu et al. proposes an adaptive tracking scheme based on online learning of the Jacobian matrix, by discussing the selection of control gain in detail, the authors prove the stability of the closed-loop system [18]. In [19], a robust controller considering actuator saturation is designed. Lyapunov theory indicates the semi-global stability of the system. In [20], a dynamic regulation controller is also established, which consists of a transposed Jacobian operator and a gravity compensator. When the required path is variable, Cheah et al. propose a passive tracking controller [21], which proves the global convergence of the tracking error. Liu et al. use the fuzzy logic system to understand the uncertainty of the robot model, and design a tracking control scheme based on sliding mode control. However, these studies require cartesian velocity or joint acceleration, which is actually difficult to obtain due to hardware constraints. Therefore, Wang et al. propose a tracking controller based on a low-pass filter, which omits the cartesian velocity measurement [22]. Similar studies can be seen in [23–25]. The above research mainly focuses on the general problems of position control, the physical uncertainty of robots and ignores the secondary tasks.

Based on the above research, this chapter studies the motion control of redundant manipulators, in which we take the uncertain kinematic parameters into account. In practice, robots are usually scheduled to perform periodic tasks, therefore, we choose repeatability as a secondary task. In order to avoid the measurement of velocity and joint acceleration in task space, a new adaptive controller is designed, which achieves secondary tasks by optimizing the functions in null space of Jacobian matrix. We also provide stability analysis and numerical simulations.

The remainder of this chapter is organized as follows. In the second part, we will introduce the basic kinematics of redundant robots and give several important properties that will be used in the following sections. In the third part, the proposed adaptive controller is discussed in detail, including an adaptive method and repeatable optimization of model parameters. The convergence analysis of tracking error is discussed. In Sect. 1.4, we provide examples and numerical simulations to verify the effectiveness of the proposed tracking method. Finally, Sect. 1.5 concludes the chapter. Before concluding this section, we emphasize the main contributions of this chapter as follows:

- This chapter mainly studies the adaptive kinematic control in the presence of model uncertainties, which is of great significance in practical engineering.
- In the controller design process, there is no need to measure the task space velocity and joint acceleration. Therefore, the controller can be easily applied in actual systems.
- The contribution also lies in that the proposed repeatability optimization scheme is guaranteed by introducing a variable coefficient, which could ensure the continuity of the joint speed.

## 1.2 Problem Formulation

Without loss of generality, the robot manipulator studied in this chapter is selected as a serial robot, which is most commonly used in industrial applications. The kinematic model of a serial robot manipulator is

$$f(\theta(t)) = x(t), \tag{1.1}$$

where $\theta(t) \in \mathbb{R}^n$ is the joint angles, and $x(t) \in \mathbb{R}^m$ is the vector describing the position and orientation of the end-effector in cartesian space. $f(\bullet) : \mathbb{R}^n \to \mathbb{R}^m$ is the forward kinematic mapping of the robot. $f(\bullet)$ is a nonlinear function. Differentiating $x(t)$ with respect to time $t$, the cartesian velocity $\dot{x}(t)$ is formulated as

$$\dot{x}(t) = J(\theta(t), a_k)\dot{\theta}(t), \tag{1.2}$$

where $J(q\theta(t), a_k) = \partial f(\theta(t), a_k)/\partial\theta(t) \in \mathbb{R}^{m\times n}$ is the Jacobian matrix. As to a redundant manipulator, $n > m$. $a_k \in \mathbb{R}^l$ denotes the vector of kinematic parameters, also called physical parameters, while in this chapter, mainly refers to length of each joint. Therefore, $a_k$ is considered as a constant vector.

The movement of the end-effector $J(\theta(t), a_k)\dot{\theta}(t)$ consists of two parts: physical parameter dependent term and joint angle-speed dependent term, and can be described in the linearization-in-parameter form [21]:

$$J(\theta(t), a_k)\dot{\theta}(t) = Y_k(\theta(t), \dot{\theta}(t))a_k, \tag{1.3}$$

where $Y_k(\theta(t), \dot{\theta}(t)) \in \mathbb{R}^{m\times l}$ is called kinematic regressor matrix.

In this chapter, to avoid measuring the task-space velocity, a low-pass filter is used as follows

$$\dot{y} + \lambda_1 y = \lambda_1 \dot{x}, \tag{1.4}$$

where $\lambda_1$ is a positive constant and $y$ is the filtered output of the task-space velocity with initial value $y(0) = 0$. Rewriting (1.4) leads to

$$y = \lambda_1 \dot{x}/(p + \lambda_1), \tag{1.5}$$

where $p$ is the Laplace variable.

Combining (1.3) and (1.5), we have

$$y = W_k(t)a_k, \quad W_k(t) = \lambda_1 Y_k(\theta, \dot{\theta})/(\lambda_1 + p), \tag{1.6}$$

where $W_k(0) = 0$. For simplicity, we write $J(\theta)$, $Y_k(\theta, \dot{\theta})$ as $J$ and $Y_k$, respectively.

**Remark 1.1** In real applications, $a_k$ has two different forms, which correspond to different meanings. The first value is the actual value of $a_k$, the other one is the nominal value of $a_k^n$. $a_k^n$ is usually a non-calibrated measurement result of $a_k$, which is usually provided by the manufacturer or manual measurement. The real value of $a_k$ is difficult to obtain in real applications, and $a_k$ is generally not the same as its nominal value of $a_k^n$. Due to assembly errors and long-term operations (such as friction, wear, *etc.,*) besides, robots may operate different tools to perform tasks, which also leads to motion uncertainty. In this case, the direct use of $a_k^n$ control method can lead to control errors, which is unacceptable in high precision tracking control.

## 1.3 Main Results

In this section, we will show the detailed process of the controller design. Firstly, the ideal case where all parameters are known is considered, then the basic idea of parameter-updating-in-realtime is designed in the case where parameters are unknown, and is then expanded to repeated optimization in null space. Finally, the stability of the closed-loop system is discussed.

### *1.3.1 Adaptive Tracking Method*

Define the tracking error in Cartesian space as

$$e(t) = x(t) - x_\mathrm{d}(t), \tag{1.7}$$

*(1) Known parameter case*

When the kinematic parameters $a_k$ is perfectly known, the accurate Jacobian matrix $J$ can be obtained, therefore, the reference trajectory can be designed as

$$\ddot{x}(t) = \ddot{x}_\mathrm{d}(t) + k_1 \dot{x}_\mathrm{d}(t) - k_2 e(t) - k_1 J \dot{\theta}(t), \tag{1.8}$$

where $k_1$ and $k_2$ are positive control gains. According to Eq. (1.2), the Eq. (1.8) can be reformulated as $\ddot{x}(t) = \ddot{x}_{\mathrm{d}}(t) + k_1\dot{x}_{\mathrm{d}}(t) - k_2 e(t) - k_1\dot{x}(t)$, by calculating the second derivative of Eq. (1.7), and substituting Eq. (1.8), we have

$$\begin{aligned} \ddot{e}(t) &= \ddot{x}(t) - \ddot{x}_{\mathrm{d}}(t) \\ &= k_1\dot{x}_{\mathrm{d}}(t) - k_2 e(t) - k_1\dot{x}(t). \end{aligned} \tag{1.9}$$

Eq. (1.9) can be rewritten as

$$\ddot{e}(t) + k_1\dot{e}(t) + k_2 e(t) = 0, \tag{1.10}$$

it is obvious that is $e(t)$ will eventually converge to zero, if $k_1$ and $k_2$ are Hurwitz. Combining Eqs. (1.8) and (1.2), and letting initial joint velocity $\dot{\theta}(0)$ be 0, one can easily derive the corresponding control signals of joint speed as below

$$\dot{\theta} = \dot{\theta}_j + \dot{\theta}_n \tag{1.11a}$$

$$\dot{\theta}_j = \int_0^t J^{\dagger}\left[(\ddot{x}_{\mathrm{d}} + k_1\dot{x}_{\mathrm{d}} - k_2 e - k_1 J\dot{\theta}) - \dot{J}\dot{\theta}\right]\mathrm{d}t \tag{1.11b}$$

$$\dot{\theta}_n = (I - J^{\dagger}J)\alpha, \tag{1.11c}$$

where $I$ is n-dimensional identity matrix, $J^{\dagger} = J^T(JJ^T)^{-1}$ is the Pseudo-Inverse of $J$, and $\dot{\theta}_n$ is a speed component in the null space of Jacobian, $\alpha$ can be selected arbitrarily. It is notable that $J\dot{\theta}_n = 0$, indicating that the speed component in the null space has no influence on the movement of end-effector. By getting the time-derivative of Eq. (1.2) and substituting Eqs. (1.11), (1.2), (1.7), one can easily verify that the error dynamics under kinematic controller Eq. (1.11) is the same as Eq. (refeq10), the tracking error will gradually converge to 0.

**Remark 1.2** Equation (1.8) gives fundamental description of reference trajectory in the Cartesian space, it is notable that all the required information except $J(\theta, a_k)$ on the right side of the equation is easy to obtain. This inspires us to design a similar control strategy with the existence of kinematic uncertainties.

*(2) Unknown parameter case*
In this situation, since the exact value of $a_k$ is unknown, $J$ is unknown. Therefore, in this case, we use $\hat{J}$ instead of $J$ by replacing $a_k$ with its estimation $\hat{a}_k$, and let $\hat{a}_k(0) = a_k^n$, then the estimated $\dot{x}(t)$ is $\hat{\dot{x}}(t) = \hat{J}\dot{\theta}$. By replacing $a_k$ by $\hat{a}_k$, according to (1.3), the estimated cartesian speed $\hat{\dot{x}}$ satisfies

$$\hat{\dot{x}} = \hat{J}\dot{q} = Y_k(q, \dot{\theta})\hat{a}_k, \tag{1.12}$$

The modified reference trajectory is thus designed as

$$\dot{\theta}(t) = \int_0^t \{\hat{J}^{\dagger}[\ddot{x}_{\mathrm{d}} + (k_1 + k_2)\dot{x}_{\mathrm{d}} - k_1 k_2 e - \dot{\hat{J}}\dot{\theta} - k_3 e] - (k_1 + k_2)\dot{\theta}\}\mathrm{d}t. \tag{1.13}$$

Since the accurate feedback of cartesian velocity $\dot{x}$ is unavailable, the derivative of tracking error $\dot{e} = \dot{x} - \dot{x}_\mathrm{d}$ is also unknown, therefore, we define the alternative value of $\dot{e}$ by using the estimated Cartesian speed $\dot{x}$:

$$\Delta\hat{\dot{x}} = \hat{\dot{x}} - \dot{x}_\mathrm{d} = \hat{J}\dot{\theta} - \dot{x}_\mathrm{d}, \tag{1.14}$$

then the updating law of kinematic parameters is designed as

$$\dot{\hat{a}}_k = k_1 Y_k^\mathrm{T}(\Delta\hat{\dot{x}} + k_1 e) + k_3 Y_k^T e - W_k^T(t)\Gamma_1(W_k(t)\hat{a}_k - y), \tag{1.15}$$

where $\Gamma_1$ is a positive definite diagonal matrix, $k_1$, $k_2$ and $k_3$ are positive control gains.

**Remark 1.3** Without loss of generality, the initial value of the estimated kinematic parameters can be selected according to the nominal value, which can be obtained from handbook or manual measurement. In fact, the adjustment of $\hat{a}_k(0)$ does affect the tracking process, which can be verified in the next section. The greater the error between $\hat{a}_k(0)$ and $a_k$, the greater the initial simulation error. However, according to (1.15), no matter what the exact value of $\hat{a}_k(0)$, the estimated value of $\hat{a}_k$ will eventually converge to $a_k$, which can be verified by stability analysis and numerical experiments.

Now, we are ready to offer a theorem about the task-space tracking problem for robots with uncertain physical parameters using the proposed adaptive controller as below.

**Theorem 1.1** The control error $e(t)$ for a redundant manipulator described by (1.7) would globally converge to 0, provided the joint speed controller described as (1.13), along with the kinematic adaptation law (1.15).

***Proof*** Differentiating (1.7) and substituting (1.3) and (1.12), we have

$$\begin{aligned}\dot{e} &= \dot{x} - \hat{\dot{x}} + \hat{\dot{x}} - \dot{x}_\mathrm{d}\\ &= Y_k a_k - Y_k\hat{a}_k + \hat{\dot{x}} - \dot{x}_\mathrm{d}\\ &= -Y_k\tilde{a}_k + \Delta\hat{\dot{x}}.\end{aligned} \tag{1.16}$$

Taking the time derivative of $\Delta\hat{\dot{x}}$ and combining Eqs. (1.14) and (1.16) derives

$$\begin{aligned}\frac{\mathrm{d}}{\mathrm{d}t}(\Delta\hat{\dot{x}}) &= \dot{\hat{J}}\dot{\theta} + \hat{J}\ddot{\theta} - \ddot{x}_\mathrm{d}\\ &= (k_1 + k_2)\dot{x}_\mathrm{d} - k_1k_2e - k_3e - k_2\hat{J}\dot{\theta} - k_1\hat{J}\dot{\theta}\\ &= k_2\dot{x}_\mathrm{d} - k_2\hat{J}\dot{\theta} - k_1k_2e - k_3e + k_1\dot{x}_\mathrm{d} - k_1(\dot{x}_\mathrm{d} + \dot{e} + Y_k\tilde{a}_k)\\ &= -k_2\Delta\hat{\dot{x}} - k_2k_1e - k_3e - k_1Y_k\tilde{a}_k - k_1\dot{e},\end{aligned} \tag{1.17}$$

where $\tilde{a}_k = a_k - \hat{a}_k$ represents the difference between the real value of physical parameters $a_k$ and the estimated one $\hat{a}_k$. Eq. (1.17) can be written as

$$\frac{\mathrm{d}}{\mathrm{d}t}(\Delta\hat{\dot{x}} - k_1 e) = -k_2(\Delta\hat{\dot{x}} + k_1 e) - k_3 e - k_1 Y_k \tilde{a}_k. \tag{1.18}$$

Select a Lyapunov function candidate as follows

$$V = (\Delta\hat{\dot{x}} + k_1 e)^{\mathrm{T}}(\Delta\hat{\dot{x}} + k_1 e)/2 + k_3 e^{\mathrm{T}} e/2 + \tilde{a}_k^T \tilde{a}_k/2. \tag{1.19}$$

By taking the time derivative of (1.19) and combing (1.15), (1.16) and (1.17), we have

$$\begin{aligned}\dot{V} &= (\Delta\hat{\dot{x}} + k_1 e)^{\mathrm{T}}\mathrm{d}(\Delta\hat{\dot{x}} + k_1 e)/\mathrm{d}t + k_3 e^{\mathrm{T}}\dot{e} + \tilde{a}_k^T \dot{\tilde{a}}_k \\ &= (\Delta\hat{\dot{x}} + k_1 e)^{\mathrm{T}}(-k_2(\Delta\hat{\dot{x}} + k_1 e) - k_3 e - k_1 Y_k \tilde{a}_k) + \tilde{a}_k^T (k_1 Y_k^{\mathrm{T}}(\Delta\hat{\dot{x}} + k_1 e) + k_3 Y_k^T e \\ &\quad - W_k^T(t)\Gamma_1(W_k(t)\hat{a}_k - y) + k_3 e^{\mathrm{T}}(-Y_k\tilde{a}_k + \Delta\hat{\dot{x}}) \\ &= -k_2(\Delta\hat{\dot{x}} + k_1 e)^{\mathrm{T}}(\Delta\hat{\dot{x}} + k_1 e) - k_1 k_3 e^{\mathrm{T}} e - \tilde{a}_k^T W_k^T(t)\Gamma_1 W_k(t)\tilde{a}_k \\ &\le 0. \end{aligned} \tag{1.20}$$

Then we can obtain that $\Delta\hat{\dot{x}}$, $e$ and $\tilde{a}_k$ are all bounded. Based on Eqs. (1.14) and (1.3), $\hat{J}\dot{\theta}$, $\hat{a}_k$ and $Y_k\tilde{a}_k$ are also bounded. Notably that $W_k(t)\tilde{a}_k$ is the output of a stable system with bounded input $Y_k(t)\tilde{a}_k$, we have $W_k(t)\tilde{a}_k$ is also bounded. Then according to Eq. (1.15), $\dot{\hat{a}}_k$ is bounded. Differentiating $W_k(t)\tilde{a}_k$ with respect to time, we have

$$\frac{\mathrm{d}}{\mathrm{d}t}(W_k(t)\tilde{a}_k) = \lambda_1(Y_k - W_k(t))\tilde{a}_k + W_k(t)\dot{\hat{a}}_k. \tag{1.21}$$

$\mathrm{d}(W_k(t)\tilde{a}_k)/\mathrm{d}t$ is also bounded. Then we have $\dot{e}$, $\mathrm{d}(\Delta\hat{\dot{x}})/\mathrm{d}t$ and $\mathrm{d}(W_k(t)\tilde{a}_k)/\mathrm{d}t$ are all bounded, which means the time derivative of (1.20), $\ddot{V}$ is bounded. Using Barbalat's Lemma, we have $\Delta\hat{\dot{x}} + k_1 e \to 0$ , $e \to 0$ , as $t \to \infty$ .

**Remark 1.4** We have proved the convergence of the tracking error under the condition of kinematic uncertainties. In fact, when $a_k$ is perfectly known, Eq. (1.13) will be degenerated as

$$\dot{\theta}(t) = \int_0^t [J^{\dagger}(\ddot{x}_{\mathrm{d}} + (k_1 + k_2)\dot{x}_{\mathrm{d}} - k_1 k_2 e - \dot{J}\dot{q} - k_3 e) - (k_1 + k_2)\dot{\theta}]\mathrm{d}t, \tag{1.22}$$

which has the similar form compared with Eq. (1.11). Therefore, known parameter case described in Eq. (1.11) can be considered as a special form of Eq. (1.13).

**Remark 1.5** The control velocity $\dot{q}$ in Eq. (1.13) is not the final result of this chapter. The velocity component in null space is ignored, although it has no effect on the movement of end-effector as well as the stability proof, this part can not be neglected, because the redundancy mechanism is of great engineering significance to the manipulator.

**Algorithm 1** The proposed tracking method

```
Input: Parameters $k_1$, $k_2$, $k_3$, $K$, $\Gamma_1$, $\epsilon$, initial states $\dot{\theta}(0) = 0$, $\theta(0)$, nominal kinematic parameter $\hat{a}_k(0)$, desired path $x_{\mathrm{d}}(t)$, $\dot{x}_{\mathrm{d}}(t)$ and $\ddot{x}_{\mathrm{d}}(t)$, task duration $T_e$, feedback of end effector $x(t)$, analytical expressions of estimated Jacobian matrix $\hat{J}$ and kinematic regressor matrix $Y_k$.
Output: To achieve task-space tracking of the redundant manipulator
    Initialize $a_k(0) \leftarrow a_k^n$.
    $x, \theta, \dot{q} \leftarrow$ Sensor readings
    Calculate $e$, $y$, and $W_k(t)$ by Equation (1.7), (1.5) and (1.6)
    Update $K$ by Equation (1.27)
    Update $\theta_j$ by $\dot{\theta}_j \leftarrow$ Equation (1.26b)
    Update $\theta_n$ by $\dot{\theta}_n \leftarrow$ Equation (1.26c)
    calculate the output $\dot{\theta}$ by Equation (1.26a)
    Update $\hat{a}_k$ by $\dot{\hat{a}}_k \leftarrow$ by Equation (1.26d)
Until($t > T_e$)
```

*(3) Repeatability optimization*
In this subsection, in order to make full use the of redundant design of a redundant manipulator, a repeatability optimization scheme is developed in the null space of the Jacobian matrix, which is helpful to improve the stability and reliability of robots in periodic tasks.

Define a following function to describe a robot's repeatability as

$$F(\theta) = -K(\theta - \theta_{ini})^{\mathrm{T}}(\theta - \theta_{ini})/2, \tag{1.23}$$

where $K$ is a positive parameter scaling the weight of repeatability optimization, $\theta_{ini}$ is the initial value of the joint angles. By using gradient descent method, a velocity component in null space can be calculated as

$$\alpha = [\partial(F(\theta))/\partial(\theta_1), \cdots, \partial(F(\theta))/\partial(\theta_n)]. \tag{1.24}$$

Combining Eqs. (1.24) and (1.23), we have

$$\alpha = [\theta_{int}(1) - \theta(1), \cdots, \theta_{int}(i) - \theta(i), \cdots, \theta_{int}(n) - \theta(n)]^{\mathrm{T}}. \tag{1.25}$$

where $\theta_{ini}(i)$ and $\theta(i)$ represent the $i$th element of $\theta_{int}$ and $\theta$, respectively, $i = 1, \cdots, n$.

Then the complete form of the proposed adaptive controller is

$$\dot{\theta} = \dot{\theta}_j + \dot{\theta}_n \tag{1.26a}$$

$$\dot{\theta}_j = \int_0^t [J^{\dagger}(\ddot{x}_{\mathrm{d}} + (k_1 + k_2)\dot{x}_{\mathrm{d}} - k_1 k_2 e - \dot{J}\dot{\theta} - k_3 e) - (k_1 + k_2)\dot{\theta}]\mathrm{d}t \tag{1.26b}$$

$$\dot{\theta}_n = (I - J^{\dagger}J)[\theta_{int}(1) - \theta(1), \cdots, \theta_{int}(i) - \theta(i), \cdots, \theta_{int}(n) - \theta(n)]^{\mathrm{T}} \tag{1.26c}$$

$$\dot{\hat{a}}_k = k_1 Y_k^{\mathrm{T}}(\Delta\dot{\hat{x}} + k_1 e) + k_3 Y_k^T e - W_k^T(t)\Gamma_1(W_k(t)\hat{a}_k - y) \tag{1.26d}$$

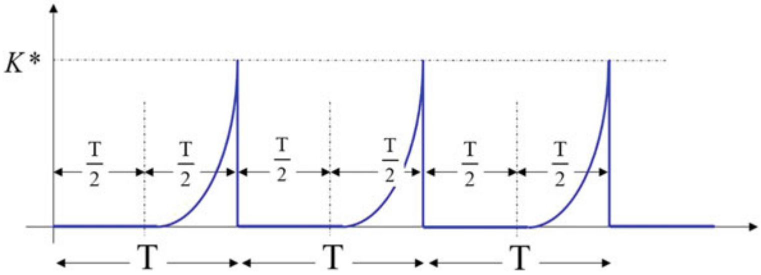


**Fig. 1.1** change curve of $K$ with time $t$

Remarkable that at the beginning stage of the tracking cycle, the repeatability is less important, and then it rises as the task continues. To this end, we set $K$ as a variable:

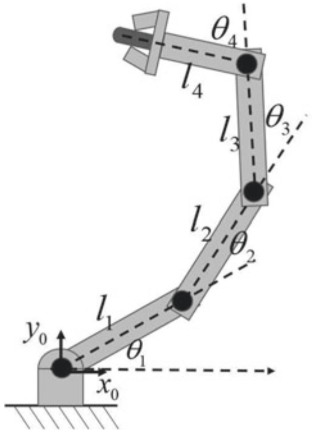
$$K = \begin{cases} 0 & NT \leq t < NT + T/2, \\ K^* & \dfrac{2NT + T}{2} < t < (N+1)T \end{cases}, \tag{1.27}$$

where $K^* = K_{max}(1 - cos(\pi(t - NT - T/2)/T)$, $N = 0, 1, 2, \ldots$ are natural numbers, $T$ is the period of cyclic motion. If $t < NT + T/2$, the robot has just left the initial state to perform a task, thus we let $K = 0$, this will cause $\alpha = 0$, the joint control velocity is the same as (1.13). When $t > NT + T/2$, $K$ increases from 0 to maximum value $K_{max}$ with time, forcing the robot to repeat the initial state. The change curve of $K$ with time is shown in Fig. 1.1.

**Remark 1.6** The main reason for this selection of $K$ is to ensure the continuity of joint speed signals during a motion cycle. Notable that the discontinuities of $K$ still appear at the moment $T = NT$. If the robot can repeat the initial joint state, $\theta - \theta_{ini}$ would converge to 0, so $\alpha$ can be also regarded as continuous. Therefore, the definition of $K$ in (1.27) is acceptable.

## 1.4 Numeral Simulations

In this section, several groups of numerical experiments are carried out to show the effectiveness of the designed controller. Firstly, a comparative simulation is given to show that the adaptive tracking law could achieve a satisfing performance with the existence of kinematic uncertainties. Secondly, we will check the performance in periodic tasks. Finally, more general trajectories are discussed to show the adaptiveness and robustness of the control algorithm.



| Joint | $a_i$ | $\alpha_i$ | $d_i$ | $\theta_i$ |
|---|---|---|---|---|
| 1 | 0.3 m | 0 rad | 0 m | 0 rad |
| 2 | 0.3 m | 0 rad | 0 m | 0 rad |
| 3 | 0.1 m | 0 rad | 0 m | 0 rad |
| 4 | 0.2 m | 0 rad | 0 m | 0 rad |

**Fig. 1.2** The 4-DOF redundant manipulator to be simulated in this chapter. Left: Physical structure of the 4-link robot manipulator. Right: D-H parameters

### 1.4.1 Simulation Settings

The vector of initial joint angles is selected as $\theta_{ini} = [\pi/2, -\pi/2, 0, 0]^{\mathrm{T}}$rad, and the corresponding cartesian position is $x_{ini} = [0.6, 0.3]^{\mathrm{T}}$. Since the exact value of kinematic parameters (see $d_i$ in Fig. 1.2), we assume the nominal values to be $a_k^n = [0.25, 0.25, 0.12, 0.18]^{\mathrm{T}}$m, and let $\hat{a}_k(0) = a_k^n$. The control gains $k_1, k_2$ and $k_3$ are set to be $k_1 = 50$, $k_2 = 50$, $k_3 = 50$, $\Gamma = 10$. As to the repeatable tasks, the parameter scaling the velocity component in the null space is selected as $K_{max} = 10$. The time constant of low-pass filter is $\lambda = 40$. It is notable that matrix $\hat{J}$ is essential in the proposed tracking controller, which is used to estimate the actual Jacobian matrix $J(q, a_k)$. To further show the detail of the proposed controller, analytical expression of $\hat{J}$ is given in Appendix.

### 1.4.2 Verification of Parameter Estimation

Comparative simulations is firstly carried out to show the effectiveness of the proposed updating law (1.15). The desired path to be tracked is defined as $x_{\mathrm{d}}(t) = 0.4 + 0.2cos(2t)$, $y_d(t) = 0.3 + 0.2sin(2t)$. In the first simulation, the nominal values are used directly in the tracking control according to Eq. (1.13). By contrast, $\hat{a}_k$ is updated using (1.15) in the comparable simulation, and $\alpha$ is set to be zero (i.e., we didn't use repeatability in this part). Simulation results are shown in Fig. 1.3. Both controllers ensure the boundedness of the tracking error. When $a_k$ is known, benefiting from the closed-loop control mechanism, the tracking errors along two axes are much less than 5 mm. The tracking errors with parameter estimation are less

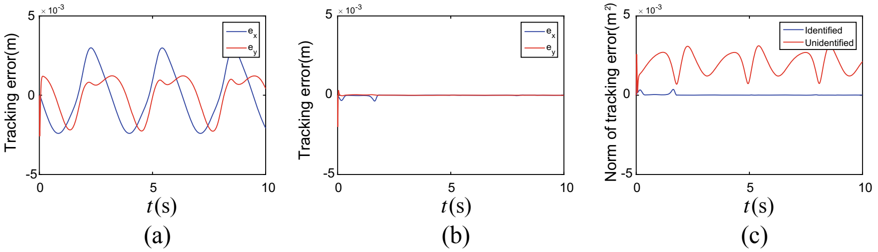


**Fig. 1.3** Error profile with and without parameter estimation when tracking a circle. **a** Tracking errors without parameter estimation. **b** Tracking errors with parameter estimation. **c** Norm of tracking errors with and without parameter estimation

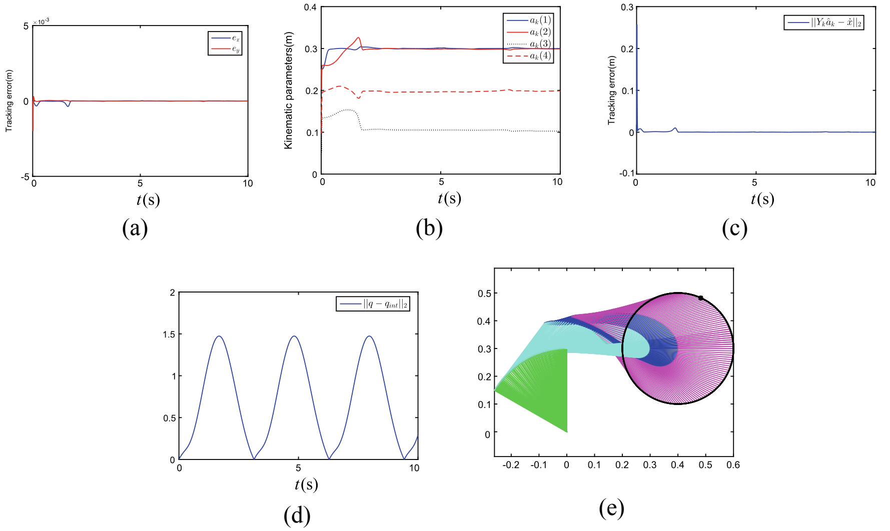
than 1 mm. Fig. 1.3c shows comparative results of tracking error norm corresponding to known and unknown $a_k$, intuitively showing the effectiveness of the proposed controller under the condition of unknown models.

### *1.4.3 Verification of Repeatability Optimization*

Then we check the effectiveness of repeatability optimization. Based on the simulation of the previous part, we introduce the proposed repeatability optimization scheme (i.e., the controller is the same as the adaptive tracking controller in the previous part except $\alpha \neq 0$.) Simulation results are shown in Fig. 1.4. The curve of tracking error $e$ is the same as the one when $\alpha = 0$, showing the property that the velocity component in null space have no influence on the cartesian movement (Fig. 1.4a). The estimated kinematic parameters $\hat{a}_k$ are shown in Fig. 1.4b, which slowly converges to $a_k$ with time. The error norm $||Y_k\hat{a}_k - \dot{x}||_2$ of the estimated cartesian speed reduced to zero quickly, as shown in Fig. 1.4c. The curve of the repeatability function is shown in Fig. 1.4d, we can observe that when $t = T, 2T, \cdots$, $||q - q_{ini}||_2$ equals to zero. This is to say, when repeatability optimization is used, $||q - q_{ini}||_2$ changes periodically. Figure 1.4e shows the motion trajectory tracked by its end-effector of the robot manipulator, illustrating the precise tracking of the circle desired trajectory.

### *1.4.4 Cardioid Tracking*

To further verify effectiveness of the proposed control scheme, the manipulator is required to track a cardioid trajectory in 2-D workspace. The desired path is defined as $x_{\mathrm{d}}(t) = 0.1(2sin(2t) - sin(4t)) + 0.6$ m, $y_{\mathrm{d}}(t) = 0.1(2cos(2t) - cos(4t)) + 0.2$ m. Simulation results are shown in Fig. 1.5. The motion trajectory achieved by its end-effector of the manipulator is shown in Fig. 1.5a. The corresponding tracking



(a) (b) (c)

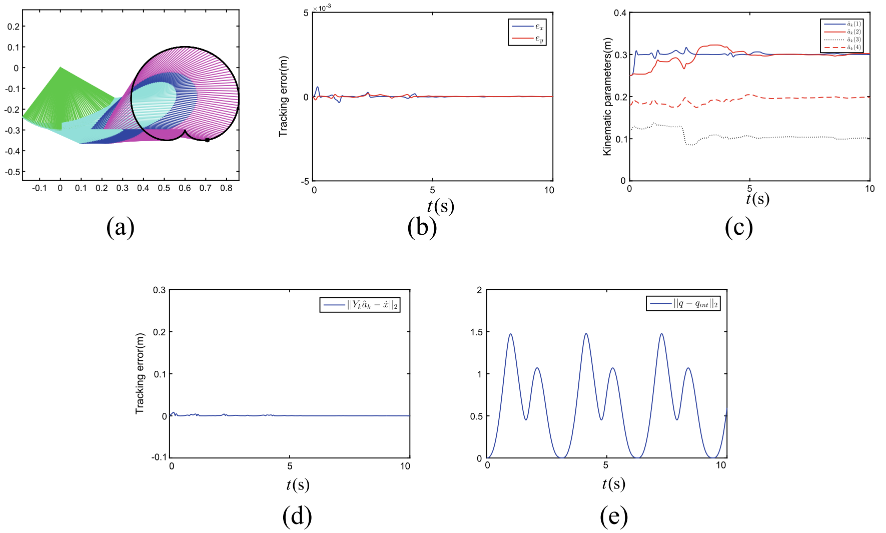
(d) (e)

**Fig. 1.4** Simulation results with parameter estimation when tracking a circle. **a** Tracking error profile. **b** Estimated parameter $\hat{a}_k$. **c** Difference between the estimated value $Y_k\hat{a}_k$ and the real one. **d** $||q - q_{ini}||_2$ with repeatability optimization. **e** Motion trajectory

errors are given in Fig. 1.5b, showing that the robot successfully tracks the given trajectory. $||q - q_{ini}||_2$ is guaranteed to 0 when $t = T, 2T, 3T$ (Fig. 1.5e), and the estimated kinematic parameters are shown in Fig. 1.5c. All in all, the proposed controller ensures stable tracking under the condition of model uncertainties, and the repeatability is also achieved.

## 1.5 Summary

In this chapter, an adaptive tracking controller is designed for redundant manipulators. Model uncertainties and repeatability are considered. The control scheme requires neither joint accelerations nor cartesian velocity, which is more suitable in practical engineering. By using the pseudo-inverse method, repeatability is optimized in the null space of Jacobian, the continuous of joint speed is also guaranteed. Future studies will concentrate on the experimental validation of the proposed controller.



(a) (b) (c)

(d) (e)

**Fig. 1.5** Simulation results when tracking a cardioid curve. **a** Motion trajectory of the manipulator. **b** Tracking error. **c** Estimated parameter $\hat{a}_k$. **d** Difference between the estimated value $Y_k\hat{a}_k$ and the real one. **e** $||q - q_{ini}||_2$ with repeatability optimization

# Appendix

Given the joint angle $\theta = [\theta_1, \theta_2, \theta_3, \theta_4]^{\mathrm{T}}$ and the estimated $\hat{a}_k = [\hat{a}_k(1), \hat{a}_k(2), \hat{a}_k(3), \hat{a}_k(4)]^{\mathrm{T}}$. By simplifying $cos(\theta_i) = c_i$, $sin(\theta_i) = s_i$, $\hat{a}_k(i) = a_i$, the analytical expression of $\hat{J}$ is given as below.

$$\hat{J}(1,1) = -a_1s_1 - a_2s_{12} - a_3s_{123} - a_4s_{1234},$$
$$\hat{J}(1,2) = -a_2s_{12} - a_3s_{123} - a_4s_{1234}$$
$$\hat{J}(1,3) = -a_3s_{123} - a_4s_{1234}$$
$$\hat{J}(1,4) = -a_4s_{1234}$$
$$\hat{J}(2,1) = a_1c_1 + a_2c_{12} + a_3c_{123} + a_4c_{1234}$$
$$\hat{J}(2,2) = a_2c_{12} + a_3c_{123} + a_4c_{1234}$$
$$\hat{J}(2,3) = a_3c_{123} + a_4c_{1234}$$
$$\hat{J}(2,4) = a_4c_{1234}.$$

Based on the analytical expression of $\hat{J}$ given above, $\dot{\hat{J}}$ can be formulated as follows.

$$\dot{\hat{J}}(1,1) = -a_1c_1\dot{\theta}_1 - a_2c_{12}(\dot{\theta}_1 + \dot{\theta}_2) - a_3c_{123}(\dot{\theta}_1 + \dot{\theta}_2 + \dot{\theta}_3) \\ - a_4c_{1234}(\dot{\theta}_1 + \dot{\theta}_2 + \dot{\theta}_3 + \dot{\theta}_4)$$
$$\dot{\hat{J}}(1,2) = -a_2c_{12}(\dot{\theta}_1 + \dot{\theta}_2) - a_3c_{123}(\dot{\theta}_1 + \dot{\theta}_2 + \dot{\theta}_3) - a_4c_{1234}(\dot{\theta}_1 + \dot{\theta}_2 + \dot{\theta}_3 + \dot{\theta}_4)$$
$$\dot{\hat{J}}(1,3) = -a_3c_{123}(\dot{\theta}_1 + \dot{\theta}_2 + \dot{\theta}_3) - a_4c_{1234}(\dot{\theta}_1 + \dot{\theta}_2 + \dot{\theta}_3 + \dot{\theta}_4)$$
$$\dot{\hat{J}}(1,4) = -a_4c_{1234}(\dot{\theta}_1 + \dot{\theta}_2 + \dot{\theta}_3 + \dot{\theta}_4)$$

$$
\begin{aligned}
\dot{\hat{J}}(2,1) &= -a_1 s_1 \dot{\theta}_1 - a_2 s_{12}(\dot{\theta}_1 + \dot{\theta}_2) - a_3 s_{123}(\dot{\theta}_1 + \dot{\theta}_2 + \dot{\theta}_3) \\
&\quad - a_4 s_{1234}(\dot{\theta}_1 + \dot{\theta}_2 + \dot{\theta}_3 + \dot{\theta}_4) \\
\dot{\hat{J}}(2,2) &= -a_2 s_{12}(\dot{\theta}_1 + \dot{\theta}_2) - a_3 s_{123}(\dot{\theta}_1 + \dot{\theta}_2 + \dot{\theta}_3) - a_4 s_{1234}(\dot{\theta}_1 + \dot{\theta}_2 + \dot{\theta}_3 + \dot{\theta}_4) \\
\dot{\hat{J}}(2,3) &= -a_3 s_{123}(\dot{\theta}_1 + \dot{\theta}_2 + \dot{\theta}_3) - a_4 s_{1234}(\dot{\theta}_1 + \dot{\theta}_2 + \dot{\theta}_3 + \dot{\theta}_4) \\
\dot{\hat{J}}(2,4) &= -a_4 s_{1234}(\dot{\theta}_1 + \dot{\theta}_2 + \dot{\theta}_3 + \dot{\theta}_4).
\end{aligned}
$$

## References

1. X. Li, Z. Xu, S. Li, H. Wu, X, Zhou, Cooperative Kinematic Control For Multiple Redundant Manipulators Under Partially Known Information Using Recurrent Neural Network. IEEE ACCESS **8**(1), 40029–40038 (2020)
2. Z. Xu, S. Li, X. Zhou, Y. Wu, T. Cheng, D. Huang, Dynamic neural networks based kinematic control for redundant manipulators with model uncertainties. Neurocomputing **329**(1), 255–266 (2019)
3. Y. Zhang, S. Li, J. Gui, X. Luo, Velocity-level control with compliance to acceleration-level constraints: a novel scheme for manipulator redundancy resolution. IEEE Trans. Industr. Informat. **14**(3), 921–930 (2018)
4. Z. Xu, S. Li, X. Zhou, T. Cheng, Dynamic neural networks based adaptive admittance control for redundant manipulators with model uncertainties. Neurocomputing **357**(1), 271–281 (2019)
5. Z. Xu, S. Li, X. Zhou, W. Yan, T. Cheng, H. Dan, Dynamic neural networks for motion-force control of redundant manipulators: an optimization perspective, IEEE Trans. Industr. Electron. Early access (2020). https://doi.org/10.1109/TIE.2020.2970635
6. J. Slotine and W. Li (2002) Adaptive manipulator control: a case study. IEEE Trans. Automatic Control 33(11): 995–1003 (2002)
7. L. Mostefai, M. Lotfi, O. Sehoon, Y. Hori, Optimal control design for robust fuzzy friction compensation in a robot joint. IEEE Trans. Industr. Electron. **56**(10), 3832–3839 (2009)
8. E. Papadopoulos, I. Poulakakis, I. Papadimitriou, Polynomial-based obstacle avoidance techniques for nonholonomic mobile manipulator systems **51**(4), 229–247 (2005)
9. L. Ellekilde, J. Perram (2005) Tool center trajectory planning for industrial robot manipulators using dynamical systems.Int. J. Robotics Res. **24**(1):385–396
10. M.Lee, S. Go, M. Lee, A robust trajectory tracking control of a polishing robot system based on CAM data. Robot. Comput. Integr. Manufact. **17**(1), 177–183 (2001)
11. B. Xian, M. Queiroz, D. Dawson, I. Walker, Task-space tracking control of robot manipulators via quaternion feedback. IEEE Trans. Robot. Automat. **20**(1), 160–167 (2004)
12. J. Ren, B. Wang, M. Cai and J. Yu, Adaptive Fast Finite-Time Consensus for Second-Order Uncertain Nonlinear Multi-Agent Systems With Unknown Dead-Zone, IEEE Access, **8**(1), 25557–25569 (2020)
13. D. Chen, S. Li, Q. Wu, X. Luo, Super-twisting ZNN for coordinated motion control of multiple robot manipulators with external disturbances suppression. Neurocomputing **371**(1), 78–90 (2020)
14. Dechao Chen, Shuai Li, A recurrent neural network applied to optimal motion control of mobile robots with physical constraints, Applied Soft Computing, to be published, 2019, https://doi.org/10.1016/j.asoc.2019.105880.
15. A. Joubair, M. Slamani, I. Bonev, Kinematic calibration of a 3−DOF planar parallel robot. Industr. Robot Int. J. **39**(4), 392–400 (2012)
16. Zhijia Zhao, Xiuyu He, Zhigang Ren, Guilin Wen, Boundary adaptive robust control of a flexible riser system with input nonlinearities. IEEE Trans. Syst. Man Cybernet. Syst. **49**(10), 1971–1980 (2019). https://doi.org/10.1109/TSMC.2018.2882734
17. A. Joubair, M. Slamani, I. Bonev, Kinematic calibration of a five-bar planar parallel robot using all working modes. Robot. Comput. Integr. Manufact. **29**(4), 15–25 (2013)

18. C. Liu, C. Cheah, Task-space adaptive setpoint control for robots with uncertain kinematics and actuator model. IEEE Trans. Automatic Control **50**(11), 1854–1860 (2004)
19. W. Dixon, Adaptive regulation of amplitude limited robot manipulators with uncertain kinematics and dynamics. IEEE Trans. Automatic Control **52**(3), 488–493 (2004)
20. M. Galicki, An Adaptive Regulator of Robotic Manipulators in the Task Space. IEEE Transactions on Automatic Control, 53(4): 1058–1062 (2008)
21. C. Cheah, C. Liu, J. Slotine, Adaptive tracking control for robots with unknown kinematic and dynamic properties. Int. J. Robotics Res. **25**(3), 283–296 (2006)
22. H.L. Wang, Y.C. Xie, Prediction error based adaptive jacobian tracking of robots with uncertain kinematics and dynamics. IEEE Trans. Automatic Control **54**(12), 2889–2894 (2009)
23. H. Wu, Y. Guan, J. Rojas, A latent state-based multimodal execution monitor with anomaly detection and classification for robot introspection. Appl. Sci. **9**(6), 1072 (2019)
24. H. Wu, Z. Xu, W. Yan, Q. Su, S. Li, T. Cheng, X. Zhou, Incremental Learning Introspective Movement Primitives From Multimodal Unstructured Demonstrations. IEEE Access **15**(7), 159022–36 (2019)
25. H. Wu, Y. Guan, J. Rojas, Analysis of multimodal Bayesian nonparametric autoregressive hidden Markov models for process monitoring in robotic contact tasks. Int. J. Advanc. Robot. Syst. **16**(2), 1729881419834840 (2019)

# Chapter 2
# RNN Based Trajectory Control for Manipulators with Uncertain Kinematic Parameters

**Abstract** Among tracking control of redundant manipulators, potential limitations such as model uncertainties and physical limitations may exist. Conventional solutions may fail when model parameters differ from nominal ones. In this chapter, a novel kinematic controller with the capability of self-adaptation is proposed to address this challenging issue. Based on the coordinate feedback, a Jacobian adaption strategy is firstly built by updating kinematic parameters online. Using Karush–Kuhn–Tucker conditions, the redundancy solution problem is then turned into a quadratic optimization one, and a recurrent neural network based controller is designed to derive the optimal solution recurrently. Theoretical analysis demonstrates the global convergence of the tracking error. Compared with existing methods, kinematic model uncertainty of the robot is allowed in this chapter, and the pseudo-inverse of Jacobian matrix is avoided, with the consideration of physical limitation in a joint framework. Numerical experiments based on Kinova JACO$_2$ show the effectiveness of the proposed controller.

## 2.1 Introduction

With the development of mechanics, electronics and computer technology advance, using robot manipulators is becoming popular in modern manufacturing such as welding, painting, assembling, *etc* [1–4]. Among these applications, tracking control of manipulators, focusing on the calculation of control actions to drive the robot to move along the user-defined trajectory in Cartesian space, is always a core problem in robot control, and has been studied intensively by researchers in recent decades.

Redundant manipulators have more degrees of freedom (DOFs) than those required to accomplish a given task [5], and have shown great potentials in enhancing robot flexibility, dexterity, and versatility, avoiding obstacles [6–9], and optimizing energy consumption [10]. However, the nonlinear function description from the joint to Cartesian space, as well as the redundancy in DOFs, makes it a challenging problem to achieve precise tracking control of redundant manipulators.

In recent decades, some results on resolving the redundancy of manipulators have been reported. In most approaches, the problem is solved at the velocity or accelera-

X. Zhou et al., *AI based Robot Safe Learning and Control*,
https://doi.org/10.1007/978-981-15-5503-9_2

tion level, namely, to derive the corresponding joint velocity or acceleration according to the trajectory description in Cartesian space. Masayuki et al. [12] propose a redundancy solution method for a S-R-S redundant manipulator at the angle level, in which analytic solutions are firstly derived, and analytical methods for joint avoidance is then considered. However, this method is effective only for robots with a specific configuration and is not scalable to manipulators with a general mechanical structure. To solve the kinematic control problem with general configurations, some controllers are proposed, including adaptive control methods [13, 14], barrier-Lyapunov-function based methods [15, 16], and Jacobian-matrix-pseudo-inverse (JMPI) methods [17–19]. In [17], an asymmetric barrier Lyapunov function-based method is introduced to handle the output limitation. This method consists of a full state feedback controller and an output feedback controller. Using the JMPI method, one can get the control signals in joint space according to the desired path and the pseudo-inverse of the Jacobian matrix. For a redundant manipulator, there is a null space for the Jacobian matrix [20], which is helpful to design controllers considering a secondary task. Therefore, JMPI based methods have been widely used in redundancy solutions. Galicki [21] proposes a JMPI based tracking controller, and an alternative method around the singular point is discussed. In [22], a weighted damped least-squares method is developed to calculate pseudo-inverse around the singularity, an appropriate damping factor is derived according to the minimum singular value. In [23], pseudo-inverse of Jacobian is calculated online by a Taylor-type discrete-time neural network, which is composed of T-ZNN-K and T-ZNN-U models. In [24], a special type of nonlinear function based neural network is designed for tracking control of a PA10 manipulator, and the finite-time convergence of tracking error is also analyzed.

Although the above-mentioned methods have achieved great success, those methods are afflicted with several major limitations in scenarios that require higher performance of real-time ability, accuracy, and self-adaptation. Firstly, the precise kinematic parameters are required in existing works. Describing the mapping from the joint movement in joint space to the movement of the end-effector in Cartesian space, the Jacobian matrix contains kinematic characteristics, such as configuration and kinematic parameters. For a specified robot, the configuration can be derived, but it is usually difficult to obtain accurate kinematic parameters. For example, because of the manufacturing error, different operation tools, etc., the DH parameters may differ to the reference ones in official guidebooks [25]. In this case, the Jacobian matrix based on the inaccurate parameters would cause errors in pseudo-inverse calculation and even instability of the system [26]. On the other hand, the calculation of pseudo-inverse operation is time-consuming, which would lead to a huge cost in real applications which requires pseudo-inverse calculation in every control cycle. Additionally, due to mechanical reasons, the robot manipulator is suffered from physical constraints, such as joint angular and speed limitations.

In terms of the kinematic control in the presence of model uncertainties, the real-time feedback of the end-effector enables closed-loop control for researchers. This can be done by high performance measuring devices such as high precision cameras and laser trackers [27]. In [28], based on the parameter linearization property, a robust controller is proposed, which shows semi-global stability in regulation control in fix-point control. As to the tracking control of manipulators, Hou proposes a neural network based control strategy, in which the position/orientation of the robot is described by a unit quaternion, and the network is used to learn the unknown nonlinear part of the system. One main contribution of the research is that singularities associated with three-parameter representation can be avoided. Cheah et al. propose several adaptive controllers for manipulators in different industrial applications, such as visual tracking, force tracking and trajectory tracking [30–34]. In [35], Chen and Zhang design a new adaptive controller in the acceleration level, the basic idea is that the Jacobian matrix is updated in realtime rather than kinematic parameters. One major drawback of the strategy is that the controller requires the actual values of end velocity and acceleration, which may contain noise in actual applications. In order to reduce the influence of noise in sensor feedback, Wang introduces a low-pass filter, and then an adaptive torque calculation controller is designed in the inner loop [36]. Xu develops a modified controller [37], in which the joint command is deigned at the acceleration level. It is verified that the controller does not require measurement of end velocity and joint acceleration. The influence of the control parameters on tracking errors and convergence rate is also discussed. The above methods mainly focus on the uncertain model parameters, and the redundancy of the manipulators is not considered. Despite the pseudo-inverse can be used instead of traditional inverse calculation of Jacobian matrix, the disadvantage of JMPI methods remains unresolved. In order to overcome the limitations based on the JMPI method, researches transform this problem into a quadratic programming problem, with the aim of obtaining an optimal solution with the specified evaluation index under the physical constraint. Physical constraints can be formulated into equation constraints or inequality constraints. Zhang et al. [38] develops a dual neural network to solve quadratic programming problems, and it is shown that this strategy is suitable for redundancy solutions. Based on the idea, a series of research is reported in eliminating the position error accumulation [39], nonconvex optimization [40], acceleration-level compliance [41], parallel robots [42] and multiple robot systems [43].

Inspired by the above literature, in this chapter, we focus on the adaptive tracking problem for redundant manipulators. The remained of this chapter is arranged as below. In Sect. 2.2, fundamental robot kinematics together with useful properties are given, we also show the control objective. In Sect. 2.3, an adaptive Jacobian method is designed by updating kinematic parameters online, and a RNN is used to achieve redundant resolution based on the estimated Jacobian matrix, convergence analysis of the tracking error in Cartesian space is also discussed. In Sect. 2.4, numerical results and comparisons are conducted on a 6DOF robot JACO$_2$. Finally, conclusions are drawn in Sect. 2.5. Before ending the introductory section, we highlight the main contributions of this chapter as below:

- As far as we know, for the first time, this chapter proposes an RNN based controller via kinematic regressing for redundant manipulator subject to model uncertainties.
- Using the Lyapunov theory, the convergence of tracking error is proved in the case of uncertain parameters.

- In this chapter, there is no need to calculate the pseudo-inverse of the Jacobian matrix, which can avoid singularity problems effectively, and also reduce computational load.
- The minimum velocity solution is derived by the proposed control scheme, thus the safety of both humans and robots can be also guaranteed.

## 2.2 Problem Formulation and Existing Results

### 2.2.1 Robot Kinematics

Without loss of generality, we consider serial robot manipulators in this chapter. The kinematic model for robot manipulators is described as follows:

$$f(\theta(t)) = x(t), \tag{2.1}$$

where $\theta(t) \in \mathbb{R}^n$ represents the vector of the joint angles at time $t$, and $x(t) \in \mathbb{R}^m$ represents the Cartesian coordinate vector of the end effector. For a specific robot manipulator, $f(\bullet) : \mathbb{R}^n \to \mathbb{R}^m$ is used to describe the forward kinematics from joint space to Cartesian space, which is a continuous nonlinear mapping containing kinematic parameters and structure information.

By differentiating $x(t)$ with respect to time $t$, we can get the relationship between Cartesian velocity $\dot{x}(t) \in \mathbb{R}^m$ and joint velocity (or joint control signal) $\dot{\theta}(t) \in \mathbb{R}^n$ as follows:

$$J(\theta(t), a_k)\dot{\theta}(t) = \dot{x}(t), \tag{2.2}$$

with $J(\theta(t), a_k) = \partial f(\theta(t), a_k)/\partial\theta(t)$ being the Jacobian matrix, and $a_k \in \mathbb{R}^l$ denotes the vector of kinematic parameters.

Once the physical structure of manipulator is determined, its kinematic equation (2.2) satisfies the following linearization property [43], which describes the relationship between the robot's end velocity and its kinematic parameters:

$$J(\theta(t), a_k)\dot{\theta}(t) = Y_k(\theta(t), \dot{\theta}(t))a_k, \tag{2.3}$$

where $Y_k(\theta(t), \dot{\theta}(t)) \in \mathbb{R}^{m\times l}$ is called kinematic regressor matrix. Remarkable that $Y_k(\theta(t), \dot{\theta}(t))$ is the function of $\theta(t)$ and $\dot{\theta}(t)$, and has no relation with $a_k$.

### 2.2.2 Control Objective

In this chapter, we consider the task space tracking problem for redundant manipulators, where precise values of kinematic parameters are unavailable. The measurable states are joint angles $\theta(t)$ and the end coordinates $x(t)$. The desired Cartesian path $x_{\mathrm{d}}(t) \in \mathbb{R}^m$ and its time derivative $\dot{x}_{\mathrm{d}}(t)$ are accessible, both $x_{\mathrm{d}}(t)$ and $\dot{x}_{\mathrm{d}}(t)$ are bounded. The nominal value of the kinematic parameter vector is also available, which is denoted as $a_k^n$.

The control objective is to generate joint velocity command in real-time, e.g., designing $\dot{\theta}(t)$ to drive the end-effector of a redundant robot to track $x_{\mathrm{d}}(t)$ , in the sense that $f(\theta(t)) = x(t) \to x_{\mathrm{d}}(t)$. During the whole tracking process, velocity of every joint $\dot{\theta}_i(t)$ should not exceed its limits $\dot{\theta}^i_{\min}$, $\dot{\theta}^i_{\max}$.

## 2.3 Main Results

In this section, we will show the detailed process of the controller design. When controlling a redundant robot, one important problem is to make use of its flexibility, such as avoiding obstacles, optimizing energy consumption, and avoiding Singularities, etc. In this chapter, when the kinematic controller is designed to achieve task-space tracking in the presence of physical uncertainties, we consider the energy-saving problem at speed level. Therefore, the secondary task in set to minimize the velocity norm $u^{\mathrm{T}}u$. The control strategy consists of three parts: a position controller in the outer-loop, a Jacobian adaption part which is capable of handling kinematic uncertainties online, and an RNN which is used to solve the redundancy resolution problem. The stability of the closed-loop system will be also discussed.

### 2.3.1 *Position Controller*

Firstly, a precise measurement of actual coordinate $x$ in real-time $t$ is taken to build the closed-loop system. The difference between the desired path and the corresponding feedback can be defined as

$$e(t) = x_{\mathrm{d}}(t) - x(t). \tag{2.4}$$

In order to make $e(t)$ converge to 0, by using the zeroing dynamics [53], the derivative of $e(t)$ is designed as

$$\dot{e}(t) = -ke(t), \tag{2.5}$$

with $k > 0$ being a positive constant scaling the convergence rate of $e(t)$. Combining (2.4) and (2.5) yields

$$\dot{x}(t) = \dot{x}_{\mathrm{d}}(t) + k(x_{\mathrm{d}}(t) - x(t)). \tag{2.6}$$

Let $\dot{\theta}(t) = u(t)$, according to (2.6), if $u(t)$ is properly designed to make the robot's end-effector move at a speed of $\dot{x}(t)$, in the sense that $\dot{x}(t) = J(\theta(t), a_k)u(t)$, the tracking error $e(t)$ in task-space would convergence to zero exponentially.

When $a_k$ is unknown, the precise Jacobian matrix described in (2.2) is unavailable. The redundancy solution can not be achieved by using $J(\theta, a_k)$. Therefore, we use the estimate Jacobian $J(\theta(t), \hat{a}_k)$ by replacing the unknown parameters $a_k$ in $J(\theta(t), a_k)$ with its estimate $\hat{a}_k$, and the initial value of $\hat{a}_k$ is set as $\hat{a}_k(0) = a_k^n$, the estimate error is defined as $\tilde{a}_k = \hat{a}_k - a_k$. Using $J(\theta(t), \hat{a}_k)$ and the control signal $\dot{\theta}(t)$, we can estimate the velocity of the end-effector as

$$\hat{\dot{x}}(t) = J(\theta(t), \hat{a}_k(t))u(t). \tag{2.7}$$

Remarkable that the linearization property described in (2.3) still holds for estimated $\hat{a}_k$:

$$J(\theta(t), \hat{a}_k(t))u(t) = Y_k(\theta(t), u(t))\hat{a}_k(t), \tag{2.8}$$

this property will be used in the following stability proof. The adaptive Jacobian method by updating its kinematic parameters $\hat{a}_k$ is thus developed as

$$\dot{\hat{a}}_k(t) = -\Gamma_1 Y_k^{\mathrm{T}}(\theta(t), u(t))e(t), \tag{2.9}$$

where $\Gamma_1 \in \mathbb{R}^{l\times l}$ is a diagonal positive definite matrix, $e(t)$ is the tracking error in Cartesian space as defined in (2.4), and $u(t)$ is the bounded joint speed vector satisfying $J(\theta(t), \hat{a}_k)u(t) = \dot{x}(t)$, which will be designed later. Unless otherwise specified, $J(\theta(t), \hat{a}_k)$ is simplified as $\hat{J}$.

**Remark 2.1** Figure 2.1 gives a brief framework of the proposed control scheme for redundant manipulators with uncertain kinematic parameters. The desired trajectory of the end-effector is specified by $x_{\mathrm{d}}(t)$ and $\dot{x}_{\mathrm{d}}(t)$. The desired trajectory together

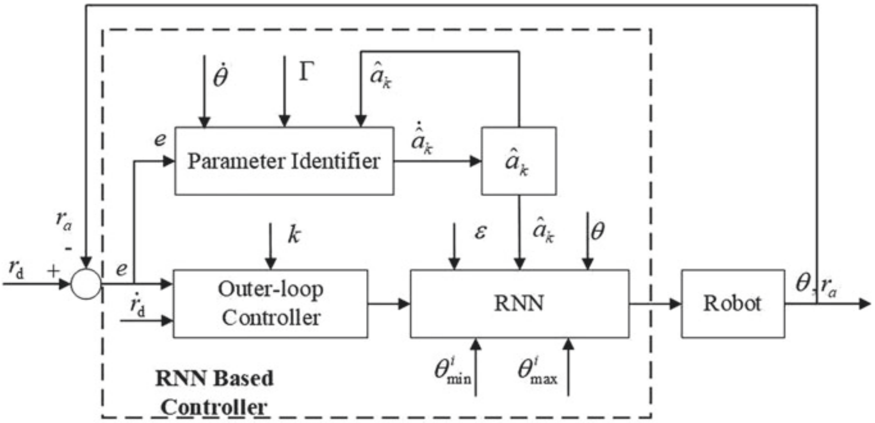


**Fig. 2.1** Framework of the proposed scheme for redundant manipulators with uncertain kinematics, in which the neural control algorithm includes three interactive modules, i.e., position control module, parameter identification module, and redundancy resolution module

with feedback $x(t)$ are fed into the position controller (2.6). The tracking error $e(t)$ and joint speed $\dot{\theta}(t)$ are used to learn the kinematic parameters online by identifier (2.9). According to the output of the position controller, identified parameter $\hat{a}_k$, feedback of the manipulator and physical limits, an RNN based controller is used to achieve the redundancy solution problem.

### *2.3.2 Redundant Solution Using RNN*

In this subsection, we focus on redundancy resolution problem based on Jacobian adaption method introduced in Sect. 2.3.1. The main purpose of redundancy resolution is to find an optimal joint speed $u(t)$ to make equation $J(\theta(t), a_k)u(t) = \dot{x}_{\mathrm{d}}(t) + k|e(t)|^{\rho}\mathrm{sgn}(e(t))$ hold, at the same time, a secondary task can be also achieved. The redundancy resolution problem can be converted into a quadratic optimization one with specified constraints. To minimize the kinetic energy of the robot, we select velocity norm $u^{\mathrm{T}}u = \dot{\theta}^{\mathrm{T}}\dot{\theta}$ as an object function to be optimized, the joint range $\theta^i_{\min} \le \theta_i \le \theta^i_{\max}$ and joint speed limits $\dot{\theta}^i_{\min} \le \dot{\theta}_i \le \dot{\theta}^i_{\max}$ are regarded as inequality constrains. Because $J(\theta, a_k)$ is unavailable, we use $\hat{J}$ instead of $J(\theta, a_k)$, and rewrite $\dot{x} = b_0$. Then the redundancy resolution problem is reformulated as the following quadratic optimization formulations:

$$\min \ u^{\mathrm{T}}u, \tag{2.10a}$$

$$\text{s.t.} \ \ b_0 = \hat{J}u, \tag{2.10b}$$

$$u \in \Omega, \tag{2.10c}$$

where $\Omega = \{u \in \mathbb{R}^n | u^i_{\min} \le u_i \le u^i_{\max}\}$ is a convex set describing the physical constraints, where $u^i_{\min} = \max\{\alpha(\theta^i_{\min} - \theta_i), \dot{\theta}^i_{\min}\}$, $u^i_{\max} = \min\{\alpha(\theta^i_{\max} - \theta), \dot{\theta}^i_{\max}\}$, $\alpha > 0$ is a positive constant. The convex set ensures the boundedness of both joint angles and speed [44]. According to the Karush−Kuhn−Tucker condition [45], an equivalent description of the optimal solution to the quadratic optimization as shown in (2.10) is described as

$$u = P_{\Omega}(u - \partial L/\partial u), \tag{2.11a}$$

$$b_0 = \hat{J}u, \tag{2.11b}$$

where $P_{\Omega}(\bullet)$ is a projection operation to the set $\Omega$, $P_{\Omega}(x) = \mathrm{argmin}_{y \in \Omega}||y - x||$, and $L = L(u, \lambda)$ is a Lagrange function defined as $L(u, \lambda) = u^{\mathrm{T}}u/2 + \lambda^{\mathrm{T}}(b_0 - \hat{J}u)$, where $\lambda \in \mathbb{R}^m$ is a Lagrange multiplier corresponding to the equality constraint.

Note that the difference between $\hat{J}$ and $J$ would lead to extra error, which will result in tracking failure. To resolve the quadratic optimization problem (2.11), we are ready to present the RNN based controller together with the updating kinematic parameters online:

**Algorithm 1** The proposed tracking method

```
Input: Parameters $k, \alpha, \Gamma, \epsilon$, joint angle limits $\theta^i_{\max}, \theta^i_{\min}$, joint speed limits $\dot{\theta}^i_{\max}, \dot{\theta}^i_{\min}$, initial states $u(0), \theta(0)$, nominal kinematic parameter $\hat{a}_k(0)$, desired path $x_{\mathrm{d}}(t), \dot{x}_{\mathrm{d}}(t)$, task duration $T$, feedback of end effector $x(t)$, analytical expressions of estimated Jacobian matrix $\hat{J}$ and kinematic regressor matrix $Y_k$.
Output: To achieve task-space tracking of the redundant manipulator
Initialize $\lambda(0), a_k(0) \leftarrow a^n_k$.
Repeat
    $x, \theta \leftarrow$ Sensor readings
    Calculate tracking error $e \leftarrow$ Equation (2.4)
    Position loop controller $b_0 \leftarrow$ Equation (2.6)
    Update control output $u$ by $\dot{u} \leftarrow$ Equation (2.12a)
    Update state variable $\lambda$ by $\dot{\lambda} \leftarrow$ Equation (2.12b)
    Update model parameter $\hat{a}_k$ by $\dot{\hat{a}}_k \leftarrow$ Equation (2.12c)
Until $(t > T)$
```

$$\varepsilon \dot{u} = -u + P_{\Omega}(-\hat{J}^{\mathrm{T}} \lambda), \tag{2.12a}$$

$$\varepsilon \dot{\lambda} = \hat{J} u - b_0, \tag{2.12b}$$

$$\dot{\hat{a}}_k = -\Gamma_1 Y_k^{\mathrm{T}}(\theta, u) e, \tag{2.12c}$$

where $\varepsilon$ is a positive factor scaling the convergence of RNN. The proposed control scheme is shown in Algorithm 2.3.2.

**Remark 2.2** It is worth pointing that although the proposed RNN in (2.12a) and (2.12b) looks similar to existing ones (*e.g.*, [46, 47]), the modification is very meaningful. The proposed RNN is capable of handling kinematic uncertainties. When the kinematic parameters $a_k$ in known, $\hat{J}$ is equal to $J$, (2.12a) and (2.12b) have the same expression with traditional ones, which shows that a known parameter case is only a special form of our control scheme, thus the proposed RNN is more general. The proposed control scheme offers an important expansion to model uncertainties, which is of universal significance in engineering applications.

**Remark 2.3** Using the proposed RNN based controller, the control command $u(t)$ can be derived according to (2.12a), which is capable of optimizing $u^{\mathrm{T}}u$, meanwhile, the projection operation $P_{\omega}(\bullet)$ handles inequality constraints. (2.12b) plays an important role in task-space tracking. By referring to (2.12c), we update the Jacobian indirectly by renewing its kinematic parameters online based on the property (2.3), which is different with other Jacobian adaption mathods (e.g., [48]), where joint acceleration is required. The necessary values of our updating law are joint angle $\theta$, joint speed $u$ and tracking error $e$, therefore, the proposed control strategy can be realized easily.

### 2.3.3 Convergence Analysis

In this part, we conduct theoretical analysis on the convergence of tracking error under the RNN based tracking controller (2.12a) and (2.12b) along with the kinematic parameter updating law described in (2.12c).

Firstly, two Lemmas are offered as below, which will be used in the proof process of convergence analysis.

**Lemma 2.1** *For any closed convex set* $\Lambda \in \mathbb{R}^p$, $(x - P_\Lambda(x))^T(P_\Lambda(x) - y) \geq 0$, $\forall y \in \Lambda$, $\forall x \in \mathbb{R}^p$, *and the equality holds only if* $x \in \Lambda$ *[49].*

**Lemma 2.2** *For any closed convex set* $\Lambda \in \mathbb{R}^p$, $(x - P_\Lambda(x))^T(x - y) \geq 0$, $\forall y \in \Lambda$, $\forall x \in \mathbb{R}^p$, *and the equality holds only if* $x \in \Lambda$ *[47].*

Based on Lemma 1 and 2, we can obtain the following theorem about convergence of tracking error under the proposed redundancy resolution scheme (2.12).

**Theorem 2.1** The control error $e(t)$ defined in (2.4) for a redundant manipulator globally converges to 0, provided the RNN based redundancy resolution (2.12a) and (2.12b), along with the kinematic adaptation law (2.12c).

*Proof:* The convergence analysis includes two parts. Firstly, we will prove that the output $u$ of proposed RNN (2.12a), (2.12b) would reach the optimal solution of (2.11). Secondly, we will show the convergence of tracking error $e$ along with the adaptation law (2.12c). Note that the proof bears similarity to that with known parameters, but the extra dynamics on parameter adaptation makes it necessary to analyze the joint stability, which constructs the main difference of this proof from existing works.

Part I. By defining $\xi = [u^{\mathrm{T}}, \lambda^{\mathrm{T}}]^{\mathrm{T}}$, controller (2.12a), (2.12b) can be reformulated as

$$\epsilon\dot{\xi} = -\xi + P_{\bar{\Omega}}(\xi - R(\xi)), \tag{2.13}$$

where $\bar{\Omega} = \{(u, \lambda)|u \in \Omega, \lambda \in \mathbb{R}^m\}$, and $R(\xi) = [u - \hat{J}^{\mathrm{T}}\lambda, -b_0 + \hat{J}u]^{\mathrm{T}}$. Define $\nabla R = \partial R(\xi)/\partial \xi$, we have

$$\nabla R = \begin{bmatrix} I & -\hat{J}^{\mathrm{T}} \\ \hat{J} & 0 \end{bmatrix},$$

where $I$ is a $n$-dimensional identity matrix, and $\nabla R \in \mathbb{R}^{(m+n)\times(m+n)}$ is a skew symmetric matrix. The transpose matrix of $\nabla R$ is defined as $\nabla^{\mathrm{T}} R$. Remarkable that $\nabla R$ satisfies the following positive semi-definite property:

$$y^{\mathrm{T}}\nabla R y = y^{\mathrm{T}}(\nabla R + \nabla^{\mathrm{T}} R)y/2 \geq 0, \ \forall y \in \mathbb{R}^{m+n}. \tag{2.14}$$

This property will be used later. Define the following Lyapunov function candidate as

$$V_1 = ||\xi - P_{\bar{\Omega}}(\xi)||_2^2/2. \tag{2.15}$$

It is obvious that $V_1 = 0$ if and only if $\xi \in \bar{\Omega}$. According to the result of reference in E.9, $\partial||\xi - P_{\bar{\Omega}}(\xi)||_2^2/\partial\xi = 2(\xi - P_{\bar{\Omega}}(\xi))$. Differentiating $V_1$ with respect to time and substituting (2.13) yield:

$$\begin{aligned}\dot{V}_1 =&(\xi - P_{\bar{\Omega}}(\xi))^{\mathrm{T}}\dot{\xi}\\ =&-(\xi - P_{\bar{\Omega}}(\xi))^{\mathrm{T}}(\xi - P_{\bar{\Omega}}(\xi - R(\xi)))/\epsilon.\end{aligned} \tag{2.16}$$

There is no doubt that $P_{\bar{\Omega}}(\xi - R(\xi)) \in \bar{\Omega}$, according to Lemma 2, $\forall\xi \in \mathbb{R}^{m+n}$ satisfies the inequality $(\xi - P_{\bar{\Omega}}(\xi))^{\mathrm{T}}(\xi - P_{\bar{\Omega}}(\xi - R(\xi))) \geq 0$. Then we have $\dot{V}_1 \leq 0$ because $\epsilon > 0$, and $\dot{V}_1 = 0$ only if $\xi \in \bar{\Omega}$. Based on to LaSalle's invariance principle [50], it can be proved that $\xi$ gradually converges into $\bar{\Omega}$, which indicates $u$ converges into $\Omega$ , the boundedness of joint angles and speed is thus guaranteed. Note that the equilibrium point $\xi^*$ satisfies

$$\xi^* = P_{\bar{\Omega}}(\xi^* - R(\xi^*)). \tag{2.17}$$

According to definition 1 and Lemma 1 in [51], $\xi^*$ satisfies the following property

$$(y - \xi^*)^{\mathrm{T}}R(\xi^*) \geq 0, \quad \forall y \in \bar{\Omega}. \tag{2.18}$$

Define function $V_2$ as

$$\begin{aligned}V_2 =&(\xi - P_{\bar{\Omega}}(\xi - R(\xi)))^{\mathrm{T}}R(\xi) + \|\xi - \xi^*\|_2^2/2\\ &-||\xi - P_{\bar{\Omega}}(\xi - R(\xi))||_2^2/2 + V_1.\end{aligned} \tag{2.19}$$

Some mathematical calculations on the first and third items of the definition (2.19) give

$$\begin{aligned}&(\xi - P_{\bar{\Omega}}(\xi - R(\xi)))^{\mathrm{T}}R(\xi) - \|\xi - P_{\bar{\Omega}}(\xi - R(\xi))\|_2^2/2\\ \geq&(\xi - P_{\bar{\Omega}}(\xi - R(\xi)))^{\mathrm{T}}R(\xi) - \|\xi - P_{\bar{\Omega}}(\xi - R(\xi))\|_2^2\\ =&(\xi - R(\xi) - P_{\bar{\Omega}}(\xi - R(\xi)))^{\mathrm{T}}(P_{\bar{\Omega}}(\xi - R(\xi)) - \xi).\end{aligned} \tag{2.20}$$

Noticing that $\xi$ would gradually converge into the convex set $\bar{\Omega}$, then we get $\xi \in \bar{\Omega}$. According to Lemma 1, inequality $(\xi - P_{\bar{\Omega}}(\xi - R(\xi)))^{\mathrm{T}} \cdot (P_{\bar{\Omega}}(\xi - R(\xi)) - \xi) \geq 0$ holds for any $\xi - R(\xi) \in \mathbb{R}^{m+n}$. Recalling the definition of $V_2$, we have

$$V_2 \geq ||\xi - \xi^*||_2^2/2 + V_1. \tag{2.21}$$

Thus $V_2$ is a Lyapunov function candidate. Differentiating $V_2$ with respect to time and combining (2.13) yields:

$$\dot{V}_2 =(\xi - P_{\bar{\Omega}}(\xi - R(\xi)))^{\mathrm{T}}\nabla R\dot{\xi} + \dot{\xi}^{\mathrm{T}}R(\xi) + (\xi - \xi^*)^{\mathrm{T}}\dot{\xi}$$

$$
\begin{aligned}
&- (\xi - P_{\bar{\Omega}}(\xi - R(\xi)))^{\mathrm{T}}\dot{\xi} + \dot{V}_1 \\
=&- (\xi - P_{\bar{\Omega}}(\xi - R(\xi)))^{\mathrm{T}}\nabla R(\xi - P_{\bar{\Omega}}(\xi - R(\xi)))/\epsilon \\
&- (\xi - R(\xi) - P_{\bar{\Omega}}(\xi - R(\xi)))^{\mathrm{T}}(P_{\bar{\Omega}}(\xi - R(\xi)) - \xi^*)/\epsilon \\
&- (\xi - \xi^*)^{\mathrm{T}} R(\xi)/\epsilon + \dot{V}_1.
\end{aligned}
\tag{2.22}
$$

Remarkable that $\xi^* \in \bar{\Omega}$ , according to Lemma 1, inequality $(\xi - R(\xi) - P_{\bar{\Omega}}(\xi - R(\xi)))^{\mathrm{T}}(P_{\bar{\Omega}}(\xi - R(\xi)) - \xi^*) \geq 0$ holds for any $\xi - R(\xi) \in \mathbb{R}^{m+n}$. Using (2.14), we have $-(\xi - P_{\bar{\Omega}}(\xi - R(\xi)))^{\mathrm{T}}\nabla R(\xi - P_{\bar{\Omega}}(\xi - R(\xi)))/\epsilon \leq 0$ since $\epsilon > 0$. According to mean value theorem, we have

$$
R(\xi) - R(\xi^*) = \nabla R(\xi')(\xi - \xi^*),
\tag{2.23}
$$

where $\xi' \in [\xi, \xi^*]$. After some mathematical calculations on the following polynomials and substituting (2.23), we have

$$
\begin{aligned}
&(\xi - \xi^*)^{\mathrm{T}} R(\xi) \\
=&(\xi - \xi^*)^{\mathrm{T}}\nabla R(\xi')(\xi - \xi^*) + (\xi - \xi^*)^{\mathrm{T}} R(\xi^*).
\end{aligned}
\tag{2.24}
$$

Using the properties (2.14) and (2.18), we have $(\xi - \xi^*)^{\mathrm{T}}\nabla R(\xi')(\xi - \xi^*) \geq 0$ and $(\xi - \xi^*)^{\mathrm{T}} R(\xi^*) \geq 0$, then

$$
(\xi - \xi^*)^{\mathrm{T}} R(\xi) \geq 0.
\tag{2.25}
$$

Combining inequalities (2.16), (2.24) and (2.25) yields $\dot{V}_2 \leq 0$, and $\dot{V}_2 = 0$ only if $\xi \in \bar{\Omega}$, which indicates $(\xi - P_{\bar{\Omega}}(\xi - R(\xi)))^{\mathrm{T}}\nabla R(\xi - P_{\bar{\Omega}}(\xi - R(\xi))) = 0$, $(\xi - R(\xi) - P_{\bar{\Omega}}(\xi - R(\xi)))^{\mathrm{T}}(P_{\bar{\Omega}}(\xi - R(\xi)) - \xi^*) = 0$ and $(\xi - \xi^*)^{\mathrm{T}} R(\xi) = 0$. From (2.24), we get $(\xi - \xi^*)^{\mathrm{T}}\nabla R(\xi')(\xi - \xi^*)$ and $(\xi - \xi^*)^{\mathrm{T}} R(\xi^*) = 0$. Notable that $\xi = \xi^*$ is the solution of the above equations. Based on LaSalle's invariance principle, we arrive at a conclusion that $\xi$ would gradually reach its equilibrium point $\xi^*$, *i.e.*, $u(t)$ would converge to its optimal solution of redundancy resolution problem (2.10).

Part II. Consider the Lyapunov function candidate

$$
V_3 = e^{\mathrm{T}} e/2 + \tilde{a}_k^{\mathrm{T}} \Gamma_1^{-1} \tilde{a}_k/2.
\tag{2.26}
$$

Differentiating $V_3$ with respect to time and substituting (2.4), (2.8) and (2.9), we have

$$
\begin{aligned}
\dot{V}_3 =& e^{\mathrm{T}}(\dot{x}_{\mathrm{d}} - \dot{x}) + \tilde{a}_k^{\mathrm{T}} \Gamma_1^{-1} \dot{\tilde{a}}_k \\
=& e^{\mathrm{T}}(\dot{x}_{\mathrm{d}} - J(\theta, a_k)u + (1-1)k|e|^{\rho}\mathrm{sgn}(e) + \tilde{a}_k^{\mathrm{T}} \Gamma_1^{-1} \dot{\tilde{a}}_k \\
=& e^{\mathrm{T}}(b_0 - Y_k(\theta, u)(\hat{a}_k - \tilde{a}_k) - k|e|^{\rho}\mathrm{sgn}(e)) - \tilde{a}_k^{\mathrm{T}} Y_k^{\mathrm{T}}(\theta, u)e \\
=& e^{\mathrm{T}}(b_0 - \hat{J}u + Y_k(\theta, u)\tilde{a}_k) - k|e|^{\rho+1} - \tilde{a}_k^{\mathrm{T}} Y_k^{\mathrm{T}}(\theta, u)e.
\end{aligned}
\tag{2.27}
$$

As proved above, using the neural network (2.12), $u^{\mathrm{T}} u$ will be minimized under the constraints $b_0 = \hat{J}u$ and $u \in \Omega$. Notable that $\tilde{a}_k^{\mathrm{T}} Y_k^{\mathrm{T}}(\theta, u)e$ is a scalar value, we have $\tilde{a}_k^{\mathrm{T}} Y_k^{\mathrm{T}}(\theta, u)e = e^{\mathrm{T}} Y_k(\theta, u)\tilde{a}_k$. Then (2.27) can be rewritten as

$$\begin{aligned}\dot{V}_3 =& e^{\mathrm{T}} Y_k(\theta, u)\tilde{a}_k - k|e|^{\rho+1} - \tilde{a}_k^{\mathrm{T}} Y_k^{\mathrm{T}}(\theta, u)e \\ =& -k|e|^{\rho+1} \leq 0.\end{aligned} \tag{2.28}$$

Then we have $e = x_{\mathrm{d}} - x$ is bounded. Taking the time derivative of $\dot{V}_3$, we have:

$$\begin{aligned}\ddot{V}_3 =& -k(\rho+1)|e|^{\rho}\mathrm{sgn}(e)\dot{e} \\ =& -k(\rho+1)|e|^{\rho}\mathrm{sgn}(e)(\dot{x}_{\mathrm{d}} - J(\theta, a_k)u).\end{aligned} \tag{2.29}$$

Since $J(\theta, a_k)$ is composed of trigonometric functions of $\theta$ and kinematic parameters $a_k$, $J(\theta, a_k)$ is bounded. $\dot{x}_{\mathrm{d}}$ is also bounded. As illustrated in part I, $u$ is bounded, thus $\ddot{V}_3$ is guaranteed to be bounded. Using Barbalat's lemma [52], we have $\dot{V}_3 \to 0$ as $t \to \infty$. Then $e \to 0$ as $t \to \infty$. This completes the Proof. □

**Remark 2.4** The convergence analysis shows the stability of the proposed control strategy. The tracking error would globally convergence to 0. The proof also illustrates that the control command $u(t)$ is ensured $u(t) \in \Omega, \forall t \geq 0$, provided $u(0) \in \Omega$, the boundedness of joint speed is thus guaranteed all the time.

## 2.4 Illustrative Examples

### 2.4.1 Numerical Setup

We consider the position tracking problem in task space, then JACO$_2$ can thus be regarded as a functional redundant manipulator. The architecture of JACO$_2$ is shown in Fig. 2.2, and the DH parameters are shown in Table 2.1. Noticing that the last 3 joints of JACO$_2$ do not intersect at a single point, these joints cannot be simplified as spherical joint, therefore the configuration of JACO$_2$ is more general than other 6DOF manipulators, e.g., the PUMA 560. The initial state of joint position vector

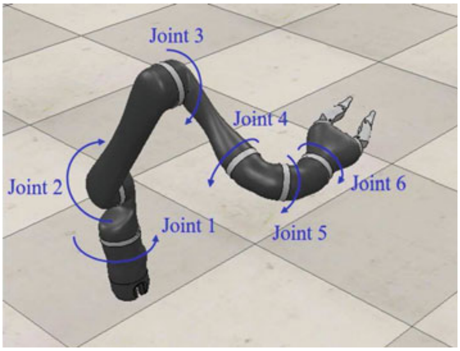


**Fig. 2.2** The physical structure of JACO$_2$

**Table 2.1** DH parameters of the Kinova JACO$_2$ robot manipulator

| Link | $a_i$ (m) | $d_i$ (m) | $\alpha_i$ (rad) | $\theta_i$ (rad) |
|---|---|---|---|---|
| 1 | 0 | 0.2755 | $\pi/2$ | $\theta_1$ |
| 2 | 0.41 | 0 | $\pi$ | $\theta_2 + \pi/2$ |
| 3 | 0 | $-0.0098$ | $\pi/2$ | $\theta_3 - \pi/2$ |
| 4 | 0 | $-0.2501$ | $\pi/3$ | $\theta_4$ |
| 5 | 0 | $-0.0856$ | $\pi/3$ | $\theta_5 + \pi$ |
| 6 | 0 | $-0.2028$ | $\pi$ | $\theta_6 - \pi/2$ |

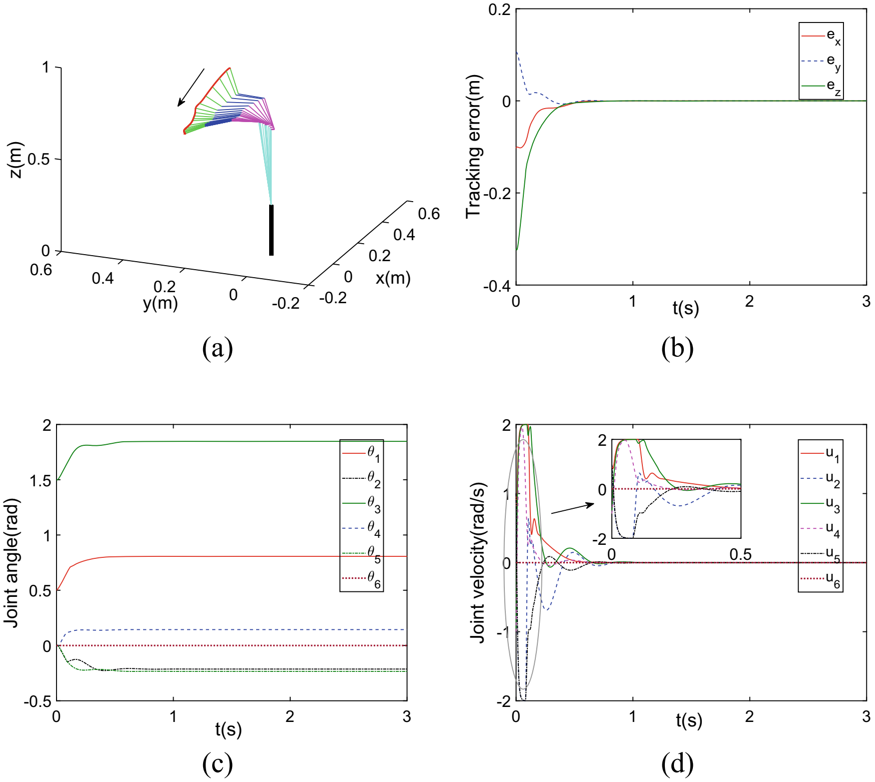
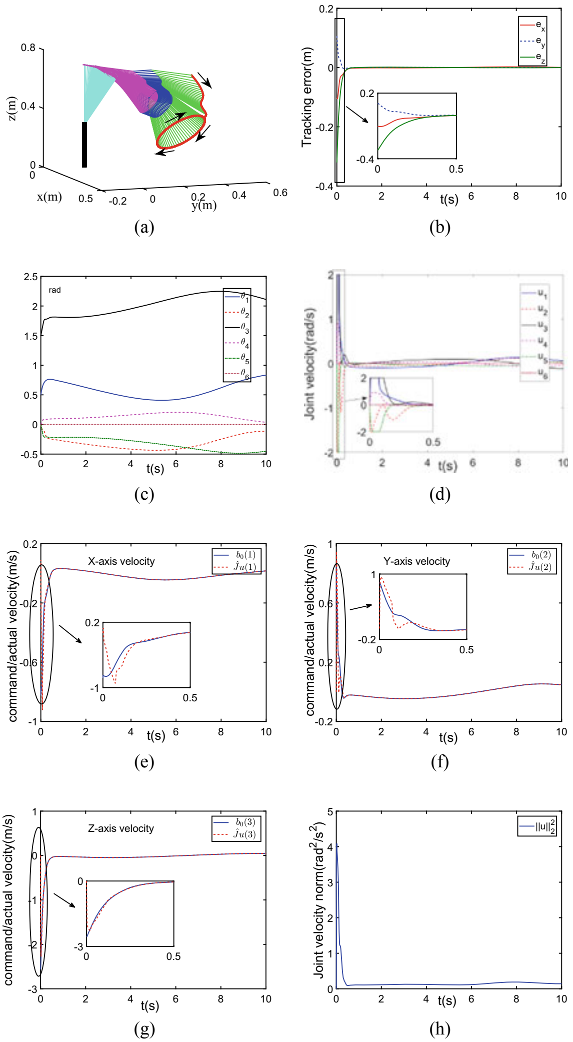


**Fig. 2.3** Results of regulation control on JACO$_2$ to a fixed point [0.3, 0.4, 0.4]m in the Cartesian space. **a** Motion trajectory of end effector (red curve) and the corresponding incremental configurations of JACO$_2$. **b** Error-time curve along three directions. **c** Angle-time curve of 6 joints. **d** Command-time curve of joint velocity $u$

**Fig. 2.4** Results when JACO$_2$ tracks a given circle in Cartesian space. **a** Motion trajectory of end effector (red curve) and the corresponding incremental configurations of JACO$_2$. **b** Error-time curve along three directions. **c** Angle-time curve of 6 joints. **d** Command-time curve of joint velocity $u$. **e** The first Cartesian velocity input $b_0$(x-axis direction) described by (2.6) and the corresponding output $\hat{J}u(1)$. **f** The second Cartesian velocity input $b_0$(y-axis direction) described by (2.6) and the corresponding output $\hat{J}u(2)$. **g** The third Cartesian velocity input $b_0$(z-axis direction) described by (2.6) and the corresponding output $\hat{J}u(3)$. **h** The Euclidean norm of the manipulator's joint velocity



$\theta(0)$ is randomly set as $[0.5, 0, 1.5, 0, 0, 0]^{\mathrm{T}}$rad, and the initial joint speed $u(0)$ is selected to be zero. The nominal values of kinematic parameters are selected as $\hat{a}_k(0) = [0.25, 0.2, 0, -0.2, -0.1, -0.2]^{\mathrm{T}}$m. The set $\Omega$ describing joint speed limits are set to be $[-2, 2]^6$rad/s. The control gain $k$ is set to be 8, and the gain matrix $\Gamma_1$ is selected as $0.5I$, where $I$ is a 6-dimensional identity matrix (Fig. 2.4).

### 2.4.2 Fixed-point Regulation

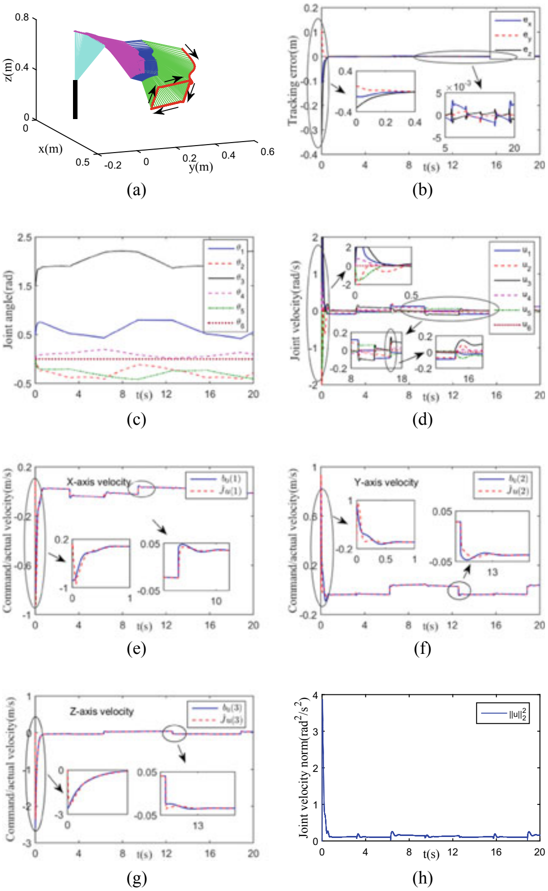
In order to verify the proposed tracking strategy, a set value adjustment experiment is carried out. Set the desired fixed position of the end-effector to $[0.3, 0.4, 0.4]^{\mathrm{T}}$m, select the zoom factor to be $\epsilon = 0.008$. The simulation results are shown in Fig. 2.3. The adjustment error converges to zero, and the convergence time is about 0.5s, as shown in Fig. 2.3b. Therefore, the joint angle $\theta$ reaches a set of constant values, as shown in Fig. 2.3c. The combined velocity $u(t)$reaches the limit at the beginning of the simulation, making the end-effector move toward the target as fast as possible, and slowing down rapidly when the end-effector approaches the target. During the whole simulation process, $u$ is guaranteed not to exceed its limit, as shown in Fig. 2.3d. Finally, as shown in Fig. 2.3a, the robot successfully reaches the fixed point under the proposed control scheme.

### 2.4.3 Circular Trajectory

In this section, tracking of a smooth circular trajectory using the control scheme is carried out. The end effector of JACO$_2$ is expected to move at an angular speed of 0.5rad/s along a circular trajectory. The desired circle is centered at $[0.3, 0.3, 0.3]^{\mathrm{T}}$ with a radius of 0.1732m, and has a revolute angle of 45° around the x-axis. The scaling coefficient is selected as $\epsilon = 0.008$ The convergence time is about 0.5s, which is similar to the regulation case. As shown in Fig. 2.4d, when the simulation begins, the robot moves at the maximum speed when the tracking error is big (Fig. 2.3a), which makes the end effector move close to the desired circle. Then the robot moves at a low speed periodically, and correspondingly, the joint angle $\theta$ changes with the same frequency (Fig. 2.4b), meanwhile, the tracking error is already close to zero (Fig. 2.4a), which means that the robot has successfully tracked the desired circular trajectory with time. According to (2.6), the reference speed vector of the end-effector $b_0$ can be derived, and its components along $x-$, $y-$, and $z-$ directions are shown as blue lines in Fig. 2.4e–g, in which red lines represent the corresponding values of $\hat{J}u$. The red lines quickly converge to blue ones, demonstrating that the proposed control strategy is able to track the given trajectory under kinematic uncertainties. The joint velocity norm $||u||_2^2$ is shown in Fig. 2.4h.

### 2.4.4 Square Trajectory

In this section, the JACO$_2$ is used to track a square trajectory. The corners of the desired square in the Cartesian space are set to be $[0.3, 0.4, 0.4]^{\mathrm{T}}$, $[0.4, 0.3, 0.4]^{\mathrm{T}}$, $[0.3, 0.2, 0.2]^{\mathrm{T}}$,and $[0.2, 0.3, 0.3]^{\mathrm{T}}$. The motion period is 12.56s. The velocity norm of desired path $||\dot{x}_{\mathrm{d}}(t)||$ keeps constant, which means that the expected velocity



(a)

(b)

(c)

(d)

(e)

(f)

(g)

(h)

**Fig. 2.5** Results when JACO$_2$ tracks a square trajectory in Cartesian space. **a** Motion trajectory of end effector (red curve) and the corresponding incremental configurations of JACO$_2$. **b** Error-time curve along three directions. **c** Angle-time curve of 6 joints. **d** Command-time curve of joint velocity $u$. **e** The first Cartesian velocity input $b_0$(x-axis direction) described by (2.6) and the corresponding output $\hat{J}u(1)$. **f** The second Cartesian velocity input $b_0$ (y-axis direction) described by (2.6) and the corresponding output $\hat{J}u(2)$. **g** The third Cartesian velocity input $b_0$(z-axis direction) described by (2.6) and the corresponding output $\hat{J}u(3)$. **h** The Euclidean norm of the manipulator's joint velocity

of end effector $\dot{x}_\mathrm{d}(t)$ remains constant between two adjacent vertices, while $\dot{x}_\mathrm{d}(t)$ changes discontinuously at the four corners. The scaling coefficient is selected as $\epsilon = 0.008$. Numerical results are shown in Fig. 2.5. In the initial stage, the tracking error approaches zero over time after a short transient state, at the same time, the joint speed remains within the set $\Omega$ at all times (Fig. 2.5d). The output of the position controller (2.6) and the resulting responses under the proposed control scheme along the $x-$, $y-$, and $z-$ directions are shown in Fig. 2.5e–g. The red lines converge to blue ones quickly both at the beginning of simulation and after discontinuous change of the desired velocity, and the joint speed also switches at these moments, as shown in Fig. 2.5d. As a result, there exist vibration on the error curve at time $t = 3.14, 6.28, 9.24, 12.56, 15.7, 18.8$ s, with the maximum value of $[4 \times 10^{-3}, 2 \times 10^{-3}, 1.5 \times 10^{-3}]^\mathrm{T}$m. The joint velocity norm $||u||_2^2$ is shown in Fig. 2.5h (Table 2.2).

**Table 2.2** Comparisons among different tracking controllers on manipulators

| Method | Exact model | JMPI | Initial | Joint | Regulation | Tracking | Real-time |
|---|---|---|---|---|---|---|---|
| | Exact model | | Position | Velocity | Error | error | Control |
| This chapter | Not required | No | Any | Limited | Zero | Zero | Yes |
| Controller in [28] | Not required | Yes | Any | Unlimited | Zero | Failed | Yes |
| Controller in [35] | Not required | Yes | Any | Unlimited | Zero | Zero | Yes |
| Controller in [39] | Required | No | Any | Limited | Zero | Zero | Yes |
| Controller in [40] | Required | No | Any | Limited | Zero | Zero | Yes |
| Controller in [61] | Required | No | Any | Unlimited | Zero | Failed | No |
| Controller in [62] | Required | No | Restrictive | Limited | Failed | Zero | Yes |

### 2.4.5 *Comparison*

In this section, we compare the proposed method with the performance of the existing redundant robot tracking controller as shown in Table 2.2. JMPI [28, 35] and RNN policy-based tracking controller [39, 40, 61, 62]. In the reference [28, 35] and our study, the exact kinematic model of the robot is not needed, which can be used to solve the kinematic uncertainty problem.The controllers proposed [28, 35, 40, 61] are calculated according to the speed level, while the controllers proposed [39, 62] are designed according to the acceleration level.In this chapter, we develop a speed-level controller. The controller obtains the control command in [28, 35] by computing the pseudo-inverse of the jacobian. These strategies cannot be used when the robot is in a singular configuration. Although DLS [22] and other improved methods are introduced, the convergence of tracking error of singular configuration cannot be guaranteed, and its physical limitations are not considered. In addition to referencing [62], the initial position of the end-effector can be set randomly in the controller, which needs to be on the desired path when referencing [62]. The controllers kin [39, 40, 62] based on RNN can guarantee the boundedness of the control command. The three controllers can track the time-varying trajectory, but the position adjustment fails in [62]. In summary, our controller can achieve stable tracking under both regulation and path tracking, and it does not need the accurate kinematic model and pseudo-inverse calculation of the Jacobian matrix, so it has good flexibility and adaptability to the uncertain environment.

## 2.5 Summary

This chapter studies the kinematic control of redundant robots with uncertain kinematics. A dynamic neural network is proposed to solve the redundancy problem by using an adaptive motion identifier to learn motion parameters online. The interaction between the adaptive online identifier and the neural controller makes it a nonlinear coupled system. The global convergence of tracking error is verified by the Lyapunov theory. Numerical experiments and comparisons based on JACO$_2$ robot arm demonstrate the effectiveness of the algorithm and its superiority over existing algorithms. This method is combined with RNN to realize static and dynamic task space tracking. The pseudo-inverse computation of the Jacobian matrix is avoided and the real-time performance of the controller is guaranteed. The boundedness of joint speed can also protect the robot and improve safety performance. Before concluding this chapter, it is worth pointing out that this is the first dynamic neural model of motion control for a manipulator with adaptive redundancy based on kinematic regression, with demonstrable convergence and guaranteed performance limits.

## Appendix

According to (2.3), the analytical expression of JACO$'_2$s kinematic regressor matrix $Y_k \in \mathbb{R}^{3\times 6}$ is given as follows.

$Y_{11} = 0,$

$Y_{12} = -\dot{\theta}_2 c_1 c_2 + \dot{\theta}_1 s_1 s_2,$

$Y_{13} = -\dot{\theta}_1 c_1,$

$Y_{14} = \dot{\theta}_2(c_1c_2c_3 + c_1s_2s_3) + \dot{\theta}_1(c_2s_1s_3 - c_3s_1s_2) - \dot{\theta}_3(c_1c_2c_3 + c_1s_2s_3),$

$Y_{15} = (\dot{\theta}_1((\sqrt{3}c_1c_4)/2 + (\sqrt{3}s_4(s_1s_2s_3 + c_2c_3s_1))/2 + (c_2s_1s_3)/2 - (c_3s_1s_2)/2) - \dot{\theta}_4((\sqrt{3}s_1s_4)/2 + (\sqrt{3}c_4(c_1c_2c_3 + c_1s_2s_3))/2) + \dot{\theta}_2((c_1c_2c_3 + c_1s_2s_3)/2) - (\sqrt{3}s_4(c_1c_2s_3 - c_1c_3s_2))/2 + (c_1s_2s_3)/2) - \dot{\theta}_3((c_1c_2c_3)/2 - (\sqrt{3}s_4(c_1c_2s_3 - c_1c_3s_2))/2),$

$Y_{16} = (\dot{\theta}_5((\sqrt{3}c_5(s_1s_4 + c_4(c_1c_2c_3 + c_1s_2s_3)))/2 + (\sqrt{3}s_5((c_4s_1)/2 - (s_4(c_1c_2c_3 + c_1s_2s_3))/2 + (\sqrt{3}(c_1c_2s_3 - c_1c_3s_2))/2))/2) - \dot{\theta}_4((\sqrt{3}s_1s_4)/4 - (\sqrt{3}c_5((s_1s_4)/2 + (c_4(c_1c_2c_3 + c_1s_2s_3))/2))/2 - (\sqrt{3}s_5(c_4s_1 - s_4(c_1c_2c_3 + c_1s_2s_3)))/2 + (\sqrt{3}c_4(c_1c_2c_3 + c_1s_2s_3))/4) + \dot{\theta}_1((\sqrt{3}c_1c_4)/4 + (\sqrt{3}s_5(c_1s_4 - c_4(s_1s_2s_3 + c_2c_3s_1)))/2 - (\sqrt{3}c_5((c_1c_4)/2 - (\sqrt{3}(c_2s_1s_3 - c_3s_1s_2))/2 + (s_4(s_1s_2s_3 + c_2c_3s_1))/2))/2 + (\sqrt{3}s_4(s_1s_2s_3 + c_2c_3s_1))/4 + (c_2s_1s_3)/4 - (c_3s_1s_2)/4) + \dot{\theta}_2(+(\sqrt{3}(c_1c_2c_3 + c_1s_2s_3))/2))/2(\sqrt{3}c_5((s_4 \cdot (c_1c_2s_3 - c_1c_3s_2))/2 - (\sqrt{3}s_4(c_1c_2s_3 - c_1c_3s_2))/4 + (c_1c_2c_3)/4 + (c_1s_2s_3)/4 + (\sqrt{3}c_4s_5(c_1c_2s_3 - c_1c_3s_2))/2) - \dot{\theta}_3((\sqrt{3}c_5((s_4(c_1c_2s_3 - c_1c_3s_2))/2 + (\sqrt{3}(c_1c_2c_3 + c_1s_2s_3))/2))/2 - (\sqrt{3}s_4(c_1c_2s_3 - c_1c_3s_2))/4 + (c_1c_2c_3)/4 + (c_1s_2s_3)/4 + (\sqrt{3}c_4s_5(c_1c_2s_3 - c_1c_3s_2))/2)),$

$Y_{21} = 0,$

$Y_{22} = -\dot{\theta}_1 c_1 s_2 - \dot{\theta}_2 c_2 s_1,$

$Y_{23} = -\dot{\theta}_1 s_1,$

$Y_{24} = \dot{\theta}_2(s_1s_2s_3 + c_2c_3s_1) - \dot{\theta}_3(s_1s_2s_3 + c_2c_3s_1) - \dot{\theta}_1(c_1c_2s_3 - c_1c_3s_2),$

$Y_{25} = \dot{\theta}_2((s_1s_2s_3)/2 + (c_2c_3s_1)/2 - (\sqrt{3}s_4(c_2s_1s_3 - c_3s_1s_2))/2) - \dot{\theta}_3((s_1s_2s_3)/2 + (c_2c_3s_1)/2 - (\sqrt{3}s_4(c_2s_1s_3 - c_3s_1s_2))/2) + \dot{\theta}_1((\sqrt{3}c_4s_1)/2 - (\sqrt{3}s_4(c_1c_2c_3 + c_1s_2s_3))/2 - (c_1c_2s_3)/2 + (c_1c_3s_2)/2) + \dot{\theta}_4((\sqrt{3}c_1s_4)/2 - (\sqrt{3}c_4(s_1s_2s_3 + c_2c_3s_1))/2),$

$Y_{26} = \dot{\theta}_1((\sqrt{3}c_4s_1)/4 - (\sqrt{3}c_5((c_4s_1)/2 - (s_4(c_1c_2c_3 + c_1s_2s_3))/2 + (\sqrt{3}(c_1c_2s_3 - c_1c_3s_2))/2))/2 + (\sqrt{3}s_5(s_1s_4 + c_4(c_1c_2c_3 + c_1s_2s_3)))/2 - (\sqrt{3}s_4(c_1c_2c_3 + c_1s_2s_3))/4 - (c_1c_2s_3)/4 + (c_1c_3s_2)/4) - \dot{\theta}_5((\sqrt{3}c_5(c_1s_4 - c_4(s_1s_2s_3 + c_2c_3s_1)))/2 + (\sqrt{3}s_5((c_1c_4)/2 - (\sqrt{3}(c_2s_1s_3 - c_3s_1s_2))/2 + (s_4(s_1s_2s_3 + c_2c_3s_1))/2))/2) + \dot{\theta}_2((s_1s_2s_3)/4 + (\sqrt{3}c_5((\sqrt{3}(s_1s_2s_3 + c_2c_3s_1))/2 + (s_4(c_2s_1s_3 - c_3s_1s_2))/2))/2 + (c_2c_3s_1)/4 - (\sqrt{3}s_4(c_2s_1s_3 - c_3s_1s_2))/4 + (\sqrt{3}c_4s_5(c_2s_1s_3 - c_3s_1s_2))/2) - \dot{\theta}_3((s_1s_2s_3)/4 + (\sqrt{3}c_5((\sqrt{3}(s_1s_2s_3 + c_2c_3s_1))/2 + (s_4(c_2s_1s_3 - c_3s_1s_2))/2))/2 + (c_2c_3s_1)/4 - (\sqrt{3}s_4(c_2s_1s_3 - c_3s_1s_2))/4 + (\sqrt{3}c_4s_5(c_2s_1s_3 - c_3s_1s_2))/2) - \dot{\theta}_4((\sqrt{3}c_5((c_1s_4)/2 - (c_4(s_1s_2s_3 + c_2c_3s_1))/2))/2 - (\sqrt{3}c_1s_4)/4 + (\sqrt{3}s_5(c_1c_4 + s_4(s_1s_2s_3 + c_2c_3s_1)))/2 + (\sqrt{3}c_4(s_1s_2s_3 + c_2c_3s_1))/4),$

$Y_{31} = 0,$

$Y_{32} = -\dot{\theta}_2s_2,$

$Y_{33} = 0,$

$Y_{34} = \dot{\theta}_3(c_2s_3 - c_3s_2) - \dot{\theta}_2(c_2s_3 - c_3s_2),$

$Y_{35} = \dot{\theta}_3((c_2s_3)/2 - (c_3s_2)/2 + (\sqrt{3}s_4(c_2c_3 + s_2s_3))/2) - \dot{\theta}_2((c_2s_3)/2 - (c_3s_2)/2 + (\sqrt{3}s_4(c_2c_3 + s_2s_3))/2) + (\sqrt{3}\dot{\theta}_4c_4(c_2s_3 - c_3s_2))/2,$

$Y_{36} = \dot{\theta}_5((\sqrt{3}s_5((\sqrt{3}(c_2c_3 + s_2s_3))/2 + (s_4(c_2s_3 - c_3s_2))/2))/2 - (\sqrt{3}c_4c_5(c_2s_3 - c_3s_2))/2) + \dot{\theta}_4((\sqrt{3}c_4(c_2s_3 - c_3s_2))/4 + (\sqrt{3}s_4s_5(c_2s_3 - c_3s_2))/2 - (\sqrt{3}c_4c_5(c_2s_3 - c_3s_2))/4) - \dot{\theta}_2((c_2s_3)/4 - (c_3s_2)/4 + (\sqrt{3}c_5((\sqrt{3}(c_2s_3 - c_3s_2))/2 - (s_4(c_2c_3 + s_2s_3))/2))/2 + (\sqrt{3}s_4(c_2c_3 + s_2s_3))/4 - (\sqrt{3}c_4s_5(c_2c_3 + s_2s_3))/2) + \dot{\theta}_3((c_2s_3)/4 - (c_3s_2)/4 + (\sqrt{3}c_5((\sqrt{3}(c_2s_3 - c_3s_2))/2 - (s_4(c_2c_3 + s_2s_3))/2))/2 + (\sqrt{3}s_4(c_2c_3 + s_2s_3))/4 - (\sqrt{3}c_4s_5(c_2c_3 + s_2s_3))/2).$

## References

1. X. Li, Z. Xu, S. Li, H. Wu, X, Zhou, Cooperative kinematic control for multiple redundant manipulators under partially known information using recurrent neural network. IEEE Access **8**(1), 40029–40038 (2020)
2. Z. Xu, S. Li, X. Zhou, Y. Wu, T. Cheng, D. Huang, Dynamic neural networks based kinematic control for redundant manipulators with model uncertainties. Neurocomputing **329**(1), 255–266 (2019)
3. Z. Xu, S. Li, X. Zhou, T. Cheng, Dynamic neural networks based adaptive admittance control for redundant manipulators with model uncertainties. Neurocomputing **357**(1), 271–281 (2019)
4. Z. Xu, S. Li, X. Zhou, W. Yan, T. Cheng, H. Dan, Dynamic neural networks for motion-force control of redundant manipulators: an optimization perspective. IEEE transactions on industrial electronics Early access (2020). https://doi.org/10.1109/TIE.2020.2970635
5. Z. Zhang, A. Beck, N. Magnenat-Thalmann, Human-Like Behavior Generation Based on Head-Arms Model for Robot Tracking External Targets and Body Parts. IEEE Transactions on Cybernetics **45**(8), 1390–1400 (2015)

6. D. Guo, Y. Zhang, "A New Inequality-Based Obstacle-Avoidance MVN Scheme and Its Application to Redundant Robot Manipulator," IEEE Transactions on Systems, Man, and Cybernetics. Part C (Applications and Reviews) **42**(6), 1326–1340 (2012)
7. H. Wu, Y. Guan, J. Rojas, A latent state-based multimodal execution monitor with anomaly detection and classification for robot introspection. Applied Sciences. **9**(6), 1072 (2019 Jan)
8. H. Wu, Z. Xu, W. Yan, Q. Su, S. Li, T. Cheng, X. Zhou, Incremental Learning Introspective Movement Primitives From Multimodal Unstructured Demonstrations. IEEE Access. **15**(7), 159022–36 (2019 Oct)
9. H. Wu, Y. Guan, J. Rojas, Analysis of multimodal Bayesian nonparametric autoregressive hidden Markov models for process monitoring in robotic contact tasks. International Journal of Advanced Robotic Systems. **16**(2), 1729881419834840 (2019 Mar 26)
10. Y. Zhang, Inverse-free computation for infinity-norm torque minimization of robot manipulators. Mechatronics **16**(3), 177–184 (2006)
11. Z. G. Hou and L. Cheng and M. Tan, Multicriteria Optimization for Coordination of Redundant Robots Using a Dual Neural Network," *IEEE Trans. Syst., Man, Cybern. B, Cybern.*, vol. 40, no. 4, pp. 1075–1087 (2010)
12. M. Shimizu, H. Kakuya, W.K. Yoon, K. Kitagaki, K. Kosuge, Analytical Inverse Kinematic Computation for 7-DOF Redundant Manipulators With Joint Limits and Its Application to Redundancy Resolution. IEEE Trans. Robot. **24**(5), 1131–1142 (2008)
13. E. Tatlicioglu, M. McIntyre, D. Dawson, I. Walker, *"Adaptive Nonlinear Tracking Control of Kinematically Redundant Robot Manipulators with Sub-Task Extensions," in Proc* (Miami, FL, USA, Dec, IEEE Int. Conf. Dec. Cont., 2008), pp. 1131–1142
14. J. Na and X. Ren and D. Zheng, "Adaptive Control for Nonlinear Pure-Feedback Systems With High-Order Sliding Mode Observer," IEEE Trans. Neur. Net., Lear., vol. 24, no. 3, pp. 370–382 (2013)
15. Z. Li, H. Su, H. Zhang, C.Y. Su, T. Chai, *"Barrier Lyapunov Based Control of dual-arm exoskeleton robots performing asymmetric bimanual tasks," in Proc* (Luoyang, China, Aug, IEEE Int. Conf. adv. Mechatron. Syst., 2014), pp. 370–382
16. W. He, Z. Yin, C. Sun, Adaptive neural network control of a marine vessel with constraints using the asymmetric barrier Lyapunov function. IEEE Trans. Cybern. **47**(7), 1641–1651 (2017)
17. X. Liang, X. Huang, M. Wang, X. Zeng, Adaptive Task-Space Tracking Control of Robots Without Task-Space- and Joint-Space-Velocity Measurements. IEEE Trans. Robot. Automat. **26**(4), 733–742 (Aug. 2010)
18. N. Kumar, J.H. Borm, V. Panwar, J. Chai, Tracking control of redundant robot manipulators using RBF neural network and an adaptive bound on disturbances. Int. J. Precis. Eng. Man. **13**(8), 1377–1386 (Aug. 2012)
19. Zhijia Zhao, Choon Ki Ahn, Han-Xiong Li. "Deadzone Compensation and Adaptive Vibration Control of Uncertain Spatial Flexible Riser Systems, *IEEE/ASME Transactions on Mechatronics*, in press, DOI:https://doi.org/10.1109/TMECH.2020.29755672020.
20. E. Castillo, A.J. Conejo, R.E. Pruneda, C. Solares, State estimation observability based on the null space of the measurement Jacobian matrix. IEEE Trans. Power Syst. **20**(3), 1656–1658 (Aug. 2005)
21. M. Galicki, Inverse-free control of a robotic manipulator in a task space. Rob. Auton. Syst. **62**(6), 131–141 (2014)
22. O. Egeland and J. R. Sagli and I. Spangelo and S. Chiaverini, " A damped least-squares solution to redundancy resolution," in *Proc. IEEE Int. Conf. Robot. Automat.*, Sacramento, Ca, USA., Apr. 1991, pp. 945-950
23. L. Jin, Y. Zhang, Discrete-time Zhang neural network of O(3) pattern for time-varying matrix pseudoinversion with application to manipulator. Neurocomputing **142**, 165–173 (2014)
24. D. Guo, Y. Zhang, Li-function activated ZNN with finite-time convergence applied to redundant-manipulator kinematic control via time varying Jacobian matrix pseudoinversion. Appl. Soft Comput. **24**, 158–168 (2014)
25. L. Cheng, Z.G. Hou, M. Tan, Adaptive neural network tracking control for manipulators with uncertain kinematics, dynamics and actuator model. Automatica **45**(10), 2312–2318 (2009)

26. S. Ma, A new formulation technique for local torque optimization of redundant manipulators. IEEE Trans. Ind. Electron. **43**(4), 462–468 (1996)
27. A. Nubiola, I. Bonev, Absolute calibration of an ABB IRB 1600 robot using a laser tracker. Robotics and Computer Integrated Manufacturing **29**(1), 236–245 (2013)
28. W.E. Dixon, Adaptive Regulation of Amplitude Limited Robot Manipulators With Uncertain Kinematics and Dynamics. IEEE Trans. Automat. Contr. **52**(3), 488–493 (2007)
29. L. Cheng, Z.G. Hou, M. Tan, *"Adaptive neural network tracking control of manipulators using quaternion feedback," in Proc* (Pasadena, CA, May, IEEE Int. Conf. Robot. Automat., 2008), pp. 3371–3376
30. C.C. Cheah, S.P. Hou, Y. Zhao, J.J.E. Slotine, Adaptive Vision and Force Tracking Control for Robots With Constraint Uncertainty. IEEE/ASME Transactions on Mechatronics **15**(3), 389–399 (2010)
31. C.C. Cheah, C. Liu, J.J.E. Slotine, *"Adaptive Vision based Tracking Control of Robots with Uncertainty in Depth Information," in Proc* (Roma, Italy, Apr, IEEE Int. Conf. Robot. Automat., 2007), pp. 2817–2822
32. J. Ren, B. Wang, M. Cai and J. Yu, "Adaptive Fast Finite-Time Consensus for Second-Order Uncertain Nonlinear Multi-Agent Systems With Unknown Dead-Zone," IEEE Access, vol. 8, No. 1, pp. 25557-25569, 2020
33. D. Chen, S. Li, Q. Wu, X. Luo, Super-twisting ZNN for coordinated motion control of multiple robot manipulators with external disturbances suppression. Neurocomputing **371**(1), 78–90 (2020)
34. Dechao Chen, Shuai Li, "A recurrent neural network applied to optimal motion control of mobile robots with physical constraints," Applied Soft Computing, to be published, 2019, doi: https://doi.org/10.1016/j.asoc.2019.105880.
35. D. Chen and Y. Zhang and S. Li, "Tracking Control of Robot Manipulators with Unknown Models: A Jacobian-Matrix-Adaption Method" *IEEE Trans Ind. Informat.*, early paper, doi: https://doi.org/10.1109/TII.2017.2766455.
36. H. Wang, Y. Xie, Adaptive inverse dynamics control of robots with uncertain kinematics and dynamics. Automatica **46**(7), 2114–2119 (2009)
37. Z. Xu, X. Zhou, T. Cheng, K. Sun, D. Huang, *"Adaptive task-space tracking for robot manipulators with uncertain kinematics and dynamics and without using acceleration," in Proc* (Macau, China, Dec, IEEE Int. Conf. on Robotics and Biomimetics., 2017), pp. 669–674
38. S. Zhang, A.G. Constantinides, Lagrange programming neural networks. IEEE Trans. Circuits Syst. II. **39**(7), 441–452 (1992)
39. S. Li, Y. Zhang, L. Jin, Kinematic Control of Redundant Manipulators Using Neural Networks. IEEE Trans. Neur. Net. Lear. **28**(10), 2243–2254 (2017)
40. Y. Zhang and S. Li and J. Gui and X. Luo, "Velocity-Level Control With Compliance to Acceleration-Level Constraints: A Novel Scheme for Manipulator Redundancy Resolution," *IEEE Trans Ind. Informat.*, vol. 14, no. 3, pp. 921-930, March. 2018
41. S. Li, S. Chen, B. Liu, Y. Li, Y. Liang, Decentralized kinematic control of a class of collaborative redundant manipulators via recurrent neural networks. Neurocomputing. **91**(1), 1–10 (2012)
42. R.F. Stengel, *Optimal Control and Estimation, Optimal Control and Estimation* (Dover, New York, NY, USA, 1994)
43. C.C. Cheah, C. Liu, J.J.E. Slotine, Adaptive tracking control for robots with unknown kinematic and Dynamic Properties. Int. J. Robt. Res. **25**(3), 283–296 (2006)
44. Y. Zhang, Z. Zhang, *Repetitive Motion Planning and Control of Redundant Robot Manipulators* (Springer-Verlag, New York, 2013)
45. S. Boyd and L. Vandenberghe, "Convex Optimization", Cambridge, U.K: Cambridge Univ. Press, 2004.
46. S. Li, Y. Zhang, L. Jin, Kinematic Control of Redundant Manipulators Using Neural Networks. IEEE Transactions on Neural Networks and Learning Systems **28**(10), 2243–2254 (2017)
47. Y. Zhang, S. Li, J. Gui, X. Luo, Velocity-Level Control With Compliance to Acceleration-Level Constraints: A Novel Scheme for Manipulator Redundancy Resolution. IEEE Transactions on Industrial Informatics **14**(3), 921–930 (2018)

48. D. Chen, Y. Zhang, S. Li, Tracking control of robot manipulators with wnknown models: a Jacobian-matrix-adaption method. IEEE Transactions on Industrial Informatics **14**(7), 3044–3053 (2018)
49. D. Kinderlehrer, G. Stampcchia, *An Introduction to Variational Inequalities and Their Applications* (Academic, New York, USA, 1980)
50. H. Khalil, *Nonlinear Systems* (Prentice Hall, New Jersey, USA, 1996)
51. D. Guo, Y. Zhang, Acceleration-level inequality-based MAN scheme for obstacle avoidance of redundant robot manipulators. IEEE Transactions on Industrial Electronics **61**(12), 6903–6914 (2014)
52. J. Slotine, W. Li, *Applied Nonlinear Control* (China Machine Press, Beijing, China, 2004)
53. Y. Zhang, L. Xiao, Z. Xiao, and M. Mao, *Zeroing Dynamics, Gradient Dynamics, and Newton Iterations.* Boca Raton, FL, USA.: Springer-Verlag, 2015
54. Y. Zhang, D. Guo, *Zhang Functions and Various Models* (CRC Press, Berlin, Germany, 2015)
55. S. Boyd, L. Vandenberghe, *Convex Optimization* (Cambridge Univ. Press, Cambridge, U.K., 2004)
56. D. Kinderlehrer and G. Stampcchia, *An Introduction to Variational Inequalities and Their Applications.* New York, USA.: Academic, 1980
57. J. Dattorro, *Convex Optimization And Euclidean Distance Geometry* (Meboo Publishing, California, USA, 2016)
58. H. Khalil, *Nonlinear systems.*New Jersey, USA.: Prentice Hall, 1996
59. D. Guo, Y. Zhang, Acceleration-Level Inequality-Based MAN Scheme for Obstacle Avoidance of Redundant Robot Manipulators. IEEE Trans. Ind. Electron. **61**(12), 6903–6914 (2014)
60. J.J.E. Slotine, W. Li, *Applied nonlinear control* (China Machine Press, Beijing, China, 2004)
61. B. Aghbali, A. Yousefi-Koma, A.G. Toudeshki, A. Shahrokhshahi, *“ZMP trajectory control of a humanoid robot using different controllers based on an off line trajectory generation,” in Proc* (Tehran, Iran, Feb, IEEE Int. Conf. Rob. Mechatron., 2013), pp. 530–534
62. Y. Xia, G. Feng, and J. Wang, “A primal-dual neural network for online resolving constrained kinematic redundancy in robot motion control,” *IEEE Trans. Syst., Man, Cybern. B, Cybern.*, vol. 35, no. 1, pp. 54-64, Feb. 2005
63. Y.S. Xia, G. Feng, J. Wang, “A Primal-Dual Neural Network for Online Resolving Constrained Kinematic Redundancy in Robot Motion Control”, IEEE Transactions on Systems, Man and Cybernetics. Part B (Cybernetics) **35**(1), 54–64 (2005)

# Chapter 3
# RNN Based Adaptive Compliance Control for Robots with Model Uncertainties

**Abstract** Position-force control is challenging for redundant manipulators, especially for the ones considering both joint physical limitations and model uncertainties. In this chapter, we considered adaptive motion-force control of redundant manipulators with uncertainties of the interaction model and physical parameters. The whole control problem is formulated as a QP equation with a set of equality and inequality constraints, where based on admittance control strategy, the desired motion-force task is combined with the kinematic property of redundant manipulators, corresponding to an equality constraint in the formed QP equation. Moreover, the uncertainties of both system model and physical parameters are also considered, together with the complicated joint physical structure constraints, formulating as a set of inequality constraints. Then an adaptive recurrent neural network is designed to solve the QP problem online. This control scheme generalizes recurrent neural network based kinematic control of manipulators to that of position-force control, which opens a new avenue to shift position-force control of manipulators from pure control perspective to cross design with both convergence and optimality consideration. Numerical results on a 7-DOF manipulator LBR iiwa and comparisons with existing methods show the validity of the proposed control method.

## 3.1 Introduction

A manipulator is called redundant if its DOFs are greater than those required to complete a task. The redundant DOFs enable the robot to maintain the position and direction of the end actuator to complete a given task and adjust its joint configuration to complete a secondary task. Take advantage of this feature, typical manipulator systems such as collaborative robots, space robotic arms, dexterous hands [1, 2] are all designed as redundant ones.

In Chaps. 1 and 2, we mainly focus on kinematic problems, in which we assume the end-effector of a manipulator could move freely in cartesian space. In fact, in industrial applications, the interaction between robot and external environment must be considered, for example, in tasks such as grinding, human-robot interaction, not

X. Zhou et al., *AI based Robot Safe Learning and Control*,
https://doi.org/10.1007/978-981-15-5503-9_3

only the high-precision motion control to a given trajectory should be guaranteed, but also the contact force exerted by the external environment should be guaranteed.

There are several approaches to achieve force control for robot manipulators. By introducing series elastic actuators as flexible units, force control can be realized by adjusting the compliance of joint angles. In [3], in order to overcome the discontinuous friction and complexity problem of traditional back-stepping based methods, a modified command-filter is introduced, and then an adaptive back-stepping controller is designed. Experimental results show the effectiveness of the proposed method. Other control schemes realize force in cartesian space. The most widely used method is called impedance control [4], where the robot and environment are regarded as impedance and admittance, respectively. The interaction model (which is also called impedance model) can be a spring-mass-damper system, spring-damper system, etc. Besides, a series of hybrid position-force controllers are designed in [5, 6], which consist of two independent loops, namely position loop, and force loop. By designing control schemes separately, the final control efforts can be formulated as the sum of the output of the independent loops. Similar research can be found in [7–9].

In industrial applications, the accurate value of the impedance model can hardly be obtained. for example, the stiffness parameter may be sensitive to environmental factors such as temperature, humidity, *etc.* Besides, the uncertainties in physical parameters would also affect control performance. In order to due with the uncertainties in the interaction model, in [10], an adaptive impedance controller is designed, in which a neural network is used to learn the nonlinear dynamics of the interaction part. In [11], by considering the influence of unknown dynamics of the external environment, and a radial basis function based controller is proposed, in which an objective function is used to regulate the torque and an adaptive admittance technique is used to minimize path tracking errors. In [12], a human-like-learning based controller is designed for interaction with environmental uncertainties. It is proved that the controller is capable of handling unstable situations such as tool switching, and can finally achieve an expected stability margin. Besides, contact force sensors are not required. Using the approximation ability of artificial neural networks, some intelligent controllers are reported in [13–17]. As to physical uncertainties, in [18], a fixed point controller is proposed based on robust adaptive control theory, the controller also ensures the bounded-ness control torque. Cheng et al. propose a unit quaternion based controller based on neural networks [19], which shows good performance in eliminating singularities, and semi-global stability is proved by theoretical results. In [20], a Jacobian adaptation method based on zeroing dynamics is proposed, in which the Jacobian matrix is updated according to the information of desired and actual accelerations. Other feasible adaptive strategies are reported in [21–24], in which the Jacobian is estimated by updating physical parameters online. As to physical constraints, in [42], an adaptive neural network control scheme is designed for systems with non-symmetric input dead-zone, as well as output constraints and model uncertainties. The output constraints are guaranteed by the barrier Lyapunov function. In [43], a boundary adaptive robust controller is established for flexible rise systems, in which an auxiliary system is introduced to suppress vibrational offset, and an estimator is constructed to observe to upper-bound of disturbances. The controller achieves the global convergence of control errors. Although the above-mentioned controllers could handle uncertainties in the interaction model or physical parameters, few studies have considered both uncertainties at the same time. More importantly, those controllers rarely consider the secondary task, let alone the redundancy resolution problem. Besides, the boundary of joint states is ignored, which is essential in protecting the robot.

In order to accomplish the secondary task in the reliable physical range, a kinematic control method of redundant manipulator based on QP is proposed [25–28]. The objective function is based on the secondary task, and the constraints describe the basic properties and physical constraints of the system [29]. Because of the high efficiency of parallel computing, the recurrent neural network is often used to solve the redundancy decomposition problem based on QP. In recent years, research shows that RNN based controller has good performance in motion control of redundant manipulator [30]. In [31], in order to achieve task space tracking, the joint velocity command is designed to ensure the boundary of joint angle, velocity, and acceleration. In the paper [32], by maximizing the indirectness of its time derivative, an operational optimization scheme is proposed. Numerical experiments show that the average increase in this method is 40%. In [33], different levels of redundancy resolution are discussed. Recently, RNN based methods have been extended to control examples of flexible robots, multi-robot systems, and methods such as [34–40]. However, as far as we know, there is no existing dynamic neural network (including RNN and DNN) protocol for the force control of redundant manipulators. It is necessary to consider not only the trajectory tracking problem of free-motion direction, but also the precise control of the contact force, especially for the system with model uncertainty. In addition, from the literature review, one of the research directions of a dynamic neural network is to extend the protocol of redundant manipulator of motion control task to those aspects that need precise control of tracking ability and contact force.

Based on the above observation results, we propose the first RNN based redundant manipulator position force controller, which considers the uncertainty of the interaction model and physical parameters. In this paper, the ideal case of the known model parameters is discussed, and then an adaptive admittance control scheme based on RNN is established. It ensures the boundary of joint angle and velocity. The effectiveness of the proposed controller is verified by the theoretical derivation and numerical results of LBR iiwa. Before concluding this chapter, the main contributions compared to the existing work are as follows

- As far as we know, this is the first time to focus on the motion-force control of redundant manipulators with model uncertainties based on the framework of RNNs, comparing to researches on kinematic control, the research is an important extension in the theoretical framework of dynamic neural networks.
- Different from traditional methods, an optimization-based controller is proposed, while ensuring the stability of the system, physical limitations are also guaranteed. Which is very helpful to improve the security of the system.

- The controller proposed in this chapter is able to learn uncertain model parameters online, which has better robustness and adaptability to uncertain working conditions.
- The proposed method is pseudo-inversion-free, which can save computing costs effectively.

## 3.2 Preliminaries

### 3.2.1 Problem Formulation

When a robot is controlled to perform a given operational task, the forward-kinematics of a serial manipulator is formulated as

$$x(t) = f(\theta(t)), \tag{3.1}$$

with $\theta \in \mathbb{R}^n$ being the generalize variable of the robot, and $x(t) \in \mathbb{R}^m$ being the description of end-effector′s coordinate in task space. Without loss of generality, in this chapter, we assume that all joint are rotational joints. Therefore, $\theta$ represents the vector of joint angles. In the velocity level, the Jacobian matrix $J(\theta, a_k) = \partial f(\theta(t), a_k)/\partial \theta(t) \in \mathbb{R}^{m \times n}$ is used to describe the relationship between $\dot{x}(t)$ and $\dot{\theta}$ as

$$\dot{x}(t) = J(\theta(t), a_k)\dot{\theta}(t), \tag{3.2}$$

where $a_k \in \mathbb{R}^l$ is a vector of physical parameters. In terms with (3.2), an important property which will be used in the controller design is given as below

$$J(\theta(t), a_k)\dot{\theta}(t) = Y(\theta, \dot{\theta})a_k, \tag{3.3}$$

with $Y(\theta, \dot{\theta}) \in \mathbb{R}^{m \times l}$ being the kinematic regressor matrix.

The physical parameters are very essential in describing the robot's kinematic model, for example, as the most common physical parameters, the length of robot links affects the DH parameters directly, which are fundamental in the controller design. In this chapter, the physical parameters refer to the length of robot links.

Figure 3.1 shows the interaction between the robot and environment, the contact force between the robot and workpiece is required to be precisely controlled. When the fixed contact surface is known, according to the idea of admittance control, the interaction model can be described as a spring-damping system as

$$F = K_p(x - x_\mathrm{d}) + K_d(\dot{x} - \dot{x}_\mathrm{d}), \tag{3.4}$$

where $K_p \in \mathbb{R}^{3\times 3}$ and $K_d \in \mathbb{R}^{3\times 3}$ are the corresponding damping and stiffness coefficients, $x_\mathrm{d}$ is the desired trajectory. If $K_p$ and $K_d$ are known, the desired contact

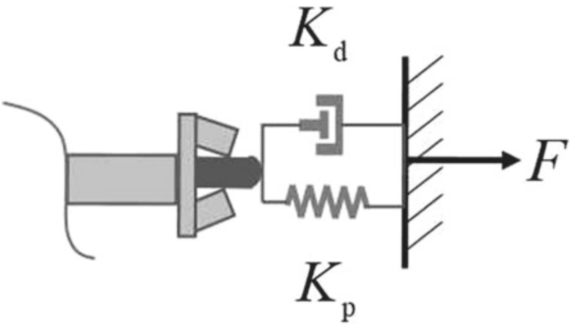


**Fig. 3.1** Spring-damper model of interaction

force $F_\mathrm{d}$ can be obtained by designing the reference velocity of the end-effector $\dot{x}_r$ based on $\dot{x}_\mathrm{d}$, $x_\mathrm{d}$, $x$ and $F_\mathrm{d}$ according to Eq. (3.4).

$$\dot{x}_r = K_d^{-1}F - K_d^{-1}K_p(x_\mathrm{d} - x) + \dot{x}_\mathrm{d}. \tag{3.5}$$

**Remark 3.1** In this chapter, we only consider the contact force on the vertical direction of the contact surface, and friction force is ignored, therefore $F$ is aligned with the normal direction of the surface. When the surface is priorly known, by defining a rotational matrix $R$ between the tool coordinate system and based coordinate system, $K_p$ and $K_d$ can be formulated as $K_p = diag(0, 0, k_p)R$, $K_p = diag(0, 0, k_d)R$, respectively. Then $K_p$ and $K_d$ can be described as single parameters.

In practical applications, the real values of system parameters such as $a_k$, $K_p$ and $K_d$ are usually unavailable. In terms of $a_k$, due to machining and installation error, the length of robot′s links may be different from the nominal value, and the robot may hold uncertain tools, which would lead to uncertainties in $a_k$. As to $K_p$ and $K_d$, the real values are even more difficult to obtain. $K_d$ and $K_p$ are related to the material and structure of the workpiece, furthermore, those parameters would change in different environmental conditions. Therefore, it is a challenging issue to achieve precise force control in the presence of parameter uncertainties.

For a redundant manipulator, the redundant DOFs can enhance the flexibility of the robot, and this property can be used to achieve a secondary task. In industrial applications, by minimizing the norm of joint speed, the kinematic energy can be optimized. Therefore, in this chapter, the objective function is selected as

$$H(\theta) = \dot{\theta}^\mathrm{T}\dot{\theta}. \tag{3.6}$$

In order to save energy consumption in the control process, a smaller value of $H(\theta)$ is preferred.

**Remark 3.2** The objective function $H(\dot{\theta})$ is a typical function to describe the secondary task in redundant resolution problems, as reported in [21, 25]. In actual implementations, this function can be defined according to the designer’s preferences or actual requirements. In this chapter, we propose a generalized RNN based force control strategy with simultaneous optimization ability. Based on the proposed control strategy, similar controllers can be easily designed by defining different objective functions.

### *3.2.2 Control Objective and QP Problem Formulation*

Before pointing out the control objective, it is noteworthy that the robot must satisfy certain constraints. For example, due to the physical structure, every joint angle $\theta_i$ must not exceed its limitations i.e., the lower bound $\theta_i^-$ and upper bound $\theta_i^+$. Furthermore, limited by actual performance of actuators, joint speed $\dot{\theta}$ is also restricted, i.e., $\dot{\theta}^- \leq \dot{\theta} \leq \dot{\theta}^+$.

When the actual parameters of interaction model are unknown, the control objective of this chapter is to design a force controller with adaptation ability, *i.e.,* to realize accurate force control along the predefined contact surface, in the sense that $F \to F_\mathrm{d}$, at the same time, physical constraints of joint angles and velocities must be ensured. According to (3.2), (3.5) and (3.6), the control objective can be described in the view of optimization as

$$\min \; H(\theta) = \dot{\theta}^\mathrm{T}\dot{\theta}, \tag{3.7a}$$

$$\text{s.t.} \; \dot{x}_r = J(\theta, a_k)\dot{\theta}, \tag{3.7b}$$

$$\dot{x}_r = K_d^{-1}F_d - K_d^{-1}K_p(x - x_\mathrm{d}) + \dot{x}_\mathrm{d}, \tag{3.7c}$$

$$\theta^- \leq \theta \leq \theta^+, \tag{3.7d}$$

$$\dot{\theta}^- \leq \dot{\theta} \leq \dot{\theta}^+. \tag{3.7e}$$

**Remark 3.3** So far, we have arrived at a generalized description of admittance control for redundant manipulators in the QP problem. Apparently, there exist parameter uncertainties in $J(\theta, a_k)$, $k_p$ and $k_d$ as formulated in (3.7b) and (3.7c). In the next chapter, we will solve the problem (3.7) with the aid of RNNs.

## 3.3 Main Results

In this chapter, an recurrent neural network based adaptive admittance controller is proposed to solve (3.7). An ideal situation where real values of system model are perfectly known is firstly considered, which lays the foundation of the later discussion. Then an adaptive RNN is proposed to achieve force control in the presence of model uncertainties. We also show the stability of the control method.

### 3.3.1 Nominal Design

In order to explain the proposed adaptive control scheme more clearly, an ideal case in which all parameters are perfectly known is firstly discussed. It can be regarded as a special case of uncertain parameter ones. In this case, both $K_d$ and $K_p$ are available, then the real value of $\dot{x}_r$ is available according to (3.5).

Let $\omega = \dot{\theta}$ and define a Lagrange function as $L_1 = \omega^{\mathrm{T}}\omega + \lambda^{\mathrm{T}}(J\omega - F_d K_d^{-1} + K_p K_d^{-1}(x - x_{\mathrm{d}}) - \dot{x}_{\mathrm{d}})$, with $\lambda$ being the Lagrange multiplier. Similar to [25], a RNN with provable convergence can be designed as

$$\varepsilon\dot{\omega} = -\omega + P_{\Omega}(-J^{\mathrm{T}}\lambda), \tag{3.8a}$$

$$\varepsilon\dot{\lambda} = -K_d^{-1}F_d + K_d^{-1}K_p(x - x_{\mathrm{d}}) - \dot{x}_{\mathrm{d}}, \tag{3.8b}$$

where $\varepsilon$ is a positive constant and $P_{\omega}(\bullet)$ is a projection operator to set $\Omega$ as $P_{\omega}(x) = \mathrm{argmin}_{y\in\Omega}||y - x||$, and the set $\Omega = \{\omega \in \mathbb{R}^n|\omega^i_{\mathrm{min}} \leq \omega_i \leq \omega^i_{\mathrm{max}}\}$ is a convex set describing the modified speed constraints based on escape velocity method [29], and $\omega_{\mathrm{min}} = \max\{\alpha(\theta_{\mathrm{min}} - \theta), \dot{\theta}_{\mathrm{min}}\}$, $u_{\mathrm{max}} = \min\{\alpha(\theta_{\mathrm{max}} - \theta), \dot{\theta}_{\mathrm{max}}\}$, $\alpha > 0$. The stability of system can be readily proved, which is similar in [25], is omitted here.

### 3.3.2 Adaptive Control Method Based on RNN

Based on the previous description, in this subchapter, by learning the uncertain parameters online, an adaptive RNN is established to solve the force control problem with gravity torque optimization under model uncertainties, the stability of the system is also proved.

#### 3.3.2.1 Adaptive RNN Design

In order to handle the uncertain interaction parameters $K_p$ and $K_d$, let $\hat{K}_p$ and $\hat{K}_d$ be their estimates. Although $K_p$ and $K_d$ are unknown, they are considered to be constant. Then the estimated reference velocity $\hat{\dot{x}}_r$ can be derived by replacing $K_p$, $K_d$ with $\hat{K}_p$ and $\hat{K}_d$ respectively according to (3.5)

$$\hat{\dot{x}}_r = \hat{K}_d^{-1}F_d - \hat{K}_d^{-1}\hat{K}_p(x - x_{\mathrm{d}}) + \dot{x}_{\mathrm{d}}. \tag{3.9}$$

Let $\eta = [x - x_{\mathrm{d}}, \hat{\dot{x}}_r - \dot{x}_{\mathrm{d}}]^{\mathrm{T}}$, $W = [K_p, K_d]^{\mathrm{T}}$ and $\hat{W} = [\hat{K}_p, \hat{K}_d]^{\mathrm{T}}$. Then we can rewrite (3.9) as $F_{\mathrm{d}} = \hat{W}^{\mathrm{T}}\eta$. However, due to the uncertainties in $K_d$ and $K_p$, in the actual process, the resulting contact force $F$ using $\hat{\dot{x}}_r$ directly is $F = W^{\mathrm{T}}\eta$, it is noteworthy that the contact force $F$ can be measured by force/torque sensors.

As to the uncertain $a_k$, the alternative Jacobian matrix $J(\theta, \hat{a}_k)$ is used by substituting $a_k$ with its estimate $\hat{a}_k$, then $J(\theta, a_k)$ in equality constraint (3.7b) is replaced by $J(\theta, \hat{a}_k)$. Therefore, the force control problem with joint speed optimization considering model uncertainties can be formulated as

$$\min \quad H(\omega) = \omega^{\mathrm{T}}\omega, \tag{3.10a}$$

$$\text{s.t.} \quad J(\theta, \hat{a}_k)\omega = \hat{K}_d^{-1}F_d - \hat{K}_d^{-1}\hat{K}_p(x - x_{\mathrm{d}}) + \dot{x}_{\mathrm{d}}, \tag{3.10b}$$

$$\theta^{-} \leq \theta \leq \theta^{+}, \tag{3.10c}$$

$$\omega^{-} \leq \omega \leq \omega^{+}. \tag{3.10d}$$

To solve (3.10), by defining a Lagrange function as $L = \omega^{\mathrm{T}}\omega + \lambda(J(\theta, \hat{a}_k)\omega - \dot{x}_r)$, the adaptive RNN is designed as

$$\varepsilon\dot{\omega} = -\omega + P_{\Omega}(-\hat{J}^{\mathrm{T}}\lambda), \tag{3.11a}$$

$$\varepsilon\dot{\lambda} = J(\theta, \hat{a}_k)\omega - \dot{\hat{x}}_r, \tag{3.11b}$$

$$\dot{\hat{W}} = -\Gamma_1\eta(F_{\mathrm{d}} - F)^{\mathrm{T}}, \tag{3.11c}$$

$$\dot{\hat{a}}_k = -\Gamma_2 Y^{\mathrm{T}}(J(\theta, \hat{a}_k)\omega - \dot{x}), \tag{3.11d}$$

where $\varepsilon$, $\Gamma_1$ and $\Gamma_2$ are positive gains. Figure 3.2 shows the framework of the proposed adaptive RNN in real-time force control with uncertain parameters. In order to learn the uncertain parameters, the neurons $\hat{W}$ and $\hat{a}_k$ update their values based on desired signals $x_{\mathrm{d}}$, $\dot{x}_{\mathrm{d}}$ and $F_{\mathrm{d}}$ and the feedback of $x$, $\dot{x}$ and $F$. The output of the RNN is exactly the joint speed command $\omega$. By designing proper updating laws, $\lambda$ and $\omega$ achieve both stability of the inner loop and the optimization of $H(\omega)$.

**Remark 3.4** In this chapter, we consider the case where $m = 6, n = 7$ (where $m$ is the dimension of the cartesian space, and $n$ is the number of joint angles). Since only the contact force on the vertical direction of the surface is considered, the dimension of $K_d$ and $K_P$ are all 1(the contact surface if known). As illustrated in Fig. 3.2, the proposed adaptive RNN has a typical one-layer architecture, and the total number of neurons is $n + l + m + 2$.

**Remark 3.5** The proposed adaptive RNN (3.11) can be regarded as a generalized form of the nominal RNN (3.8), when $\hat{W} = W$ and $\hat{a}_k = 0$, it can be obtained that $\dot{\hat{W}} = 0$ and $\dot{\hat{a}}_k = 0$ from (3.3) and (3.9). Then (3.11) is the same as (3.8). However, it is remarkable that adaptive RNN is capable of dealing with model uncertainties. On the other hand, unlike the adaptive RNNs based kinematic control strategies in [21, 22], the proposed controller can achieve both precise control of both position and contact force.

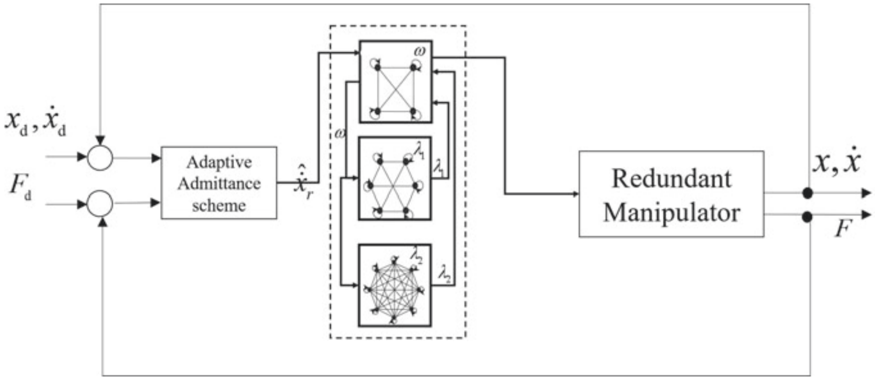


**Fig. 3.2** A schematic framework of the proposed RNN based force controller

#### 3.3.2.2 Stability Analysis

So far, a theorem about the convergence of the force control problem using proposed adaptive RNN in presence of model uncertainties can be summarised as below

**Theorem 3.1** Consider the force control problem for a category of redundant manipulators described in (3.1)–(3.4) with model uncertainties, the state variable $\omega$ of the proposed adaptive RNN will converge the optimal solution of (3.7), i.e., the force control error will converge to 0, and the norm of joint speed will be optimized simultaneously.

***Proof*** The proof consists of three steps. Firstly, we will prove that $\hat{W}$ and $\hat{a}_k$ could learn the model parameters online, and then the stability in the inner-loop is also analyzed.

Step 1. Define the estimate error of concatenated form of $W$ as $\tilde{W} = \hat{W} - W$, and $e_f = F - F_d$ be the error between the contact force and the desired signal. From (3.9), $e_f$ can be formulated as $e_f = \hat{W}^{\mathrm{T}}\eta - W^{\mathrm{T}}\eta = \tilde{W}^{\mathrm{T}}\eta$. Consider the Lyapunov function as $V_1 = tr(\tilde{W}^{\mathrm{T}}\tilde{W})/2$, which $tr(\bullet)$ is the trace of a matrix. Calculating the time derivative of $V_1$ yields

$$\begin{aligned}\dot{V}_1 &= tr(\tilde{W}^{\mathrm{T}}\dot{\tilde{W}}) = tr(-\Gamma_1\tilde{W}^{\mathrm{T}}\eta(F_{\mathrm{d}} - F)^{\mathrm{T}}) \\ &= tr(-\Gamma_1 e_f e_f^{\mathrm{T}}) = -\Gamma_1 e_f^{\mathrm{T}} e_f \leq 0. \end{aligned} \tag{3.12}$$

From (3.12) and (3.10) and using LaSalle′s invariance principle [41], we have $e_f^{\mathrm{T}}e_f = 0$, as $t \to \infty$. In other words, the state variable $\hat{W}$ ensures the convergence of force error $e_f$ by modifying the end-effector′s reference speed $\hat{\dot{x}}_r$ according to (3.5).

Step 2. Define the estimate error of $a_k$ as $\tilde{a}_k = \hat{a}_k - a_k$, and let $V_2 = \tilde{a}_k^{\mathrm{T}}\tilde{a}_k/2$. It is notable that during the control process, $a_k$ can be regarded as constant, then we have $\dot{\tilde{a}}_k = \dot{\hat{a}}_k$. Actually, using the property described in Eq. (3.3), based on linearization

**Algorithm 3** The proposed adaptive RNN based force controller

```
Input: Nominal values of interaction model $K_p^n$, $K_d^n$ and physical parameters $a_k^n$. Physical range of joint angles and joint velocities $\theta_{\max}^i$, $\theta_{\min}^i$, $\dot{\theta}_{\max}^i$, $\dot{\theta}_{\min}^i$. The desired trajectory $x_{\mathrm{d}}$, $\dot{x}_{\mathrm{d}}$ and the desired contact force $F_{\mathrm{d}}$. Positive control gains $\alpha$, $\Gamma_1$, $\Gamma_2$, $\varepsilon$. Sensor readings of contact force $F$ and movement of the end-effector $x$, $\dot{x}_d$. Task duration $T$.
Output: To achieve position-force control in presence of model uncetainties
Initialize $\lambda(0)$, $\hat{a}_k(0) \leftarrow a_k$, $\hat{W}(0) \leftarrow [K_p^n; K_p^n]$
Repeat
    $x, \dot{x}, \theta, \dot{\theta}, F \leftarrow$ Sensor readings
    Obtain the Jacobian matrix $J(\theta, \hat{a}_k)$ and kinematic regressor matrix $Y(\theta, \dot{\theta})$
    Calculate the modified reference trajectory $\hat{\dot{x}}_r$ by Equation (3.9)
    Update interaction parameters $\hat{W}$ by Equation (3.11c)
    Update physical parameters $\hat{a}_k$ by Equation (3.11d)
    Update state variable $\lambda$ by Equation (3.11b)
    Update joint velocity command $\omega$ by Equation (3.11a)
Until $(t > T)$
```

descriptions of $\dot{\theta}$ and $a_k$, respectively, the task-space velocity $\dot{x}$ has two equivalent descriptions, namely $J(\theta, a_k)\dot{\theta}$ and $Y(\theta, \dot{\theta})a_k$. As a result, the estimated value $\hat{\dot{x}}$ also has two similar descriptions, depending on the estimated value of kinematic parameter $\hat{a}_k$. Therefore, the updating law Eq. (3.11) is equivalent to $-\Gamma_2 Y^{\mathrm{T}}(Y(\theta, \omega)\hat{a}_k - \dot{x})$. Then it follows from (3.2) and (3.3) that

$$\begin{aligned}\dot{\hat{a}}_k &= -\Gamma_2 Y^{\mathrm{T}}(J(\theta, \hat{a}_k)\omega - J(\theta, a_k)\omega) \\ &= -\Gamma_2 Y(\theta, \omega)^{\mathrm{T}} Y(\theta, \omega)\tilde{a}_k.\end{aligned} \tag{3.13}$$

In light of (3.13), $\dot{V}_2$ can be rewritten as

$$\begin{aligned}\dot{V}_2 &= \tilde{a}_k^{\mathrm{T}}\dot{\hat{a}}_k \\ &= -\Gamma_2 \tilde{a}_k^{\mathrm{T}} Y(\theta, \omega)^{\mathrm{T}} Y(\theta, \omega)\tilde{a}_k \le 0.\end{aligned} \tag{3.14}$$

Then it can be readily obtained that $Y(\theta, \omega)\tilde{a}_k \to 0$ as $t \to \infty$. From (3.3) and and definition of $\tilde{a}_k$, $J(\theta, \hat{a}_k)\omega$ will eventually converge to $J(\theta, a_k)\omega$, i.e., the equality constraint (3.10b) will eventually be equivalent to (3.7b).

*Step 3.* Then we will prove the stability of inner-loop system. According to (3.11), the dynamics of $\omega$ and $\lambda$ can be reformulated as

$$\varepsilon\dot{\xi} = -\xi + P_{\bar{\Omega}}[\xi - F(\xi)], \tag{3.15}$$

with $\xi = [\omega^{\mathrm{T}}, \lambda^{\mathrm{T}}]^{\mathrm{T}}$, $\bar{\Omega} = \{(\omega, \lambda) | \omega \in \Omega, \lambda \in \mathbb{R}^m\}$, and

$$F = \begin{bmatrix} \omega + J^{\mathrm{T}}\lambda \\ -\hat{J}\omega - F_d\hat{K}_d^{-1} + \hat{K}_p\hat{K}_d^{-1}(x - x_{\mathrm{d}}) - \dot{x}_{\mathrm{d}} \end{bmatrix}.$$

Define $\nabla F = \partial F(\xi)/\partial\xi$, we have

$$\nabla R = \begin{bmatrix} I & -\hat{J}^{\mathrm{T}} \\ \hat{J} & 0 \end{bmatrix},$$

with $I$ being the identity matrix. Then it can be readily obtained that $\nabla F + (\nabla F)^{\mathrm{T}}$ is positive semi-definite. According to the definition in [32], $F$ is a monotone function of $\xi$. From the description of (3.11) and (3.15), $P_{\bar{\Omega}}$ can be formulated as $P_{\bar{\Omega}} = [P_\Omega; P_R]$, where $P_R \in \mathbb{R}^m$ is a projection operator of $\lambda$ to set $R$, with the upper and lower bounds being $\pm\infty$. Therefore, $P_{\bar{\Omega}}$ is a projection operator to closed set $\bar{\Omega}$. Based on Lemma 1 in [32], the adaptive RNN (3.11) is stable, and the will be ultimately equivalent to the solution of (3.7). This completes the Proof. □

**Remark 3.6** Till now, we have shown the stability of the proposed RNN based adaptive admittance control strategy in the presence of uncertain model parameters. The established adaptive RNN is capable of maintaining the boundedness of system states and avoiding calculating the pseudo-inversion of the Jacobian matrix.

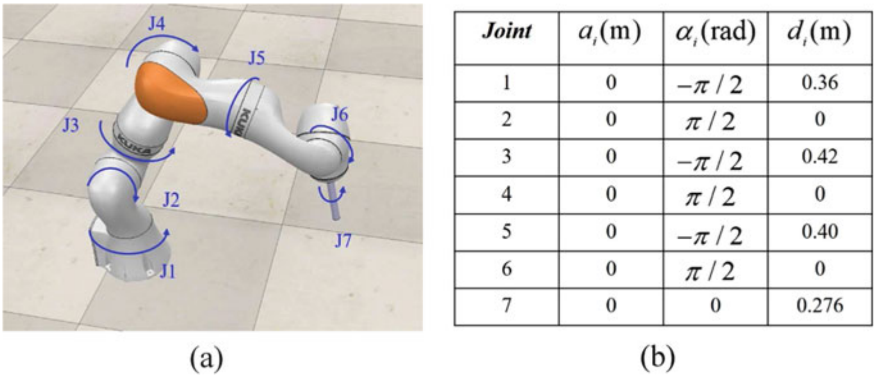
## 3.4 Illustrative Examples

In this chapter, numerical results on a 7-DOF robot manipulator LBR iiwa are carried out. The physical structure and D-H parameters of iiwa are shown in Fig. 3.3. All we all know, up to 6 DOFs (3 DOFs of position and another 3 DOFs of orientation) are required to fulfill a given task in engineering applications, therefore, the iiwa is a typical redundant manipulator in the force control when considering both the position and orientation of the end-effector. As to the contact force, the contact surface is selected as a plane in the workspace, as shown in Fig. 3.3a. The end-effector is controlled to offer a desired contact force on the contact surface while tracking a given path on it. In the control process, the orientation of the end-effector is wished to keep constant.

This chapter mainly consists of three parts, firstly, a comparative simulation between the proposed controller and pseudo-inverse of Jacobian matrix(PJMI) based method is firstly discussed, and then the effectiveness of the proposed adaptive controller is checked via more cases. In addition, more discussion about the superiority of the proposed method is carried out to enlighten the contribution of this chapter.

### *3.4.1 Simulation Setup*

In this chapter, the initial value of joint angles is set as $\theta_0 = [0, \pi/3, 0, \pi/3, 0, \pi/3, 0]^{\mathrm{T}}$rad, and the corresponding coordinate of the end-effector is noted as $P0$. The initial value of joint velocity is set as $\dot{\theta}_0 = [0, 0, 0, 0, 0, 0, 0]^{\mathrm{T}}$rad/s. The contact surface is defined as a horizontal plane with $z = 0.094$m, and the physical parameters of the interaction model are set as $K_p = 5000$, $K_d = 20$, respectively. The limitations of



| *Joint* | $a_i$(m) | $\alpha_i$(rad) | $d_i$(m) |
|---|---|---|---|
| 1 | 0 | $-\pi/2$ | 0.36 |
| 2 | 0 | $\pi/2$ | 0 |
| 3 | 0 | $-\pi/2$ | 0.42 |
| 4 | 0 | $\pi/2$ | 0 |
| 5 | 0 | $-\pi/2$ | 0.40 |
| 6 | 0 | $\pi/2$ | 0 |
| 7 | 0 | 0 | 0.276 |

**Fig. 3.3** The architecture of 7-DOF manipulator iiwa. **a** Physical structure. **b** Table of D-H parameters

joint angles and velocities are selected as $\theta^- = [-2, -2, -2, -2, -2, -2, -2]^\mathrm{T}$rad, $\theta^+ = [2, 2, 2, 2, 2, 2, 2]^\mathrm{T}$rad, $\dot{\theta}^- = [-2, -2, -2, -2, -2, -2, -2]^\mathrm{T}$rad/s and $\dot{\theta}^+ = [2, 2, 2, 2, 2, 2, 2]^\mathrm{T}$rad/s, respectively. The control gains of the proposed ARNN are set as $\varepsilon = 0.002$, $\Gamma_1 = diag(5000, 3000)$, $\Gamma_2 = 100I$, $\alpha = 8$, respectively.

## 3.4.2 Comparative Simulation Between PJMI Methods

Firstly, a comparative simulation between the proposed control strategy and traditional Jacobian-inverse based methods is carried out to show the superiority of the RNN based controller. The robot is expected to provide a contact force of 20N at a fixed point $P1 = [0.2, 0.6, 0.094]^\mathrm{T}$m, without considering the orientation control of the end-effector. In traditional PJMI based methods, the joint commands are obtained by directly calculating the inverse of the Jacobian matrix online, and only the special solution is considered. Simulation results are shown in Fig. 3.4. Both controllers can guarantee the convergence of positional and force errors. Using the same control gain in the outer loop, although the controller based on PJMI achieves a faster convergence of control errors, its output is big at the beginning of the simulation (with the Euclidean norm of joint velocity being about 20 rad/s), moreover, as shown in Fig. 3.4c, the joint angle $\theta_6$ exceeds its upper bound during 0.2–1 s. In contrast, using the RNN based controller, both joint angles and velocities are ensured not to exceed their limits. It is worth noting that at about $t = 0.6$s, $\theta_6$ reaches its upper limit (Fig. 3.4e), correspondingly, the joints move a relatively big range, as shown in Fig. 3.4f, as a result, $\theta_6$ stops increasing and then converges to a group of certain values via self-motion.

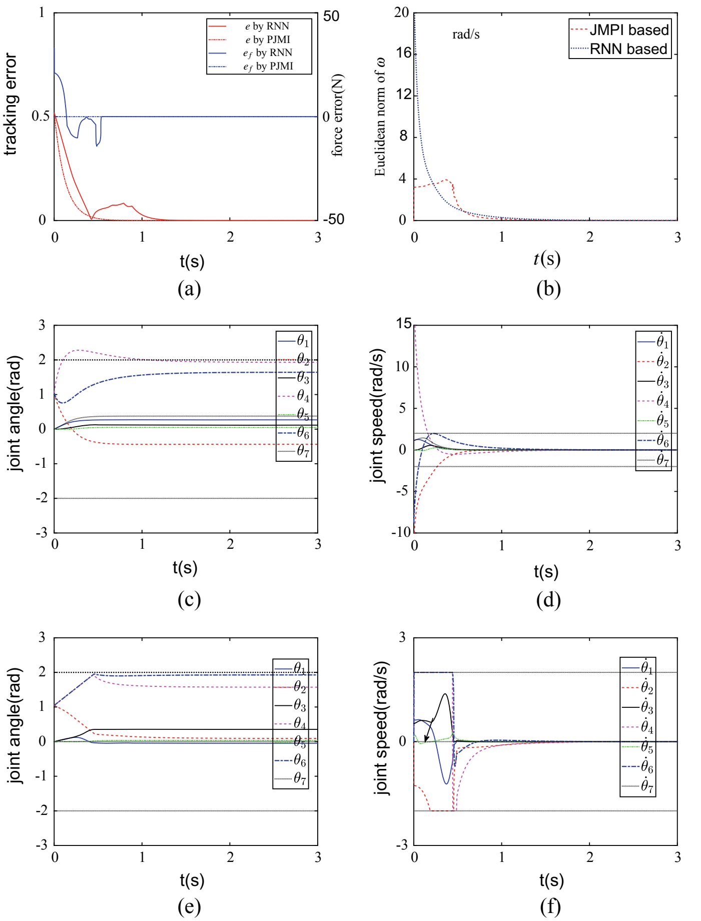


**Fig. 3.4** Numerical results of comparative simulation between the proposed scheme and PJMI based methods. **a** Profile of position and force errors. **b** Euclidean norm of joint velocities. **c** Profile of joint angles using PJMI method. **d** Profile of joint velocities using PJMI method. **e** Profile of joint angles using the proposed method. **f** Profile of joint velocities using the proposed method

## 3.4.3 Force Control Along Predefined Trajectories with Model Uncertainties

In this subchapter, we will carry out a group of experimental tests to further verify the validity of RNN based admittance controller (3.11). In terms of the interaction parameters, we assume the nominal values of $K_p$ and $K_d$ is 4500 and 15, respectively. As to the kinematic parameters, the nominal value of $a_k$ is set to be $\hat{a}_k(0) = [\hat{D}_1(0), \hat{D}_3(0), \hat{D}_5(0), \hat{D}_7(0)]^{\mathrm{T}} = [0.36, 0.4, 0.42, 0.25]^{\mathrm{T}}\mathrm{m}$.

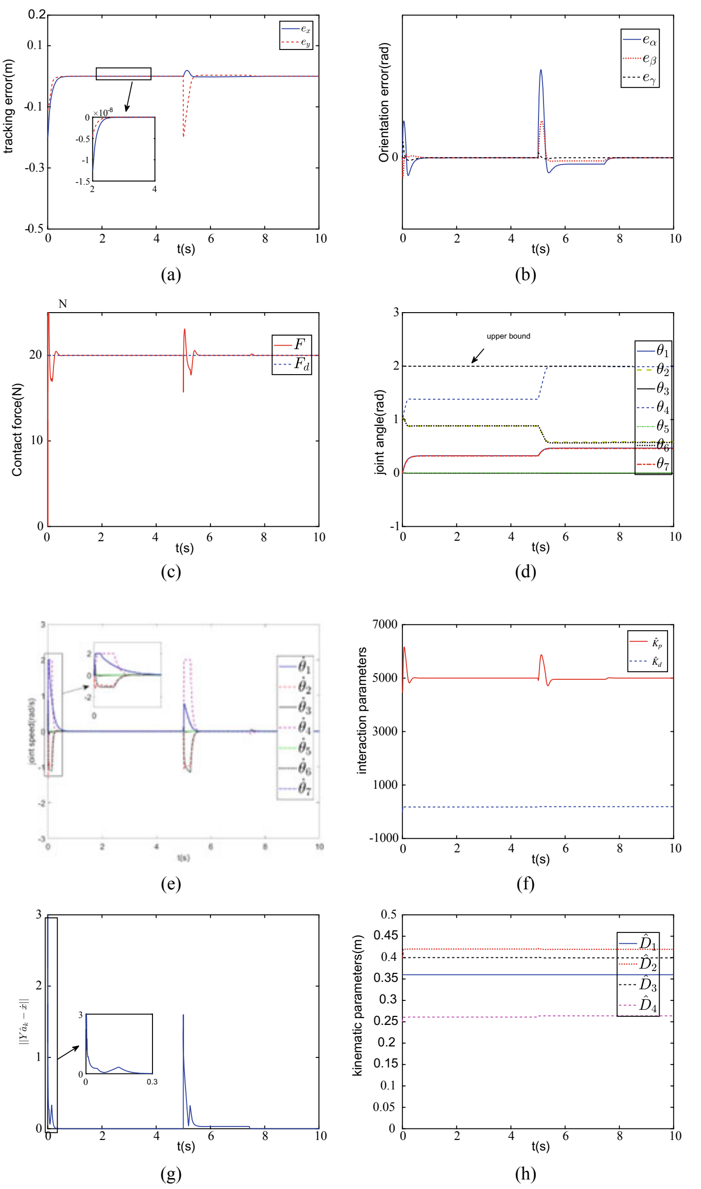
*(1) Force Control On Fixed Points*
Similar to Sect. 3.4.2, a motion-force control at fixed points is studied first. When simulation begins, the target point is set as $P1 = [0.2, -0.6, 0.094]^{\mathrm{T}}$m, at $t = 5$s, the target point is reset to $P2 = [0.2, -0.4, 0.094]^{\mathrm{T}}$m. During the simulation, the contact force between the end-effector and contact surface is selected as $F_{\mathrm{d}} = 20$N. Numerical results are shown in Figs. 3.5 and 3.6.

The position error when the simulation begins is about 0.2 m, accordingly, the proposed RNN based controller generates a large output, which ensures the quick convergence of both motion and force errors. The stabilization time is about 0.5 s. At $t = 5$s, the target point is switched to $P2$, leading to an instantaneous change of position error. Using the adaptive admittance controller (3.11), the robot adjusts its joint configurations quickly and then slows down with the convergence of errors. It is remarkable that the second joint reaches its maximum value, and during the whole process, the joint velocities are guaranteed not to exceed the predefined limits. The estimated values of $\hat{K}_p$ and $\hat{K}_d$ are shown in Fig. 3.5f, although the exact values of $K_d$ and $K_p$ are unknown, by updating $\hat{K}_p$ and $\hat{K}_p$ online according to (3.11), precise control of control is achieved. The difference between the task-space speed $\dot{x}$ and its estimate value $Y\hat{a}_k$ is shown in Fig. 3.5g, correspondingly, $\hat{D}_1$, $\hat{D}_2$, $\hat{D}_3$ and $\hat{D}_4$ converge to a group of constant value.

*(2) Force Control Along A Circular Path*
In this example, the end-effector is controlled to offer constant contact force $F_{\mathrm{d}} = 20$N while tracking a circular trajectory on the contact surface, this trajectory is defined as $x_{\mathrm{d}} = [-0.1 + 0.1cos(0.5t), -0.6 - 0.1sin(0.5t), 0.094]^{\mathrm{T}}$m, and the orientation is required to remain the same as the initial state. Numerical results are shown in Figs. 3.7 and 3.8. As shown in Fig. 3.7a, the robot tracks the desired path successfully, and both position and orientation errors converge to zero in less than 1 s, the expected contact force is also obtained. Because of the periodicity of the desired commands, the robot′s joint angles and angular velocities change periodically, at the same time, boundedness of $\theta$ and $\dot{\theta}$ is also guaranteed. On the other hand, the smooth change of $\theta$ and $\dot{\theta}$ shows that the proposed controller is very stable. Based on the adaptive strategy (3.11d), the system shows great robustness against uncertain system parameters.

*(3) Force Control Along A Rhodonea Path*
In this example, we consider the case where the robot provides a time-varying contact force while tracking a Rhodonea path. The desired contact force is set to be $F_{\mathrm{d}} = 20 + 5sin(0.2t)$N, and the Rhodonea path is defined as

$$\begin{aligned}
x_{\mathrm{d}X}(t) &= 0.1sin(0.4t)cos(0.2t),\\
x_{\mathrm{d}Y}(t) &= 0.15sin(0.4t)sin(0.2t) - 0.6,\\
x_{\mathrm{d}Z}(t) &= 0.094.
\end{aligned}$$



(a)

(b)

(c)

(d)

(e)

(f)

(g)

(h)

**Fig. 3.5** Numerical results of force control at fixed points with uncertain model parameters. **a** Profile of positional error. **b** Profile of orientational error. **c** Profile of contact force. **d** Profile of joint angles. **e** Profile of joint speed. **f** Profile of the estimated interaction coefficients. **g** Profile of $||Y\hat{a}_k - \dot{x}||_2^2$. **h** Profile of the estimated physical parameters

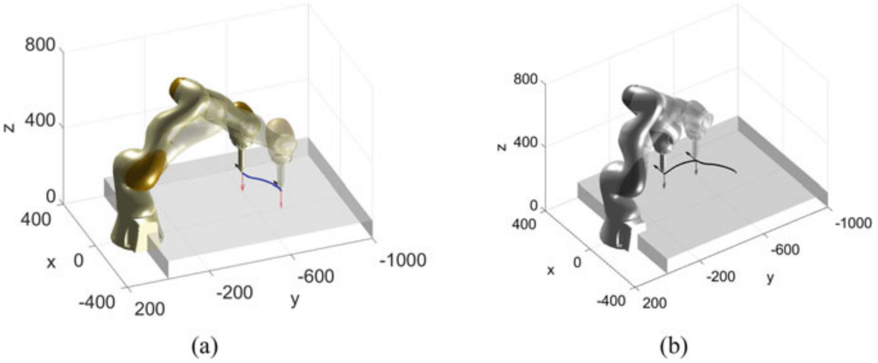
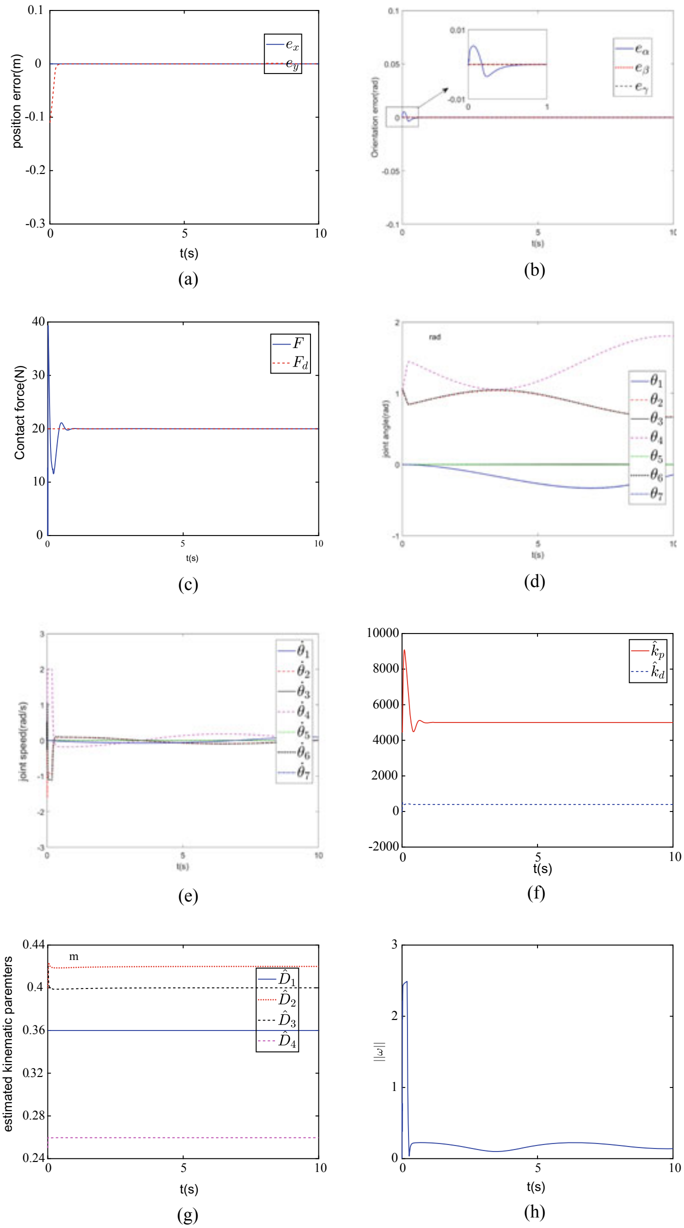


**Fig. 3.6** Snapshots when iiwa offers a constant contact force at fixed points. **a** Snapshot when $t = 2$s. **b** Snapshot when $t = 7$s

Numerical results are shown in Figs. 3.9 and 3.10. Figure 3.9a, b show the positional and orientational errors the end-effector respect to the desired path, respectively. In the steady-state, accurate motion control is realized using the proposed controller, and the operation force between the end-effector and contact surface is shown in Fig. 3.9c. At the beginning stage, the joint speed is high, which enables the end-effector to move toward the desired path rapidly. As the end-effector approached the expected path, the robot moves at a low speed periodically and smoothly, correspondingly, the joint angles changes at the same frequency. The Euclidean norm of $Y\hat{a}_k - \dot{x}$ is illustrated in Fig. 3.9g, the proposed RNN based control strategy could calculate control command $\omega$ online with subject to model uncertainties. The estimated model parameters are given in Fig. 3.10f, h.

### *3.4.4 Comparison*

To further illustrate the contribution of the proposed force control strategy, we provide a comparison between the proposed method and the existing related methods, as shown in the Table 3.1. In [11], an adaptive admittance control scheme based on neural network approximation capability is proposed and the admittance adaptive tracking error is optimized. However, no physical constraints are considered. In [10], although the established impedance controller can guarantee input saturation, the controller still needs to calculate the pseudo-inverse of the jacobian. In [25, 32], a pseudo-inverse-free controller based on RNN is designed to realize the task space tracking of redundant robots, and its convergence is proved. Convex optimization and non-convex optimization are also obtained. Considering physical uncertainties,



(a)

(b)

(c)

(d)

(e)

(f)

(g)

(h)

**Fig. 3.7** Numerical results of force control along a circular curve with uncertain model parameters. **a** Profile of positional error. **b** Profile of orientational error. **c** Profile of contact force. **d** Profile of joint angles. **e** Profile of joint speed. **f** Profile of the estimated interaction coefficients. **g** Profile of the estimated physical parameters. **h** Profile of the objective function $||\omega||_2^2$

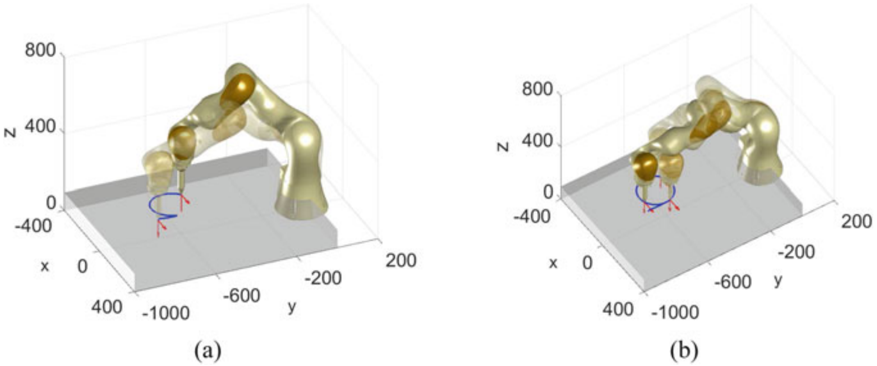


**Fig. 3.8** Snapshots when iiwa offers a constant contact force along a circular curve. **a** Snapshot when $t = 8$s. **b** Snapshot when $t = 15$s

**Table 3.1** Comparisons among different tracking controllers on manipulators

| Method | Handling model uncertainties | Pseudo-inverse Required | Force tracking versus trajectory tracking | Handling inequality constraints | Convergence | Secondary task |
|---|---|---|---|---|---|---|
| This chapter | Yes | No | Force | Yes | Yes | Yes |
| [10] | Yes | Yes | Force | Restrictive[a] | Yes | Yes |
| [11] | Yes | Yes | Force | No | Yes | No |
| [21, 22] | Restrictive[b] | No | Kinematic | Yes | Yes | Yes |
| [25, 32] | No | No | Kinematic | Yes | Yes | Yes |

[a] In [10], only the input saturation is considered
[b] In [21, 22], the authors only consider the uncertainties of physical parameters, while the contact force is ignored

two different adaptive strategies are proposed in [21, 22]. This is the first RNN-based force controller in this chapter. On the other hand, this control scheme is suitable for the case of model uncertainty, so it is no longer necessary to calculate the pseudo-inverse of the Jacobian matrix and has great application potential in force control.

## 3.5 Summary

In this chapter, we propose an adaptive admittance control method for redundant robots based on a recursive neural network. The convergence of the adaptive RNN is proved by the theoretical derivation of the Lyapunov technique, and the effectiveness of the control strategy is verified by numerical simulation on the 7-DOF robot

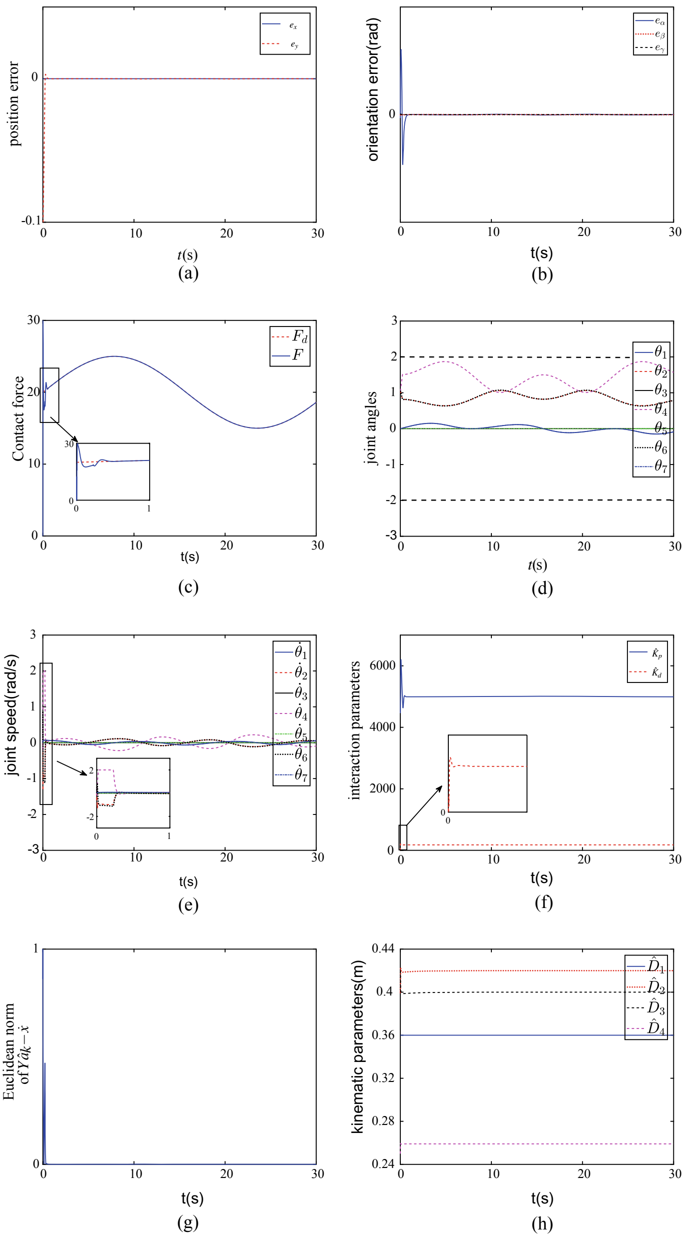


**Fig. 3.9** Numerical results of force control along a Rhodonea curve with uncertain model parameters. **a** Profile of positional error. **b** Profile of orientational error. **c** Profile of contact force. **d** Profile of joint angles. **e** Profile of joint speed. **f** Profile of the estimated interaction coefficients. **g** Profile of $||Y\hat{a}_k - \dot{x}||_2^2$. **h** Profile of the estimated physical parameters

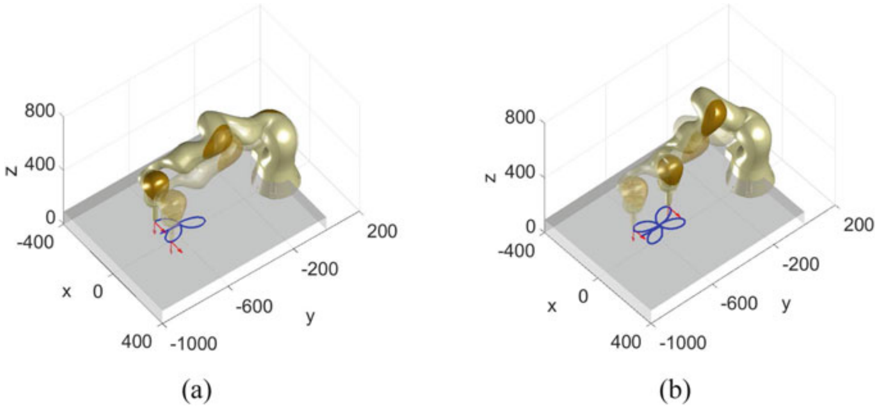


**Fig. 3.10** Snapshots when iiwa offers a constant contact force along a Rhodonea curve. **a** Snapshot when $t = 4$s. **b** Snapshot when $t = 19$s. **c** Snapshot when $t = 27$s

iiwa. Compared with the existing control methods, the controller not only has better performance in dealing with physical constraints but also has better performance in eliminating pseudo-inversion calculation. Finally, it is worth noting that this is the first time that an RNN-based approach has been extended to force control for the redundant manipulator, especially for model uncertainty manipulators. The research of this subject is of great significance to grinding robot, assembly robot and industrial application.

## References

1. C. Yang, Y. Jiang, W. He, J. Na, Z. Li, B. Xu, Adaptive Parameter Estimation and Control Design for Robot Manipulators With Finite-Time Convergence. IEEE Transactions on Industrial Electronics **65**(10), 8112–8123 (2018)
2. L. Cheng, Z.G. Hou, M. Tan, Adaptive neural network tracking control for manipulators with uncertain kinematics, dynamics and actuator model. Automatica **45**(10), 2312–2318 (2009)
3. Y. Pan, H. Wang, X. Li, H. Yu, Adaptive Command-Filtered Backstepping Control of Robot Arms With Compliant Actuators. IEEE Transactions on Control Systems Technology **26**(3), 1149–1156 (2018)
4. N. Hogan, Impedance control - An approach to manipulation. I - Theory. II - Implementation. III - Applications. Asme Transactions Journal of Dynamic Systems & Measurement Control B **107**(1), 304–313 (1985)
5. J. Craig, Ping Hsu and S. Sastry, “Adaptive control of mechanical manipulators,” Proceedings. 1986 IEEE International Conference on Robotics and Automation, pp. 190-195, 1986
6. M.H. Raibert, J.J. Craig, Hybrid Position/Force Control of Manipulator. Asme Journal of Dynamic Systems Measurement & Control **102**(2), 126–133 (1981)

7. Y.P. Pan, X. Li, H.M. Wang, H.Y. Yu, Continuous sliding mode control of compliant robot arms: A singularly perturbed approach. Mechatronics **52**(1), 127–134 (2018)
8. W. He, S.S. Ge, Y.N. Li, E. Chew, Neural Network Control of a Rehabilitation Robot by State and Output Feedback. Journal of Intelligent & Robotic Systems **80**(1), 15–31 (2015)
9. J. Ren, B. Wang, M. Cai and J. Yu, "Adaptive Fast Finite-Time Consensus for Second-Order Uncertain Nonlinear Multi-Agent Systems With Unknown Dead-Zone," IEEE Access, vol. 8, No. 1, pp. 25557-25569, 2020
10. W. He, Y. Dong, C. Sun, Adaptive Neural Impedance Control of a Robotic Manipulator With Input Saturation. IEEE Transactions on Systems, Man, and Cybernetics: Systems **46**(3), 334–344 (2016)
11. C. Yang, G. Peng, Y. Li, R. Cui, L. Cheng, Z. Li, Neural Networks Enhanced Adaptive Admittance Control of Optimized Robot-Environment Interaction. IEEE Transactions on Cybernetics **49**(7), 2568–2579 (2019)
12. C. Yang, G. Ganesh, S. Haddadin, S. Parusel, A. Albu-Schaeffer, E. Burdet, Human-Like Adaptation of Force and Impedance in Stable and Unstable Interactions. IEEE Transactions on Robotics **27**(5), 918–930 (2011)
13. H. Wu, Y. Guan, J. Rojas, A latent state-based multimodal execution monitor with anomaly detection and classification for robot introspection. Applied Sciences. **9**(6), 1072 (2019). Jan
14. H. Wu, Z. Xu, W. Yan, Q. Su, S. Li, T. Cheng, X. Zhou, Incremental Learning Introspective Movement Primitives From Multimodal Unstructured Demonstrations. IEEE Access. **15**(7), 159022–36 (2019). Oct
15. H. Wu, Y. Guan, J. Rojas, Analysis of multimodal Bayesian nonparametric autoregressive hidden Markov models for process monitoring in robotic contact tasks. International Journal of Advanced Robotic Systems. **16**(2), 1729881419834840 (2019). Mar 26
16. B. Xu, D. Yang, Z. Shi, Y. Pan, B. Chen, F. Sun, Online Recorded Data-Based Composite Neural Control of Strict-Feedback Systems With Application to Hypersonic Flight Dynamics. IEEE Transactions on Neural Networks and Learning Systems **29**(8), 3839–3849 (2018)
17. B. Xu, D. Wang, Y. Zhang, Z. Shi, DOB-Based Neural Control of Flexible Hypersonic Flight Vehicle Considering Wind Effects. IEEE Transactions on Industrial Electronics **64**(11), 8676–8685 (2017)
18. W.E. Dixon, Adaptive Regulation of Amplitude Limited Robot Manipulators With Uncertain Kinematics and Dynamics. IEEE Transactions on Automatic Control **52**(3), 488–493 (2007)
19. L. Cheng , Z. G. Hou and M. Tan, "Adaptive neural network tracking control of manipulators using quaternion feedback," 2008 IEEE International Conference on Robotics and Automation, pp. 3371-3376, 2008
20. D. Chen, Y. Zhang, S. Li, Tracking Control of Robot Manipulators with Unknown Models: A Jacobian-Matrix-Adaption Method. IEEE Transactions on Industrial Informatics **14**(7), 3044–3053 (2018)
21. Z. Xu, S. Li, X. Zhou, W. Yan, T. Cheng, D. Huang, Dynamic neural networks based kinematic control for redundant manipulators with model uncertainties. Neurocomputing **329**(1), 255–266 (2019)
22. Y. Zhang, S. Chen, S. Li, Z. Zhang, Adaptive Projection Neural Network for Kinematic Control of Redundant Manipulators With Unknown Physical Parameters. IEEE Transactions on Industrial Electronics **65**(6), 4909–4920 (2018)
23. J. Na, M. Nasiruddin, H. Guido, X. Ren, B. Phil, Robust adaptive finite-time parameter estimation and control for robotic systems. International Journal of Robust & Nonlinear Control **25**(16), 345–3071 (2015)

24. H. Wang, P. Shi, H. Li, Q. Zhou, Adaptive Neural Tracking Control for a Class of Nonlinear Systems With Dynamic Uncertainties. IEEE Transactions on Cybernetics **47**(10), 3075–3087 (2017)
25. S. Li, Y. Zhang, L. Jin, Kinematic Control of Redundant Manipulators Using Neural Networks. IEEE Transactions on Neural Networks and Learning Systems **28**(10), 2243–2254 (2017)
26. Y. Zhang, Inverse-free computation for infinity-norm torque minimization of robot manipulators. Mechatronics **16**(3), 177–184 (2006)
27. D. Chen, S. Li, Q. Wu, X. Luo, Super-twisting ZNN for coordinated motion control of multiple robot manipulators with external disturbances suppression. Neurocomputing **371**(1), 78–90 (2020)
28. Dechao Chen, Shuai Li, "A recurrent neural network applied to optimal motion control of mobile robots with physical constraints," Applied Soft Computing, to be published, 2019, doi: https://doi.org/10.1016/j.asoc.2019.105880.
29. Y. Zhang, S. Ge, T. Lee, "A unified quadratic-programming-based dynamical system approach to joint torque optimization of physically constrained redundant manipulators," IEEE Transactions on Systems, Man, & Cybernetics. Part B (Cybernetics) **34**(5), 2126–2132 (2004)
30. S. Li, M. Zhou, X. Luo, Z. You, Distributed Winner-Take-All in Dynamic Networks. IEEE Transactions on Automatic Control **62**(2), 577–589 (2017)
31. Y. Zhang, S. Li, J. Gui, X. Luo, Velocity-Level Control With Compliance to Acceleration-Level Constraints: A Novel Scheme for Manipulator Redundancy Resolution. IEEE Transactions on Industrial Informatics **14**(3), 921–930 (2018)
32. L. Jin, S. Li, H.M. La, X. Luo, Manipulability Optimization of Redundant Manipulators Using Dynamic Neural Networks. IEEE Transactions on Industrial Electronics **64**(6), 4710–4720 (2017)
33. B. Cai, Y. Zhang, Different-Level Redundancy-Resolution and Its Equivalent Relationship Analysis for Robot Manipulators Using Gradient-Descent and Zhang's Neural-Dynamic Methods. IEEE Transactions on Industrial Electronics **59**(8), 3146–3155 (2012)
34. J. Na, X.M. Ren, D.D. Zheng, Adaptive Control for Nonlinear Pure-Feedback Systems With High-Order Sliding Mode Observer. IEEE Transactions on Neural Networks and Learning Systems **24**(3), 370–382 (2013)
35. S. Li, S. Chen, B. Liu, Y. Li, Y. Liang, Decentralized kinematic control of a class of collaborative redundant manipulators via recurrent neural networks. Neurocomputing **91**(1), 1–10 (2012)
36. H. Wang, K. Liu, X. Liu, B. Chen, C. Lin, Neural-Based Adaptive Output-Feedback Control for a Class of Nonstrict-Feedback Stochastic Nonlinear Systems. IEEE Transactions on Cybernetics **45**(9), 1977–1987 (2015)
37. Y. Li, S. Li, B. Hannaford, A Model based Recurrent Neural Network with Randomness for Efficient Control with Applications. IEEE Transactions on Industrial Informatics **15**(4), 2054–2063 (2019)
38. X. Li, Z. Xu, S. Li, H. Wu, X, Zhou, "Cooperative Kinematic Control For Multiple Redundant Manipulators Under Partially Known Information Using Recurrent Neural Network". IEEE ACCESS **8**(1), 40029–40038 (2020)
39. Z. Xu, S. Li, X. Zhou, T. Cheng, Dynamic Neural Networks Based Adaptive Admittance Control for Redundant Manipulators with Model Uncertainties. Neurocomputing **357**(1), 271–281 (2019)
40. Z. Xu, S. Li, X. Zhou, W. Yan, T. Cheng, H. Dan, "Dynamic Neural Networks for Motion-Force Control of Redundant Manipulators: An Optimization Perspective", IEEE transactions on industrial electronics. Early access (2020). https://doi.org/10.1109/TIE.2020.2970635
41. H. Khalil, *Nonlinear Systems* (Prentice Hall, New Jersey, USA, 1996)

42. Zhijia Zhao, Xiuyu He, Zhigang Ren, Guilin Wen, Boundary Adaptive Robust Control of a Flexible Riser System with Input Nonlinearities. IEEE Transactions on Systems, Man, and Cybernetics: Systems **49**(10), 1971–1980 (2019). https://doi.org/10.1109/TSMC.2018.2882734
43. Zhijia Zhao, Choon Ki Ahn, Han-Xiong Li. "Deadzone Compensation and Adaptive Vibration Control of Uncertain Spatial Flexible Riser Systems". IEEE/ASME Transactions on Mechatronics, in press, DOI:https://doi.org/10.1109/TMECH.2020.29755672020.

# Chapter 4
# Deep RNN Based Obstacle Avoidance Control for Redundant Manipulators

**Abstract** In this chapter, we consider the obstacle avoidance problem of redundant robot manipulators with physical constraints compliance, where static and dynamic obstacles are investigated. Both the robot and obstacles are abstracted as two critical point sets, respectively, relying on the general class-K functions, the obstacle avoidance problem is formulated into an inequality in speed level. The minimal-velocity-norm (MVN) is regarded as the cost function, converting the kinematic control problem of redundant manipulators considering obstacle avoidance into a constraint-quadratic-programming problem, in which the joint angles and joint velocity constraints are built in velocity level in form of inequality. To solve it, a novel deep recurrent neural network based controller is proposed. Theoretical analyses and the corresponding simulative experiments are given successively, showing that the proposed neural controller does not only avoid collision with obstacles, but also track the desired trajectory correctly.

## 4.1 Introduction

With development of intelligent manufacturing and automation, the research on robot manipulators is obtaining increasing attention from a large number of scholars, numerous fruits have been reported on painting, welding and assembly [1, 2] and so on. With the popularization of robots, higher requirements such as flexibility and execution ability are imposed on robots, especially working in the complicated environment [3]. Consequently, more and more scholars cast light on redundant robots which show better flexibility, responsiveness [4, 5].

Stem from the consideration of human-machine collaboration, robots are no longer arranged in a separate area [6–8], which makes the obstacle avoidance for robots become an important part of kinematic control of the robot manipulators. There has reported many obstacle avoidance methods applicable to robot manipulators. A modified RRT based method, namely Smoothly RRT, was proposed in [9]. This paper established a maximum curvature constraint to obtain a smooth curve when avoiding obstacles. Compared to the traditional RRT based method, the proposed method achieves faster convergence. In [10], Hsu investigated the probabilistic foundations

X. Zhou et al., *AI based Robot Safe Learning and Control*,
https://doi.org/10.1007/978-981-15-5503-9_4

of PRM based methods, obtaining a conclusion that the visibility properties has a heavier impact on the probability, and the convergence would be faster if extract partial knowledge could be introduced. However, due to the heavy computational burdens, those methods can be hardly used online.

Apart from stochastic sampling based algorithms mentioned above, the artificial potential field method is also a potential method for obstacle avoidance, and have been found their application in [11–15]. Taking advantage of redundant DOFs, obstacles can be avoided by the self-motion in the null space. Using pseudo-inverse of Jacobian matrix, the solution can be built as the sum of a minimum-norm particular solution and homogeneous solutions [16–18].

With parallelism and easier to implement in hardware, neural networks have been a powerful tool in robot control. Artificial intelligence algorithms based on neural networks provide a new view for robotic control, these methods are very promising due to neural networks' excellent learning ability [19]. For example, in [20], a neural network based learning scheme was proposed to handle functional uncertainties. In [21], a bio-mimetic hybrid controller was designed, where the control strategy consist of an RBF neural network based feed-forward predictive machine and a feedback servo machine based on a proportional-derivative controller. In [22], a fuzzy logic controller is proposed for long-term navigation of quad-rotor UAV systems with input uncertainties. Experiment results show that the controller can achieve better control performance when compared to their singleton counterparts. In [23], an online learning mechanism is built for visual tracking systems. The controller uses both positive and negative sample importance as input, and it is shown that the proposed weighted multiple instance learning scheme achieves wonderful tracking performance in challenging environments. The system model of robot manipulators is highly nonlinear, however, if the prior information of the model is known in advance, the neural network can be optimized. This is to say, on one hand, the number of nodes in neural networks can be reduced. In addition, the excellent learning efficiency is maintained simultaneously [24]. Therefore, to achieve the real-time control of robot manipulators, a series of dynamic neural network are proposed, such as [25–27]. For kinematic control of redundant manipulators, such a time-varying problem will be transformed into a quadratic programming from perspective of optimization, where nonlinear mapping from joint space to cartesian space is abstracted as a linear equation. Dynamic neural networks can be used to solve the quadratic-programming problem online, therefore, the kinematic control of manipulators is achieved when the formulated linear equation is ensured. More importantly, these methods can also handle inequality constraints considering joint physical constraints, and model uncertainties [28–32]. There are few works on obstacle avoidance using dynamic neural network. In [33], the obstacle avoidance problem is considered as an equality constraint, however the parameters of the escape velocity is not easy to get. In [34], the distance between the robot and obstacles are formulated as a group of distances from critical points and robot links. On this basis, an improved method is proposed by Guo et. al. in [35], which can suppress undesirable discontinuity in the original solutions.

Motivated by the above observations, in this chapter, a RNN-based obstacle avoidance strategy was proposed for redundant robot manipulators. Both the robot and obstacles are abstracted as two critical point sets, respectively, relying on the class-K functions, the obstacle avoidance problem is formulated into an inequality in speed level. The minimal velocity-norm (MVN) is regarded as the cost function, converting the kinematic control problem of redundant manipulators considering obstacle avoidance into a constraint-quadratic-programming problem, in which the joint angles and joint velocity constraints are built in velocity level in form of inequality. To solve it, a novel deep recurrent neural network based controller is proposed. Theoretical analyses and the corresponding simulative experiments are given successively, showing that the proposed neural controller does not only avoid collision with obstacles, but also track the desired trajectory correctly. The main contributions of this chapter are summarized as below:

- A deep RNN-based controller is proposed. Four objectives are achieved at the same time, i.e, the desired path tracking, obstacle avoidance, physical constraints compliance considering joint angles and velocity constraints and optimality of cost functions.
- Relying on a class-K function, a novel obstacle avoidance strategy is given, where robots and obstacles are abstracted into a set of critical point sets, the distance between them can be described as the point-to-point distance.
- Numerical results show that the effectiveness of the designed RNN controller, i.e, the static and dynamic obstacles can be avoided while tracking the desired motion trajectory.

## 4.2 Problem Formulation

### *4.2.1 Basic Description*

When a redundant robot is controlled to track a particular trajectory in the cartesian space, the positional description of the end-effector can be formulated as

$$x = f(\theta), \tag{4.1}$$

where $x \in \mathbb{R}^m$ and $\theta \in \mathbb{R}^n$ are the end-effector′s positional vector and joint angles, respectively. In the velocity level, the kinematic mapping between $\dot{x}$ and $\dot{\theta}$ can be described as

$$\dot{x} = J(\theta)\dot{\theta}, \tag{4.2}$$

where $J(\theta) \in \mathbb{R}^{m\times n}$ is the Jacobian matrix from the end-effector to joint space.

In engineering applications, obstacles are inevitable in the workspace of a robot manipulator. For example, robot manipulators usually work in a limited workspace restricted by fences, which are used to isolate robots from humans or other robots.

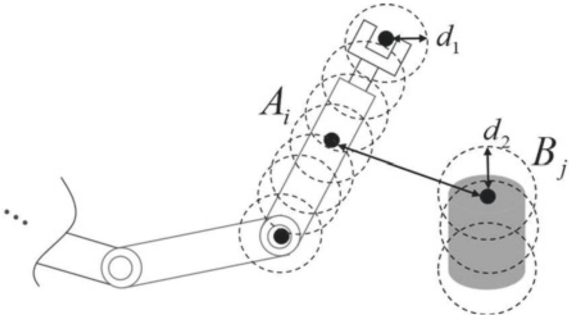


**Fig. 4.1** The basic idea of obstacle avoidance in this chapter

This problem could be even more acute in tasks which require collaboration of multiple robots. Let $C_1$ be the set of all the points on a robot body, and $C_2$ be the set of all the points on the obstacles, then the purpose of obstacle avoidance of a robot manipulator is to ensure $C_1 \cup C_2 = \emptyset$ at all times. By introducing $d$ as a safety distance between the robot and obstacles, the obstacle avoidance is reformulated as

$$|O_j A_i| \geq d, \qquad \forall A_i \in C_1, \forall O_i \in C_2. \tag{4.3}$$

where $|O_j A_i| = \sqrt{(A_i - O_j)^{\mathrm{T}}(A_i - O_j)}$ is the Euclidean norm of the vector $A_i O_j$.

Equation (4.3) gives a basic description of obstacle avoidance problem in form of inequalities. However, there are too many elements in sets $C_1$ and $C_2$, the vast majority of which are actually unnecessary. Therefore, by uniformly selecting points of representative significance from $C_1$ and $C_2$, and increasing $d$ properly, Eq. (4.3) can be approximately described as below

$$|O_j A_i| \geq d, \tag{4.4}$$

with $A_i, i = 1, \ldots, a$ and $O_j, j = 1, \ldots, b$ being the representative points of the robot and obstacles, respectively. The schematic diagram of Eq. (4.4) in shown in Fig. 4.1.

### 4.2.2 Reformulation of Inequality in Speed Level

In order to guarantee the inequality (4.4), by defining $D = |O_j A_i| - d$, an inequality is rebuilt in speed level as

$$\mathrm{d}(|O_j A_i|)/\mathrm{d}t \geq -\mathrm{sgn}(D) g(|D|), \tag{4.5}$$

in which $g(\bullet)$ belongs to class-$K$. Remarkable that the velocities of critical points $A_i$ can be described by joint velocities

$$\dot{A}_i = J_{ai}(\theta)\dot{\theta}, \tag{4.6}$$

where $J_{ai} \in \mathbb{R}^{m\times n}$ is the Jacobian matrix from the critical point $A_i$ to joint space. If $O_j$ is prior known, the left-side of Eq. (4.5) can be reformulated as

$$\begin{aligned}\frac{\mathrm{d}}{\mathrm{d}t}(|O_jA_i|) =& \frac{\mathrm{d}}{\mathrm{d}t}(\sqrt{(A_i - O_j)^{\mathrm{T}}(A_i - O_j)})\\ =& \frac{1}{|O_jA_i|}(A_i - O_j)^{\mathrm{T}}(\dot{A}_i - \dot{O}_j)\\ =& \overrightarrow{|O_jA_i|}^{\mathrm{T}} J_{ai}(\theta)\dot{\theta} - \overrightarrow{|O_jA_i|}^{\mathrm{T}}\dot{O}_j,\end{aligned} \tag{4.7}$$

where $\overrightarrow{|O_jA_i|} = (A_i - O_j)^{\mathrm{T}}/|O_jA_i| \in \mathbb{R}^{1\times m}$ is the unit vector of $\overrightarrow{A_i - O_j}$. Therefore, the collision between critical point $A_i$ and object $O_j$ can be obtained by ensuring the following inequality

$$J_{oi}\dot{\theta} \leq \mathrm{sgn}(D)g(|D|) - \overrightarrow{|O_jA_i|}^{\mathrm{T}}\dot{O}_j, \tag{4.8}$$

where $J_{oi} = -\overrightarrow{|O_jA_i|}^{\mathrm{T}} J_{ai} \in \mathbb{R}^{1\times n}$. Based on the inequality description (4.8), the collision between robot and obstacle can be avoided by ensuring

$$J_o\dot{\theta} \leq B, \tag{4.9}$$

where $J_o = [\underbrace{J_{o1}^{\mathrm{T}}, \cdots, J_{o1}^{\mathrm{T}}}_{b}, \cdots, \underbrace{J_{oa}^{\mathrm{T}}, \cdots, J_{oa}^{\mathrm{T}}}_{b}]^{\mathrm{T}} \in \mathbb{R}^{ab\times n}$ is the concatenated form of $J_{oi}$ considering all pairs between $A_i$ and $O_j$, $B = [B_{11}, \cdots, B_{1b}, \cdots, B_{a1}, \cdots, B_{ab}]^{\mathrm{T}} \in \mathbb{R}^{ab}$ is the vector of upper-bounds, in which $B_{ij} = \mathrm{sgn}(D)g(|D|) - \overrightarrow{|O_jA_i|}^{\mathrm{T}}\dot{O}_j$.

**Remark 4.1** According to Eq. 4.5 and the definition of class-K functions, the original escape velocity based obstacle avoidance methods in [34, 35] can be regarded as a special case of 4.5, in which $G(|D|)$ is selected as $G(|D|) = k|D|$. Therefore, in this chapter, the proposed obstacle avoidance strategy is more general than traditional methods.

### *4.2.3 QP Type Problem Description*

As to redundant manipulators, in order to take full advantage of the redundant DOFs, the robot can be designed to fulfill a secondary task when tracking a desired trajectory. In this chapter, the secondary task is set to minimize joint velocity while avoiding obstacles. In real implementations, both joint angles and velocities are limited because of physical limitations such as mechanical constraints and actuator saturation. Because of the fact that $\mathrm{rank}(J) < n$, there might be infinity solutions to achieve kinematic control. In this chapter, we aim to design a kinematic controller which is capable of avoiding obstacles while tracking a pre-defined trajectory in the Cartesian space. For safety, the robot is wished to move at a low speed, in addition, lower energy consumption is guaranteed. By defining an objective function scaling joint velocity as $\dot{\theta}^{\mathrm{T}}\dot{\theta}/2$, the tracking control of a redundant manipulator while avoiding obstacles can be described as

$$
\begin{aligned}
\min \quad & \dot{\theta}^{\mathrm{T}}\dot{\theta}/2, && \text{(4.10a)}\\
s.t. \quad & x = x_{\mathrm{d}}, && \text{(4.10b)}\\
& \theta^{-} \leq \theta \leq \theta^{+}, && \text{(4.10c)}\\
& \dot{\theta}^{-} \leq \dot{\theta} \leq \dot{\theta}^{+}, && \text{(4.10d)}\\
& J_o\dot{\theta} \leq B. && \text{(4.10e)}
\end{aligned}
$$

It is remarkable that the constraints Eq. (4.10b)–(4.10e) and the objective function (4.10a) to be optimized are built in different levels, which is very difficult to solve directly. Therefore, we will transform the original QP problem (4.10) in the velocity level. In order to realize precise tracking control to the desired trajectory $x_{\mathrm{d}}$, by introducing a negative feedback in the outer-loop, the equality constraint can be ensured by letting the end-effector moves at a velocity of $\dot{x} = \dot{x}_{\mathrm{d}} + k(x_{\mathrm{d}} - x)$. In terms with (4.10c), according to the escape velocity method, it can be obtained by limiting joint speed as $\alpha(\theta^{-} - \theta) \leq \dot{\theta} \leq \alpha(\theta^{+} - \theta)$, where $\alpha$ is a positive constant. Combing the kinematic property (4.2), the reformulated QP problem is

$$
\begin{aligned}
\min \quad & \dot{\theta}^{\mathrm{T}}\dot{\theta}/2, && \text{(4.11a)}\\
s.t. \quad & J(\theta)\dot{\theta} = \dot{x}_{\mathrm{d}} + k(x_{\mathrm{d}} - x), && \text{(4.11b)}\\
& \max(\alpha(\theta^{-} - \theta), \dot{\theta}^{-}) \leq \dot{\theta} \leq \min(\dot{\theta}^{+}, \alpha(\theta^{+} - \theta)), && \text{(4.11c)}\\
& J_o\dot{\theta} \leq B. && \text{(4.11d)}
\end{aligned}
$$

It is noteworthy that both the formula (4.11a) and (4.11d) are nonlinear. The QP problem cannot be solved directly by traditional methods. Using the parallel computing and learning ability, a deep RNN will be established later.

## 4.3 Deep RNN Based Solver Design

In this chapter, a deep RNN is proposed to solve the obstacle avoidance problem (4.11) online. To ensure the constraints (4.11b), (4.11c), and (4.11d), a group of state variables are introduced in the deep RNN. The stability is also proved by Lyapunov theory.

### 4.3.1 Deep RNN Design

Firstly, by defining a group of state variables $\lambda_1 \in \mathbb{R}^m, \lambda_2 \in \mathbb{R}^{ab}$, a Lagrange function is selected as

$$L = \dot{\theta}^{\mathrm{T}}\dot{\theta}/2 + \lambda_1^{\mathrm{T}}(\dot{x}_{\mathrm{d}} + k(x_{\mathrm{d}} - x) - J(\theta)\dot{\theta}) + \lambda_2^{\mathrm{T}}(J_o\dot{\theta} - B), \tag{4.12}$$

$\lambda_1$ and $\lambda_2$ are the dual variables corresponding to the constraints (4.11b) and (4.11d). According to Karush-Kuhn-Tucker conditions, the optimal solution of the optimization problem (4.12) can be equivalently formulated as

$$\dot{\theta} = P_{\Omega}(\dot{\theta} - \frac{\partial L}{\partial \dot{\theta}}), \tag{4.13a}$$

$$J(\theta)\dot{\theta} = \dot{x}_{\mathrm{d}} + k(x_{\mathrm{d}} - x), \tag{4.13b}$$

$$\begin{cases} \lambda_2 > 0 & \text{if} \quad J_o\dot{\theta} = B, \\ \lambda_2 = 0 & \text{if} \quad J_o\dot{\theta} \leq B, \end{cases} \tag{4.13c}$$

where $P_{\Omega}(x) = \operatorname{argmin}_{y \in \Omega}||y - x||$ is a projection operator to a restricted interval $\Omega$, which is defined as $\Omega = \{\dot{\theta} \in \mathbb{R}^n | \max(\alpha(\theta^- - \theta), \dot{\theta}^-) \leq \dot{\theta} \leq \min(\dot{\theta}^+, \alpha(\theta^+ - \theta))\}$. Notable that Equation (4.13c) can be simply described as

$$\lambda_2 = (\lambda_2 + J_o\dot{\theta} - B)^+, \tag{4.14}$$

where $(\bullet)^+$ is a projection operation to the non-negative space, in the sense that $y^+ = \max(y, 0)$.

Although the solution of (4.13) is exact the optimal solution of the constrained-optimization problem (4.11), it is still a challenging issue to solve (4.13) online since its inherent nonlinearity. In this chapter, in order to solve (4.13), a deep recurrent neural network is designed as

$$\varepsilon\ddot{\theta} = -\dot{\theta} + P_{\Omega}(J^{\mathrm{T}}\lambda_1 - J_o^{\mathrm{T}}\lambda_2), \tag{4.15a}$$

$$\varepsilon\dot{\lambda}_1 = \dot{x}_{\mathrm{d}} + k(x_{\mathrm{d}} - x) - J(\theta)\dot{\theta}, \tag{4.15b}$$

$$\varepsilon\dot{\lambda}_2 = -\lambda_2 + (\lambda_2 + J_o\dot{\theta} - B)^+, \tag{4.15c}$$

with $\varepsilon$ is a positive constant scaling the convergence of (4.15).

**Remark 4.2** As to the established deep RNN (4.15), the first dynamic equation is also the output dynamics, which combines the dynamics of state variables $\lambda_1$ and $\lambda_2$, as well as the physical limitations including joint angles and velocities. State variable $\lambda_1$ is used to ensure the equality constraint (4.11b), as shown in (4.15b), $\lambda_1$ is updated according to the difference between reference speed $\dot{x}_{\mathrm{d}} + k(x_{\mathrm{d}} - x)$ and actually speed $J(\theta)\dot{\theta}$. Similarly, $\lambda_2$ is used to ensure the inequality constraint (4.11d), which will be further discussed later. It is remarkable that $\varepsilon$ plays an important role in the convergence of the deep RNN. The smaller $\varepsilon$, the faster the deep RNN converges.

**Remark 4.3** By introducing the model information such as $J$, $J_o$, etc., the established deep RNN consists of three class of nodes, namely $\dot{\theta}$, $\lambda_1$ and $\lambda_2$, and the total number of nodes is $n + m + ab$. Comparing to traditional neural networks in [19], the complexity of neural networks is greatly reduced.

### *4.3.2 Stability Analysis*

In this part, we offer stability analysis of the obstacle avoidance method based on deep RNN based. First of all, some basic Lemmas are given as below.

**Definition 4.1** A continuously differentiable function $F(\bullet)$ is said to be monotone, if $\nabla F + \nabla F^{\mathrm{T}}$ is positive semi-definite, where $\nabla F$ is the gradient of $F(\bullet)$.

**Lemma 4.1** *A dynamic neural network is said to converge to the equilibrium point if it satisfies*

$$\kappa \dot{x} = -\dot{x} + P_S(x - \rho F(x)), \tag{4.16}$$

*where $\kappa > 0$ and $\rho > 0$ are constant parameters, and $P_S = argmin_{y \in S} ||y - x||$ is a projection operator to closed set $S$.*

**Lemma 4.2** *[37] Let $V : [0, \infty) \times B_d \to \mathbb{R}$ be a $C^1$ function, $\alpha_1$, $\alpha_2$ be class-K functions defined on $[0, d)$ which satisfy $\alpha_1(||x||) \le V(t, x) \le \alpha_2(||x||)$, $\forall (t, x) \in [0, d) \times B_d$, then $x = 0$ is a uniformly asymptotically stable equilibrium point if there exist some class-K function g defined on $[0, d)$, to make the following inequality hold*

$$\frac{\partial V}{\partial t} + \frac{\partial V}{\partial x} f(t, x) \le -\alpha(||x||), \forall (t, x) \in [0, \infty) \times B_d, \tag{4.17}$$

About the stability of the closed-loop system, we offer the following theorem.

**Theorem 4.1** *Given the obstacle avoidance problem for a redundant manipulator in kinematic control tasks, the proposed deep recurrent neural network is stable and will globally converge to the optimal solution of (4.10).*

***Proof*** The stability analysis consists of two parts: firstly, we will show that the equilibrium of the deep RNN is exactly the optimal solution of the control objective described in (4.11), which prove that the output of deep RNN will realize obstacle avoidance while tracking a given trajectory, and then we will prove that the deep recurrent neural network is stable in sense of Lyapunov.

*Part I.* As to the deep recurrent neural network (4.15), let $(\dot{\theta}^*, \lambda_1^*, \lambda_2^*)$ be the equilibrium of the deep RNN, then $(\dot{\theta}^*, \lambda_1^*, \lambda_2^*)$ satisfies

$$-\dot{\theta}^* + P_\Omega(J^{\mathrm{T}}\lambda_1^* - J_o^{\mathrm{T}}\lambda_2^*) = 0, \tag{4.18a}$$

$$\dot{x}_{\mathrm{d}} + k(x_{\mathrm{d}} - x) - J(\theta)\dot{\theta}^* = 0, \tag{4.18b}$$

$$-\lambda_2^* + (\lambda_2^* + J_o\dot{\theta}^* - B)^+ = 0, \tag{4.18c}$$

with $\dot{\theta}^*$ be the output of deep RNN. By comparing Equation (4.18) and (4.13), we can readily obtain that they are identical, i.e., the equilibrium point satisfies the KKT condition of (4.10).

Then we will show that the equilibrium point (output of the proposed deep RNN) is capable of dealing with kinematic tracking as well as obstacle avoidance problem. Define a Lyapunov function $V$ about the tracking error $e = x_\mathrm{d} - x$ as $V = e^\mathrm{T}e/2$, by differentiating $V$ with respect to time and combining (4.11b), we have

$$\begin{aligned}\dot{V} &= e^\mathrm{T}\dot{e} = e^\mathrm{T}(\dot{x}_\mathrm{d} - J(\theta)\dot{\theta}^*)\\ &= -ke^\mathrm{T}e \leq 0,\end{aligned} \tag{4.19}$$

in which the dynamic equation (4.18b) is also used. It can readily obtained that the tracking error would eventually converge to zero.

It is notable that the dynamic equation (4.18c) satisfies

$$\lambda_2^* + J_o\dot{\theta}^* - B - (\lambda_2^* + J_o\dot{\theta}^* - B)^+ = J_o\dot{\theta}^* - B. \tag{4.20}$$

According to the property of projection operator $(\bullet)^+$, $y - (y)^+ \leq 0$ holds for any $y$, then we have $J_o\dot{\theta}^* - B \leq 0$, together with (4.5), the inequality (4.5) is satisfied. Notable that (4.5) can be rewritten as

$$\dot{D} \geq -\mathrm{sgn}(D)g(|D|). \tag{4.21}$$

As to (4.21), we first consider the situation when equality holds. Since $g(|D|)$ is a function belonging to class K, it can be easily obtained that $D = 0$ is the only equilibrium of $\dot{D} = -\mathrm{sgn}(D)g(|D|)$. Define a Lyapunov function as $V_2(t, D) = D^2/2$, and select two functions as $\alpha_1(|D|) = \alpha_2(|D|) = D^2/2$. It is obvious that $\alpha_1(|D|) = \alpha_2(|D|)$ belongs to class-K, and the following inequality will always hold

$$\alpha_1(|D|) \leq V_2(t, D) \leq \alpha_2(|D|). \tag{4.22}$$

Taking the time derivative of $V_2(t, D)$, we have

$$\frac{\partial V_2}{\partial \mathrm{t}} + \frac{\partial V}{\partial D}\dot{D} = -|D|g(|D|) \leq 0. \tag{4.23}$$

According to Lemma 4.2, the equilibrium $x = 0$ is uniformly asymptotically stable. Then we arrive at the conclusion that if the equality $\mathrm{d}(|O_jA_i|)/\mathrm{d}t = -\mathrm{sgn}(D)g(|D|)$ holds, $|D| = 0$ will be guaranteed, i.e., $|O_jA_i| - d$ for all $i = 1\cdots a, = 1\cdots b$. Based on comparison principle, we can readily obtain that $|O_jA_i| \geq d$ when $\mathrm{d}(|O_jA_i|)/\mathrm{d}t \geq -\mathrm{sgn}(D)g(|D|)$.

*Part II.* Then we will show the stability of the deep RNN (4.15). Let $\xi = [\dot{\theta}^\mathrm{T}, \lambda_1^\mathrm{T}, \lambda_2^\mathrm{T}]^\mathrm{T}$ be the a concatenated vector of state variables of the proposed deep RNN, then (4.15) can be rewritten as

$$\varepsilon\dot{\xi} = -\xi + P_{\bar{\Omega}}[\xi - F(\xi)], \tag{4.24}$$

where $P_S(\bullet)$ is a projection operator to a set $S$, and $F(\xi) = [F_1(\xi), F_2(\xi), F_3(\xi)]^{\mathrm{T}} \in \mathbb{R}^{n+m+ab}$, in which

$$\begin{bmatrix} F_1 \\ F_2 \\ F_3 \end{bmatrix} = \begin{bmatrix} \dot{\theta} - J^{\mathrm{T}}\lambda_1 + J_o^{\mathrm{T}}\lambda_2 \\ J\dot{\theta} - \dot{x}_{\mathrm{d}} - k(x_{\mathrm{d}} - x) \\ -J_o\dot{\theta}^* - B \end{bmatrix}.$$

Let $\nabla F = \partial F/\partial\xi$, we have

$$\nabla F(\xi) = \begin{bmatrix} I & -J^{\mathrm{T}} & J_o^{\mathrm{T}} \\ J & 0 & 0 \\ -J_o^{\mathrm{T}} & 0 & 0 \end{bmatrix}. \tag{4.25}$$

According to the definition of monotone function, we can readily obtain that $F(\xi)$ is monotone. From the description of (4.24), the projection operator $P_S$ can be formulated as $P_S = [P_\Omega; P_R; P_\Lambda]$, in which $P_\Omega$ is defined in (4.13), $P_R$ can be regarded as a projection operator of $\lambda_1$ to $R$, with the upper and lower bounds being $\pm\infty$, and $P_\Lambda = (\bullet)^+$ is a special projection operator to closed set $\mathbb{R}_+^{ab}$. Therefore, $P_S$ is a projection operator to closed set $[\Omega; \mathbb{R}^m; \mathbb{R}_+^{ab}]$. Based on Lemma 4.1, the proposed neural network (4.15) is stable and will globally converge to the optimal solution of (4.10). The proof is completed. □

## 4.4 Numerical Results

In this chapter, the proposed deep RNN based controller is applied on a planar 4-DOF robot. Firstly, a basic case where the obstacle is described as a single point is discussed, and then the controller is expanded to multiple obstacles and dynamic ones. Comparisons with existing methods are also listed to indicate the superiority of the deep RNN based scheme.

### *4.4.1 Simulation Setup*

The physical structure of the 4-link planar robot to be simulated is shown in Fig. 4.2, in which the critical points of the robot are also marked. As shown in Fig. 4.2, critical points $A_2$, $A_4$, $A_6$ are selected at the joint centers, and $A_1$, $A_3$, $A_5$, $A_7$ are selected at the center of robot links. It is notable that $A_i$ and the Jacobian matrix $J_{oi}$ are essential in the proposed control scheme. Based on the above description of $A_i$, the D-H parameters of $A_1$ is $a_1 = 0.15$, $a_2 = a_3 = 0$, $\alpha_1 = \alpha_2 = \alpha_3 = 0$, $d_1 = d_2 = d_3 = 0$, then both the position and Jacobian matrix $J_{a1}$ of $A_1$ can be calculated readily. Based on the definition in Eq. (4.8), $J_{o1}$ can be obtained. $A_i$ and $J_{oi}$ can

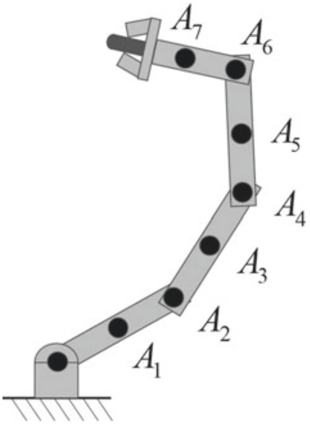


**Fig. 4.2** The planar robot to be simulated in this chapter

be calculated similarly. The control parameters are set as $\varepsilon = 0.001$, $\alpha = 8$, $k = 8$. As to the physical constraints, the limits of joint angles and velocities are selected as $\theta_i^- = -3$rad, $\theta_i^+ = 3$rad, $\dot{\theta}_i^- = -1$rad/s, $\dot{\theta}_i^+ = 1$rad/s for $i = 1 \ldots 4$. The safety distance $d$ is set to be 0.1 m.

### *4.4.2 Single Obstacle Avoidance*

In this simulation, the obstacle is assumed to be centered at $[-0.1, 0.2]^{\mathrm{T}}$m, the desired path is set as $x_{\mathrm{d}} = [0.4 + 0.1cos(0.5t), 0.4 + 0.1sin(0.5t)]^{\mathrm{T}}$m, and the initial joint angles are set to be $\theta_0 = [\pi/2, -\pi/3, -\pi/4, 0]^{\mathrm{T}}$rad. The class-K function is selected as $G(|D|) = K_1|D|$ with $K_1 = 200$. In order to show the effectiveness of the proposed obstacle avoidance method, contrast simulations with and without inequality constraint (4.10e) are conducted. Simulation results are shown in Fig. 4.3. When ignoring the obstacle, the end-effector trajectories and the corresponding incremental configurations are shown in Fig. 4.3a, although the robot achieves task space tracking to $x_{\mathrm{d}}$, obviously the first link of the robot would collide with the obstacle. After introducing obstacle avoidance scheme, the robot moves other joints rather than the first joint, and then avoids the obstacle effectively (Fig. 4.3b). Simultaneously, the tracking errors when tracking the given circle are shown in Fig. 4.3c. From the initial state, the end-effector moves towards the circle quickly and smoothly, after that, the tracking error in stable state keeps less than $1 \times 10^{-4}$m, showing that the robot could achieve kinematic control as well as obstacle avoidance tasks. To show more details of the proposed deep RNN based method, some important process data is given. As the obstacle is close to the first joint, critical points $A_1$ and $A_2$ are more likely to collide with the obstacle, therefore, as a result, the distances between the obstacle $O_1$ and $A_1$, $A_2$ are shown in Fig. 4.3d, from $t = 2$s to $t = 6.5$s, $||A_1O_1||$ remains at the minimum value $d = 0.1$, that is to say, using the proposed obstacle

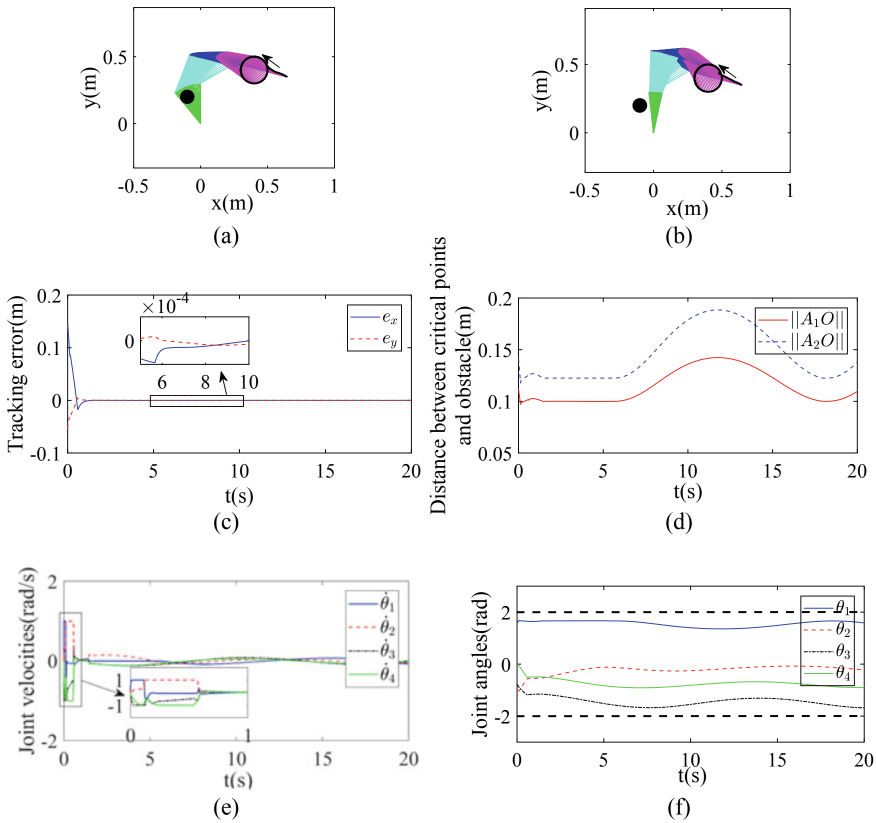


**Fig. 4.3** Numerical results of single obstacle avoidance. **a** is the motion trajectories when ignoring obstacle avoidance scheme, **b** is the motion trajectories when considering obstacle avoidance scheme, **c** is the profile of tracking errors when considering obstacle avoidance scheme, **d** is the profile of distances between critical points and obstacle, **e** is the profile of joint velocities, **f** is the profile of joint angles

avoidance method, the robot maintains minimum distance from the obstacle. The profile of joint velocities are shown in Fig. 4.3e, at the beginning of simulation, the robot moves at maximum speed, which leads to the fast convergence of tracking errors. The curve of joint angles change over time is shown in Fig. 4.3f.

### *4.4.3 Discussion on Class-K Functions*

In this part, we will discuss the influence of different class-K functions in the avoidance scheme (4.5). Four functions are selected as $G_1(|D|) = K|D|^2$, $G_2(|D|) = K|D|, G_3(|D|) = K\tanh(5|D|), G_4(|D|) = K\tanh(10|D|)$, Fig. 4.4a shows the com-

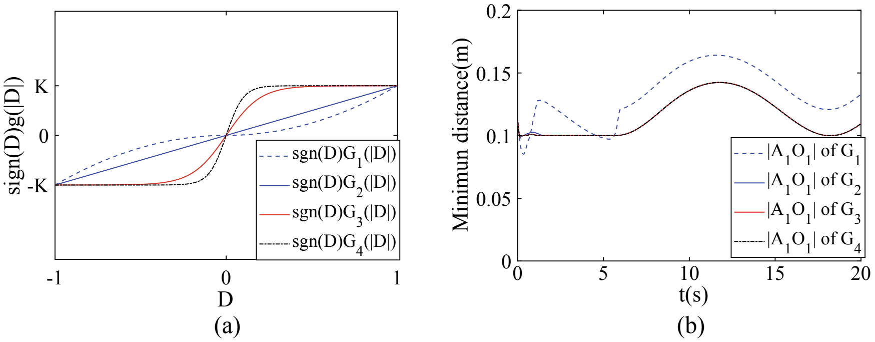


**Fig. 4.4** Discussions on different obstacle avoidance functions. **a** is the comparative curves of different obstacle avoidance functions. **b** is the profile of minimum distance of the robot and obstacle using different obstacle avoidance functions

parative curves the these functions. Other simulation settings are the same as the previous one. Simulation results are shown in Fig. 4.4b. When selecting the same positive gain $K$, the minimum distance is about 0.08 m, which shows the robot can avoid colliding with the obstacle using the avoidance scheme (4.5). The close-up graph of the tracking error is also shown, it is remarkable that the minimum distance deceases, as the gradient of the class-K function increases near zero. Therefore, one conclusion can be drawn that the function can be more similar with sign function, to achieve better obstacle avoidance.

### *4.4.4 Multiple Obstacles Avoidance*

In this part, we consider the case where there are two obstacles in the workspace. The obstacles are set at $[0.1, 0.25]^{\mathrm{T}}$m and $[0, 0.4]^{\mathrm{T}}$m, respectively. Simulation results are shown in Fig. 4.5. The desired path is defined as $x_{\mathrm{d}} = [0.45 + 0.1cos(0.5t), 0.4 + 0.1sin(0.5t)]^{\mathrm{T}}$. The initial joint angle of the robot is selected as $\theta_0 = [1.5, -1 - 1, 0]^{\mathrm{T}}$. To further show the effectiveness of the proposed obstacle avoidance strategy 4.5, $g|D|$ is selected as $g|D| = K_1/(1 + e^{-|D|}) - K_1/2$ with $K_1 = 200$. When $\lambda_2$ is set to 0, as shown in Fig. 4.5a, the inequality constraint (4.11d) will not work, in other words, only kinematic tracking problem is considered rather than obstacle avoidance, in this case, the robot would collide with the obstacles. After introducing online training of $\lambda_2$, the simulation results are given in Fig. 4.5b–h. The tracking errors are shown in Fig. 4.5c, with the transient time being about 4s, and steady state error less than $1 \times 10^{-3}$m. Correspondingly, the robot moves fast in the transient stage, ensuring the quick convergence of the tracking errors. It is remarkable that the distances between the critical points and obstacle points are kept larger than 0.1m at all times, showing the effectiveness of the proposed method. At $t = 14$s,

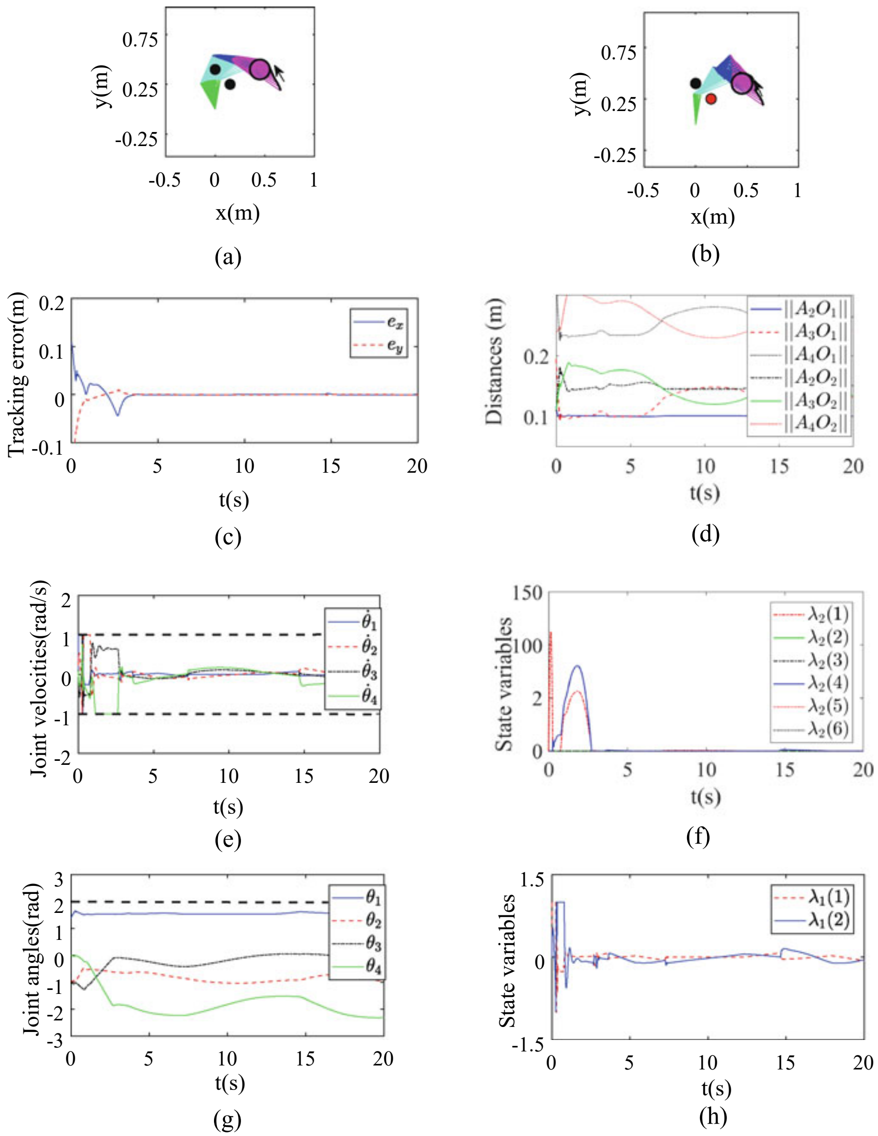


**Fig. 4.5** Numerical results of multiple obstacle avoidance. **a** is the motion trajectories when ignoring obstacle avoidance scheme. **b** is the motion trajectories when considering obstacle avoidance scheme. **c** is the profile of tracking errors when considering obstacle avoidance scheme. **d** is the profile of distances between critical points and obstacles. **e** is the profile of joint velocities. **f** is the profile of $\lambda_2$. **g** is the profile of joint angles. **h** is the profile of $\lambda_1$

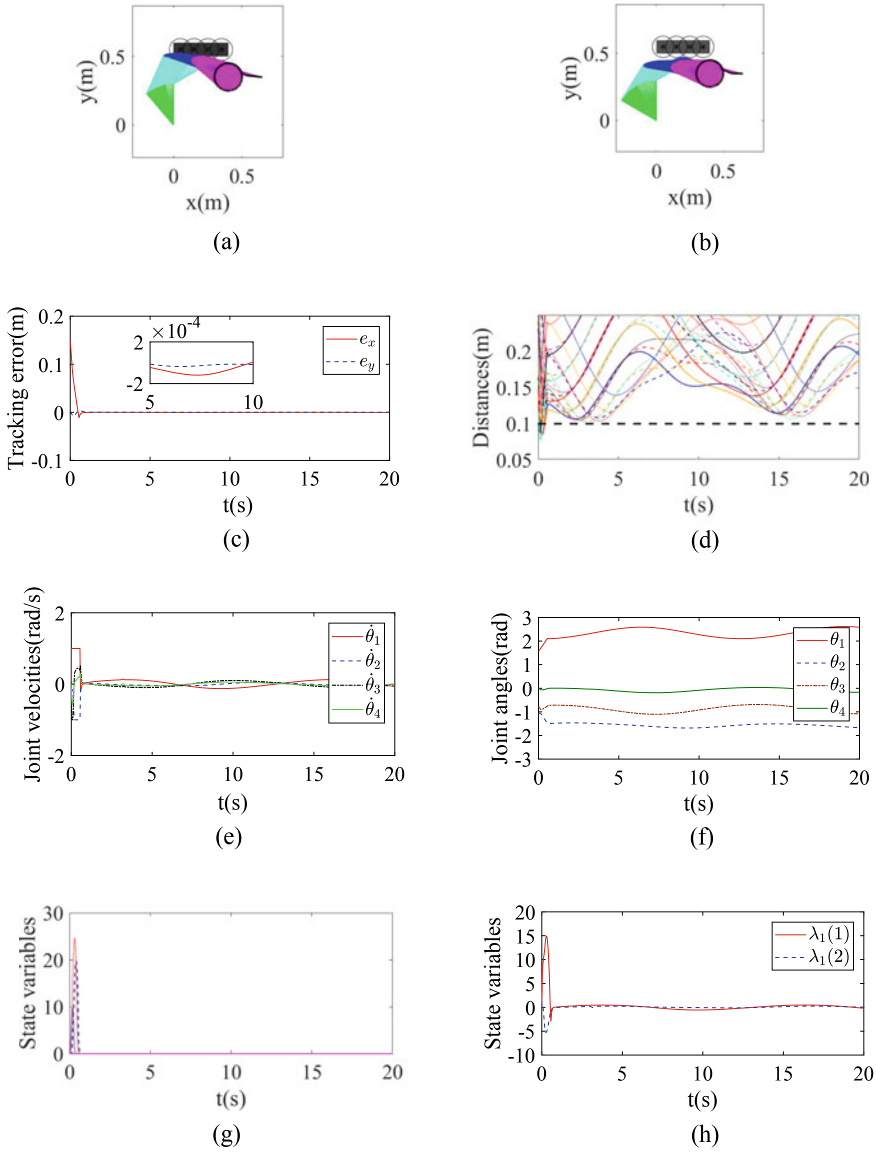
from Fig. 4.5d and g, when the distance between $A_3$ and $O_1$ is close to 0.1m, the corresponding dual variable $\lambda_2$ becomes positive, making the inequality constraint (4.11d) hold, and the boundary between the robot and obstacle is thus guaranteed. After $t = 18$s, $||A_3O_1||$ becomes greater, then $\lambda_2$ converges to aero. Notable that although $\lambda_1$ and $\lambda_2$ do not converge to certain values, the dynamic change of $\lambda_1$ and $\lambda_2$ ensures the regulation of the proposed deep RNN.

### 4.4.5 Enveloping Shape Obstacles

In this part, we consider obstacles of general significance. Suppose that there is a rectangular obstacle in the workspace, with the vertices being $[0, 0.5]^{\mathrm{T}}$, $[0.4, 0.5]^{\mathrm{T}}$, $[0.4, 0.6]^{\mathrm{T}}$ and $[0.5, 0.6]^{\mathrm{T}}$, respectively. By selecting the safety distance $d = 0.1$m, and obstacle points as $O_1 = [0.05, 0.55]^{\mathrm{T}}$, $O_2 = [0.15, 0.55]^{\mathrm{T}}$, $O_3 = [0.25, 0.55]^{\mathrm{T}}$ and $O_4 = [0.35, 0.55]^{\mathrm{T}}$. It can be readily obtained that the rectangular obstacle is totally within the envelope defined by $O_i$ and $d$. The incremental configurations when tracking the path while avoiding the obstacle are shown in Fig. 4.6b, in which a local amplification diagram is also given, showing that the critical points $A_3$ is capable of avoiding $O_2$ and $O_3$. It is worth noting that by selecting proper point group and safety distance, the obstacle can be described by the envelope shape effectively. Figure 4.6c, h also give important process data of the system under the proposed controller, including tracking errors, joint angles, angular velocities, and state variables of deep RNNs. We can observe that the physical constraints as well as kinematic control task are realized using the controller.

### 4.4.6 Comparisons

To illustrate the priority of the proposed scheme, a group of comparisons are carried out. As shown in Table 4.1, all the controllers in [12, 16, 34, 35] achieve the avoidance of obstacles. Comparing to APF method in [12, 16] and JP based method in [12, 16], the proposed controller can realize a secondary task, at the same time, we present a more general formulation of the obstacle avoidance strategy, which is helpful to gain a deeper understanding of the mechanism for avoidance of obstacles. Moreover, in this chapter, both dynamic trajectories and obstacles are considered. The comparisons above also highlight the main contributions of this paper.



(a) (b)

(c) (d)

(e) (f)

(g) (h)

**Fig. 4.6** Numerical results of enveloping shape obstacles. **a** is the motion trajectories when ignoring obstacle avoidance scheme. **b** is the motion trajectories when considering obstacle avoidance scheme. **c** is the profile of tracking errors when considering obstacle avoidance scheme. **d** is the profile of distances between critical points and obstacles. **e** is the profile of joint velocities. **f** is the profile of joint angles. **g** is the profile of $\lambda_2$. **h** is the profile of $\lambda_1$

**Table 4.1** Comparisons among different obstacle avoidance controllers on manipulators

| Method | Convergence | Secondary task | Physical constraints satisfied | Dynamic obstacles | Obstacle avoidance description |
|---|---|---|---|---|---|
| This paper | Yes | Considered | Yes | Considered | Inequalities |
| [35] | Yes | Considered | Yes | * | Inequalities** |
| [34] | Yes | Considered | Yes | * | Inequalities** |
| [12] | Yes | Not considered | No | Considered | Repulsion |
| [16] | Yes | Not considered | No | * | Null space |

* In [34, 35] and [16], dynamic obstacles are not considered

** Regular escape velocity method is used, which is only a special case of 4.5

## 4.5 Summary

In this chapter, a novel obstacle avoidance strategy is proposed based on a deep recurrent neural network. The robots and obstacles are presented by sets of critical points, then the distance between the robot and obstacle can be approximately described as point-to-points distances. By understanding the nature escape velocity methods, a more general description of obstacle avoidance strategy is proposed. Using minimum-velocity-norm (MVN) scheme, the obstacle avoidance together with path tracking problem is formulated as a QP problem, in which physical limits are also considered. By introducing model information, a deep RNN with simple structure is established to solve the QP problem online. Simulation results show that the proposed method can realize the avoidance of static and dynamic obstacles.

## References

1. C. Yang, Y. Jiang, W. He, J. Na, Z. Li, B. Xu, Adaptive Parameter Estimation and Control Design for Robot Manipulators With Finite-Time Convergence. IEEE Transactions on Industrial Electronics **65**(10), 8112–8123 (2018)
2. L. Cheng, Z.G. Hou, M. Tan, Adaptive Parameter Estimation and Control Design for Robot Manipulators With Finite-Time Convergence. Automatica **45**(10), 2312–2318 (2009)
3. Y. Pan, C. Yang, L. Pan, H. Yu, Integral Sliding Mode Control: Performance, Modification, and Improvement. IEEE Transactions on Industrial Informatics **14**(7), 3087–3096 (2018)
4. Y. Zhang, Singularity-conquering Tracking Control of A Class of Chaotic Systems Using Zhang-gradient Dynamics. IET Control Theory & Applications **9**(6), 871–881 (2015)
5. J. Ren, B. Wang, M. Cai and J. Yu, "Adaptive Fast Finite-Time Consensus for Second-Order Uncertain Nonlinear Multi-Agent Systems With Unknown Dead-Zone," ?IEEE Access, vol. 8, No. 1, pp. 25557-25569, 2020
6. H. Wu, Y. Guan, J. Rojas, A latent state-based multimodal execution monitor with anomaly detection and classification for robot introspection. Applied Sciences. **9**(6), 1072 (2019). Jan

7. H. Wu, Z. Xu, W. Yan, Q. Su, S. Li, T. Cheng, X. Zhou, Incremental Learning Introspective Movement Primitives From Multimodal Unstructured Demonstrations. IEEE Access. **15**(7), 159022–36 (2019). Oct
8. H. Wu, Y. Guan, J. Rojas, Analysis of multimodal Bayesian nonparametric autoregressive hidden Markov models for process monitoring in robotic contact tasks. International Journal of Advanced Robotic Systems. **16**(2), 1729881419834840 (2019). Mar 26
9. K. Wei, B. Ren, A Method on Dynamic Path Planning for Robotic Manipulator Autonomous Obstacle Avoidance Based on an Improved RRT Algorithm. Sensors **18**(2), 571–578 (2018)
10. D. Hsu, J. Latombe, H. Kurniawati, On the Probabilistic Foundations of Probabilistic Roadmap Planning. International Journal of Robotics Research **25**(7), 627–643 (2006)
11. O. Khatib, Real-Time Obstacle Avoidance System for Manipulators and Mobile Robots. International Journal of Robotics Research **5**(1), 90–98 (1986)
12. A. Csiszar, M. Drust, T. Dietz, A. Verl, C. Brisan, Dynamic and Interactive Path Planning and Collision Avoidance for an Industrial Robot Using Artificial Potential Field Based Method. Mechatronics **1**(1), 413–421 (2011)
13. C. Tsai, J. Lee, J. Chuang, "Path planning of 3-D objects using a new workspace model," IEEE Transactions on Systems, Man, and Cybernetics. Part C (Applications and Reviews) **31**(3), 405–410 (2001)
14. T. Tsuji, Y. Tanaka, P. Morasso, V. Sanguineti, M. Kaneko, "Bio-mimetic trajectory generation of robots via artificial potential field with time base generator," IEEE Transactions on Systems, Man, and Cybernetics. Part C (Applications and Reviews) **32**(4), 426–439 (2002)
15. G. Wen, S. Ge, F. Tu, Y. Choo, Artificial Potential Based Adaptive H$\infty$ Synchronized Tracking Control for Accommodation Vessel. IEEE Transactions on Industrial Electronics **64**(7), 5640–5647 (2017)
16. T. Krzysztof, R. Joanna, Dynamically consistent Jacobian inverse for non-holonomic robotic systems. Nonlinear Dynamics **85**(1), 107–122 (2016)
17. B. Cao, G. Dodds, G. Irwin, Redundancy resolution and obstacle avoidance for cooperative industrial robots. Journal of Robotic Systems **16**(7), 405–417 (1999)
18. S. Ali, A. Moosavian, E. Papadopoulos, Modified transpose Jacobian control of robotic systems. Automatica **43**(7), 1226–1233 (2007)
19. S. Jung, S. Kim, Hardware Implementation of a Real-Time Neural Network Controller With a DSP and an FPGA for Nonlinear Systems. IEEE Transactions on Industrial Electronics **54**(1), 265–271 (2007)
20. Y. Pan, T. Sun, Y. Liu, H. Yu, Composite Learning from Adaptive Backstepping Neural Network Control. Neural Networks **95**(1), 134–142 (2017)
21. Y. Pan, H. Yu, Biomimetic Hybrid Feedback Feedforward Neural-Network Learning Control. IEEE Transactions on Neural Networks and Learning Systems **28**(6), 1481–1487 (2017)
22. C. Fu, A. Sarabakha, E. Kayacan, C. Wagner, R. John, J. Garibaldi, Input Uncertainty Sensitivity Enhanced Non-Singleton Fuzzy Logic Controllers for Long-Term Navigation of Quadrotor UAVs. IEEE/ASME Transactions on Mechatronics **23**(2), 725–734 (2018)
23. C. Fu, R. Duan, E. Kayacan, Visual tracking with online structural similarity-based weighted multiple instance learning. Information Sciences **481**(1), 292–310 (2019)
24. J. Fontaine, A. Germain, Model-based neural networks. Computers & Chemical Engineering **25**(7), 1045–1054 (2001)
25. D. Chen, S. Li, Q. Wu, X. Luo, Super-twisting ZNN for coordinated motion control of multiple robot manipulators with external disturbances suppression. Neurocomputing **371**(1), 78–90 (2020)
26. Dechao Chen, Shuai Li, "A recurrent neural network applied to optimal motion control of mobile robots with physical constraints," Applied Soft Computing, to be published, 2019, https://doi.org/10.1016/j.asoc.2019.105880.
27. S. Li, Y. Zhang, L. Jin, Kinematic Control of Redundant Manipulators Using Neural Networks. IEEE Transactions on Neural Networks and Learning Systems **28**(10), 2243–2254 (2017)
28. Z. Xu, S. Li, X. Zhou, W. Yan, T. Cheng, D. Huang, Dynamic neural networks based kinematic control for redundant manipulators with model uncertainties. Neurocomputing **329**(1), 255–266 (2019)

29. Y. Zhang, S. Chen, S. Li, Z. Zhang, Adaptive Projection Neural Network for Kinematic Control of Redundant Manipulators With Unknown Physical Parameters. IEEE Transactions on Industrial Electronics **65**(6), 4909–4920 (2018)
30. Y. Li, S. Li, B. Hannaford, A Model based Recurrent Neural Network with Randomness for Efficient Control with Applications. IEEE Transactions on Industrial Informatics **15**(4), 2054–2063 (2019)
31. Z. Xu, S. Li, X. Zhou, W. Yan, T. Cheng, H. Dan, "Dynamic Neural Networks for Motion-Force Control of Redundant Manipulators: An Optimization Perspective", IEEE transactions on industrial electronics. Early access (2020). https://doi.org/10.1109/TIE.2020.2970635
32. X. Li, Z. Xu, S. Li, H. Wu, X, Zhou, "Cooperative Kinematic Control For Multiple Redundant Manipulators Under Partially Known Information Using Recurrent Neural Network". IEEE Access **8**(1), 40029–40038 (2020)
33. F. Cheng, T. Chen, Y. Wang and Y. Sun, "Obstacle avoidance for redundant manipulators using the compact QP method," IEEE International Conference on Robotics and Automation, pp. 41-50, 1993
34. Y. Zhang, J. Wang, "Obstacle avoidance for kinematically redundant manipulators using a dual neural network," IEEE Transactions on Systems, Man, and Cybernetics. Part B (Cybernetics) **34**(1), 752–759 (2004)
35. D. Guo, Y. Zhang, "A New Inequality-Based Obstacle-Avoidance MVN Scheme and Its Application to Redundant Robot Manipulators," IEEE Transactions on Systems, Man, and Cybernetics. Part C (Applications and Reviews) **42**(6), 1326–1340 (2012)
36. Z. Xu, S. Li, X. Zhou, T. Cheng, Dynamic Neural Networks Based Adaptive Admittance Control for Redundant Manipulators with Model Uncertainties. Neurocomputing **357**(1), 271–281 (2019)
37. Slotine and W. Li, "Applied Nonlinear Control," China Machine Press, Beijing, China, 2004

# Chapter 5
# Optimization-Based Compliant Control for Manipulators Under Dynamic Obstacle Constraints

**Abstract** The research on force control among manipulators has attracted more and more attention from a large of scholars and researcher. In this chapter, from perspective of optimization, we investigated the collision-free compliance control of redundant robot manipulators using recurrent neural network. The position-force control is constructed as an equality constraint in velocity level together with the kinematic property of robots. Both the joint angles and joint speed limitations on robots physical structure are also considered, and are described by a group of inequality constraints. To avoid collision between robots and obstacles, they are described as two set of points, the Euclidean norm of distance between robots and obstacles, greater than zero, is established as the condition of collision-free occurrence. Minimizing joint velocities as the secondary task, a time-varying QP-type problem description is given with equality and inequality constraints, then an RNN-based controller is designed to solve it. Based on theoretical analysis and simulative experiments, the effectiveness of the designed controller is validated.

## 5.1 Introduction

With development of industry society, robot manipulators are required to be more flexible and intelligent, to satisfy the increasing personalized and customized production requirement [1]. Compared to non-redundant manipulators, redundant ones show more flexibility due to its more DOFs that exceed the required number to accomplish the given task [2]. On the other hand, position control scheme on robots would show lower performance for some complicated tasks [3]. For example, the control methods that only considers position usually ignore the contact-force between robot and workpieces, with high safety challenge, resulting from the excessive system stiffness would bring the unpredictable responses [4]. Therefore, control of contact force between redundant robots and workpieces should be considered.

In the light of different robot structure and control signals, until now, a number of methods have been proposed. Imitating the muscle-tendon tissue of animal joints, compliance units such as series elastic actuators (SEA), variable stiffness actuators and so on, are introduced into the robots. In [5], Pan et al. proposed a compliance

X. Zhou et al., *AI based Robot Safe Learning and Control*,
https://doi.org/10.1007/978-981-15-5503-9_5

controller for SEA-based systems, achieving torque output control. As to the interaction between the robot and workpieces, Hogan proposes a basic idea of impedance control, in which the robot and environment usually bear as an impedance and admittance, respectively [6]. Generally speaking, the contact force and relative movement of the robot and workpieces can be described as a combination of mass-spring-damper systems. Therefore, the contact force can be controlled by designing motion commands indirectly. Another representative approach is hybrid position-force control, the controller is usually designed in the torque loop of the joint space, in which both contact forces and movement of the robot are modelled based on dynamic analysis. Then the controller can be described as a combination of control efforts which achieve position and force control respectively [7]. Similar research can be found in literature such as [8–13].

During the operation, the robot may collide with the environment because the manipulator usually needs to keep in touch with the workpiece. In addition, the working space of the robot is limited [14]. For example, in a production line with multiple manipulators, each robot is in a fixed position. In order to avoid interference, the working space of the robot is limited by hardware (fences, obstacles, [emph], etc.) or software constraints (pre planned space). In the case of human-computer cooperation, the robot shall not collide with people. Therefore, it is very important to avoid obstacles in the process of operation. In current reports, the desired trajectory is usually obtained by offline programming, which is limited by the efficiency of programming. In order to achieve real-time obstacle avoidance control, the artificial potential field method has been widely used. The basic idea is that when an obstacle repels the robot, the target acts as an attractive pole, and then the robot will be controlled to converge to the target without colliding with the obstacle [15]. In [16], a modified method is proposed, which describes the obstacles by different geometrical forms, both theoretical conduction and experimental tests validate the proposed method. Considering the local minimum problem that may caused by multi-link structures, in [17], a two minima is introduced to construct potential field, such that a dual attraction between links enables faster maneuvers comparing with traditional methods. Other improvements to artificial potential field method can be found in [18, 19]. A series of pseudo-inverse methods are constructed for redundant manipulators in [20], in which the control efforts consists of a minimum-norm particular solution and homogeneous solutions, and the collision can be avoided by calculating a escape velocity as homogeneous solutions. By understanding the limited workspace, the obstacle avoidance can be described in forms of inequalities, which opens a new way in realtime collision avoidance. In [21], the robot is regarded as the sum of several links, and the distances between the robot and obstacle is obtained by calculating distances between points and links. Then Guo [22] improves the method by modifying obstacle avoidance MVN scheme, and simulation results show that the modified control strategy can suppress the discontinuity of angular velocities effectively.

To solve the problem of robot compliance control, the controller should be designed according to the required command and system characteristics. That is to say, robots must follow constraints to achieve compliance control, while ensuring unequal constraints to avoid obstacles. Obviously, the control problem involves several constraints, including equal constraints and unequal constraints. By using the idea of constraint optimization, the control problem with multiple constraints can be well dealt with. In recent years, the application of recurrent neural network in robot control has been studied extensively, and it shows the great efficiency of real-time processing [23–27]. In those literatures, analysis in dual space and a convex projection are introduced to handle inequality constraints.

Recently, taking advantage of parallel computing, neural networks are used to solve the constraint-optimization, and have shown great efficiency in real-time processing. In [28, 29], controllers are established in joint velocity/acceleration level, to fulfill kinematic tracking problem for robot manipulators. In [30], tracking problem with model uncertainties is considered, and an adaptive RNN based controller is proposed for a 6DOF robot Jaco$_2$. Discussions on multiple robot systems, parallel manipulators, time-delay systems using RNN can be found in [30–33].

Based on the above observations, a RNN based collision-free compliance control strategy is proposed for redundant manipulators. The remainder of this chapter is organized as follows. In Chap. 2, the control objective including the position-force control as well as collision avoidance is pointed out, and then rewritten as a QP problem. In Chap. 3, the RNN based controller is proposed, and the stability of the system is also analyzed. A number of numerical experiments on a 4-DOF redundant manipulator including model uncertainties and narrow workspace are carried out in Chap. 4, to further verify effectiveness of the proposed control strategy. Chapter 5 concludes the chapter. The contributions of this chapter are summarized as below

- To the best of the author′s knowledge, there is few research on compliance control using recurrent neural networks, the study in this chapter is of great significance in enriching the theoretical frame of RNN.
- The proposed controller is capable of handling compliance control, as well as avoiding obstacles in realtime, which does make sense in industrial applications, besides, physical constraints are also guaranteed.
- Comparing to traditional neural-network-based controllers used in robotics, not only control errors but model information is considered, therefore, the proposed RNN has a simple structure, and the global convergence can be ensured.

## 5.2 Problem Formulation

### *5.2.1 Robot Kinematics and Impedance Control*

Without loss of generality, we consider series robot manipulators with redundant DOFs, and the joints are assumed as rotational joints. Let $\theta \in \mathbb{R}^n$ be the vector of joint angles, the description of the end-effector in the cartesian space is

$$x = f(\theta), \tag{5.1}$$

where $x \in \mathbb{R}^m$ is the coordination of the end-effector. In the velocity level, the forward kinematic model can be formulated as

$$\dot{x} = J(\theta)\dot{\theta}, \tag{5.2}$$

in which $J(\theta) = \partial x/\partial\theta$ is Jacobian matrix. As to redundant manipulators, $J \in \mathbb{R}^{m\times n}$, $rank(J) < n$.

In industrial applications, position control based operation mode has many limitations: due to the lack of compliance, pure kinematic control methods may cause unexpected consequences, especially when the robot is in contact with the environment. To enhance the compliance and achieve precise control of contact force, according to impedance control technology, the interaction between robot and environment can be described as a damper-spring system [34].

$$F = K_p \Delta x + K_d \mathrm{d}(\Delta x)/\mathrm{d}t, \tag{5.3}$$

where $K_p$ and $K_d$ are interaction coefficients, and $\Delta x = x - x_\mathrm{d}$ is the difference between the actual response $x$ and desired trajectory $x_\mathrm{d}$. By referring to Eqs. (5.2) and (5.3), we have

$$\dot{x} = K_d^{-1}F - K_p K_d^{-1}\Delta x + \dot{x}_\mathrm{d}. \tag{5.4}$$

When the real values of $K_p$ and $K_d$ are known, $F$ can be obtained by adjusting the velocity $\dot{x}$ of the end-effector according to Eq. (5.4).

### *5.2.2 Obstacle Avoidance Scheme*

In the process of robot force control, there is a risk of collision as the robot may contact with workpieces. Besides, robot manipulators usually work in a limited workspace restricted by fences, which are used to isolated robots from humans or other robots. This problem could be even more acute in tasks which require collaboration of multiple robots. Therefore, obstacle avoidance problem must be taken into consideration. When collision does not happens, the distance between robot and obstacles keeps positive. By describing the robot and obstacles as two separated sets, namely $S_A = \{A_1, \ldots, A_a\}$, $S_B = \{B_1, \ldots, B_b\}$, where $A_i, i = 1, \cdots, a$ and $B_j$, $j = 1, \cdots, b$ are points on the robot and obstacles, respectively. Then the sufficient and necessary condition of obstacle avoidance problem is that the intersection of A and B is an empty set. That is to say, for any point pair $A_i$ on the robot and $B_j$ on the obstacle, the distance between $A_i$ and $B_j$ is always positive, i.e., $||A_i B_j||_2^2 > 0$, for all $i = 1, \ldots, a$, $j = 1, \cdots, b$, where $|| \bullet ||_2^2$ is the Euclidean norm of vector $A_i B_j$. For sake of safety, let $d > 0$ be a proper value describing the minimum distance between robot and obstacles, the collision can be avoided b ensuring $||A_i B_j||_2^2 \geq d$.

**Remark 5.1** In fact, both $S_A$ and $S_B$ consist of infinite points. However, by evenly selecting representative points from the robot link and obstacles, $S_A$ and $S_B$ can be simplified significantly. Besides, the safety distance $d$ can be appropriately increased. Despite that this treatment will sacrifice some workspace of the robot (the inequality $||A_i B_j||_2^2 \geq d$ would consider some areas that collisions do not happen, due to a bigger $d$ is considered), this sacrifice is meaningful: the number of inequality constraints can be reduced greatly, which is helpful for constraint description and solution.

In real applications, the key points of the robot manipulator is easy to select. Cylindrical envelopes are usually used to describe the robotic links, then the key points can be selected on the axes of the cylinders uniformly, and the distance between those points can be defined the same as the radius of the cylinder. As to the obstacles with irregular shapes, the key points can be selected based on image processing techniques, such as edge detection, corrosion, etc.

### 5.2.3 Problem Reformulation in QP Type

From the above description, the purpose of this chapter is to build a collision-free force controller for redundant manipulators, to achieve precise force control along a predefined trajectory, in the sense that $F \to F_\mathrm{d}$, $x \to x_\mathrm{d}$, and $||A_i B_j||_2^2 \geq d$ for all $i = 1, \cdots, a$, $j = 1, \ldots, b$.

As to a redundant manipulator, there exist redundant DOFs, which can be used to enhance the flexibility of the robot. When the robot gets close to the obstacles, the robot must avoid the obstacle without affecting the contact force and tracking errors. In addition, when there is no risk of collision, the robot may work in an economic way, by minimizing the joint velocities, energy consumption can be reduced effectively. Therefore, by defining an objective function as $||\dot{\theta}||_2^2$, the control objective can be summarized as

$$\begin{aligned} \min \quad & ||\dot{\theta}||_2^2, && (5.5a)\\ s.t. \quad & x = x_\mathrm{d}, && (5.5b)\\ & F = F_\mathrm{d}, && (5.5c)\\ & ||A_i B_j||_2^2 \geq d, && (5.5d) \end{aligned}$$

where $||\dot{\theta}||_2^2$ is the Euclidean norm of $\dot{\theta}$. It is noteworthy that in actual industrial applications, the robot is also limited by its own physical structures. For instance, the joint angles are limited in a fixed range, and the upper/lower bounds of joint velocities are also constrained due to actuator saturation. By combing Eq. (5.4), the control objective rewrites as

$$\begin{aligned}
\min \quad & ||\dot{\theta}||_2^2, && (5.6a)\\
s.t. \quad & J\dot{\theta} = K_d^{-1}F - K_pK_d^{-1}\Delta x + \dot{x}_{\mathrm{d}}, && (5.6b)\\
& ||A_iB_j||_2^2 \geq d, && (5.6c)\\
& \theta^- \leq \theta \leq \theta^+, && (5.6d)\\
& \dot{\theta}^- \leq \dot{\theta} \leq \dot{\theta}^+, && (5.6e)
\end{aligned}$$

with $\theta^-$, $\theta^+$, $\dot{\theta}^-$, $\dot{\theta}^+$ being the upper/lower bounds of joint angles and velocities, respectively. However, the optimization problem is described in different levels, i.e., joint speed level or joint angle level, which remains challenging to solve Eq. (5.6) directly. Therefore, we will rewrite this formula in velocity level. As to the key points $A_i$ on the robot, let $x_{Ai}$ be the coordination of $A_i$ in the cartesian space, both $x_{Ai}$ and $\dot{x}_{Ai}$ are available:

$$\begin{aligned}
x_{Ai} &= f_{Ai}(\theta), && (5.7a)\\
\dot{x}_{Ai} &= J_{Ai}\dot{\theta}, && (5.7b)
\end{aligned}$$

where $f_{Ai}(\bullet)$ is the forward kinematics of point $A_i$, and $J_{Ai}$ is the corresponding Jacobian matrix from $A_i$ to joint space. Let us consider the following equality

$$\frac{\mathrm{d}}{\mathrm{d}t}(||A_iB_j||_2^2) = k(||A_iB_j||_2^2 - d), \tag{5.8}$$

in which $k$ is a positive constant. It is obviously that the equilibrium point of Eq. (5.8) is $||A_iB_j||_2^2 = d$. By letting $\frac{\mathrm{d}}{\mathrm{d}t}(||A_iB_j||_2^2) \geq 0$, the inequality (5.5d) can be readily guaranteed. Taking the time-derivative of $||A_iB_j||_2^2$ yields

$$\begin{aligned}
\frac{\mathrm{d}}{\mathrm{d}t}(||B_jA_i||_2^2) =& \frac{\mathrm{d}}{\mathrm{dt}}(\sqrt{(A_i - B_j)^{\mathrm{T}}(A_i - B_j)})\\
=& \frac{1}{||B_jA_i||_2^2}(A_i - B_j)^{\mathrm{T}}(\dot{A}_i - \dot{B}_j)\\
=& \overrightarrow{|B_jA_i|}^{\mathrm{T}} J_{Ai}(\theta)\dot{\theta} - \overrightarrow{|B_jA_i|}^{\mathrm{T}}\dot{B}_j,
\end{aligned} \tag{5.9}$$

where $\overrightarrow{|B_jA_i|} = (A_i - B_j)^{\mathrm{T}}/||\dot{\theta}||_2^2$ is a unit vector from $B_j$ to $A_i$, and $\dot{B}_j$ is the velocity of key point $B_j$ on the obstacles. By Eqs. (5.9) and (5.6c), the inequality description of obstacle avoidance strategy is

$$\overrightarrow{|B_jA_i|}^{\mathrm{T}} J_{Ai}(\theta)\dot{\theta} \geq k(||A_iB_j||_2^2 - d) + \overrightarrow{|B_jA_i|}^{\mathrm{T}}\dot{B}_j, \tag{5.10}$$

**Remark 5.2** In this part, we have shown the basic idea of obstacle avoidance scheme in velocity level, whose equilibrium point is described in Eq. (5.8). It is notable that the right-hand side of Eq. (5.8) is only a common form to realize obstacle avoidance.

Generally speaking, the right-hand side of Eq. (5.8) may have different forms, such as $k(||A_iB_j||_2^2-d)$, $k(||A_iB_j||_2^2-d)^3$, *etc.* From Eq. (5.10), the value of the response velocity to avoid obstacles is related to the two parts, the first part is the difference between the actual and safety distance, the other part depends on the movement of the obstacles.

In terms of the physical constraints of joint angles, according to escape velocity method, inequalities (5.6d) and (5.6e) can be uniformly described as max $(\alpha(\theta^- - \theta), \dot{\theta}^-) \le \dot{\theta} \le \min(\dot{\theta}^+, \alpha(\theta^+ - \theta))$. So far, the position-force control problem together with obstacle avoidance strategy in velocity level is as below

$$\min \quad ||\dot{\theta}||_2^2, \tag{5.11a}$$

$$s.t. \quad J\dot{\theta} = K_d^{-1}F - K_pK_d^{-1}\varDelta x + \dot{x}_\mathrm{d}, \tag{5.11b}$$

$$\max(\alpha(\theta^- - \theta), \dot{\theta}^-) \le \dot{\theta} \le \min(\dot{\theta}^+, \alpha(\theta^+ - \theta)), \tag{5.11c}$$

$$J_o\dot{\theta} \le B. \tag{5.11d}$$

where (5.11c) is a rewritten inequality considering (5.6d) and (5.6e) based on escape velocity scheme , $J_o = [\underbrace{|\overrightarrow{B_1A_1}|^\mathrm{T}J_{A1}; \cdots ; |\overrightarrow{B_bA_1}|^\mathrm{T}J_{Ab}}_{b}, \cdots , \underbrace{|\overrightarrow{B_1A_a}|^\mathrm{T}J_{Aa}^\mathrm{T}; \cdots ; |\overrightarrow{B_bA_a}|^\mathrm{T}J_{Ab}}_{b}] \in \mathbb{R}^{ab\times n}$ is the concatenated form of $J_{Ai}$ considering all pairs between $A_i$ and $B_j$, $B = [B_{11}, \cdots , B_{1b}, \cdots , B_{a1}, \cdots , B_{ab}]^\mathrm{T} \in \mathbb{R}^{ab}$ is the vector of upper-bounds, in which $-k(||A_iB_j||_2^2 - d) - |\overrightarrow{B_jA_i}|^\mathrm{T}\dot{B}_j$. From the definition of $J_o$, $B$, inequality (5.11d) in equivalent to $|\overrightarrow{B_1A_1}|^\mathrm{T}J_{A1}(\theta)\dot{\theta} \ge k(||A_1B_1||_2^2 - d) + |\overrightarrow{B_1A_1}|^\mathrm{T}\dot{B}_1$,... $|\overrightarrow{B_bA_a}|^\mathrm{T}J_{Aa}(\theta)\dot{\theta} \ge k(||A_aB_b||_2^2 - d) + |\overrightarrow{B_bA_a}|^\mathrm{T}\dot{B}_b$, which is the cascading form of the inequality description (5.10) for all points pairs $A_iB_j$, i.e., if (5.11d) holds, the obstacle avoidance can be achieved. It is notable that a larger number of key points do help to describe the information of the obstacle more clearly, but it would lead to a computational burden, since the number of inequality constraints also increases. Therefore, the distance of the key points on the obstacle can be selected similar to those of the manipulator.

## 5.3 RNN Based Controller Design

In previous parts, we have transform the compliance control as well as obstacle avoidance problem into a constraint-optimization one. However, because that the QP problem described in Eq. (5.11) contains equality and inequality constraints, moreover, both Eq. (5.11b, d) are nonlinear, it is difficult to solve directly, especially in industrial applications in realtime. Based on the parallel computation ability, an RNN is established to solve Eq. (5.11) online, and the stability of the closed-loop system is also discussed.

### 5.3.1 RNN Design

In terms with the QP problem Eq. (5.11), although the analytical solution can be hardly obtained, by defining a Lagrange function as

$$L = ||\dot{\theta}||_2^2 + \lambda_1^{\mathrm{T}}(K_d^{-1}F - K_p K_d^{-1}\Delta x + \dot{x}_{\mathrm{d}} - J(\theta)\dot{\theta}) + \lambda_2^{\mathrm{T}}(J_o\dot{\theta} - B), \quad (5.12)$$

where $\lambda_1$ and $\lambda_2$ are state variables, respectively. According to Karush-Kuhn-Tucker (KKT) conditions, the inherent solution of Eq. (5.11) satisfies

$$\dot{\theta} = P_{\Omega}(\dot{\theta} - \frac{\partial L}{\partial \dot{\theta}}), \quad (5.13a)$$

$$J\dot{\theta} = K_d^{-1}F - K_p K_d^{-1}\Delta x + \dot{x}_{\mathrm{d}}, \quad (5.13b)$$

$$\lambda_2 = (\lambda_2 + J_o\dot{\theta} - B)^+, \quad (5.13c)$$

where $P_{\Omega}(x) = \mathrm{argmin}_{y\in\Omega}||y - x||$ is a projection operator of $\dot{\theta}$ to convex $\Omega$, and $\Omega = \{\dot{\theta} \in \mathbb{R}^n | \max(\alpha(\theta^- - \theta), \dot{\theta}^-) \leq \dot{\theta} \leq \min(\dot{\theta}^+, \alpha(\theta^+ - \theta))\}$. In Eq. (5.13c), the operation function $(\bullet)^+$ is defined as a mapping to the non-negative space. Equation(5.13c) can be rewritten as

$$\begin{cases} \lambda_2 > 0 & \text{if} \quad J_o\dot{\theta} = B, \\ \lambda_2 = 0 & \text{if} \quad J_o\dot{\theta} \leq B, \end{cases} \quad (5.14)$$

When $J_o\dot{\theta} \leq B$, the inequality Eq. (5.11d) holds, then $\lambda_2$ stays zero. Instead, if the inequality reaches a critical state, $\lambda_2$ becomes positive to ensure $J_o\dot{\theta} = B$. In order to obtain the inherent solution in real time, a recurrent neural network is built as follows

$$\varepsilon\ddot{\theta} = -\dot{\theta} + P_{\Omega}(\dot{\theta} - \dot{\theta}/||\dot{\theta}||_2^2 + J^{\mathrm{T}}\lambda_1 - J_o^{\mathrm{T}}\lambda_2), \quad (5.15a)$$

$$\varepsilon\dot{\lambda}_1 = K_d^{-1}F - K_p K_d^{-1}\Delta x + \dot{x}_{\mathrm{d}} - J(\theta)\dot{\theta}, \quad (5.15b)$$

$$\varepsilon\dot{\lambda}_2 = -\lambda_2 + (\lambda_2 + J_o\dot{\theta} - B)^+, \quad (5.15c)$$

with $\varepsilon$ being a positive constant scaling the convergence of Eq. (5.15).

The proposed RNN based algorithm is shown in Algorithm 5.3.1. Based on escape velocity method, the convex set of joint speed can be obtained based on the positive constant $\alpha$ and physical constraints $\theta^-, \theta^+, \dot{\theta}^-, \dot{\theta}^-$. After initializing state variables $\lambda_1$ and $\lambda_2$, the reference velocity can be obtained based on the desired command and actual responses according to Eq. (5.4) then the output of RNN (which is also the control command) can be calculated based on Eq. (5.15a), at the same time, both $\lambda_1$ and $\lambda_2$ can be updated according to Eqs. (5.15b) and (5.15c).

In real applications, the nonlinear system can be hardly approximated completely. Therefore, the approximate error is inevitable, which would influence the performance of the proposed controller. However, the approximate error is a small value

**Algorithm 4** Collision-Free position-force controller based on RNN

```
Input: Positive control gains $\alpha$, $\varepsilon$, and interaction coefficients $K_p$, $K_d$. Initial states $\dot{q}(0) = 0, q(0)$,
    desired path $x_{\mathrm{d}}(t)$, $\dot{x}_{\mathrm{d}}(t)$ and operation force $F_{\mathrm{d}}(t)$, task duration $T_e$, feedback of end effector's
    coordination $x(t)$ and contact force $F$, joint angles $\theta$, Jacobian matrix $J(\theta)$, information of the
    obstacles $B_j$ and $\dot{B}_j = 1, \cdots, b$. Location of key points $A_i$, $i = 1, \cdots, a$ on the robot, and the
    corresponding Jacobian matrices $J_{Ai}$. Physical limitations $\theta^-$, $\theta^+$, $\dot{\theta}^-$, $\dot{\theta}^+$. Safety distance $d$.
Output: To achieve position-force control without colliding with obstacles
    Initialize $\lambda_1 = 0$, $\lambda_2 = 0$.
    $x, q, F, \dot{\theta} \leftarrow$ Sensor readings
    Calculate $x_{Ai}$, $\dot{x}_{Ai}$ and $J_{Ai}$ by Equation (5.7)
    Calculate matrices $J_o$, $B$ by Equation (5.11d)
    Update upper and lower bounds of joint velocities by Equation (5.11c)
    Update output of RNN (joint velocity) by $\ddot{\theta}$ using Equation (5.15a)
    Update $\lambda_1$ by $\dot{\lambda}_1$ using Equation (5.15b)
    Update $\lambda_2$ by $\dot{\lambda}_2$ using Equation (5.15c)
Until($t > T_e$)
```

of higher order, and the influence can be suppressed based on the negative feedback scheme in the outer-loop, as shown in Eq. (5.4).

**Remark 5.3** The output dynamics of the proposed RNN is given in Eq. (5.15a), in which the projection operator $P_{\Omega}(\bullet)$ plays an important rule in handling physical constraints Eq. (5.11c), the updating of $\dot{\theta}$ depends on three parts: the first part $-\dot{\theta}/||\dot{\theta}||_2^2$ in used to optimize the objective function $||\dot{\theta}||_2^2$, and the second item $J^{\mathrm{T}}\lambda_1$ guarantees the equality constraint Eq. (5.11b) by adjusting the dual state variable $\lambda_1$ according to Eq. (5.15b), and the last item $-J_o^{\mathrm{T}}\lambda_2$ ensures the inequality constraint Eq. (5.11d). The RNN consists of three kinds of nodes, namely, $\ddot{\theta}$, $\lambda_1$ and $\lambda_2$, with the number of neurons being $n + ab + m$.

It is remarkable that the proposed controller is based on the information of system models such as $J$, $J_o$, $K_p$, etc., which is helpful to reduce computational cost. As to the constraint-optimization problem Eq. (5.11), the main challenge is to solve it in real-time, since the parameters in constraints Eqs. (5.11b) and (5.11d) are time varying. From Eq. (5.15), the control effort is obtained by calculating its updating law, which is based on the historical data and model information, i.e., it is no longer necessary to solve the solution of Eq. (5.11) as every step, and the computational cost is thus reduced. In the following section, we will also show the convergence of the RNN based controller.

In this chapter, we mainly concern the obstacle avoidance problem in force control tasks. It is notable that force control is mainly based on the idea of impedance control theory, which is similar to existing methods in [35, 36]. The main challenge of the proposed control scheme lies in the limitation of sampling ability of cameras, which are used to capture the obstacles. To handle the measurement noise or disturbances, a larger safety distance $d$ can be introduced to ensure the performance of obstacle avoidance.

### 5.3.2 Stability Analysis

**Lemma 1** (Convergence for a class of neural networks) *[37] A dynamic neural network is said to converge to its equilibrium point if it satisfies*

$$\kappa \dot{x} = -\dot{x} + P_S(x - \varrho F(x)), \tag{5.16}$$

*where $\kappa > 0$ and $\varrho > 0$ are constant parameters, and $P_S = argmin_{y \in S} ||y - x||$ is a projection operator to closed set $S$.*

**Definition 1** For a given function $F(\bullet)$ which is continuously differentiable, with its gradient defined as $\nabla F$, if $\nabla F + \nabla F^{\mathrm{T}}$ is positive semi-definite, $F(\bullet)$ is called a monotone function.

About the stability of the closed-loop system, we offer the following theorem.

**Theorem 1** *Given the collision-free position-force controller based on a recurrent neural network, the RNN will converge to the inherent solution (optimal solution) of Eq. (5.11), and the stability of the closed-loop system is also ensured.*

***Proof*** Define a vector $\xi$ as $\xi = [\dot{\theta}; \lambda_1; \lambda_2] \in \mathbb{R}^{n+m+ab}$, according to Eq.(5.15), the time derivative of $\xi$ satisfies

$$\varepsilon \dot{\xi} = -\xi + P_{\bar{\Omega}}[\xi - F(\xi)], \tag{5.17}$$

in which $\varepsilon > 0$, and $F(\xi) = [F_1(\xi), F_2(\xi), F_3(\xi)]^{\mathrm{T}}$, where $F_1 = \dot{\theta}/||\dot{\theta}||_2^2 - J^{\mathrm{T}}\lambda_1 + J_o^{\mathrm{T}}\lambda_2$, $F_2 = J\dot{\theta} - K_d^{-1}F + K_p K_d^{-1}\Delta x - \dot{x}_{\mathrm{d}}$, $F_3 = -J_o\dot{\theta} + B$. By calculating the gradient of $F(\xi)$, we have

$$\nabla F(\xi) = \begin{bmatrix} I/||\dot{\theta}||_2^2 & -J^{\mathrm{T}} & J_o^{\mathrm{T}} \\ J & 0 & 0 \\ -J_o^{\mathrm{T}} & 0 & 0 \end{bmatrix}. \tag{5.18}$$

It is obviously that $\nabla F(\xi)$ is positive definite. According to Definition 1, $F(\xi)$ is a monotone function. From the description of (5.17), the projection operator $P_S$ can be formulated as $P_S = [P_\Omega; P_R; P_\Lambda]$, in which $P_\Omega$ is defined in (5.13a), $P_R$ can be regarded as a projection operator of $\lambda_1$ to $R$, with the upper and lower bounds being $\pm\infty$, and $P_\Lambda = (\bullet)^+$ is a special projection operator to closed set $\mathbb{R}_+^{ab}$. Therefore, $P_S$ is a projection operator to closed set $[\Omega; \mathbb{R}^m; \mathbb{R}_+^{ab}]$. Based on Lemma 5.1, the proposed neural network (5.15) is stable and will globally converge to the optimal solution of (5.11).

Notable that the equality constraint (5.11b) describes the impedance controller, and the convergence can be found in [38]. Similarly, the establishment of inequality constraint enables obstacle avoidance during the whole process. The proof is completed. □

**Remark 5.4** It is remarkable that the original impedance controller described in (5.11b) bears similar with traditional methods in [39] the main contribution of the proposed controller is that the controller can not only realize the force control, but also realize the obstacle avoidance, besides, the control strategy is capable of handling inequality constraints, including joint angles and velocities.

## 5.4 Numerical Results

In this part, a series of numerical simulations are carried out to verify the effectiveness of the proposed control scheme. First, the pure force control experiment is carried out to show the effectiveness of the force controller, and then the control scheme is further verified by testing the system response after the introduction of obstacles. Then, we examine the control performance in more general cases, including model uncertainty and multiple obstacles.

### *5.4.1 Simulation Settings*

First of all, the planar robot used in the simulation is the same as the previous chapters. It is worth noting that in the force control task, the final actuator needs to keep contact with the workpiece, so it is necessary to distinguish between necessary contact and unnecessary collision. In this chapter, the proposed controller can handle this problem by properly selecting key points. As a result, the final effector is not considered critical in order to be in contact with an obstacle (or external environment). In order to avoid obstacles, the set of key points of the robot is defined as $A_1, \cdots, A_7$, in which $A_1$, $A_3$, $A_5$ and $A_7$ locate at the center of the links, and $A_2$, $A_4$ and $A_6$ are defined to be at $J2$, $J3$ and $J4$. The lower and upper bounds of joint angles and joint velocities are defined as $\theta_i^- = -3$rad, $\theta_i^+ = 3$rad, $\dot{\theta}_i^- = -1$rad/s, $\dot{\theta}_i^+ = 1$rad/s for $i = 1 \ldots 4$, respectively. The safety margin is selected as 0.01m. The coefficients describing the contact force are selected as $K_d = 50$, $K_p = 5000$. For simplicity, let $b_0 = K_d^{-1}F - K_pK_d^{-1}\Delta x + \dot{x}_\text{d}$.

### *5.4.2 Force Control Without Obstacles*

First of all, an ideal case where there is no obstacles in the workspace is considered, and the parameters $K_d$ and $K_p$ are assumed to be known. The robot is wished to offer a constant contact force on a given plane. The contact force is set to be 20 N, while the direction of contact force is aligned with the y-axis of the tool coordination system. In this example, the y-axis of is $[1, -1]^\text{T}$ in the base coordination. The pre-defined path on the contact plane is $x_\text{d} = [0.4 + 0.1cos(0.5t), 0.5 + 0.1cos(0.5t)]$. The initial

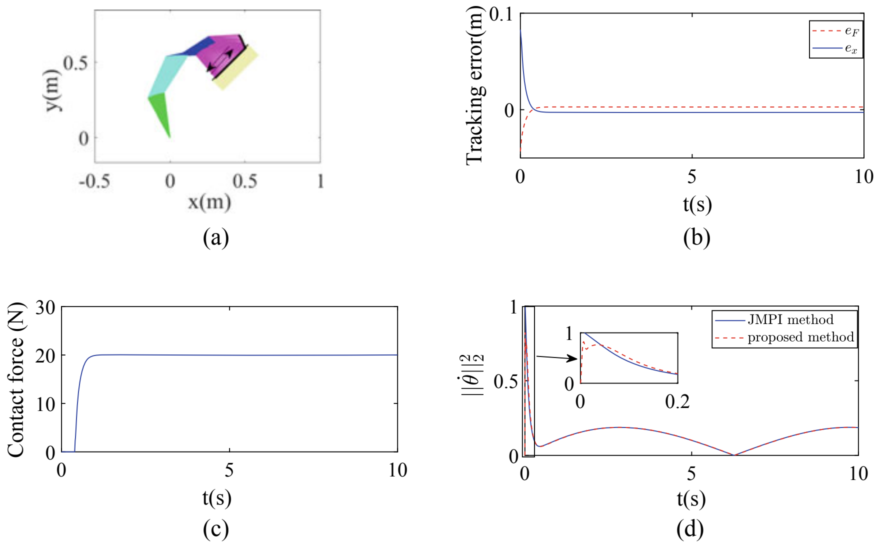


**Fig. 5.1** Numerical results of compliance control without obstacles. **a** is the robot′s tracking path and the corresponding joint configurations. **b** is the profile of position error along the free-motion direction. **c** is the profile of contact force. **d** is the profile of $||\dot{\theta}||_2^2$

state of the robot system is set as $\theta_0 = [1.57, -0.628, -0.524, -0.524]^{\mathrm{T}}$rad, $\dot{\theta}_0 = [0, 0, 0, 0]^{\mathrm{T}}$rad/s. The control gains of the proposed RNN controller are $\alpha = 8$,$\varepsilon = 0.02$, respectively. Numerical results are shown in Fig. 5.1. The tracking error along the contact plane is given in Fig. 5.1b, the transient is about 1 s. At the beginning stage, since the end-effector is not in contact with the surface, the contact force stays zero before 0.5 s. As the end-effector approaches the surface, the contact force converges to 20 N, showing the convergence of both positional and force errors. The Euclidean norm of joint velocities (which is also output of the established RNN) is shown in Fig. 5.1d, $||\dot{\theta}||$ changes periodically, with the same cycle as the expected trajectory. The time history of the end-effector′s motion trajectory and the corresponding joint configurations are shown in Fig. 5.1a, in which the red arrow indicates the direction of the contact force, and the blue arrow shows the direction of the end-effector′s free-motion. All in all, the proposed controller can achieve the position-force control precisely.

### *5.4.3 Force Control with Single Obstacles*

In this chapter, a stick obstacle is introduced into the workspace, which is defined as $x = -0.05$ m. The initial states and expected values of $x_{\mathrm{d}}$, $F_{\mathrm{d}}$ are the same as Chap. 5.4.2.

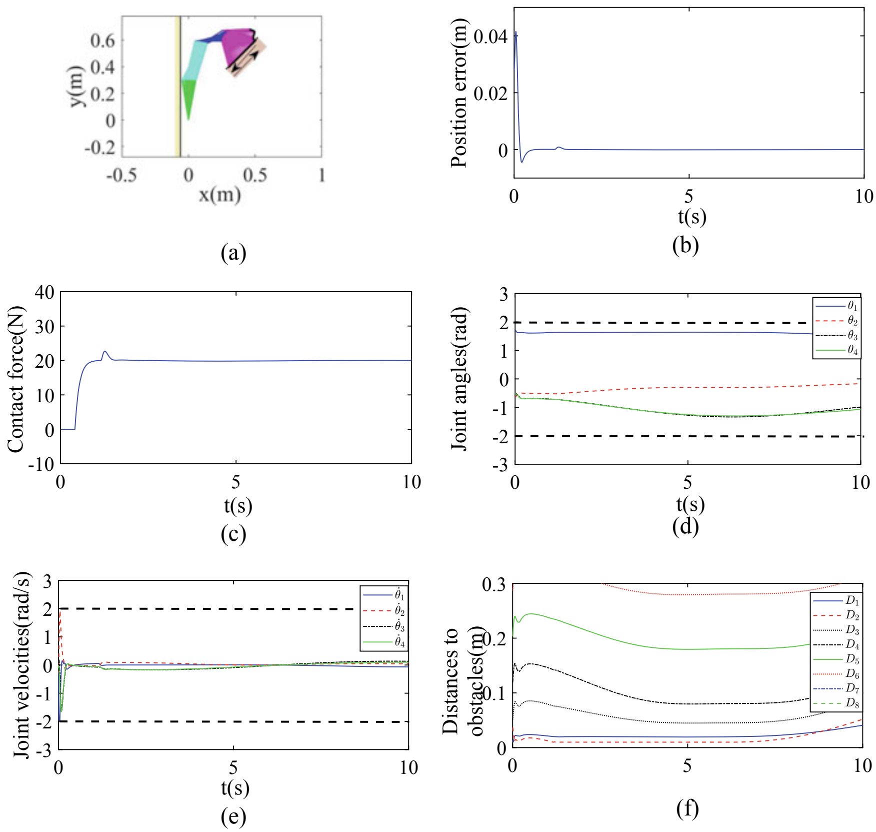
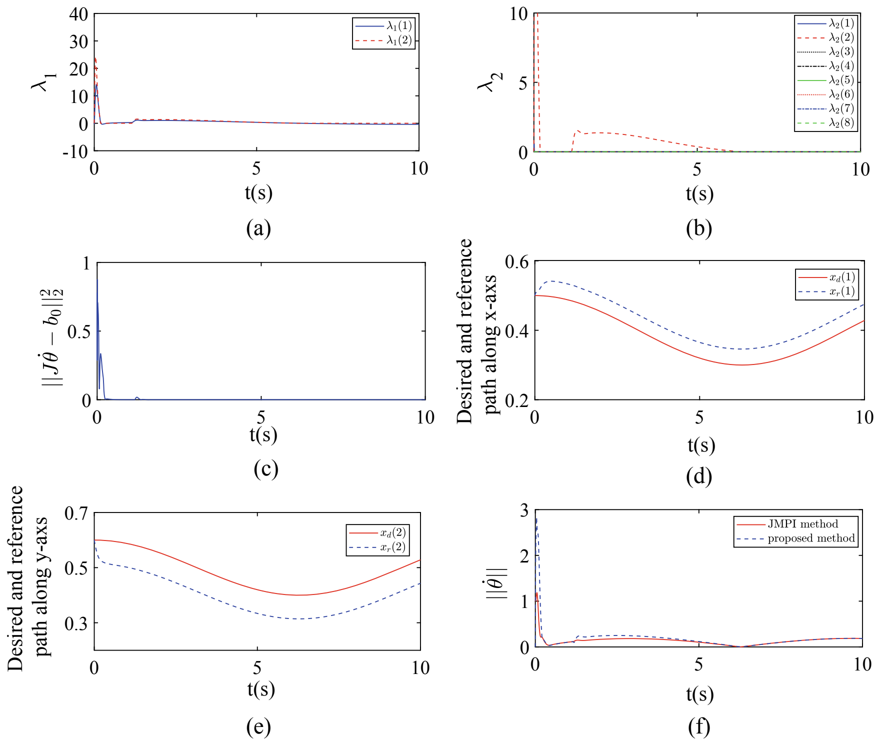


**Fig. 5.2** Control performance of the proposed controller while avoiding a wall obstacle. **a** is the robot′s tracking path and the corresponding joint configurations. **b** is the profile of position error along the free-motion direction. **c** is the profile of contact force. **d** is the profile of joint angles, **r** is the profile of joint velocities. **f** is the profile of the closest distance to the obstacle of each key points $A_i, i = 1, \cdots, 7$

**Remark 5.5** In Eq. (5.10), we have shown the basic idea of calculating the distance between the robot and obstacles, i.e., by abstracting key points form the robot and obstacles, the distances can be the robot and obstacle can be described approximately at a set of point-to-point distances. In this example, the distance can be obtained in a simpler way. However, the obstacle avoidance strategy is essentially consistent with Eq. (5.10).

Simulation results are given in Figs. 5.2 and 5.3. The output of RNN is shown in Fig. 5.2e, when simulation begins, $\dot{\theta}$ reaches its maximum value, driving the end-effector to move towards the desired path. And then the robot slows down quickly (after $t \approx 0.5$s), the robot move smoothly, as a result, the position error successfully converges to 0, and simultaneously, the contact force converges to 20 N. It is notable



(a)

(b)

(c)

(d)

(e)

(f)

**Fig. 5.3** Simulation results of the established RNN while avoiding a wall obstacle. **a** is the profile of $\lambda_1$. **b** is the profile of $\lambda_2$. **c** is the profile of $||J\dot{\theta} - b_0||_2^2$. **d** is the profiles of the desired and reference trajectory along x-axis. **e** is the profiles of the desired and reference trajectory along y-axis. **f** is the profiles of the objective function of the proposed controller and JPMI based method

that at $t = 1.2$ s, the key point $A_2$ of the robot gets close to the obstacle, as shown in Fig. 5.2f. Based on the obstacle avoidance strategy Eq. (5.15c), the state variable $\lambda_2(2)$ becomes positive, and then the output of the RNN varies with $\lambda_2$ (Fig. 5.3b). Correspondingly, an error (about $1 \times 10^{-3}$m) occurs in the positional tracking, and so as the contact force (force error is about 2N). However, the RNN converges to the new equilibrium point(since the equilibrium point would change when the inequality constraint works), and both $e_x$ and $e_f$ converges to 0. By comparing Figs. 5.2a and 5.1a, after introducing the obstacle, the robot is capable of adjusting its joint configuration to avoid the obstacle. The distances between the key points $A_1 - A_7$ to the obstacle are shown in Fig. 5.2d, a minimum value of about 0.01m is ensured during the whole process. Using impedance model, the force control problem is transferred into a kinematic control one by modifying the reference speed Eq.(5.4). Consequently, the resulting trajectory $x_r$ together with $x_{\mathrm{d}}$ are as shown in Fig. 5.3d, e. As an important index in the proposed control scheme, the norm of joint speed $||\dot{\theta}||_2^2$ is wished as small as possible. Therefore, we introduce a comparative simulation, in which the solution is obtained based on pseudo-inverse of Jacobian matrix and physical limitations are not considered. Comparative curves of the objective functions are as shown in Fig. 5.3f. The RNN based controller can optimize the objective function, it is remarkable that a difference appears at about $t = 1.2 - 5$ s, which is mainly caused by obstacle avoidance (which is not considered in JMPI based method). Since the output of RNN $\dot{\theta}$ is used to approximate the reference speed $b_0$, the approximate error $||J\dot{\theta} - b_0||_2^2$ is shown in 5.3c, demonstrating the effectiveness of the established RNN.

### 5.4.4 Force Control with Uncertain Parameters

In this example, we check the control performance of the proposed control scheme in presence of model uncertainties. Similar with previous simulations, the initial states of the robot are also $\theta_0 = [1.57, -0.628, -0.524, -0.524]^{\mathrm{T}}$rad, $\dot{\theta}_0 = [0, 0, 0, 0]^{\mathrm{T}}$rad/s. In real implementations, the interaction model is usually unknown, and the nominal values of $K_d$ and $K_p$ are not accurate. Without loss of generality, we select the nominal values of $K_d$ and $K_p$ as $\hat{K}_d = 80$, $\hat{K}_p = 4000$, respectively. In order to handle model uncertainties in the interaction coefficients, an extra node is introduced into (5.15). Then the modified RNN can be formulated as

$$\begin{aligned}
\varepsilon\ddot{\theta} &= -\dot{\theta} + P_{\Omega}(\dot{\theta} - \dot{\theta}/||\dot{\theta}||_2^2 + J^{\mathrm{T}}\lambda_1 - J_o^{\mathrm{T}}\lambda_2),\\
\varepsilon\dot{\lambda}_1 &= K_d^{-1}F - K_pK_d^{-1}\Delta x + \dot{x}_{\mathrm{d}} - J(\theta)\dot{\theta},\\
\varepsilon\dot{\lambda}_2 &= -\lambda_2 + (\lambda_2 + J_o\dot{\theta} - B)^{+},\\
\dot{\hat{W}} &= -K_{in}\eta(F_{\mathrm{d}} - F)^{\mathrm{T}},
\end{aligned}$$

in which $W = [K_p; K_d]$, $\eta = [x - x_{\mathrm{d}}; \dot{x} - \dot{x}_{\mathrm{d}}]$, and the positive coefficient $K_{in}$ scaling the updating rate is defined as $K_{in} = diag(500, 20)$. Simulation results are shown in Fig. 5.4 and Fig. 5.5. Although the exact values of $K_d$ and $K_p$ are unknown, the closed-loop system is still stable, which can be shown from the convergence of tracking error $e_x$ and contact force $F$ in Figs. 5.4a and 5.4b. The change curves of joint angles and joint velocities with respect to time are shown in Fig. 5.4c, d, in which the bounded-ness of joint angles and velocities are guaranteed. The observed interaction coefficients $\hat{K}_d$ and $\hat{K}_p$ are shown in Fig. 5.4e, indicating that both $\hat{K}_d$ and $\hat{K}_p$ converge to their real values. Figure 5.5a shows the distances between the key points and the obstacle, it is obvious that all key points keep at a safe distance from the obstacle (the closest key point is $A_2$). Euclidean norm of $b_0 - J\dot{\theta}$ is illustrated in Fig. 5.5c, despite fluctuation occurs at about $t = 1.5$s, the proposed controller could handle model uncertainties. The impedance model based reference trajectory and the original desired trajectory are shown in Fig. 5.5d and Fig. 5.5e. Although $x_r$ and $x_{\mathrm{d}}$ are different, the tracking error $e_x$ along the direction of free motion and force error $e_F$ converges to zero, as shown in Fig. 5.4a, b. The objective function $||\dot{\theta}||_2^2$ to

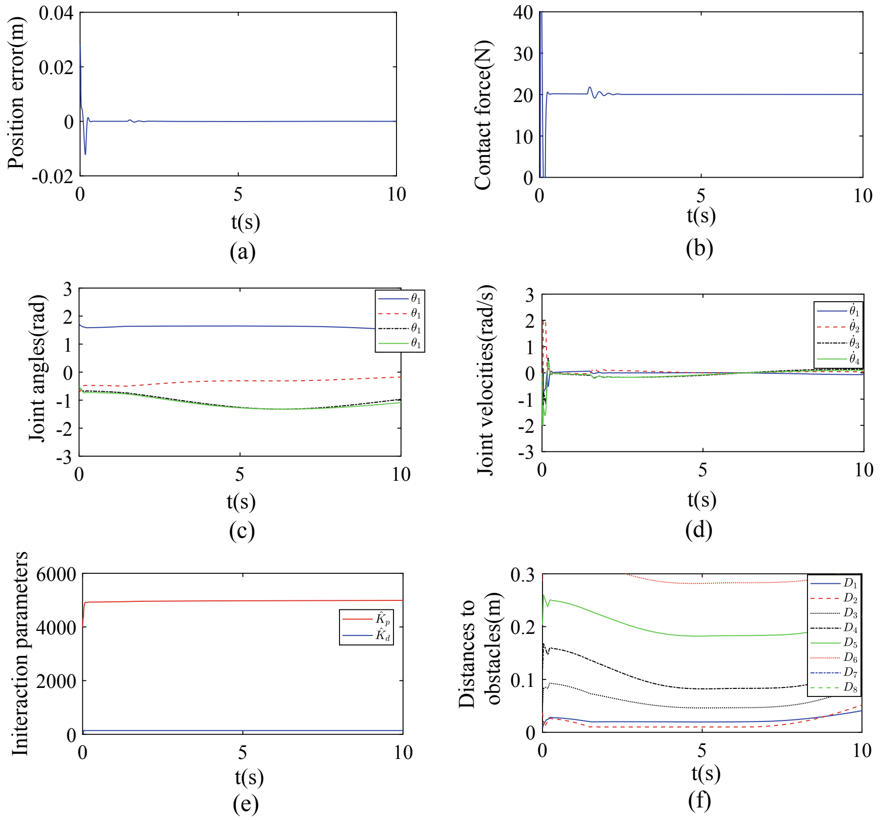


**Fig. 5.4** Control performance of the proposed controller while avoiding a wall obstacle with uncertain $K_p$ and $K_d$. **a** is the robot′s tracking path and the corresponding joint configurations. **b** is the profile of position error along the free-motion direction. **c** is the profile of contact force. **d** is the profile of joint angles. **e** is the profile of joint velocities. **f** is the profile of the closest distance to the obstacle of each key points $A_i, i = 1, \ldots, 7$

be optimized is given in Fig. 5.5f. The convergence of the established RNN is shown in Fig. 5.5c, despite the uncertain parameters, using the adaptive updating law, the established RNN is capable of learning the optimal solution. The spikes are mainly because of the change of $\lambda_2$ when obstacle avoidance scheme is activated.

### *5.4.5 Manipulation in Narrow Space*

In this part, we discuss a more general case of motion-force control task, in which the workspace is defined in a limited narrow space. The robot is limited by two parallel lines, namely, $y_1 = 0.15$ and $y_2 = -0.15$ m. Considering the safety distance, all

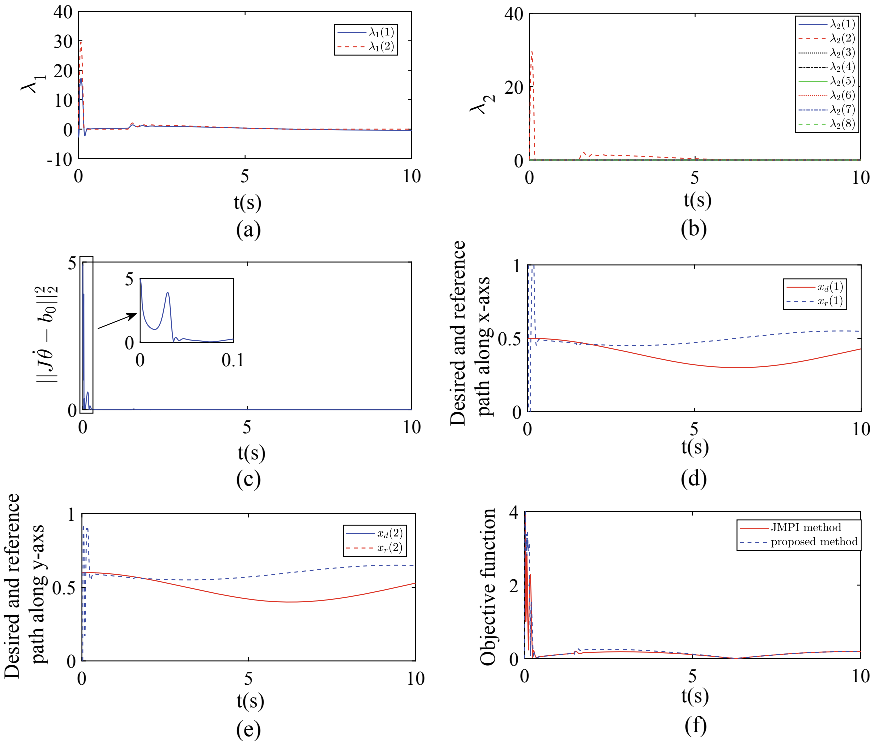


**Fig. 5.5** Simulation results of the established RNN while avoiding a wall obstacle with uncertain $K_p$ and $K_d$. **a** is the profile of $\lambda_1$. **b** is the profile of $\lambda_2$. **c** is the profile of $||J\dot{\theta}-b_0||_2^2$. **d** is the profiles of the desired and reference trajectory along x-axis. **e** is the profiles of the desired and reference trajectory along y-axis. **f** is the profiles of the objective function of the proposed controller and JPMI based method

key points except $A_8$ must satisfy the workspace description $-0.14 \leq y \leq 0.14$m. The initial joint angles are set to be $\theta_0 = [0.393, -1.05, 1.57, -0.785]^{\mathrm{T}}$ rad, and $\dot{\theta}_0 = [0, 0, 0, 0]^{\mathrm{T}}$ rad/s. The desired path is selected as $x_{\mathrm{d}} = [0.8 + 0.1cos(0.5t), -0.15]^{\mathrm{T}}$m, and the expected contact force is $F_{\mathrm{d}} = 20$ N, with the direction vector being $[0, -1]^{\mathrm{T}}$. Simulation results are given in Figs. 5.6 and 5.7. When simulation begins, the initial position error is about 0.1 m, and the converges to zero, with the transient being about 0.5 s. Simultaneously, the contact force also converges to 20 N. In Fig. 5.7a, minimum distances between the key points to $y_1$ and $y_2$ are represented by blue and red curves, respectively. The tracking trajectory and the corresponding joint configurations are shown in Fig. 5.6a. During $t = 1 - 1.5$s and $t = 6 - 13$s, point $A_2$ gets close to $y_1$, during $t = 4 - 7$s, $A_4$ is close to $y_2$. Remarkable that there exist fluctuations in positional and force errors at $t = 1$s and $t = 4$s, (i.e., when $A_2$ and $A_4$ get close to the bounds), respectively. Similar to the previous simulations, the reference trajectories are given in Fig. 5.5c, d, and the objective functions are shown

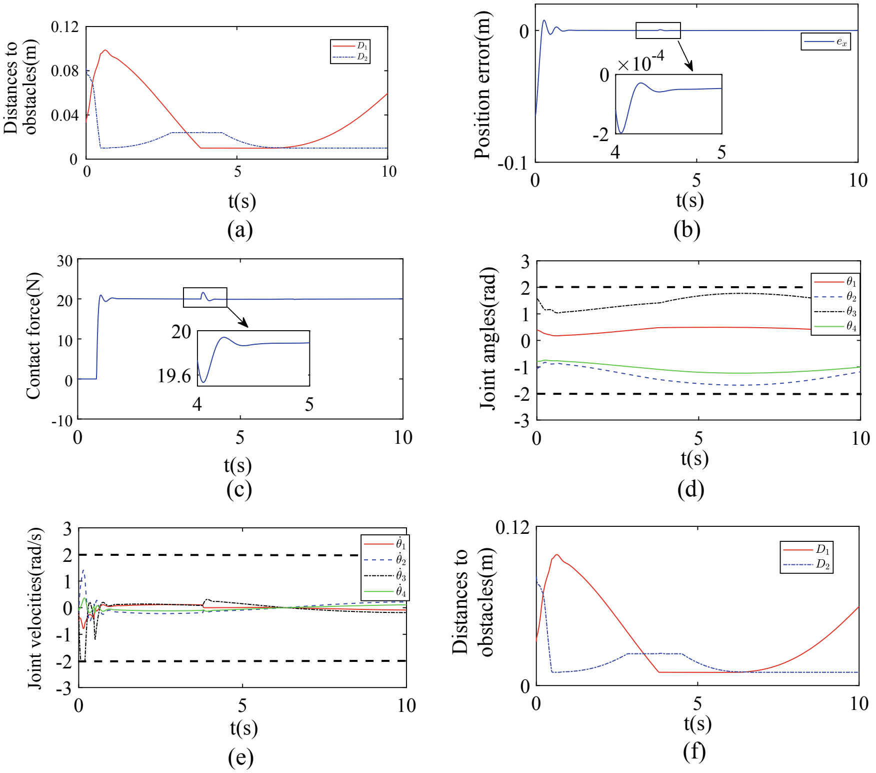


**Fig. 5.6** Control performance of the proposed controller in a narrow workspace. **a** is the robot′s tracking path and the corresponding joint configurations. **b** is the profile of position error along the free-motion direction. c) is the profile of contact force, **d** is the profile of joint angles. (e) is the profile of joint velocities. (f) is the profile of the closest distance to the obstacle of each key points $A_i, i = 1, \cdots, 7$

in Fig. 5.5e. Using the proposed RNN controller, the robot can realize both position and force control in limited narrow space.

### 5.4.6 Comparisons

In this part, comparisons among the proposed control scheme and existing methods are given to show the superiority of the RNN based strategy. The comparisons are shown in Table 5.1. In [22], an RNN based controller is designed for redundant manipulators, both obstacle avoidance and physical constraints are considered. However, the controller only focus on kinematic control problem. In [40] and [16], force control together with obstacle avoidance are taken into account, but the physical constraints are ignored. [23] develop an adaptive admittance control strategy, which is capable

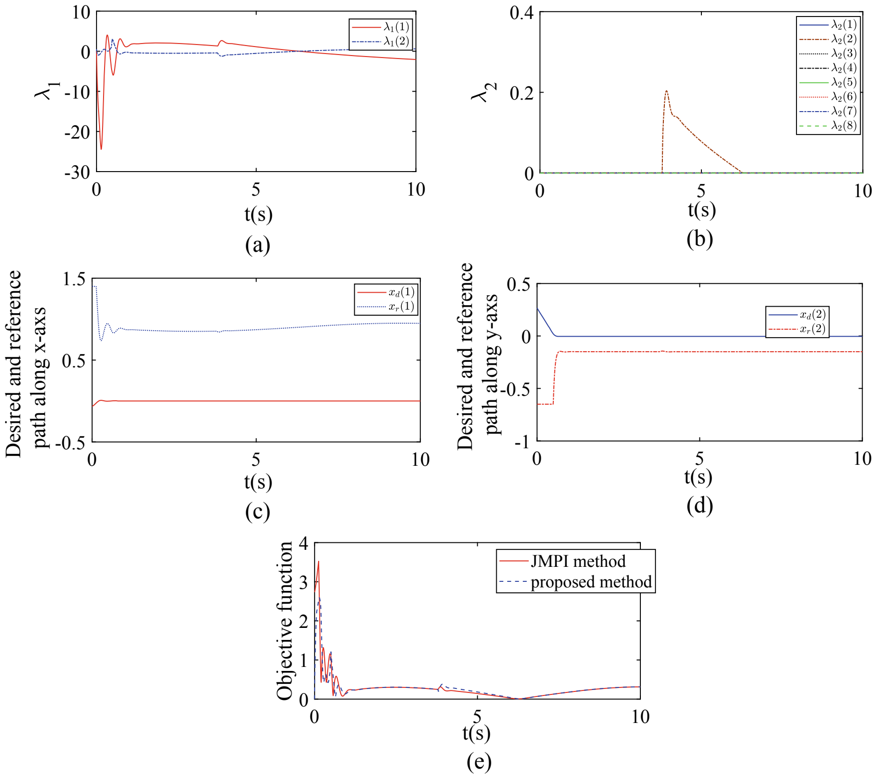


**Fig. 5.7** Simulation results of the established RNN in a narrow workspace. **a** is the profile of $\lambda_1$, **b** is the profile of $\lambda_2$, **c** is the profiles of the desired and reference trajectory along x-axis. **d** is the profiles of the desired and reference trajectory along y-axis. **e** is the profiles of the objective function of the proposed controller and JPMI based method

**Table 5.1** Comparisons among The Proposed Controller and Existing Methods

| Method | Provable convergence | Optimize in realtime | Physical limitations | Force versus kinematic control | Collison avoidance |
|---|---|---|---|---|---|
| This chapter | Yes | Yes | Considered | Force control | Yes |
| Method in [22] | Yes | Yes | Considered | kinematic control | Yes |
| Method in [40] | Yes | Yes | Ignored | Force control | Yes |
| method in [23] | Yes | Yes | Considered | Force control | No |
| method in [16] | Yes | No | Ignored | Force control | Yes |

of dealing with force control under model uncertainties, physical constraints and real-time optimization. It is remarkable that the proposed strategy focus on real-time obstacle avoidance in force control tasks, and the controller is capable of ensuring the boundedness of joint angles and velocities. At the same time, simulations have shown the potential of optimization ability of norm of joint speed.

## 5.5 Summary

This chapter constructs a new collision free compliance controller based on the concept of QP programming and neural network. Different from the existing methods, this chapter describes the control problem from the perspective of optimization, taking compliance control and conflict avoidance as equal or inequality constraints. Physical constraints, such as joint angle and speed constraints, are also considered. Before concluding this chapter, it is worth noting that it is the first RNN based compliance control method, which considers collision avoidance in real time and shows great potential in dealing with physical constraints. In this chapter, Matlab is simulated to verify the efficiency of the controller. In the future, we will check the control framework of different impedance models in the physical real simulation environment, and then consider the machine vision technology and system delay on the physical experiment platform.

## References

1. A. Gonzalez, M. Alves, G. Viana, L. Carvalho, J. Basilio, Supervisory Control-Based Navigation Architecture: A New Framework for Autonomous Robots in Industry 4.0 Environments. IEEE Transactions on Industrial Informatics **14**(4), 1732–1743 (2018)
2. Y. Zhang, Singularity-conquering Tracking Control of A Class of Chaotic Systems Using Zhang-gradient Dynamics. IET Control Theory & Applications **9**(6), 871–881 (2015)
3. W. He, S.S. Ge, Y.N. Li, E. Chew, Neural Network Control of a Rehabilitation Robot by State and Output Feedback. Journal of Intelligent & Robotic Systems **80**(1), 15–31 (2015)
4. M. Cai, Z. Xiang, Adaptive finite-time control of a class of non-triangular nonlinear systems with input saturation. Neural Computing & Applications **29**(7), 565–576 (2018)
5. Y. Pan, H. Wang, X. Li, H. Yu, Adaptive Command-Filtered Backstepping Control of Robot Arms With Compliant Actuators. IEEE Transactions on Control Systems Technology **26**(3), 1149–1156 (2018)
6. N. Hogan, Impedance control - An approach to manipulation. I - Theory. II - Implementation. III - Applications. Asme Transactions Journal of Dynamic Systems & Measurement Control B **107**(1), 304–313 (1985)
7. M.H. Raibert, J.J. Craig, Hybrid Position/Force Control of Manipulators. Asme Journal of Dynamic Systems Measurement & Control **102**(2), 126–133 (1981)
8. Y. Pan, X. Li, H. Wang, H. Yu, Continuous sliding mode control of compliant robot arms: A singularly perturbed approach. Mechatronics **52**(1), 127–134 (2018)
9. H. Wu, Y. Guan, J. Rojas, A latent state-based multimodal execution monitor with anomaly detection and classification for robot introspection. Applied Sciences. **9**(6), 1072 (2019)

10. H. Wu, Z. Xu, W. Yan, Q. Su, S. Li, T. Cheng, X. Zhou, Incremental Learning Introspective Movement Primitives From Multimodal Unstructured Demonstrations. IEEE Access. **15**(7), 159022–36 (2019)
11. H. Wu, Y. Guan, J. Rojas, Analysis of multimodal Bayesian nonparametric autoregressive hidden Markov models for process monitoring in robotic contact tasks. International Journal of Advanced Robotic Systems. **16**(2), 1729881419834840 (2019 Mar 26)
12. Zhijia Zhao, Xiuyu He, Zhigang Ren, Guilin Wen, "Boundary Adaptive Robust Control of a Flexible Riser System with Input Nonlinearities". IEEE Transactions on Systems, Man, and Cybernetics: Systems **49**(10), 1971–1980 (2019). https://doi.org/10.1109/TSMC.2018.2882734
13. Zhijia Zhao, Choon Ki Ahn, Han-Xiong Li. "Deadzone Compensation and Adaptive Vibration Control of Uncertain Spatial Flexible Riser Systems". IEEE/ASME Transactions on Mechatronics, in press, https://doi.org/10.1109/TMECH.2020.29755672020
14. O. Khatib, Real-Time Obstacle Avoidance System for Manipulators and Mobile Robots. International Journal of Robotics Research **5**(1), 90–98 (1986)
15. S. Wang, J. Zhang, J. Zhang, Artificial Potential Field Algorithm for Path Control of Unmanned Ground Vehicles Formation in Highway. Electronics Letters **54**(20), 1166–1168 (2018)
16. A. Csiszar, M. Drust, T. Dietz, A. Verl, C. Brisan, Dynamic and Interactive Path Planning and Collision Avoidance for an Industrial Robot Using Artificial Potential Field Based Method. Mechatronics **1**(1), 413–421 (2011)
17. A. Badawy, Dual-well Potential Field Function for Articulated Manipulator Trajectory Planning. Alexandria Engineering Journal **55**(2), 1235–1241 (2016)
18. C. Tsai, J. Lee, J. Chuang, "Path Planning of 3-D Objects Using A New Workspace Model," IEEE Transactions on Systems, Man, and Cybernetics. Part C (Applications and Reviews) **31**(3), 405–410 (2001)
19. T. Tsuji, Y. Tanaka, P. Morasso, V. Sanguineti, M. Kaneko, "Bio-mimetic Trajectory Generation of Robots Via Artificial Potential Field With Time Base Generator," IEEE Transactions on Systems, Man, and Cybernetics. Part C (Applications and Reviews) **32**(4), 426–439 (2002)
20. L. Sciavicco, B. Siciliano, A Solution Algorithm to the Inverse Kinematic Problem for Redundant Manipulators. IEEE Journal on Robotics and Automation **1**(4), 403–410 (1988)
21. Y. Zhang, J. Wang, "Obstacle Avoidance for Kinematically Redundant Manipulators Using A Dual Neural Network," IEEE Transactions on Systems, Man, and Cybernetics. Part B (Cybernetics) **34**(1), 752–759 (2004)
22. D. Guo, Y. Zhang, "A New Inequality-Based Obstacle-Avoidance MVN Scheme and Its Application to Redundant Robot Manipulators," IEEE Transactions on Systems, Man, and Cybernetics. Part C (Applications and Reviews) **42**(6), 1326–1340 (2012)
23. Z. Xu, S. Li, X. Zhou, T. Cheng, Dynamic Neural Networks Based Adaptive Admittance Control for Redundant Manipulators with Model Uncertainties. Neurocomputing **357**(1), 271–281 (2019)
24. J. Ren, B. Wang, M. Cai and J. Yu, "Adaptive Fast Finite-Time Consensus for Second-Order Uncertain Nonlinear Multi-Agent Systems With Unknown Dead-Zone," ?IEEE Access, vol. 8, No. 1, pp. 25557-25569, 2020
25. H. Wang, X. Liu, K. Liu, Robust Adaptive Neural Tracking Control for a Class of Stochastic Nonlinear Interconnected Systems. IEEE Transactions on Neural Networks and Learning Systems **27**(3), 510–523 (2015)
26. Z. Xu, S. Li, X. Zhou, W. Yan, T. Cheng, H. Dan, "Dynamic Neural Networks for Motion-Force Control of Redundant Manipulators: An Optimization Perspective", IEEE transactions on industrial electronics. Early access (2020). https://doi.org/10.1109/TIE.2020.2970635
27. X. Li, Z. Xu, S. Li, H. Wu, X, Zhou, "Cooperative Kinematic Control For Multiple Redundant Manipulators Under Partially Known Information Using Recurrent Neural Network". IEEE Access **8**(1), 40029–40038 (2020)
28. S. Li, Y. Zhang, L. Jin, Kinematic Control of Redundant Manipulators Using Neural Networks. IEEE Transactions on Neural Networks and Learning Systems **28**(10), 2243–2254 (2017)

29. C. Yang, G. Peng, Y. Li, R. Cui, L. Cheng, Z. Li, Neural Networks Enhanced Adaptive Admittance Control of Optimized Robot-Environment Interaction. IEEE Transactions on Cybernetics **49**(7), 1–12 (2019)
30. Z. Xu, S. Li, X. Zhou, W. Yan, T. Cheng, D. Huang, Dynamic neural networks based kinematic control for redundant manipulators with model uncertainties. Neurocomputing **329**(1), 255–266 (2019)
31. Y. Li, S. Li, B. Hannaford, A model based recurrent neural network with randomness for efficient control with applications. IEEE Trans. Industr. Informat. **15**(4), 2054–2063 (2019)
32. D. Chen, S. Li, Q. Wu, X. Luo, Super-twisting ZNN for coordinated motion control of multiple robot manipulators with external disturbances suppression. Neurocomputing **371**(1), 78–90 (2020)
33. Dechao Chen, Shuai Li, "A recurrent neural network applied to optimal motion control of mobile robots with physical constraints," Applied Soft Computing, to be published, 2019, https://doi.org/10.1016/j.asoc.2019.105880
34. T. Senoo, M. Koike, K. Murakami, M. Ishikawa, Impedance control design based on plastic deformation for a robotic arm. IEEE Robot. Automat. Lett. **2**(1), 209–2061 (2017)
35. T. Zhang, J. Xia, Interconnection and damping assignment passivity-based impedance control of a compliant assistive robot for physical human-robot interactions. IEEE Robotics Automat. Lett. **4**(2), 538–545 (2019)
36. B. Huang, Z. Li, X. Wu, A. Ajoudani, A. Bicchi, J. Liu, Coordination control of a dual-arm exoskeleton robot using human impedance transfer skills. IEEE Trans. Syst Man Cybernet. Syst. **49**(5), 954–963 (2019)
37. X. Gao, Exponential stability of globally projected dynamic systems. IEEE Trans. Neural Netw. **14**(2), 426–431 (2003)
38. J. Na, M. Nasiruddin, H. Guido, X. Ren, B. Phil, Robust adaptive finite-time parameter estimation and control for robotic systems. Int. J. Robust Nonlinear Control **25**(16), 345–3071 (2015)
39. C. Yang, Y. Jiang, W. He, J. Na, Z. Li, B. Xu, Adaptive parameter estimation and control design for robot manipulators with finite-time convergence. IEEE Trans. Industr. Electron. **65**(10), 8112–8123 (2018)
40. D. Nanayakkara, K. Kiguchi, T. Murakami, K. Watanabe and K. Izumi, Skillful Adaptation of A 7-DOF Manipulator to Avoid Moving Obstacles in A Teleoperated Force Control Task, 2001 IEEE International Symposium on Industrial Electronics Proceedings (Cat. No.01TH8570), pp. 1982-1988 (2001)

# Chapter 6
# RNN for Motion-Force Control of Redundant Manipulators with Optimal Joint Torque

**Abstract** Precise position force control is the core and difficulty of robot technology, especially for robots with redundant degrees of freedom. For example, track-based controls often fail to grind the robot due to the intolerable impact force applied to the end-effector. The main difficulties lie in the coupling of motion and contact forces, redundancy analysis and physical constraints. In this chapter, we propose a new motion force control strategy under the framework of recursive neural network. The tracking error and contact force are described respectively in the orthogonal space. By choosing the minimum joint torque as the secondary task, the control problem is transformed into the QP problem under multi-constraint conditions. In order to obtain real-time optimization of the joint torque relative to the non-convex joint angle, the original QP is reconstructed at the velocity level, and the original objective function is replaced by the time derivative. Then a convergent dynamic neural network is established to solve the improved QP problem online. The robot position control based on recursive neural network is extended to the robot position control based on position force, which opens a new way for the robot to turn from simple control angle to crossover design with convergence and optimality. Numerical results show that the proposed method can realize precise position force control, deal with inequality constraints such as joint angular velocity and torque limitation, and reduce joint torque consumption by 16% on average.

## 6.1 Introduction

Redundant manipulators, which have more DOFs than those required to complete a given task, are more flexible than non-redundant ones. The redundant DOFs enable manipulators to realize the fault tolerant control, improve operation performance and enhance reliability. Therefore, redundant manipulators have been widely used in industry, agriculture, military, space exploration, etc. Consequently, the research on the redundant manipulator has been studied intensively [1–4].

Motion control and force control are two main modes of redundant manipulator control. In motion control problems, a basic assumption is that there is no contact between the robot and the environment, that is, the robot can move freely in the workspace [5]. This problem is well-reflected in coating, welding, stacking, stacking

X. Zhou et al., *AI based Robot Safe Learning and Control*,
https://doi.org/10.1007/978-981-15-5503-9_6

and other applications. Then the core problem is to design control commands that drive the robot to follow a predetermined trajectory. The control command may be joint angular sequence [6], velocity sequence [7], acceleration sequence [8, 9] or torque sequences [10–12]. The redundancy resolution is usually used to achieve a secondary task, such as avoiding obstacles [13], avoiding singularities [14], etc. Different from motion control, force control involves the direct interaction between a robot and its environment. The control of contact force can enhance the robustness and flexibility of the robot in the weak structure environment, so as to enhance the robot′s operation ability [15]. Corresponding typical applications in polishing, grinding, assembly, polishing, polishing and other fields [16, 17]. In [18], the theoretical framework of impedance control is proposed. The basic idea is to use the environment as admittance and the robot as impedance.By maintaining the dynamic relationship between force and motion, the controller is represented as a spring-mass-damping system. In [19], a hybrid position force controller is proposed which combines position information with force information to realize the simultaneous control of position constraint and force constraint. Based on these two control frameworks, a series of controllers are proposed and verified by simulation or experiments [20–22].

Although the above work has achieved great success in motion force control of non-redundant robots, the control of redundant robots has not attracted enough attention. It is worth noting that the redundancy of the manipulator provides an opportunity to meet secondary objectives, but also sets up mathematical difficulties. In [23], in order to realize the flexible control of unknown obstacles, a position force control strategy is proposed. The motion of the robot is completely decoupled into two parts, namely the motion of the end-effector and the internal motion. The motion of the end-effector is controlled to achieve positional force control over the environment, while the internal motion is designed to avoid obstacles by minimizing impact. In [24], a robust control strategy with the ability to adjust contact force and apparent impedance is designed. The controller has strong robustness in dynamic and kinematic uncertainties. In [25], Patel et al. proposed a hybrid impedance control scheme based on pseudo inverse jacobian. The limitation of joint angle is avoided by defining a function that scales the difference between joint angle and its boundary. However, these literatures require continuous computation of the pseudo inverse of the jacobian matrix, which brings huge computational burden and is difficult to deal with multiple constraints [26].

In order to solve the redundant solution problem of redundant robots, a feasible method is to transform the control problem into an optimization problem under constraints [27]. The objective function is established according to the secondary task, and the constraint conditions are established according to the primary task and physical constraints. This optimization problem is often described as a QP problem [28]. Because of the high efficiency of parallel computing, recursive neural network is often used to solve QP-based redundant solutions online [29]. In recent years, the controller based on recursive neural network (RNN) has been introduced into the motion control of redundant robots. A new redundancy decomposition method is proposed, which constructs a robust neural network at the velocity level to guarantee the boundary of joint acceleration [30]. In [31], RNN is improved to allow projection operations on non-convex sets, avoiding the accumulation of tracking errors to system noise. In [32], a method is proposed to realize operational optimization through indirect maximization of time derivative. In [33], cooperative control of distributed multi-robot is studied. Recently, RNN has been extended to control flexible robots, model uncertainties and other issues [34–37]. Although the motion control of redundant robot based on RNN has obtained good research results, as far as we know, there is no research report on the application of RNN in motion force control of robot.

On this basis, we propose a position force control scheme based on RNN, which is an important extension of recursive neural network in robot control. Table 6.1 provides a brief comparison between the proposed and existing programmes. Unlike [25] and [23], in this chapter the motion force controller is established in the joint velocity stage and allows for multiple inequality constraints. The non-convex optimization problem is studied without loss of generality. With [31, 32, 34] the existing similar motion controller, the proposed motion force controller is no longer needed the pseudo inverse Jacobian.

This chapter mainly studies the following aspects. In the second part, the tracking error and contact force are modeled, and the control problem is written as QP problem. In Chap. 3, QP is reformulated at the velocity level by rewriting the objective function and constraints. In the fourth part, we set up an RNN to solve the redundant resolution problem. Stability has also been demonstrated. In the fifth part, the numerical experiments of 4-DOF planar manipulator are carried out. Finally, the sixth part carries on the summary to the full text. By the end of this section, the main contributions are as follows:

- As far as we know, this is the first research to study the motion force control of redundant robots in the framework of RNNs. It is worth noting that force sensitive manipulators, such as milling robots and polishing robots, cannot successfully control the use of existing results in RNNs, while the RNN model built in this work can do this.
- In the proposed control scheme, the motion-force control problem as well as redundancy resolution problem are reconstructed to facilitate practical implementations.
- The controller is capable of handling multiple inequality constraints, including but not limited to angle constraints, angular velocity constraints and torque constraints. This is of great significance in improving system security.
- The contribution of this chapter also lies in the realization of real-time optimization of joint torque, which is helpful to save energy in industrial applications.

## 6.2 Problem Formulation

### *6.2.1 Problem Formulation*

In this chapter, we focus on position-force control problem for redundant manipulators. Figure 6.1 gives a brief introduction of a redundant robot and its operation on an

**Table 6.1** Comparisons among the Proposed Motion-force Control Scheme and Existing Ones

| | Nonconvex | Motion-force Control | Control Command | Pseudo-inverse Required | Handling Multiple Inequality Constraints |
|---|---|---|---|---|---|
| This chapter | No | Yes | Joint Velocity | No | Yes |
| Methods [31, 32] | No | No | Joint Velocity | No | Yes |
| Method [25] | † | Yes | Joint Torque | Yes | No |
| Method [23] | † | Yes | Joint Torque | Yes | Restrictive‡ |
| Methods [34] | Yes | No | Joint Velocity | No | Yes |

† In [25] and [23], no projection operations are introduced in those control strategies
‡ In [23], only some certain kinds of inequalities can be handled by [23]

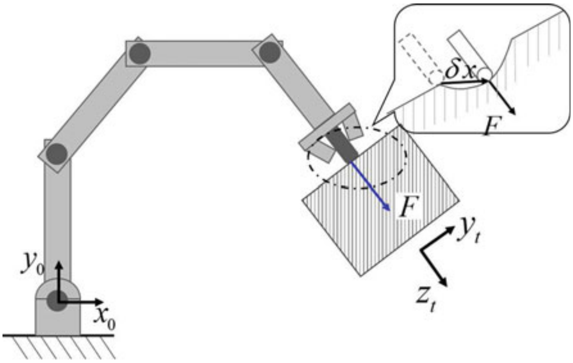


**Fig. 6.1** A brief diagram of a planar redundant manipulator and its operation on a workpiece

workpiece. The robot is expected to offer a desired contact force in the vertical direction of the contact surface, at the same time, the end-effector is required to track a predefined trajectory along the surface. In the base coordinate frame $R_0(0_0, x_0, y_0, z_0)$, forward kinematics of a serial manipulator can be written as

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

where $\theta \in \mathbb{R}^n$ is the vector of joint angles, and $x \in \mathbb{R}^m$ represents the end-effector$'$s coordinate vector in frame $R_0$, $f(\bullet) : \mathbb{R}^n \rightarrow \mathbb{R}^m$ is used to describe the forward kinematics operator. For a redundant manipulator, we have $n > m$.

By differentiating $x(t)$ with respect to time $t$, we can get the relationship between Cartesian velocity $\dot{x}(t) \in \mathbb{R}^m$ and joint velocity (or joint control signal) $\dot{\theta}(t) \in \mathbb{R}^n$ as follows:

$$J(\theta(t))\dot{\theta}(t) = \dot{x}(t), \tag{6.2}$$

where $J(\theta(t)) = \partial f(\theta(t))/\partial\theta(t)$ is called Jacobian matrix.

In position-force control tasks, the end-effector′s motion is constrained by the contact surface. For simplicity, we define a tool coordinate system as $R_t(x_t, y_t, z_t)$, in which the axis $z_t$ is set in alignment with the vertical direction of the contact surface. Obviously, the motion of end-effector can be specified along $x_t$ and $y_t$. In this chapter, frictional force between the robot and contact surface is ignored, therefore, the contact force $F$ is in alignment with $z_t$.

In the tool coordinate system $R_t$, let $\delta X_t$ be the displacement between effector and its position command, then the contact force $F_t$ can be formulated as

$$F_t = k_f \Sigma_t \delta X_t, \tag{6.3}$$

where $k_f > 0$ is the stiffness coefficient, $\Sigma_t = diag(0, 0, 1)$. Diagonal matrix $\Sigma_t$ describes the relationship between the contact force and relative displacement along different axes: 1 means that displacement component along $z_t$ affects the contact force, and 0 otherwise.

Similarly, in tool coordinate system $R_t$, the position tracking error $e_t$ can be written as

$$e_t = \bar{\Sigma}_t \delta X_t, \tag{6.4}$$

where $\bar{\Sigma}_t = I - \Sigma_f = diag(1, 1, 0)$, 1 means there is a DOF of movement along the corresponding direction, and 0 otherwise.

When the contact surface is prior known, $R_t$ can be obtained from $R_0$ by a rotation matrix $S_t$. Let $F$, $e_0$ and $\delta X$ be the corresponding description of $F_t$, $e_t$ and $\delta X_t$ in coordinate frame $R_0$, then we have $F = S_t^{\mathrm{T}} F_t, e_t = S_t e_0$ and $\delta X_t = S_t \delta X$. Therefore, $F$ and $e_0$ can be rewritten as

$$\begin{cases} F = k_f S_t^{\mathrm{T}} \Sigma_t S_t \delta X, \\ e_0 = S_f^{\mathrm{T}} \bar{\Sigma}_t S_f \delta X. \end{cases} \tag{6.5}$$

Notable that in frame $R_0$, the displacement $\delta X$ can be described as $\delta X = x - x_{\mathrm{d}}$, where $x_{\mathrm{d}}$ is the desired position signals described in $R_0$. Using (6.1), (6.5) can be rewritten as

$$\begin{cases} F = k_f S_t^{\mathrm{T}} \Sigma_f S_t (f(\theta) - x_{\mathrm{d}}), \\ e_0 = S_t^{\mathrm{T}} \bar{\Sigma}_f S_t (f(\theta) - x_{\mathrm{d}}). \end{cases} \tag{6.6}$$

**Remark 6.1** Equation (6.6) gives the unified description of the relationship between the contact force $F$, position tracking error $e_0$ and displacement $\delta X$ in $R_0$. $\delta X$ will lead to contact force $F$ in the vertical direction, and position tracking error $e_0$ along the contact surface.

In real implementations, given the desired contact force $F_{\mathrm{s}}$ and trajectory command $x_{\mathrm{d}}$, the manipulator′s end-effector is expected to offer contact force $F_{\mathrm{d}}$ while tracking $x_{\mathrm{d}}$, *i.e.*, $F \to F_{\mathrm{d}}$, $e_0 \to 0$. For the convenience of writing in the following sections, let $A=[k_f S_t^{\mathrm{T}} \Sigma_t S_t;\ S_t^{\mathrm{T}} \bar{\Sigma}_f S_t]$, $r=[F^{\mathrm{T}}, e_0^{\mathrm{T}}]^{\mathrm{T}}$, and $r_{\mathrm{d}}=[F_{\mathrm{d}}; 0]$. Then (6.6) can be reformulated as

$$A(f(\theta)-x_{\mathrm{d}})=r. \tag{6.7}$$

Therefore, the control objective of position-force control is to adjust the joint angles $\theta$, to ensure $r \rightarrow r_{\mathrm{d}}$.

### 6.2.2 Joint Torque and Physical Constraints

When the end-effector offers a contact force $F$, the corresponding torque is provided by motors at every joint. The relationship between contact force $F$ and the joint torque $\tau$ can be formulated as

$$\tau=J^{\mathrm{T}}(\theta) F. \tag{6.8}$$

In the control of redundant manipulators, there would be infinite groups of solutions to a certain control task. In order to save energy during the control process, we select a objective function scaling energy consumption as $\tau^{\mathrm{T}} \tau / 2$. The smaller $\tau^{\mathrm{T}} \tau / 2$, the less energy consumption.

In real implementations, the system is limited by physical constraints. For example, the joint angles $\theta$ and velocities $\dot{\theta}$ must not exceed their limits: $\theta^{\min}$,$\theta^{\max}$, $\dot{\theta}^{\min}$, $\dot{\theta}^{\max}$, since the possible collisions or overheating of motor would lead to irreversible damages. At the same time, considering the bounded torque output of the motors, the limitation of joint torque $\tau$ is described as $\tau^{\min} \leq \tau \leq \tau^{\max}$.

### 6.2.3 Optimization Problem Formulation

According to the descriptions above, the position-force control problem for redundant manipulators considering torque optimization can be formulated as

$$\begin{aligned}
\min \quad & G_1=\tau^{\mathrm{T}} \tau / 2, && \text{(6.9a)}\\
\text{s.t.} \quad & \tau=J^{\mathrm{T}} F, && \text{(6.9b)}\\
& r_{\mathrm{d}}=A(f(\theta)-x_{\mathrm{d}}), && \text{(6.9c)}\\
& \theta^{\min} \leq \theta \leq \theta^{\max}, && \text{(6.9d)}\\
& \dot{\theta}^{\min} \leq \dot{\theta} \leq \dot{\theta}^{\max}, && \text{(6.9e)}\\
& \tau^{\min} \leq \tau \leq \tau^{\max}, && \text{(6.9f)}
\end{aligned}$$

with $\theta$ being the decision variable. Equation (6.9a) is the cost function to be minimized, the equality constraint (6.9b) describes the relationship between the resulting joint torque $\tau$ and contact force $F$. The force and motion tasks of the robot are described in (6.9c), and inequality constraints (6.9e), (6.9d) and (6.9f) show the physical limitations to be satisfied. By substituting (6.9b) into (6.9a), the optimization problem can be rewritten as

$$\begin{aligned} \min \quad & G_1 = F^{\mathrm{T}} J(\theta) J^{\mathrm{T}}(\theta) F/2, && (6.10a)\\ \text{s.t.} \quad & r_{\mathrm{d}} = A(f(\theta) - x_{\mathrm{d}}), && (6.10b)\\ & \theta^{\min} \le \theta \le \theta^{\max}, && (6.10c)\\ & \dot{\theta}^{\min} \le \dot{\theta} \le \dot{\theta}^{\max}, && (6.10d)\\ & \tau^{\min} \le \tau \le \tau^{\max}. && (6.10e) \end{aligned}$$

To solve (6.10), there are two main challenges. Firstly, as an objective function to be minimized, $F^{\mathrm{T}} J(\theta) J^{\mathrm{T}}(\theta) F/2$ is usually non-convex relative to $\theta$, because it is a function of $J(\theta)$. Secondly, the equation constrain (6.10b) is highly nonlinear, and at the same time, it remains difficult to handle the inequality constraints, especially (6.10d) and (6.10e).

## 6.3 Reconstruction of Optimization Problem

In this section, in order to overcome the above difficulties, the redundancy resolution problem (6.10) will be reconstructed. The objective function is firstly redefined, and both equality and inequality constrains are rebuilt in velocity level.

### *6.3.1 Reconstruction of Objective Function*

As to $F^{\mathrm{T}} J(\theta) J^{\mathrm{T}}(\theta) F/2$, we will replace $F$ with the desired value $F_{\mathrm{d}}$. Therefore, the optimization function can be formulated as $G_2 = F_{\mathrm{d}}^{\mathrm{T}} J(\theta) J^{\mathrm{T}}(\theta) F_{\mathrm{d}}/2$.

**Remark 6.2** There are two main reasons: firstly, according to the control objective, the contact force $F$ is expected to track $F_{\mathrm{d}}$, if the controller is proper designed, $F$ will eventually converge to $F_{\mathrm{d}}$, consequently, $F_{\mathrm{d}}^{\mathrm{T}} J(\theta) J^{\mathrm{T}}(\theta) F_{\mathrm{d}}/2$ will be equivalent to $F^{\mathrm{T}} J(\theta) J^{\mathrm{T}}(\theta) F/2$. Secondly, $F_{\mathrm{d}}$ is independent of $\theta$, this replacement will reduce the computational complexity in the control process.

Differentiating $G_2$ with respect to time, we have

$$\dot{G}_2 = (J^{\mathrm{T}}(\theta) F_{\mathrm{d}})^{\mathrm{T}} \frac{\mathrm{d}(J^{\mathrm{T}}(\theta) F_{\mathrm{d}})}{\mathrm{d}t}. \tag{6.11}$$

Obviously, $\dot{G}_2$ describes the change of $G_2$. By minimizing $\dot{G}_2$, the system will be compelled to develop in the direction of decreasing $G_2$. Therefore, in this chapter, we use $\dot{G}_2$ instead of $G_2$ as the new objective function. Notable that $\mathrm{d}(J^{\mathrm{T}}(\theta)F_{\mathrm{d}})/\mathrm{d}t$ can be formulated as

$$\begin{aligned}\frac{\mathrm{d}}{\mathrm{d}t}(J^{\mathrm{T}}(\theta)F_{\mathrm{d}}) &= \sum_{i=1}^{n}\frac{\partial(J^{\mathrm{T}}(\theta)F_{\mathrm{d}})}{\partial\theta_i}\dot{\theta}_i + J^{\mathrm{T}}(\theta)\dot{F}_{\mathrm{d}} \\ &= [H_i, \cdots, H_n]\dot{\theta} + J^{\mathrm{T}}(\theta)\dot{F}_{\mathrm{d}},\end{aligned} \tag{6.12}$$

where $H_i \in \mathbb{R}^n$ is

$$H_i = \frac{\partial(J^{\mathrm{T}}(\theta)F_{\mathrm{d}})}{\partial\theta_i} = \begin{bmatrix}\sum_{j=1}^{m}(\partial(J(j,1)F_{\mathrm{d}}(j))/\partial\theta_i) \\ \sum_{j=1}^{m}(\partial(J(j,2)F_{\mathrm{d}}(j))/\partial\theta_i) \\ \cdots \\ \sum_{j=1}^{m}(\partial(J(j,n)F_{\mathrm{d}}(j))/\partial\theta_i)\end{bmatrix}.$$

Let $H = [H_1, \cdots, H_n]$, then (6.11) can be converted as

$$\dot{G}_2 = F_{\mathrm{d}}^{\mathrm{T}}JH\dot{\theta} + F_{\mathrm{d}}^{\mathrm{T}}JJ^{\mathrm{T}}\dot{F}_{\mathrm{d}}. \tag{6.13}$$

It is worth pointing that the second term of (6.13) is independent of $\dot{\theta}$, therefore, the objective function is equivalent to $F_{\mathrm{d}}^{\mathrm{T}}JH\dot{\theta}$.

### 6.3.2 Reconstruction of Constraints

In this part, we will transform the constrains into velocity level. First of all, we define a concatenated vector describing force and position errors as $e = r - r_{\mathrm{d}} = [F - F_{\mathrm{d}}; e_0]$, according to (6.7), $e$ can be formulated as

$$e = A(f(\theta) - x_{\mathrm{d}}) - r_{\mathrm{d}}. \tag{6.14}$$

Differentiating $e$ and combing (6.2) yields

$$\dot{e} = A(J\dot{\theta} - \dot{x}_{\mathrm{d}}) - \dot{r}_{\mathrm{d}}. \tag{6.15}$$

To ensure that $e$ converges to zero, a simple controller can be designed as $\dot{e} = -ke$, where $k > 0$ is a positive constant. According to (6.14), (6.15), the equality constrains can be converted in velocity level as

$$AJ\dot{\theta} = \dot{r}_{\mathrm{d}} + A\dot{x}_{\mathrm{d}} - k(Af(\theta) - x_{\mathrm{d}}). \tag{6.16}$$

As to the inequality constraints (6.10c) and (6.10d), according to [27], let $\omega = \dot{\theta}$ and define $\alpha \geq 0$ as a constant parameter to scale the negative feedback to conform the joint constraints, these two constraints can be formulated in the speed level as

$$\omega^{\min} \le \omega \le \omega^{\max}, \tag{6.17}$$

where $\omega^{\min} = \max\{\alpha(\theta^{\min} - \theta), \dot{\theta}^{\min}\}$, and $\omega^{\max} = \min\{\alpha(\theta^{\max} - \theta), \dot{\theta}^{\max}\}$.

Similarly, (6.10e) can be built indirectly by limiting its derivative: $\beta(\tau^{\min} - \tau) \le \dot{\tau} \le \beta(\tau^{\max} - \tau)$, where $\beta$ is a positive constant. By combing (6.12), the boundedness of joint torque can be rewritten as an inequality constraint about a function $g(\omega)$ as

$$g(\omega) \le 0, \tag{6.18}$$

where $g(\omega) = [\beta(\tau^{\min} - \tau) - J^{\mathrm{T}}\dot{F}_{\mathrm{d}} - H\omega, H\omega - \beta(\tau^{\max} - \tau)J^{\mathrm{T}} + \dot{F}_{\mathrm{d}}]^{\mathrm{T}} \in \mathbb{R}^{2n}$.

### 6.3.3 Reformulation and Convexification

According to the above description, in order to achieve position-force control of redundant manipulators, instead of solving (6.10) directly, one feasible solution is to solve the optimization problem in velocity level as

$$\begin{aligned} \min \quad & F_{\mathrm{d}}^{\mathrm{T}} J H\omega, && (6.19a)\\ \text{s.t.} \quad & r_r = AJ\omega, && (6.19b)\\ & g(\omega) \le 0, && (6.19c)\\ & \omega \in \Omega, && (6.19d) \end{aligned}$$

where $r_r = \dot{r}_d + A\dot{x}_{\mathrm{d}} - k(Af(\theta) - x_{\mathrm{d}})$, $\Omega = \{\omega \in \mathbb{R}^n | \omega_i^{\min} \le \omega_i \le \omega_i^{\max}\}$ is a convex set. It is remarkable that the objective function described in (6.19a) is non-convex relative to $\omega$. Therefore, (6.19b) is introduced to convexity (6.19a). The final form of optimization problem is described as

$$\begin{aligned} \min \quad & F_{\mathrm{d}}^{\mathrm{T}} J H\omega + (AJ\omega - r_r)^{\mathrm{T}}(AJ\omega - r_r), && (6.20a)\\ \text{s.t.} \quad & r_r = AJ\omega, && (6.20b)\\ & g(\omega) \le 0, && (6.20c)\\ & \omega \in \Omega. && (6.20d) \end{aligned}$$

So far, we have reconstructed the position-force control with joint torque optimization problem into a quadratic programming issue under constraints. However, the QP problem (6.20) cannot be solved directly.

## 6.4 RNN Based Redundancy Resolution

In this chapter, in order to solve the optimization problem (6.20), an expanded recurrent neural network is built to obtain the optimal solution of (6.20). Stability will be also discussed.

### 6.4.1 RNN Design

Firstly, let $\lambda_1 \in \mathbb{R}^{2m}$ and $\lambda_2 \in \mathbb{R}^{2n}$ be dual variables to constraints (6.20b) and (6.20c), a Lagrange function is defined as

$$\begin{aligned} L =& F_{\mathrm{d}}^{\mathrm{T}} J H \omega + (AJ\omega - r_r)^{\mathrm{T}}(AJ\omega - r_r) \\ &+ \lambda_1^{\mathrm{T}}(r_r - AJ\omega) + \lambda_2^{\mathrm{T}} g(\omega). \end{aligned} \tag{6.21}$$

According to Karush−Kuhn−Tucker condition, the optimal solution of the optimization problem (6.20) can be equivalently formulated as

$$\omega = P_{\Omega}(\omega - \frac{\partial L}{\partial \omega}), \tag{6.22a}$$

$$r_r = AJ\omega, \tag{6.22b}$$

$$\lambda_2 = (\lambda_2 + g(\omega))^+, \tag{6.22c}$$

where $P_{\Omega}(x) = \mathrm{argmin}_{y\in\Omega}||y - x||$ is a projection operation to convex set $\Omega$, and $(x)^+ = (x_1^+, \cdots, x_{2n}^+)^{\mathrm{T}}, x_i^+ = \max(x_i, 0)$.

In order to solve (6.22), an expanded recurrent neural network is designed as

$$\begin{aligned} \varepsilon\dot{\omega} =& -\omega + P_{\Omega}(\omega - H^{\mathrm{T}} J^{\mathrm{T}} F_{\mathrm{d}} - J^{\mathrm{T}} A^{\mathrm{T}}(AJ\omega - r_r) \\ &+ J^{\mathrm{T}} A^{\mathrm{T}} \lambda_1 - \nabla g \lambda_2), \end{aligned} \tag{6.23a}$$

$$\varepsilon\dot{\lambda}_1 = r_r - AJ\omega, \tag{6.23b}$$

$$\varepsilon\dot{\lambda}_2 = (\lambda_2 - (\lambda_2 + g(\omega))^+), \tag{6.23c}$$

where $\nabla g = (\frac{\partial g_1}{\partial \omega}, \cdots, \frac{\partial g_{2m}}{\partial \omega}) = [-H^{\mathrm{T}}, H^{\mathrm{T}}] \in \mathbb{R}^{n\times 2n}$, $\varepsilon$ is a positive constant scaling the convergence of (6.23). The pseudo code of the RNN-based strategy is shown in Algorithm 5.

**Algorithm 5** The RNN based position-force controller

```
Input: Control parameters $\epsilon$, $k$, $\alpha$, stiffness coefficient $k_f$, prior knowledge of the contact surface $S_t$, $\Sigma_t$ and $\bar{\Sigma}_t$. The robot' joint limits $\theta_i^{\max}$, $\theta_i^{\min}$ and joint speed limits $\dot{\theta}_i^{\max}$, $\dot{\theta}_i^{\min}$, initial joint angles $\dot{\theta}(0)$, desired tracking trajectory $r_{\rm d}(t)$, $\dot{r}_{\rm d}(t)$, and contact force $F_{\rm d}$. Feedback of actual coordinate $x(t)$, contact force $F(t)$, and joint angle $\theta$, task duration $T$.
Output: To achieve position-force control of the redundant manipulator and optimize joint torque
1: Initialize $\lambda_1(0)$, $\lambda_2(0)$
2: Repeat
3:     On-line feedback of $F, \theta, x \leftarrow$ from sensors
4:     Calculate the weight matrix $A$
5:     Calculate the reference command $r_r \leftarrow$ Eq. (6.19)
6:     Update joint velocity command $\omega$ by $\dot{\omega} \leftarrow$ Eq. (6.23a)
7:     Update state variable $\lambda_1$ by $\dot{\lambda}_1 \leftarrow$ Eq. (6.23b)
8:     Update state variable $\lambda_2$ by $\dot{\lambda}_2 \leftarrow$ Eq. (6.23c)
Until $(t > T)$
```

### 6.4.2 *Stability Analysis*

In this part, theoretical analysis of stability and convergence of closed-loop system using the proposed neural network (6.23).

First of all, several important definitions and Lammas are presented, which is very useful in stability analysis.

**Definition 6.1** A continuously differentiable function $F(\bullet)$ is said to be monotone if $\nabla F + \nabla F^{\rm T}$ is positive semi-definite, where $\nabla F$ is the gradient of $F(\bullet)$.

**Lemma 6.1** *[38] A dynamic neural network is said to converge to the equilibrium point if it satisfies*

$$\kappa \dot{x} = -\dot{x} + P_S(x - \varrho F(x)), \tag{6.24}$$

*where $\kappa > 0$ and $\varrho > 0$ are constant parameters, and $P_S = argmin_{y \in S}||y - x||$ is a projection operator to closed set $S$.*

So far, a theorem about the convergence of the redundancy resolution problem can be described as follows

**Theorem 6.1** *Given the motion-force control problem for redundant manipulators with torque optimization under physical constraints as (6.20), the recurrent neural network (6.23) is stable and will globally converge to the optimal solution of (6.20).*

***Proof*** Let $\xi = [\omega^{\rm T}, \lambda_1^{\rm T}, \lambda_2^{\rm T}]^{\rm T}$, the proposed RNN (6.23) can be written as

$$\varepsilon \dot{\xi} = -\xi + P_{\bar{\Omega}}[\xi - F(\xi)], \tag{6.25}$$

where $F(\xi) = [F_1(\xi), F_2(\xi), F_3(\xi)]^{\mathrm{T}} \in \mathbb{R}^{2m+3n}$, in which

$$\begin{bmatrix} F_1 \\ F_2 \\ F_3 \end{bmatrix} = \begin{bmatrix} H^{\mathrm{T}} J^{\mathrm{T}} F_{\mathrm{d}} + J^{\mathrm{T}} A^{\mathrm{T}} (AJ\omega - r_r) - J^{\mathrm{T}} A^{\mathrm{T}} \lambda_1 + \nabla g \lambda_2 \\ \lambda_1 - r_r + AJ\omega \\ -g(\omega) \end{bmatrix}.$$

Let $\nabla F(\xi) = \partial F / \partial \xi$, we have

$$\nabla F(\xi) = \begin{bmatrix} J^{\mathrm{T}} A^{\mathrm{T}} AJ & -J^{\mathrm{T}} A^{\mathrm{T}} & \nabla g \\ AJ & I & 0 \\ -(\nabla g)^{\mathrm{T}} & 0 & 0 \end{bmatrix}. \tag{6.26}$$

It is remarkable that

$$\nabla F(\xi) + (\nabla F(\xi))^{\mathrm{T}} = \begin{bmatrix} 2J^{\mathrm{T}} A^{\mathrm{T}} AJ & 0 & 0 \\ 0 & 2I & 0 \\ 0 & 0 & 0 \end{bmatrix}. \tag{6.27}$$

From Definition 6.1, $F(\xi)$ is a monotone function of $\xi$.

According to the description of (6.23) and (6.25), $P_{\bar{\Omega}}$ can be formulated as $P_{\bar{\Omega}} = [P_{\Omega}; P_R; P_{\Lambda}]$, where $P_R \in \mathbb{R}^{2m}$ is a projection operator of $\lambda_1$ to set $R$, with the upper and lower bounds being $\pm\infty$. Furthermore, $(\bullet)^+$ is a special case of $P_{\Lambda}$, in which $\Lambda = \mathbb{R}^{2n}_{+}$ is the nonnegative quadrant of $\mathbb{R}^{2n}$. Therefore, $P_{\bar{\Omega}}$ is a projection operator to closed set $\bar{\Omega}$. Based on Lemma 6.1, the proposed neural network (6.23) is stable and will globally converge to the optimal solution of (6.20). The proof is completed. □

## 6.5 Illustrative Examples

In this chapter, taking the planar 4-DOF manipulator as an example, numerical calculation is carried out to verify the effectiveness of the proposed control scheme. First, we will check the control performance without joint torque optimization by making $H^{\mathrm{T}} J^{\mathrm{T}} F_{\mathrm{d}}$ in (6.23a) be 0. Secondly, a dynamic simulation example of joint torque optimization is introduced to illustrate the superiority of the control strategy. Finally, the adaptability and optimization performance of this method are verified by simulation experiments under different initial conditions.

### *6.5.1 Simulation Setup*

As shown in Fig. 6.2, a contact surface in the workspace can be described as $y = 0$, the end-effector can move freely along the horizontal axis, and the desired contact force $F_{\mathrm{d}}$ is aligned with the vertical direction. The stiffness coefficient $k_f$ is

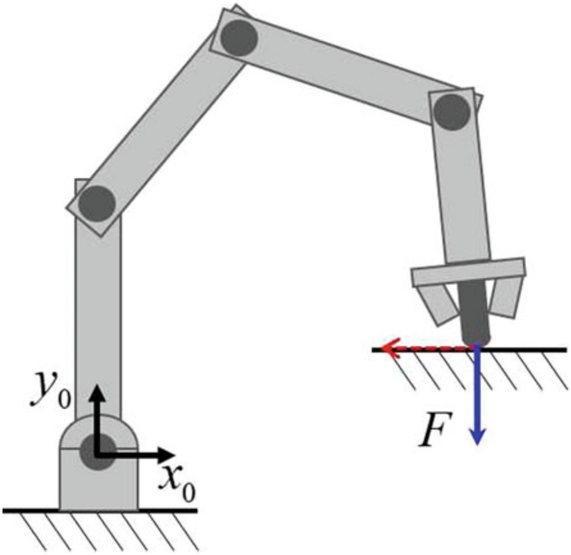


**Fig. 6.2** Physical structure of the 4-link robot manipulator

set to be 1000N/mm. Control positive control gains are set as $\alpha = 10$, $\beta = 10$, $k = 8$, $\varepsilon = 0.005$, respectively. Physical constraints of joint angles, velocities and torque are defined as $\theta^{\min} = [-2, -2, -2, -2]^{\mathrm{T}}$ rad, $\theta^{\max} = [2, 2, 2, 2]^{\mathrm{T}}$ rad, $\dot{\theta}^{\min} = [-2, -2, -2, -2]^{\mathrm{T}}$ rad/s, $\dot{\theta}^{\max} = [2, 2, 2, 2]^{\mathrm{T}}$ rad/s, $\tau^{\min} = [-10, -10, -10, -10]^{\mathrm{T}}$ Nm, $\tau^{\max} = [10, 10, 10, 10]^{\mathrm{T}}$ Nm, respectively.

### 6.5.2 Position-Force Control Without Optimization

In this part, the robot is controlled to offer a constant contact force on the surface while tracking a given trajectory. It is remarkable that joint torque optimization is not investigated yet. The initial joint angles are selected as $\theta_0 = [1.57, -1.26, -0.52, -0.52]^{\mathrm{T}}$ rad. The desired trajectory is defined as $x_{\mathrm{d}} = [0.25 + 0.1cos(0.5t), 0]^{\mathrm{T}}$, and the contact force is defined as $F_{\mathrm{d}} = [0, -1]^{\mathrm{T}}$ N. Numerical results are shown in Fig. 6.3. When simulation begins ($t < 0.5s$), the position error is big, and there is no contact between the robot and surface. Correspondingly, both contact force and result torque are zero. Under the RNN based controller (6.23), joint velocities reach the maximum value, the end-effector approaches to the surface rapidly from the initial position, and the tracking error converges to zero quickly, the corresponding joint configurations are shown in Fig. 6.3a. As the robot approaches the contact surface, the robot slows down quickly, and the contact force rises from zero, the then converges to the desired value smoothly. In the stable state ($t > 2$ s), both contact force $F$ and end-effector track the desired command, the tracking errors are zero, which means the robot tracks

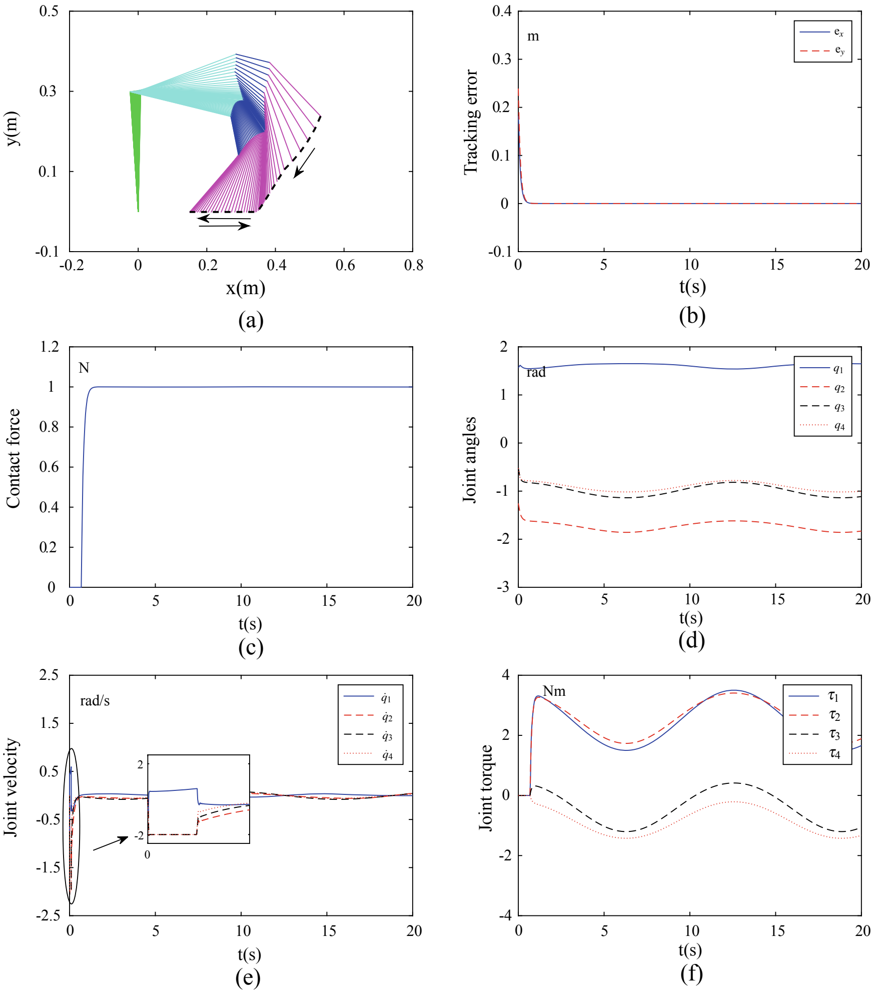


**Fig. 6.3** Numerical results when tracking a time varying force command along a trajectory without optimization. **a** Profiles of the end-effector (black dashed line) and the corresponding joint configurations. **b** Profiles of position error. **c** Profiles of contact force. **d** Profiles of joint angles. **e** Profiles of joint velocities. **f** Profiles of joint torque

both desired trajectory and force successfully. Correspondingly, joint angles change periodically, which enables the robot to achieve dynamic tracking. This will also lead to a periodic change in result torque in joint space, as shown in Fig. 6.3f. During the whole process, boundary of joint angles, velocities and torque are all ensured.

### *6.5.3 Position-Force Control with Optimization*

In this part, joint torque optimization is introduced to make full use of redundancy resolution. The proposed position-force control scheme is firstly validated in a fixed point case, and then extended to dynamic cases.

*(1) Position-Force Control on A Fixed Point*
In this simulation, the robot is wished to offer a constant contact force $F_{\mathrm{d}} = [0, -10]^{\mathrm{T}}$ at a fixed point $x_{\mathrm{d}} = [0.3, 0]^{\mathrm{T}}$. The initial values of joint angles are also $\theta_0 = [1.57, -1.26, -0.52, -0.52]^{\mathrm{T}}$rad. Numerical results are shown in Fig. 6.4. At the beginning of simulation, the robot moves at its maximum speed (2 rad/s), making the regulation error converge quickly. Then it slows down as the regulation error is small. At $t = 0.5\,s$, robot touches the surface, which leads to the emergence of contact force. Using the proposed controller, both control error of motion and force converge to zero smoothly. Correspondingly, the Euclidean norm of joint torque also converges to a constant value (3.7 $\mathrm{N}^2/\mathrm{m}^2$). From Fig. 6.4e, f, joint angles and velocities do not exceed their limits, showing that the proposed scheme could handle inequality constraints effectively. To further demonstrate the validity of the optimization scheme, comparative simulations without optimization are also carried out. The obtained Euclidean norm of joint torque without optimization is shown as the red dashed line (4.3 $N^2/m^2$ in stable state). After introducing joint torque optimization strategy, 16% off of torque consumption is achieved.

*(2) Position-Force Control Along A Straight Line*
Then we check the optimization scheme in dynamic control. Both the desired path $x_{\mathrm{d}}$ and desired force $F_{\mathrm{d}}$ are time varying. The expected signals are defined as $x_{\mathrm{d}} = [0.25 + 0.1cos(0.5t), 0]^{\mathrm{T}}$, $F_{\mathrm{d}} = [0, 20 - 2cos(0.5t)]^{\mathrm{T}}$ N, respectively. The initial values of joint angles are the same as the previous case. Numerical results are shown in Fig. 6.5. We also define a index to scale the torque consumption as $J_\tau = \int_0^T ||\tau(t)||_2^2 \mathrm{d}t$.

When simulation begins, high joint speed ensures the fast convergence of tracking error, which is very similar to the previous simulation. After $t = 2s$, high precision trajectory tracking is realized by the control strategy, as well as the contact force. Comparative simulation without joint torque optimization is carried out. Figure 6.5d shows the comparison of Euclidean norms of joint torque with and without optimization. Correspondingly, $J_\tau$ decreases 16.2% from 142 to 119, showing the validity of the proposed scheme. It is notable that all physical constraints are guaranteed. Dynamic change of joint configurations is shown in Fig. 6.5a.

*(3) Position-Force Control Along An Arc Surface*
In this part, the end-effector is controlled to track a quarter-circular surface, which is centered at $[0.3, 0.3]^{\mathrm{T}}$m with radius 0.2 m, and is provided a constant force of 10 N in the vertical direction. The initial values of joint angles are selected as $\theta_0 = [1.5708, -0.9851, -1.1714, 0]^{\mathrm{T}}$rad. Numerical results are shown in Fig. 6.6. The trajectory of end-effector is shown in Fig. 6.6a, while in Fig. 6.6b, optimization is not introduced. The proposed controller enables the robot to achieve precision control of both position and force, and at the same time, by adjusting its joint angles, the

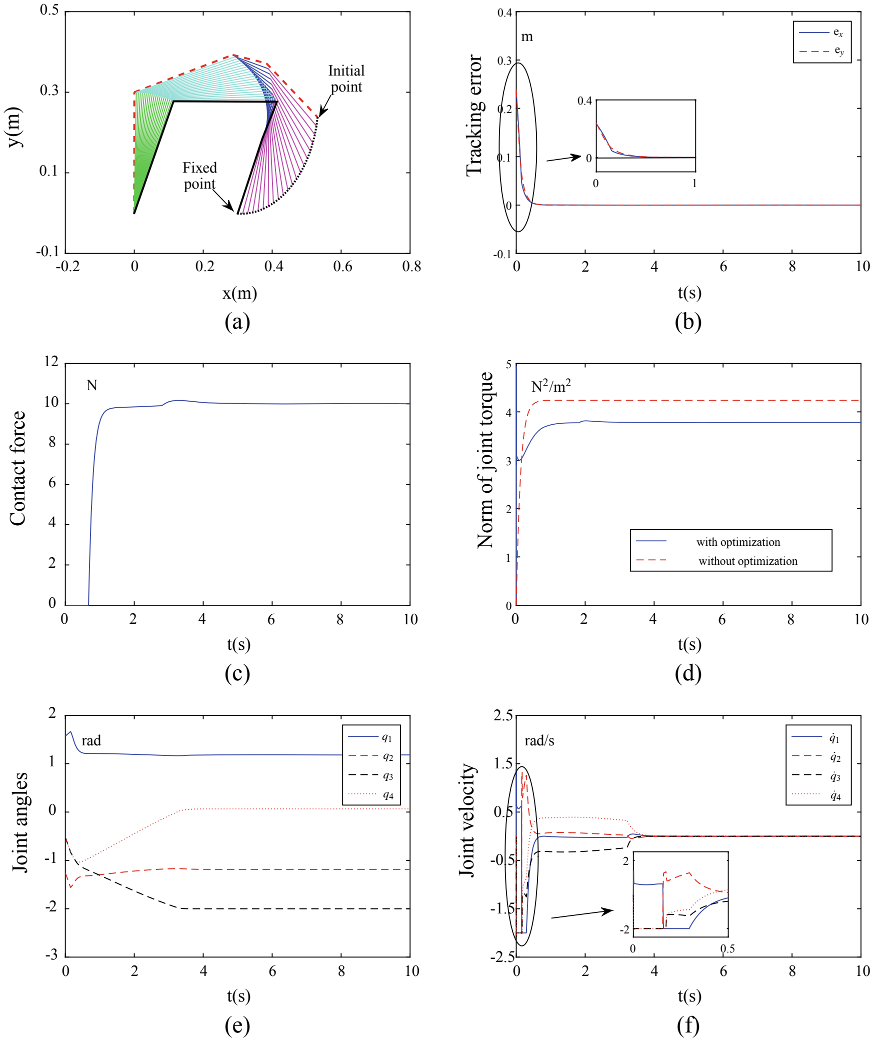
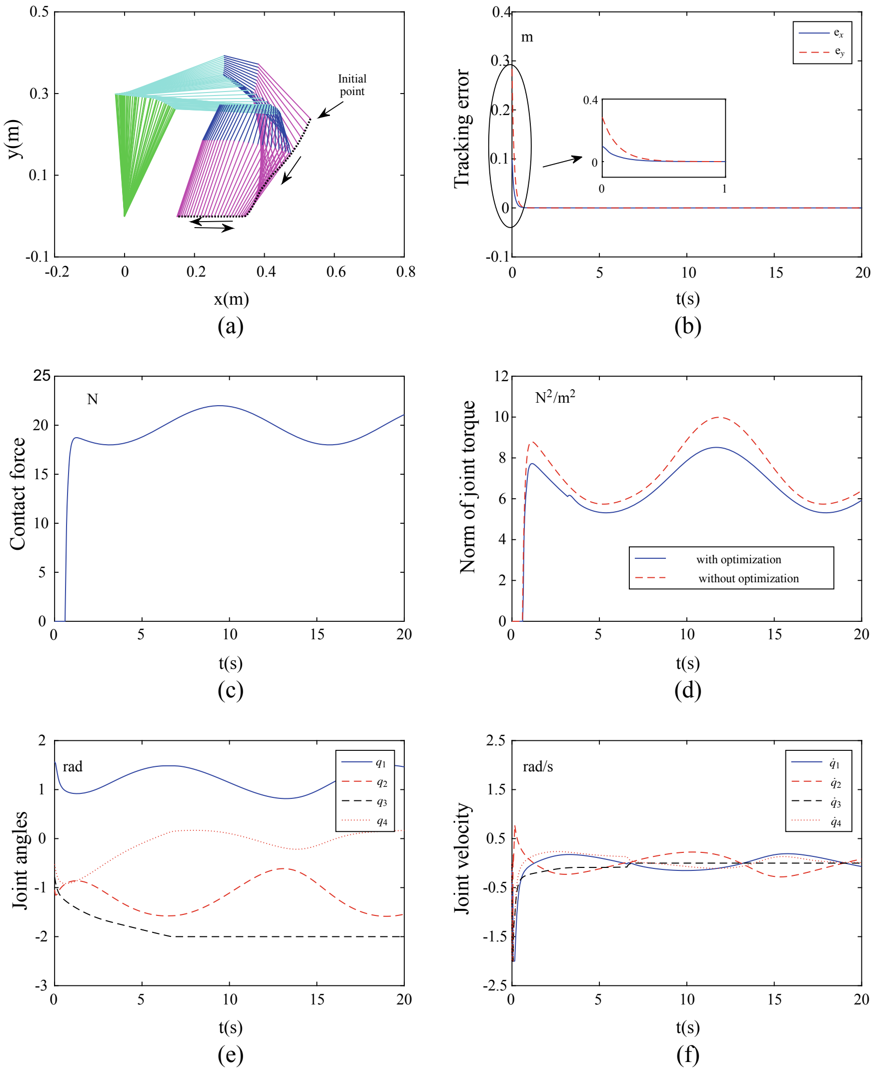


**Fig. 6.4** Numerical results when the robot is controlled to offer constant force at a fixed point. **a** Profiles of the end-effector (black dashed line) and the corresponding joint configurations. **b** Profiles of position error. **c** Profiles of contact force. **d** Comparison of Euclidean norm of joint torque with and without optimization. **e** Profiles of joint angles. **f** Profiles of joint velocities

joint torque consumption is reduced, i.e., $J_\tau$ decreases 17.6% from 88.1 to 72.6. It is remarkable that the physical constraints are also guaranteed.

*(4) Adaptability to Different Initial Settings*

To further illustrate the joint optimization scheme, another fixed-point control is presented. Desired signals are set as $x_{\rm d} = [0.3, 0]^{\rm T}$ and $F_{\rm d} = [0, -10]^{\rm T}$. The initial



(a) (b) (c) (d) (e) (f)

**Fig. 6.5** Numerical results when tracking a time varying force command along a straight line with optimization. **a** Profiles of the end-effector (black curve) and the corresponding joint configurations. **b** Profiles of position error. **c** Profiles of contact force. **d** Comparison of Euclidean norm of joint torque with and without optimization. **e** Profiles of joint anges. **f** Profiles of joint velocities

values of joint angles is selected as $\theta_0 = [1.8850, -1.8850, -1.2566, 0]^{\mathrm{T}}$ rad, consequently, the corresponding position of the end-effector is exactly the same as $x_{\mathrm{d}}$. As shown in Fig. 6.7a, the robot adjusts its posture and stops in final state, while keeping its end-effector on the fixed point. This phenomenon is similar to the null-space movement based on pseudo-inverse methods. However, different with pseudo-inverse

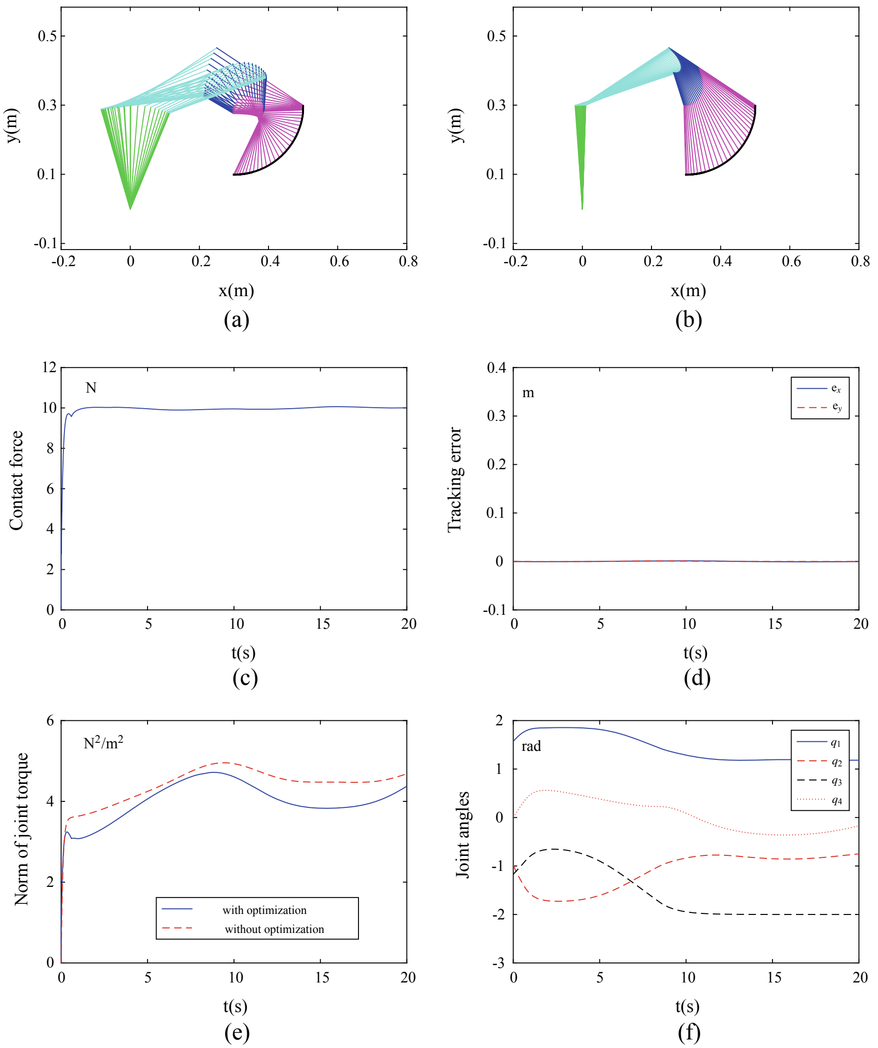


**Fig. 6.6** Numerical results when tracking a time varying force command along an arc surface with optimization. **a** Time history of the end-effector (black curve) and the corresponding joint configurations with optimization. **b** Time history of the end-effector (black curve) and the corresponding joint configurations without optimization. **c** Profiles of contact force. **d** Profiles of position error. **e** Comparison of Euclidean norm of joint torque with and without optimization. **f** Profiles of joint angles

based method, the RNN based motion-force controller is capable of handling physical inequalities, at the same time, joint torque optimization is achieved from 4.3 to 3.7. Further more, there in no need to calculate pseudo-inverse of Jacobian matrix, which will save computing cost effectively.

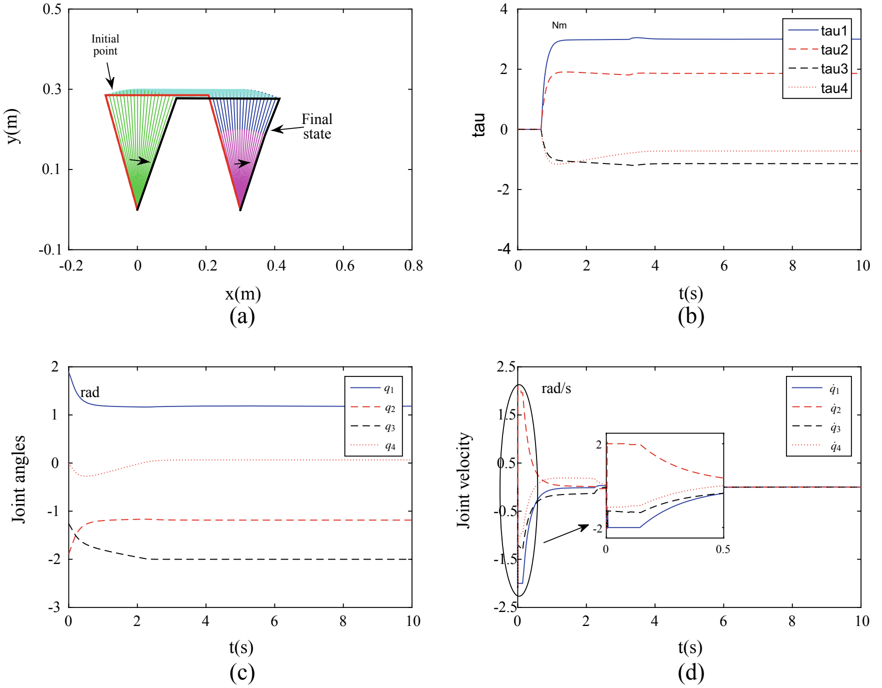


**Fig. 6.7** Numerical results when the initial position of end-effector locates on the desired fixed point. **a** Profiles of joint configurations. **b** Profiles of Euclidean norm of joint torque with and without optimization. **c** Profiles of joint angles. **d** Profiles of joint velocities

Finally, a group of verifications for fixed point position-force control with different initial joint angles are carried out, the desired signals are the same as the previous simulation. As shown in Fig. 6.8, although the initial joint angles are different, at steady state, the robot reaches the same joint angle, which shows the adaptability of the RNN based control strategy.

## 6.6 Question and Answer

**Q1:** "*What's the complexity of the proposed RNN?*"

**Answer:** The network is organized in a one-layer architecture, which consists of $2m + 3n$ neurons, namely $\omega \in \mathbb{R}^n$, $\lambda_1 \in \mathbb{R}^{2m}$ and $\lambda_2 \in \mathbb{R}^{2n}$. Despite the difference between the proposed neural network with traditional recurrent neural networks, from both the mathematical description Eq. (23) and the architecture, one characteristic of the established neural network can be found that the neural network uses its historical information to calculate the output at current moment, which is also a typical feature of recurrent neural networks.

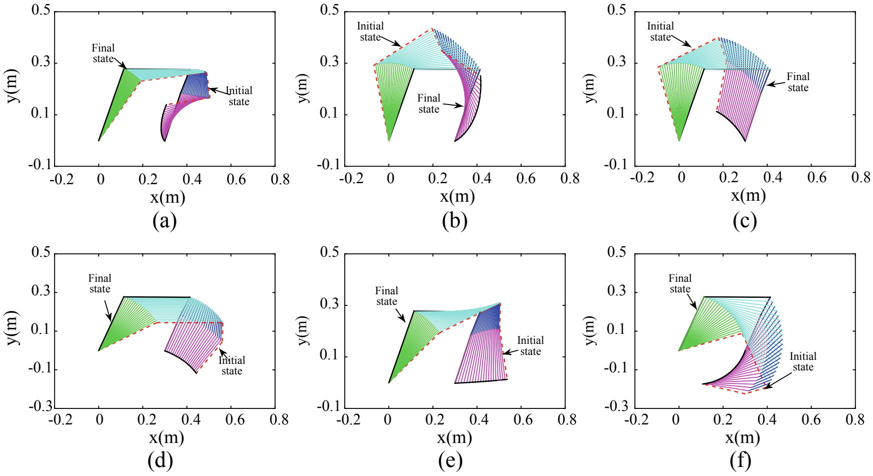


**Fig. 6.8** Time history of robot′s joint configurations in fixed point control from different initial joint angle $\theta_0$. **a** $\theta_0 = [0.9, -0.75, -1.5, -1.6]^{\mathrm{T}}$ rad. **b** $\theta_0 = [1.8, -0.3, -1.6, 0.6]^{\mathrm{T}}$ rad. **c** $\theta_0 = [1.9, -1.5, -1.6, -0.6]^{\mathrm{T}}$ rad. **d** $\theta_0 = [0.5, -0.5, -1.6, -0.6]^{\mathrm{T}}$ rad. **e** $\theta_0 = [0.7, -0.3, -2, 0.2]^{\mathrm{T}}$ rad. **f** $\theta_0 = [0.3, -1.5, -1.6, -0.6]^{\mathrm{T}}$ rad

**Q2:** "*As described in Section II, the matrices $\Sigma_f$ and $\bar{\Sigma}_f$ are crucial in the controller design, however, the authors didnt show the details. How to obtain those matrices in actual applications requires detailed description.*"

**Answer:** $\Sigma_f$ and $\bar{\Sigma}_f$ are used to realize the decoupling of the contact force and tracking error of the end-effector. When the contact surface is known, the combination of $\Sigma_f$, $\bar{\Sigma}_f$ and $S_t$ enables the normalized description of the control tasks.

**Q3:** "Limited stiffness of the manipulator elements can lead to state variables oscillations. Have you observed such work of the object?"

**Answer:** The limited stiffness of the manipulator elements can lead to state variables oscillations. In this chapter, the QP type formulation is obtained based on static modeling method, and inertial force is not taken into account. The condition of the modeling method is that the process is quasi-static. In other words, the relative motion of the end-effector and the workpiece is very slow. In the experiment tests, we also found that some oscillation would occur if some parameters are appropriately tuned. In this case, a damping coefficient can be introduced to handle the oscillations.

**Q4:** "*In real manipulator significant issue is related to control of electric drives. In mentioned structures, internal control loop, related to torque control, introduces some delays for external speed controllers. Have you considered such problem?*"

**Answer:** In this chapter, we mainly focus on projection RNN based controller design in kinematic level, and the control command is selected as joint velocity signals. Therefore, we assume that the robot controller can provide an ideal response to the joint velocity command. Although the delay is unavoidable for real systems, when the control frequency is set as 100 Hz, experimental results could show the effectiveness of the proposed controller. From Eq. (23), it can be observed that the force control is realized by adjusting the joint velocities base on the RNN, which is consistent with the idea of admittance control. In our experiment, the velocity control in the inner loop is done by the robot controller, and we assume that it provide an ideal response to the joint velocity command. It is remarkable that the uncertainties in the dynamic level such as friction and disturbances do affect the performance of position-force control in the outer loop, but these uncertainties can be suppressed by the closed-loop control mechanism of the controller itself.

**Q5:** "*Could you explain real impact of projection operator PR on work of the control system?*"

**Answer:** The projection operator $P_\Omega$ plays an important role in guaranteeing the bounded-ness of the output of neural network i.e., the boundedness of $\omega$ can be ensured introducing $P_\Omega$. As described in Eq. (17), based on escape velocity method, both the boundedness of joint angles and velocities are guaranteed.

**Q6:** "*RNN uses delays during data processing, so the calculation step size seems to be important for overall work. Have you considered such issue?*"

**Answer:** We did consider this problem. The faster the RNN calculates, the better performance can be achieved. But at the same time, it would also lead to a increase of computational burden, which may make the system unstable. In our experiment, the control period is set to be 10 ms.

## 6.7 Summary

This chapter focuses on motion-force control problem for redundant manipulators, while physical constraints and torque optimization are taken into consideration. Firstly, tracking error and contact force are modeled in orthogonal spaces respectively, and then the control problem is turned into a QP problem, which is further rewritten in velocity level by rewriting objective function and constraints. To handle multiple physical constraints, an RNN based scheme is designed to solve the redundancy resolution online. Numerical experiment results show the validity of the proposed control scheme. Before ending this chapter, it is noteworthy that this is the first chapter to deal with motion-force control of redundant manipulators in the framework of RNNs and redundant manipulators with force sensitivity, e.g., grinding robots, can be readily controlled with the proposed RNN model but cannot with existing RNN models in this field.

## References

1. C. Yang, Y. Jiang, J. Na, Z. Li, L. Cheng and C. Su, "Finite-time Convergence Adaptive Fuzzy Control for Dual-Arm Robot with Unknown Kinematics and Dynamics," *IEEE Trans. Fuzzy Syst.*, to be published, https://doi.org/10.1109/tfuzz.2018.2864940, 2018
2. H. Wu, Y. Guan, J. Rojas, A latent state-based multimodal execution monitor with anomaly detection and classification for robot introspection. Applied Sciences. **9**(6), 1072 (2019)

3. H. Wu, Z. Xu, W. Yan, Q. Su, S. Li, T. Cheng, X. Zhou, Incremental Learning Introspective Movement Primitives From Multimodal Unstructured Demonstrations. IEEE Access. **15**(7), 159022–36 (2019)
4. Wu H, Guan Y, Rojas J. "Analysis of multimodal Bayesian nonparametric autoregressive hidden Markov models for process monitoring in robotic contact tasks." International Journal of Advanced Robotic Systems. 2019 26;16(2):1729881419834840
5. L. Cheng, Z. Hou, M. Tan, Adaptive Neural Network Tracking Control for Manipulators with Uncertain Kinematics, Dynamics and Actuator Model. Automatica **45**(10), 2312–2318 (2009)
6. L. Cheng, W. Liu, Z. Hou, Z. Li, M. Tan, Neural Network Based Nonlinear Model Predictive Control for Piezoelectric Actuators. IEEE Trans. Ind. Electron. **62**(12), 7717–7727 (2015)
7. H. Wang, W. Liu, J. Qiu, P. Liu, Adaptive Fuzzy Decentralized Control for A Class of Strong Interconnected Nonlinear Systems with Unmodeled Dynamics. IEEE Trans. Fuzzy Syst. **26**(2), 836–846 (2018)
8. D. Guo, Y. Zhang, Acceleration-Level Inequality-Based MAN Scheme for Obstacle Avoidance of Redundant Robot Manipulators. IEEE Trans. Ind. Electron. **61**(12), 6903–6914 (2014)
9. J. Ren, B. Wang, M. Cai and J. Yu, "Adaptive Fast Finite-Time Consensus for Second-Order Uncertain Nonlinear Multi-Agent Systems With Unknown Dead-Zone," ?IEEE Access, vol. 8, No. 1, pp. 25557-25569, 2020
10. J. Na, X. Ren, D. Zheng, Adaptive Control for Nonlinear Pure-feedback Systems with High-order Sliding Mode Observer. IEEE Trans. Neur. Net. Lear. **24**(3), 370–382 (2013)
11. C. Yang, Y. Jiang, W. He, J. Na, Z. Li, B. Xu, Adaptive Parameter Estimation and Control Design for Robot Manipulators with Finite-Time Convergence. IEEE Trans. Ind. Electron. **65**(10), 8112–8123 (2018)
12. Y. Liu, S. Tong, Barrier Lyapunov Functions-based Adaptive Control for A Class of Nonlinear Pure-feedback Systems with Full State Constraints. Automatica **64**(1), 70–75 (2016)
13. D. Guo and Y Zhang, "A New Inequality-Based Obstacle-Avoidance MVN Scheme and Its Application to Redundant Robot Manipulators," *IEEE Trans. Syst., Man, Cybern. C, Appl. Rev.*, vol. 42, no. 6, pp. 1326-1340, 2012
14. W. Xu, J. Zhang, B. Liang, B. Li, Singularity Analysis and Avoidance for Robot Manipulators with Non-Spherical Wrists. IEEE Trans. Ind. Electron. **63**(1), 277–290 (2016)
15. N. Hogan, On the Stability of Manipulators Performing Contact Tasks. IEEE J. Robot. Automat. **4**(6), 677–686 (1988)
16. O. Khatib, A Unified Approach for Motion and Force Control of Robot Manipulators: The Oerational Space Formulation. IEEE J. Robot. Automat. **3**(1), 43–53 (2003)
17. K. Kiguchi, T. Fukuda, Position/force Control of Robot Manipulators for Geometrically Unknown Objects Using Fuzzy Neural Networks. IEEE Trans. Ind. Electron. **47**(3), 641–649 (2000)
18. N. Hogan, "Impedance Control: An Approach to Manipulation, Parts I, II, III," in *J. Dyn. Syst., Meas. Control.*, vol. 107, no. 1, pp. 1-23, 1985
19. J. Craig, P. Hsu and S. Sastry "Adaptive Control of Mechanical Manipulators," in *Proc. IEEE Int. Conf. Robotics and Automation.*, San Francisco, Ca, USA., 1986, pp. 190-195
20. Z. Li, J. Liu, Z. Huang, Y. Peng, H. Pu, L. Ding, Adaptive Impedance Control of HumanCRobot Cooperation Using Reinforcement Learning. IEEE Trans. Ind. Electron. **64**(10), 8013–8022 (2017)
21. D. Chen, S. Li, Q. Wu, X. Luo, Super-twisting ZNN for coordinated motion control of multiple robot manipulators with external disturbances suppression. Neurocomputing **371**(1), 78–90 (2020)
22. Dechao Chen, Shuai Li, "A recurrent neural network applied to optimal motion control of mobile robots with physical constraints," Applied Soft Computing, to be published, 2019, https://doi.org/10.1016/j.asoc.2019.105880
23. B. Nemec, L. Zlajpah, Force Control of Redundant Robots in Unstructured Environment. IEEE Trans. Ind. Electron. **49**(1), 233–240 (2002)
24. R. Patel, H. Talebi, J. Jayender, F. Shadpey, A Robust Position and Force Control Strategy for 7-DOF Redundant Manipulators. IEEE/ASME Trans. Mechatron. **14**(5), 575–589 (2009)

25. R. Patel, F. Shadpey, *“Control of Redundant Robot Manipulators,” Lecture Notes in Control* (Springer, Information Sciences., 2005)
26. W. He, Z. Yan, C. Sun, Adaptive Neural Network Control of A Marine Vessel with Constraints Using The Symmetric Barrier Lyapunov Function. IEEE Trans. Cybern. **47**(7), 1641–1651 (2017)
27. Y. Zhang, J. Wang, Y. Xia, A Dual Neural Network For Redundancy Resolution of Kinematically Redundant Manipulators Subject to Joint Limits and Joint Velocity Limits. IEEE Trans. Neural Networks **14**(4), 658–667 (2003)
28. L. Jin, Y. Zhang, G2-type SRMPC Scheme For Synchronous Manipulation of Two Redundant Robot Arms. IEEE Trans. Cybern. **45**(2), 153–164 (2015)
29. B. Cai, Y. Zhang, Different-Level Redundancy-Resolution and Its Equivalent Relationship Analysis for Robot Manipulators Using Gradient-Descent and Zhang 's Neural-Dynamic Methods. IEEE Trans. Ind. Electron. **59**(8), 3146–3155 (2012)
30. Y. Zhang, S. Li, J. Gui, X. Luo, Velocity-Level Control with Compliance to Acceleration-Level Constraints: A Novel Scheme for Manipulator Redundancy Resolution. IEEE Trans Ind. Informat. **14**(3), 921–930 (2018)
31. S. Li, Y. Zhang, L. Jin, Kinematic Control of Redundant Manipulators Using Neural Networks. IEEE Trans. Neur. Net. Lear. **28**(10), 2243–2254 (2017)
32. L. Jin, S. Li, H. La, X. Luo, Manipulability Optimization of Redundant Manipulators Using Dynamic Neural Networks. IEEE Trans. Ind. Electron. **64**(6), 4710–4720 (2017)
33. S. Li, S. Chen, B. Liu, Y. Li, Y. Liang, Decentralized Kinematic Control of A Class of Collaborative Redundant Manipulators via Recurrent Neural Networks. Neurocomputing. **91**(1), 1–10 (2012)
34. Z. Xu, S. Li, X. Zhou, Y. Wu, T. Cheng and D. Huang, “Dynamic Neural Networks Based Kinematic Control for Redundant Manipulators with Model Uncertainties,” *Neurocomputing.*, to be published, https://doi.org/10.1016/j.neucom.2018.11.001
35. Z. Xu, S. Li, X. Zhou, T. Cheng, Dynamic Neural Networks Based Adaptive Admittance Control for Redundant Manipulators with Model Uncertainties. Neurocomputing **357**(1), 271–281 (2019)
36. Z. Xu, S. Li, X. Zhou, W. Yan, T. Cheng, H. Dan, “Dynamic Neural Networks for Motion-Force Control of Redundant Manipulators: An Optimization Perspective”, IEEE transactions on industrial electronics. Early access (2020). https://doi.org/10.1109/TIE.2020.2970635
37. X. Li, Z. Xu, S. Li, H. Wu, X, Zhou, “Cooperative Kinematic Control For Multiple Redundant Manipulators Under Partially Known Information Using Recurrent Neural Network”. IEEE Access **8**(1), 40029–40038 (2020)
38. X. Gao, Exponential stability of globally projected dynamic systems. IEEE Trans. Neural Netw. **14**(2), 426–431 (2003)