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 JACO2 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 acceler-
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 designed 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 JACO2. 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:

\[ f(\theta(t)) = x(t), \]  

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, \( f(\bullet) : \mathbb{R}^n \rightarrow \mathbb{R}^m \) 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:

\[ J(\theta(t), a_k)\dot{\theta}(t) = \dot{x}(t), \]  

with \( J(\theta(t), a_k) = \frac{\partial f(\theta(t), a_k)}{\partial \theta(t)} \) 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:

\[ J(\theta(t), a_k)\dot{\theta}(t) = Y_k(\theta(t), \dot{\theta}(t))a_k, \]  

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 $x_d(t) \in \mathbb{R}^m$ and its time derivative $\dot{x}_d(t)$ are accessible, both $x_d(t)$ and $\dot{x}_d(t)$ 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 $x_d(t)$, in the sense that $f(\theta(t)) = x(t) \rightarrow x_d(t)$. 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 $u^T u$. 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

$$ e(t) = x_d(t) - x(t). \quad (2.4) $$

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

$$ \dot{e}(t) = -k e(t), \quad (2.5) $$

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

$$ \dot{x}(t) = \dot{x}_d(t) + k (x_d(t) - x(t)). \quad (2.6) $$

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 $\dot{x}(t) = J(\dot{\theta}(t), a_k) u(t)$, 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 cannot 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 \( \hat{a}_k = \hat{a}_k - a_k \). Using \( J(\theta(t), \hat{a}_k) \) and the control signal \( \dot{\theta}(t) \), we can estimate the velocity of the end-effector as

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

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

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

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 \( x_d(t) \) and \( \dot{x}_d(t) \). The desired trajectory together
2.3 Main Results

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 $J(\theta(t), a_k)u(t) = \dot{x}_d(t) + k|e(t)|^\rho \text{sgn}(e(t))$ 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 $u^T u = \dot{\theta}^T \dot{\theta}$ as an object function to be optimized, the joint range $\theta_{\text{min}} \leq \theta_i \leq \theta_{\text{max}}$ and joint speed limits $\dot{\theta}_{\text{min}} \leq \dot{\theta}_i \leq \dot{\theta}_{\text{max}}$ 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:

$$\min u^T u,$$  \hspace{1cm} (2.10a)

$$\text{s.t.} \quad b_0 = \hat{J} u,$$ \hspace{1cm} (2.10b)

$$u \in \Omega,$$ \hspace{1cm} (2.10c)

where $\Omega = \{u \in \mathbb{R}^n | u_{\text{min}}^i \leq u_i \leq u_{\text{max}}^i \}$ is a convex set describing the physical constraints, where $u_{\text{min}}^i = \max \{\alpha(\theta_{\text{min}}^i - \theta_i), \dot{\theta}_{\text{min}}^i \}$, $u_{\text{max}}^i = \min \{\alpha(\theta_{\text{max}}^i - \theta_i), \dot{\theta}_{\text{max}}^i \}$, $\alpha > 0$ 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

$$u = P_\Omega (u - \partial L / \partial u),$$ \hspace{1cm} (2.11a)

$$b_0 = \hat{J} u,$$ \hspace{1cm} (2.11b)

where $P_\Omega (\bullet)$ is a projection operation to the set $\Omega$, $P_\Omega (x) = \arg \min_{y \in \Omega} ||y - x||$, and $L = L(u, \lambda)$ is a Lagrange function defined as $L(u, \lambda) = u^T u / 2 + \lambda^T (b_0 - \hat{J} u)$, 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 $\theta_i^{\text{max}}$, $\theta_i^{\text{min}}$, joint speed limits $\dot{\theta}_i^{\text{max}}$, $\dot{\theta}_i^{\text{min}}$, initial states $u(0)$, $\theta(0)$, nominal kinematic parameter $\hat{a}_k(0)$, desired path $x_d(t)$, $\dot{x}_d(t)$, 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

1: Initialize $\lambda(0)$, $a_k(0) \leftarrow a_k^n$.
2: Repeat
3: $x$, $\theta \leftarrow$ Sensor readings
4: Calculate tracking error $e \leftarrow$ Equation (2.4)
5: Position loop controller $b_0 \leftarrow$ Equation (2.6)
6: Update control output $u$ by $\dot{u} \leftarrow$ Equation (2.12a)
7: Update state variable $\lambda$ by $\dot{\lambda} \leftarrow$ Equation (2.12b)
8: Update model parameter $\hat{a}_k$ by $\dot{\hat{a}}_k \leftarrow$ Equation (2.12c)
Until ($t > T$)

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

\[ \varepsilon \dot{\lambda} = \hat{J} u - b_0, \]  
(2.12b)

\[ \dot{\hat{a}}_k = -\Gamma_1 Y_k^T(\theta, u) e, \]  
(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 $u^T u$, 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 $\dot{\theta}$, joint speed $u$ and tracking error $e$, therefore, the proposed control strategy can be realized easily.
2.3 Main Results

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 $\xi = [u^T, \lambda^T]^T$, controller (2.12a), (2.12b) can be reformulated as

$$
\dot{\xi} = -\xi + P_{\tilde{\Omega}}(\xi - R(\xi)),
$$

(2.13)

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

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

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 $\nabla^T R$. Remarkable that $\nabla R$ satisfies the following positive semi-definite property:

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

(2.14)

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

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

(2.15)
It is obvious that $V_1 = 0$ if and only if $\xi \in \tilde{\Omega}$. According to the result of reference in E.9, $\partial ||\xi - P_{\tilde{\Omega}}(\xi)||_2^2 / \partial \xi = 2(\xi - P_{\tilde{\Omega}}(\xi))$. Differentiating $V_1$ with respect to time and substituting (2.13) yield:

$$
\dot{V}_1 = (\xi - P_{\tilde{\Omega}}(\xi))^T \dot{\xi} \\
= - (\xi - P_{\tilde{\Omega}}(\xi))^T (\xi - P_{\tilde{\Omega}}(\xi - R(\xi))) / \epsilon.
$$

(2.16)

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

$$
\xi^* = P_{\tilde{\Omega}}(\xi^* - R(\xi^*)).
$$

(2.17)

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

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

(2.18)

Define function $V_2$ as

$$
V_2 = (\xi - P_{\tilde{\Omega}}(\xi - R(\xi)))^T R(\xi) + \|\xi - \xi^*\|_2^2 / 2 \\
- \|\xi - P_{\tilde{\Omega}}(\xi - R(\xi))\|_2^2 / 2 + V_1.
$$

(2.19)

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

$$(\xi - P_{\tilde{\Omega}}(\xi - R(\xi)))^T R(\xi) - \|\xi - P_{\tilde{\Omega}}(\xi - R(\xi))\|_2^2 / 2 \\
\geq (\xi - P_{\tilde{\Omega}}(\xi - R(\xi)))^T R(\xi) - \|\xi - P_{\tilde{\Omega}}(\xi - R(\xi))\|_2^2 \\
= (\xi - R(\xi) - P_{\tilde{\Omega}}(\xi - R(\xi)))^T (P_{\tilde{\Omega}}(\xi - R(\xi)) - \xi).
$$

(2.20)

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

$$
V_2 \geq ||\xi - \xi^*||_2^2 / 2 + V_1.
$$

(2.21)

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

$$
\dot{V}_2 = (\xi - P_{\tilde{\Omega}}(\xi - R(\xi)))^T \nabla R(\xi) + \dot{\xi}^T R(\xi) + (\xi - \xi^*)^T \dot{\xi}
$$
2.3 Main Results

Differentiating (2.24), we get

\[ R(\xi) - R(\xi^*) = \nabla R(\xi')(\xi - \xi^*), \]

(2.23)

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

\[ (\xi - \xi^*)^T R(\xi) = (\xi - \xi^*)^T \nabla R(\xi')(\xi - \xi^*) + (\xi - \xi^*)^T R(\xi^*). \]

(2.24)

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

\[ (\xi - \xi^*)^T R(\xi) \geq 0. \]

(2.25)

Combining inequalities (2.16), (2.24) and (2.25) yields \( \dot{V}_2 \leq 0 \), and \( \dot{V}_2 = 0 \) only if \( \xi \in \widehat{\mathcal{O}} \), which indicates \( (\xi - P_{\widehat{\mathcal{O}}}(\xi - R(\xi)))^T \nabla R(\xi - P_{\widehat{\mathcal{O}}}(\xi - R(\xi))) = 0 \), \( (\xi - R(\xi) - P_{\widehat{\mathcal{O}}}(\xi - R(\xi)))^T (P_{\widehat{\mathcal{O}}}(\xi - R(\xi)) - \xi^*) = 0 \) and \( (\xi - \xi^*)^T R(\xi) = 0 \). From (2.24), we get \( (\xi - \xi^*)^T \nabla R(\xi')(\xi - \xi^*) \) and \( (\xi - \xi^*)^T R(\xi^*) = 0 \). 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

\[ V_3 = e^T e/2 + \tilde{a}_k^T \Gamma_1^{-1} \tilde{a}_k/2. \]

(2.26)

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

\[ \dot{V}_3 = e^T (\dot{x}_d - \dot{x}) + \tilde{a}_k^T \Gamma_1^{-1} \dot{\tilde{a}}_k \]

\[ = e^T (\dot{x}_d - J(\theta, a_k)u + (1 - 1)k|e|^\rho \text{sgn}(e) + \tilde{a}_k^T \Gamma_1^{-1} \dot{\tilde{a}}_k \]

\[ = e^T (b_0 - Y_k(\theta, u)(\tilde{a}_k - \bar{a}_k) - k|e|^\rho \text{sgn}(e)) - \tilde{a}_k^T Y_k^T(\theta, u)e \]

\[ = e^T (b_0 - \dot{J}u + Y_k(\theta, u)\bar{a}_k) - k|e|^\rho + 1 - \tilde{a}_k^T Y_k^T(\theta, u)e. \]

(2.27)

As proved above, using the neural network (2.12), \( u^T u \) will be minimized under the constraints \( b_0 = \dot{J}u \) and \( u \in \mathcal{O} \). Notable that \( \tilde{a}_k^T Y_k^T(\theta, u)e \) is a scalar value, we have \( \tilde{a}_k^T Y_k^T(\theta, u)e = e^T Y_k(\theta, u)\bar{a}_k \). Then (2.27) can be rewritten as...
\[ \dot{V}_3 = e^T Y_k(\theta, u) \tilde{a}_k - k|e|^\rho + 1 - \tilde{a}_k^T Y_k^T(\theta, u) e \]
\[ = - k|e|^\rho + 1 \leq 0. \quad (2.28) \]

Then we have \( e = x_d - x \) is bounded. Taking the time derivative of \( \dot{V}_3 \), we have:

\[ \ddot{V}_3 = - k(\rho + 1)|e|^{\rho} \text{sgn}(e) \dot{e} \]
\[ = - k(\rho + 1)|e|^{\rho} \text{sgn}(e)(\dot{x}_d - J(\theta, a_k) u). \quad (2.29) \]

Since \( J(\theta, a_k) \) is composed of trigonometric functions of \( \theta \) and kinematic parameters \( a_k, J(\theta, a_k) \) is bounded. \( \dot{x}_d \) is also bounded. As illustrated in part I, \( u \) is bounded, thus \( \dot{V}_3 \) 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. \( \square \)

**Remark 2.4** The convergence analysis shows the stability of the proposed control strategy. The tracking error would globally converge 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 JACO2 can thus be regarded as a functional redundant manipulator. The architecture of JACO2 is shown in Fig. 2.2, and the DH parameters are shown in Table 2.1. Noticing that the last 3 joints of JACO2 do not intersect at a single point, these joints cannot be simplified as spherical joint, therefore the configuration of JACO2 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 JACO2
Table 2.1 DH parameters of the Kinova JACO2 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 JACO2 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 JACO2. 
(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 JACO2 tracks a given circle in Cartesian space. a Motion trajectory of end effector (red curve) and the corresponding incremental configurations of JACO2. 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 $\|u\|^2$.

$\theta(0)$ is randomly set as $[0.5, 0, 1.5, 0, 0, 0]^T\text{rad}$, and the initial joint speed $u(0)$ is selected to be zero. The nominal values of kinematic parameters are selected as $\hat{a}_k(0) = [0.25, 0.2, 0, -0.2, -0.1, -0.2]^T\text{m}$. The set $\Omega$ describing joint speed limits are set to be $[-2, 2]^6\text{rad/s}$. 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 Illustrative Examples

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 $[0.3, 0.4, 0.4]^T$ 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 JACO2 is expected to move at an angular speed of 0.5rad/s along a circular trajectory. The desired circle is centered at $[0.3, 0.3, 0.3]^T$ 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 JACO2 is used to track a square trajectory. The corners of the desired square in the Cartesian space are set to be $[0.3, 0.4, 0.4]^T$, $[0.4, 0.3, 0.4]^T$, $[0.3, 0.2, 0.2]^T$, and $[0.2, 0.3, 0.3]^T$. The motion period is 12.56s. The velocity norm of desired path $||\dot{x}_d(t)||$ keeps constant, which means that the expected velocity
Fig. 2.5 Results when JACO2 tracks a square trajectory in Cartesian space. 

a Motion trajectory of end effector (red curve) and the corresponding incremental configurations of JACO2. 

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 \( b_0(x-axis \text{ direction}) \) described by (2.6) and the corresponding output \( \hat{J}u(1) \). 

f The second Cartesian velocity input \( b_0(y-axis \text{ direction}) \) described by (2.6) and the corresponding output \( \hat{J}u(2) \). 

g The third Cartesian velocity input \( b_0(z-axis \text{ direction}) \) 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 \( \dot{x}_d(t) \) remains constant between two adjacent vertices, while \( \dot{x}_d(t) \) 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 \([4 \times 10^{-3}, 2 \times 10^{-3}, 1.5 \times 10^{-3}]^T \text{m} \). 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 Position | Joint Velocity | Regulation Error | Tracking error | Real-time Control |
|-------------------------|-------------|------|------------------|----------------|------------------|----------------|------------------|
| This chapter            | Not required|      | No               | 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\textsubscript{2} robot arm dem-
onstrate 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 con-
trol 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\textsubscript{2} s kinematic regressor matrix
\( Y_k \in \mathbb{R}^{3\times6} \) is given as follows.

\[
Y_{11} = 0,
\]
\[
Y_{12} = -\dot{\theta}_2 c_1 c_2 + \dot{\theta}_1 s_1 s_2,
\]
\[
Y_{13} = -\dot{\theta}_1 c_1,
\]
\[
Y_{14} = \dot{\theta}_2 (c_1 c_2 c_3 + c_1 s_2 s_3) + \dot{\theta}_1 (c_2 s_1 s_3 - c_3 s_1 s_2) - \dot{\theta}_3 (c_1 c_2 c_3 + c_1 s_2 s_3),
\]
\[
Y_{15} = (\dot{\theta}_1 ((\sqrt{3} c_1 c_4) / 2 + (\sqrt{3} s_4 (s_1 s_2 s_3 + c_2 c_3 s_1)) / 2 + (c_2 s_1 s_3) / 2 - (c_3 s_1 s_2) / 2) - \dot{\theta}_4 ((\sqrt{3} s_1 s_4) / 2 + (\sqrt{3} c_4 (c_1 c_2 c_3 + c_1 s_2 s_3)) / 2) + \dot{\theta}_2 (c_1 c_2 c_3 + c_1 s_2 s_3) / 2) - (\sqrt{3} s_4 (c_1 c_2 s_3 - c_1 c_3 s_2) / 2) - (\sqrt{3} s_4 (c_1 c_2 c_3 - c_1 c_3 s_2) / 2),
\]
\[
Y_{16} = (\dot{\theta}_3 (\sqrt{3} c_5 (s_1 s_4 + c_4 (c_1 c_2 c_3 + c_1 s_2 s_3))) / 2 + (\sqrt{3} s_5 ((c_4 s_1) / 2 - (s_4 (c_1 c_2 c_3 + c_1 s_2 s_3))) / 2 + (\sqrt{3} (c_1 c_2 s_3 - c_1 c_3 s_2)) / 2) - \dot{\theta}_4 ((\sqrt{3} s_1 s_4) / 4 - (\sqrt{3} c_5 ((s_1 s_4) / 2 + (c_4 (c_1 c_2 c_3 + c_1 s_2 s_3))) / 2 - (\sqrt{3} s_5 (c_4 s_1 - s_4 (c_1 c_2 s_3 + c_1 s_2 s_3) - c_2 c_3 s_1))) / 2 - (\sqrt{3} c_5 ((c_1 s_4) / 2 - (\sqrt{3} s_5 (s_1 s_4 - c_4 (s_1 s_2 s_3 + c_2 c_3 s_1))) / 2 + (\sqrt{3} s_5 (c_1 c_2 s_3 - c_1 c_3 s_2))) / 2 + (\sqrt{3} c_5 (c_1 c_2 s_3 - c_1 c_3 s_2)) / 2) - (\sqrt{3} c_5 ((s_4 - (c_1 c_2 s_3 - c_1 c_3 s_2))) / 2 - (\sqrt{3} c_5 (c_1 c_2 s_3 - c_1 c_3 s_2)) / 2 - (\sqrt{3} c_5 ((s_4 - (c_1 c_2 s_3 - c_1 c_3 s_2))) / 2 + (c_1 c_2 s_3) / 4 + (\sqrt{3} c_5 (c_1 c_2 s_3 - c_1 c_3 s_2)) / 2) - (\sqrt{3} c_5 ((s_4 - (c_1 c_2 s_3 - c_1 c_3 s_2))) / 2 - (\sqrt{3} c_5 ((s_4 (c_1 c_2 s_3 - c_1 c_3 s_2))) / 4 + (c_1 c_2 c_3 / 4 + (c_1 c_2 s_3) / 4 + (\sqrt{3} c_5 (s_1 s_4 - c_4 (s_1 s_2 s_3 + c_2 c_3 s_1))) / 2) - (\sqrt{3} c_5 ((s_4 - (c_1 c_2 s_3 - c_1 c_3 s_2))) / 2 - (\sqrt{3} c_5 ((s_4 (c_1 c_2 s_3 - c_1 c_3 s_2))) / 4 + (c_1 c_2 c_3) / 4 + (c_1 s_2 s_3) / 4 + (\sqrt{3} c_5 (c_1 c_2 s_3 - c_1 c_3 s_2)) / 2) - (\sqrt{3} c_5 ((s_4 (c_1 c_2 s_3 - c_1 c_3 s_2))) / 2 - (\sqrt{3} c_5 ((s_4 - (c_1 c_2 s_3 - c_1 c_3 s_2))) / 4 + (c_1 c_2 c_3) / 4 + (c_1 s_2 s_3) / 4 + (\sqrt{3} c_5 (c_1 c_2 s_3 - c_1 c_3 s_2)) / 2),
\]
\[
Y_{21} = 0,
\]
\[
Y_{22} = -\dot{\theta}_1 c_1 s_2 - \dot{\theta}_2 c_2 s_1,
\]
\[
Y_{23} = -\dot{\theta}_1 s_1,
\]
\[
Y_{24} = \dot{\theta}_2 (s_1 s_2 s_3 + c_2 c_3 s_1) - \dot{\theta}_3 (s_1 s_2 s_3 + c_2 c_3 s_1) - \dot{\theta}_1 (c_1 c_2 s_3 - c_1 c_3 s_2),
\]
2.5 Summary

\[ Y_{25} = \dot{\theta}_2 ((s_1 s_2 s_3) / 2 + (c_2 c_3 s_1) / 2 - (\sqrt{3} s_4 (c_2 s_1 s_3 - c_3 s_1 s_2)) / 2) - \dot{\theta}_3 ((s_1 s_2 s_3) / 2 + (c_2 c_3 s_1) / 2 - (\sqrt{3} s_4 (c_2 s_1 s_3 - c_3 s_1 s_2)) / 2) + \dot{\theta}_1 ((\sqrt{3} c_4 s_1) / 2 - (\sqrt{3} s_4 (c_1 c_2 c_3 + c_1 s_2 s_3)) / 2 - (c_1 c_2 s_3) / 2 + (c_1 c_3 s_2) / 2) + \dot{\theta}_4 ((\sqrt{3} c_1 s_4) / 2 - (\sqrt{3} c_4 (s_1 s_2 s_3 + c_2 c_3 s_1))) / 2), \]

\[ Y_{26} = \dot{\theta}_1 ((\sqrt{3} c_4 s_1) / 4 - (\sqrt{3} c_5 ((c_4 s_1) / 2 - (s_4 (c_1 c_2 c_3 + c_1 s_2 s_3)) / 2 + (\sqrt{3} (c_4 s_1 s_3 - c_1 c_3 s_2))) / 2)) / 2 + (\sqrt{3} s_5 (s_1 s_4 + c_4 (c_1 c_2 c_3 + c_1 s_2 s_3))) / 2 - (\sqrt{3} s_4 (c_1 c_2 s_3 - c_1 s_2 s_3)) / 4 - (c_1 c_3 s_2) / 4 + (c_1 c_3 s_2) / 4 - \dot{\theta}_2 ((\sqrt{3} c_3 ((c_4 s_1 - c_4 (s_1 s_2 s_3 + c_2 c_3 s_1))) / 2) / 2 + (\sqrt{3} s_5 ((c_1 c_4) / 2 - (\sqrt{3} (c_2 s_1 s_3 - c_3 s_1 s_2))) / 2 + (s_4 (s_1 s_2 s_3 + c_2 c_3 s_1))) / 2) / 2 + (\sqrt{3} s_5 ((c_1 c_4 - (\sqrt{3} c_4 s_1 s_3 - c_3 s_1 s_2)) / 2 + (\sqrt{3} c_4 s_5 (c_2 s_1 s_3 - c_3 s_1 s_2))) / 2) / 2 + (\sqrt{3} c_5 ((c_4 s_1 / 2 - (c_4 (s_1 s_2 s_3 + c_2 c_3 s_1))) / 2) / 2 - (\sqrt{3} c_5 (s_1 s_2 s_3 + c_2 c_3 s_1))) / 2 + (\sqrt{3} c_4 (s_1 s_2 s_3 + c_2 c_3 s_1)) / 4), \]

\[ Y_{31} = 0, \]

\[ Y_{32} = -\dot{\theta}_2 s_2, \]

\[ Y_{33} = 0, \]

\[ Y_{34} = \dot{\theta}_3 (c_2 s_3 - c_3 s_2) - \dot{\theta}_2 (c_2 s_3 - c_3 s_2), \]

\[ Y_{35} = \dot{\theta}_3 ((c_2 s_3) / 2 - (c_3 s_2) / 2 + (\sqrt{3} s_4 (c_2 c_3 + s_2 s_3)) / 2) / 2 + (\sqrt{3} s_4 (c_2 c_3 + s_2 s_3)) / 2) / 2 + (\sqrt{3} s_4 (s_4 (c_2 s_3 - c_3 s_2))) / 2, \]

\[ Y_{36} = \dot{\theta}_5 ((\sqrt{3} s_5 ((\sqrt{3} (c_2 c_3 + s_2 s_3))) / 2 + (s_4 (c_2 s_3 - c_3 s_2))) / 2) / 2 - (\sqrt{3} c_4 c_5 (c_2 s_3 - c_3 s_2)) / 2 + \dot{\theta}_4 ((\sqrt{3} c_4 (c_2 s_3 - c_3 s_2)) / 4 + (\sqrt{3} s_5 (c_2 s_3 - c_3 s_2)) / 2) / 2 - (\sqrt{3} c_4 c_5 (c_2 s_3 - c_3 s_2)) / 4 - (\sqrt{3} s_4 (c_2 s_3 + s_2 s_3)) / 2 + (\sqrt{3} s_4 (s_4 (c_2 s_3 + s_2 s_3)) / 4) / 2 + (\sqrt{3} s_4 (c_2 s_3 - c_3 s_2)) / 4, \]

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 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 unknown 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. Shahrokshahi, “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)

Open Access This chapter is licensed under the terms of the Creative Commons Attribution 4.0 International License (http://creativecommons.org/licenses/by/4.0/), which permits use, sharing, adaptation, distribution and reproduction in any medium or format, as long as you give appropriate credit to the original author(s) and the source, provide a link to the Creative Commons license and indicate if changes were made.

The images or other third party material in this chapter are included in the chapter’s Creative Commons license, unless indicated otherwise in a credit line to the material. If material is not included in the chapter’s Creative Commons license and your intended use is not permitted by statutory regulation or exceeds the permitted use, you will need to obtain permission directly from the copyright holder.