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
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

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

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

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

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 \[21\]:

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

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

\[ \dot{y} + \lambda_1 y = \lambda_1 \dot{x}, \]  

where \( \lambda_1 \) is the filter constant.
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

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

where $p$ is the Laplace variable.

Combining (1.3) and (1.5), we have

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

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 $a^n_k$. $a^n_k$ 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 $a^n_k$. 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 $a^n_k$ 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

$$e(t) = x(t) - x_d(t), \quad (1.7)$$

**(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

$$\ddot{x}(t) = \ddot{x}_d(t) + k_1 \dot{x}_d(t) - k_2 e(t) - k_1 J \dot{\theta}(t), \quad (1.8)$$
where \( k_1 \) and \( k_2 \) are positive control gains. According to Eq. (1.2), the Eq. (1.8) can be reformulated as 
\[
\ddot{\varepsilon}(t) = \ddot{x}(t) - \ddot{x}_d(t) = k_1 \dot{x}_d(t) - k_2 e(t) - k_1 \dot{x}(t).
\] (1.9)

Eq. (1.9) can be rewritten as
\[
\ddot{\varepsilon}(t) + k_1 \dot{\varepsilon}(t) + k_2 e(t) = 0,
\] (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
\[
\dot{\theta} = \dot{\theta}_j + \dot{\theta}_n
\] (1.11a)
\[
\dot{\theta}_j = \int_0^t J^\dagger [\ddot{x}_d + k_1 \dot{x}_d - k_2 e - k_1 J \dot{\theta}] - \dot{J} \dot{\theta}] \, dt
\] (1.11b)
\[
\dot{\theta}_n = (I - J^\dagger J)\alpha,
\] (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 \( \hat{a}_k(0) = a_k^0 \), then the estimated \( \dot{x}(t) \) is \( \hat{x}(t) = \hat{J} \dot{\theta} \). By replacing \( a_k \) by \( \hat{a}_k \), according to (1.3), the estimated cartesian speed \( \hat{x} \) satisfies
\[
\hat{x} = \hat{J} \dot{q} = Y_k(q, \dot{\theta}) \hat{a}_k,
\] (1.12)

The modified reference trajectory is thus designed as
\[
\dot{\theta}(t) = \int_0^t \left[ \hat{J}^\dagger [\ddot{x}_d + (k_1 + k_2) \dot{x}_d - k_1 k_2 e - \dot{\theta} - k_3 e] - (k_1 + k_2) \dot{\theta} \right] dt.
\] (1.13)
Since the accurate feedback of cartesian velocity \( \dot{x} \) is unavailable, the derivative of tracking error \( \dot{e} = \dot{x} - \dot{x}_d \) is also unknown, therefore, we define the alternative value of \( \dot{e} \) by using the estimated Cartesian speed \( \dot{x} \):

\[
\Delta \dot{x} = \dot{x} - \dot{x}_d = \dot{J}\dot{\theta} - \dot{x}_d, \quad (1.14)
\]

then the updating law of kinematic parameters is designed as

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

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

\[
\dot{e} = \dot{x} - \dot{\hat{x}} + \dot{\hat{x}} - \dot{x}_d \\
= Y_k \hat{a}_k - Y_k \hat{\hat{a}}_k + \dot{\hat{x}} - \dot{x}_d \\
= -Y_k \tilde{a}_k + \Delta \dot{x}. \quad (1.16)
\]

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

\[
\frac{d}{dt}(\Delta \dot{x}) = \dot{J}\dot{\theta} + \dot{J}\dot{\theta} - \dot{x}_d \\
= (k_1 + k_2) \dot{\dot{x}}_d - k_1 k_2 e - k_3 e - k_2 \dot{J}\dot{\theta} - k_1 \dot{J}\dot{\theta} \\
= k_2 \dot{x}_d - k_2 \dot{J}\dot{\theta} - k_1 k_2 e - k_3 e + k_1 \dot{x}_d - k_1 (\dot{x}_d + \dot{\dot{e}} + Y_k \hat{a}_k) \\
= -k_2 \Delta \dot{x} - k_2 k_1 e - k_3 e - k_1 Y_k \hat{a}_k - k_1 \dot{\dot{e}}, \quad (1.17)
\]

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
\[
\frac{d}{dt} (\Delta \hat{x} - k_1 e) = -k_2 (\Delta \hat{x} + k_1 e) - k_3 e - k_1 Y_k \hat{a}_k. \quad (1.18)
\]

Select a Lyapunov function candidate as follows
\[
V = (\Delta \hat{x} + k_1 e)^T (\Delta \hat{x} + k_1 e)/2 + k_3 e^T e/2 + \hat{a}_k^T \hat{a}_k/2. \quad (1.19)
\]

By taking the time derivative of (1.19) and combing (1.15), (1.16) and (1.17), we have
\[
\dot{V} = (\Delta \hat{x} + k_1 e)^T d(\Delta \hat{x} + k_1 e)/dt + k_3 e^T e + \hat{a}_k^T \hat{a}_k
\]
\[
= (\Delta \hat{x} + k_1 e)^T (-k_2 (\Delta \hat{x} + k_1 e) - k_3 e - k_1 Y_k \hat{a}_k) + \hat{a}_k^T (k_1 Y_k^T (\Delta \hat{x} + k_1 e) + k_3 Y_k^T e
\]
\[
- W_k^T (t) \Gamma_1 (W_k(t) \hat{a}_k - y) + k_3 e^T (-Y_k \hat{a}_k + \Delta \hat{x})
\]
\[
= -k_2 (\Delta \hat{x} + k_1 e)^T (\Delta \hat{x} + k_1 e) - k_1 k_3 e^T e - \hat{a}_k^T W_k^T (t) \Gamma_1 W_k(t) \hat{a}_k
\]
\[
\leq 0. \quad (1.20)
\]

Then we can obtain that \( \Delta \hat{x}, e \) and \( \hat{a}_k \) are all bounded. Based on Eqs. (1.14) and (1.3), \( \dot{\hat{\theta}}, \hat{a}_k \) and \( Y_k \hat{a}_k \) are also bounded. Notably that \( W_k(t) \hat{a}_k \) is the output of a stable system with bounded input \( Y_k(t) \hat{a}_k \), we have \( W_k(t) \hat{a}_k \) is also bounded. Then according to Eq. (1.15), \( \hat{a}_k \) is bounded. Differentiating \( W_k(t) \hat{a}_k \) with respect to time, we have
\[
\frac{d}{dt} (W_k(t) \hat{a}_k) = \lambda_1 (Y_k - W_k(t)) \hat{a}_k + W_k(t) \dot{\hat{a}}_k. \quad (1.21)
\]

\( d(W_k(t) \hat{a}_k)/dt \) is also bounded. Then we have \( \dot{\hat{e}}, d(\Delta \hat{x})/dt \) and \( d(W_k(t) \hat{a}_k)/dt \) are all bounded, which means the time derivative of (1.20), \( \dot{V} \) is bounded. Using Barbalat’s Lemma, we have \( \Delta \hat{x} + k_1 e \to 0, e \to 0 \), 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
\[
\hat{\theta}(t) = \int_0^t [J^T (\dot{x}_d + (k_1 + k_2) \dot{x}_d) - k_1 k_2 e - \dot{J} \dot{q} - k_3 e] - (k_1 + k_2) \dot{\theta}] dt. \quad (1.22)
\]
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 \( x_d(t), \dot{x}_d(t) \) and \( \ddot{x}_d(t) \), 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

1. Initialize \( a_k(0) \leftarrow a^n_k \).
2. \( x, \theta, \dot{q} \leftarrow \) Sensor readings
3. Calculate \( e, y, \) and \( W_k(t) \) by Equation (1.7), (1.5) and (1.6)
4. Update \( K \) by Equation (1.27)
5. Update \( \theta_j \) by \( \dot{\theta}_j \leftarrow \) Equation (1.26b)
6. Update \( \theta_n \) by \( \dot{\theta}_n \leftarrow \) Equation (1.26c)
7. calculate the output \( \dot{\theta} \) by Equation (1.26a)
8. 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

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

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

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

where \( \theta_{int}(i) \) 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

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

\[
\dot{\theta}_j = \int_0^t [J^T(\ddot{x}_d + (k_1 + k_2)\dot{x}_d - k_1k_2e - J\dot{\theta} - k_3e) - (k_1 + k_2)\dot{\theta}]dt \tag{1.26b}
\]

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

\[
\dot{a}_k = k_1Y_k^T(\Delta \dot{x} + k_1e) + k_3Y_k^Te - W_k^T(t)\Gamma_1(W_k(t)\hat{a}_k - y) \tag{1.26d}
\]
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:

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

(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 $\dot{K}$ 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 satisfying 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.
1.4.1 Simulation Settings

The vector of initial joint angles is selected as $\theta_{ini} = [\pi/2, -\pi/2, 0, 0]^T \text{rad}$, and the corresponding cartesian position is $x_{ini} = [0.6, 0.3]^T$. Since the exact value of kinematic parameters (see $d_i$ in Fig. 1.2), we assume the nominal values to be $a^n_k = [0.25, 0.25, 0.12, 0.18]^T \text{m}$, and let $\hat{a}_k(0) = a^n_k$. 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 $x_d(t) = 0.4 + 0.2\cos(2t), y_d(t) = 0.3 + 0.2\sin(2t)$. 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
1.4 Numeral Simulations

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 \( ||Y_{\hat{a}_k} - \dot{x}||_2 \) 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

\[
x_d(t) = 0.1(2\sin(2t) - \sin(4t)) + 0.6 \text{ m}, \quad y_d(t) = 0.1(2\cos(2t) - \cos(4t)) + 0.2 \text{ m}
\]

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
1. Adaptive Jacobian Based Trajectory Tracking for Redundant Manipulators

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.
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 $\theta = [\theta_1, \theta_2, \theta_3, \theta_4]^T$ and the estimated $\hat{a}_k = [\hat{a}_k(1), \hat{a}_k(2), \hat{a}_k(3), \hat{a}_k(4)]^T$. 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, $\dot{J}$ can be formulated as follows.

$$\dot{\hat{J}}(1, 1) = -a_1c_1\dot{\theta}_1 - a_2c_{12}(\dot{\theta}_1 + \dot{\theta}_2) - a_3c_{123}(\dot{\theta}_1 + \dot{\theta}_2 + \dot{\theta}_3) - a_4c_{1234}(\dot{\theta}_1 + \dot{\theta}_2 + \dot{\theta}_3 + \dot{\theta}_4),$$
$$\dot{\hat{J}}(1, 2) = -a_2c_{12}(\dot{\theta}_1 + \dot{\theta}_2) - a_3c_{123}(\dot{\theta}_1 + \dot{\theta}_2 + \dot{\theta}_3) - a_4c_{1234}(\dot{\theta}_1 + \dot{\theta}_2 + \dot{\theta}_3 + \dot{\theta}_4),$$
$$\dot{\hat{J}}(1, 3) = -a_3c_{123}(\dot{\theta}_1 + \dot{\theta}_2 + \dot{\theta}_3) - a_4c_{1234}(\dot{\theta}_1 + \dot{\theta}_2 + \dot{\theta}_3 + \dot{\theta}_4),$$
$$\dot{\hat{J}}(1, 4) = -a_4c_{1234}(\dot{\theta}_1 + \dot{\theta}_2 + \dot{\theta}_3 + \dot{\theta}_4).$$
\[
\dot{J}(2, 1) = -a_1s_1\dot{\theta}_1 - a_2s_{12}(\dot{\theta}_1 + \dot{\theta}_2) - a_3s_{123}(\dot{\theta}_1 + \dot{\theta}_2 + \dot{\theta}_3) - a_4s_{1234}(\dot{\theta}_1 + \dot{\theta}_2 + \dot{\theta}_3 + \dot{\theta}_4)
\]
\[
\dot{J}(2, 2) = -a_2s_{12}(\dot{\theta}_1 + \dot{\theta}_2) - a_3s_{123}(\dot{\theta}_1 + \dot{\theta}_2 + \dot{\theta}_3) - a_4s_{1234}(\dot{\theta}_1 + \dot{\theta}_2 + \dot{\theta}_3 + \dot{\theta}_4)
\]
\[
\dot{J}(2, 3) = -a_3s_{123}(\dot{\theta}_1 + \dot{\theta}_2 + \dot{\theta}_3) - a_4s_{1234}(\dot{\theta}_1 + \dot{\theta}_2 + \dot{\theta}_3 + \dot{\theta}_4)
\]
\[
\dot{J}(2, 4) = -a_4s_{1234}(\dot{\theta}_1 + \dot{\theta}_2 + \dot{\theta}_3 + \dot{\theta}_4).
\]

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. 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)

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.