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

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

6.1 Introduction

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

Motion control and force control are two main modes of redundant manipulator control. In motion control problems, a basic assumption is that there is no contact between the robot and the environment, that is, the robot can move freely in the workspace [5]. This problem is well-reflected in coating, welding, stacking, stacking
and other applications. Then the core problem is to design control commands that drive the robot to follow a predetermined trajectory. The control command may be joint angular sequence [6], velocity sequence [7], acceleration sequence [8, 9] or torque sequences [10–12]. The redundancy resolution is usually used to achieve a secondary task, such as avoiding obstacles [13], avoiding singularities [14], etc. Different from motion control, force control involves the direct interaction between a robot and its environment. The control of contact force can enhance the robustness and flexibility of the robot in the weak structure environment, so as to enhance the robot’s operation ability [15]. Corresponding typical applications in polishing, grinding, assembly, polishing, polishing and other fields [16, 17]. In [18], the theoretical framework of impedance control is proposed. The basic idea is to use the environment as admittance and the robot as impedance. By maintaining the dynamic relationship between force and motion, the controller is represented as a spring-mass-damping system. In [19], a hybrid position force controller is proposed which combines position information with force information to realize the simultaneous control of position constraint and force constraint. Based on these two control frameworks, a series of controllers are proposed and verified by simulation or experiments [20–22].

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

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

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

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

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

6.2 Problem Formulation

6.2.1 Problem Formulation

In this chapter, we focus on position-force control problem for redundant manipulators. Figure 6.1 gives a brief introduction of a redundant robot and its operation on an
Table 6.1  Comparisons among the Proposed Motion-force Control Scheme and Existing Ones

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

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

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

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

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

(6.1)

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

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

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

(6.2)

where \( J(\theta(t)) = \partial f(\theta(t))/\partial \theta(t) \) is called Jacobian matrix.
In position-force control tasks, the end-effector’s motion is constrained by the contact surface. For simplicity, we define a tool coordinate system as $R_t(x_t, y_t, z_t)$, in which the axis $z_t$ is set in alignment with the vertical direction of the contact surface. Obviously, the motion of end-effector can be specified along $x_t$ and $y_t$. In this chapter, frictional force between the robot and contact surface is ignored, therefore, the contact force $F$ is in alignment with $z_t$.

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

$$F_t = k_f \Sigma_t \delta X_t, \quad (6.3)$$

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

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

$$e_t = \bar{\Sigma}_t \delta X_t, \quad (6.4)$$

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

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

$$\begin{cases}
F = k_f S_t^T \Sigma_t S_t \delta X, \\
e_0 = S_t^T \bar{\Sigma}_t S_t \delta X.
\end{cases} \quad (6.5)$$

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

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

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

In real implementations, given the desired contact force $F_s$ and trajectory command $x_d$, the manipulator’s end-effector is expected to offer contact force $F_d$ while tracking $x_d$, i.e., $F \rightarrow F_d$, $e_0 \rightarrow 0$. For the convenience of writing in the following
sections, let $A = [k_f S_f^T S_f; S_f^T S_f]$, $r = [F^T, e_0^T]^T$, and $r_d = [F_d; 0]$. Then (6.6) can be reformulated as

$$A(f(\theta) - x_d) = r.$$

(6.7)

Therefore, the control objective of position-force control is to adjust the joint angles $\theta$, to ensure $r \rightarrow r_d$.

### 6.2.2 Joint Torque and Physical Constraints

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

$$\tau = J^T(\theta) F.$$

(6.8)

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

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

### 6.2.3 Optimization Problem Formulation

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

$$\begin{align*}
\min_{\theta} & \quad G_1 = \tau^T \tau/2, \\
\text{s.t.} & \quad \tau = J^T(\theta) F, \\
& \quad r_d = A(f(\theta) - x_d), \\
& \quad \theta_{\min} \leq \theta \leq \theta_{\max}, \\
& \quad \dot{\theta}_{\min} \leq \dot{\theta} \leq \dot{\theta}_{\max}, \\
& \quad \tau_{\min} \leq \tau \leq \tau_{\max},
\end{align*}$$

(6.9a-6.9f)

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

$$
\text{min } G_1 = F^T J(\theta) J^T(\theta) F / 2,
$$
(6.10a)

s.t.

$$
r_d = A (f(\theta) - x_d),
$$
(6.10b)

$$
\dot{\theta}^\text{min} \leq \dot{\theta} \leq \dot{\theta}^\text{max},
$$
(6.10c)

$$
\ddot{\theta}^\text{min} \leq \ddot{\theta} \leq \ddot{\theta}^\text{max},
$$
(6.10d)

$$
\tau^\text{min} \leq \tau \leq \tau^\text{max}.
$$
(6.10e)

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

### 6.3 Reconstruction of Optimization Problem

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

#### 6.3.1 Reconstruction of Objective Function

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

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

Differentiating $G_2$ with respect to time, we have

$$
\dot{G}_2 = (J^T(\theta) F_d)^T \frac{d(J^T(\theta) F_d)}{dt}.
$$
(6.11)

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

$$\frac{d}{dt}(J^T(\theta)F_d) = \sum_{i=1}^{n} \frac{\partial(J^T(\theta)F_d)}{\partial \theta_i} \dot{\theta}_i + J^T(\theta)\dot{F}_d$$  \hspace{1cm} (6.12)

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

$$H_i = \frac{\partial(J^T(\theta)F_d)}{\partial \theta_i} = \left[ \frac{\sum_{j=1}^{m}(\partial(J(j, 1)F_d(j))/\partial \theta_i)}{\sum_{j=1}^{m}(\partial(J(j, 2)F_d(j))/\partial \theta_i)} \cdots \frac{\sum_{j=1}^{m}(\partial(J(j, n)F_d(j))/\partial \theta_i)}{\sum_{j=1}^{m}(\partial(J(j, n)F_d(j))/\partial \theta_i)} \right].$$

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

$$\dot{G}_2 = F_d^T J H \dot{\theta} + F_d^T J J^T \dot{F}_d.$$  \hspace{1cm} (6.13)

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

### 6.3.2 Reconstruction of Constraints

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

$$e = A(f(\theta) - x_d) - r_d.$$  \hspace{1cm} (6.14)

Differentiating $e$ and combing (6.2) yields

$$\dot{e} = A(J \dot{\theta} - \dot{x}_d) - \dot{r}_d.$$  \hspace{1cm} (6.15)

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

$$AJ \dot{\theta} = \dot{r}_d + A \dot{x}_d - k(Af(\theta) - x_d).$$  \hspace{1cm} (6.16)

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

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

$$g(\omega) \leq 0,$$

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

### 6.3.3 Reformulation and Convexification

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

$$\begin{align*}
\min & \quad F_d^T J H \omega, \\
\text{s.t.} & \quad r_r = AJ \omega, \\
& \quad g(\omega) \leq 0, \\
& \quad \omega \in \Omega,
\end{align*}$$

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

$$\begin{align*}
\min & \quad F_d^T J H \omega + (AJ \omega - r_r)^T (AJ \omega - r_r), \\
\text{s.t.} & \quad r_r = AJ \omega, \\
& \quad g(\omega) \leq 0, \\
& \quad \omega \in \Omega.
\end{align*}$$

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

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

6.4.1 RNN Design

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

\[
L = F_d^T J H \omega + (AJ \omega - r_r)^T (AJ \omega - r_r) + \lambda_1^T (r_r - AJ \omega) + \lambda_2^T g(\omega).
\]  
(6.21)

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

\[
\omega = P_\Omega(\omega - \frac{\partial L}{\partial \omega}), \quad \text{(6.22a)}
\]

\[
r_r = AJ \omega, \quad \text{(6.22b)}
\]

\[
\lambda_2 = (\lambda_2 + g(\omega))^+, \quad \text{(6.22c)}
\]

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

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

\[
\varepsilon \dot{\omega} = -\omega + P_\Omega(\omega - H^T J^T F_d - J^T A^T (AJ \omega - r_r)) + J^T A^T \lambda_1 - \nabla g \lambda_2, \quad \text{(6.23a)}
\]

\[
\varepsilon \dot{\lambda}_1 = r_r - AJ \omega, \quad \text{(6.23b)}
\]

\[
\varepsilon \dot{\lambda}_2 = (\lambda_2 - (\lambda_2 + g(\omega))^+), \quad \text{(6.23c)}
\]

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

Input: Control parameters $\epsilon$, $k$, $\alpha$, stiffness coefficient $k_f$, prior knowledge of the contact surface $S_t$, $\Sigma_t$ and $\bar{\Sigma}_t$. The robot’s joint limits $\theta^i_{\text{max}}, \theta^i_{\text{min}}$ and joint speed limits $\dot{\theta}^i_{\text{max}}, \dot{\theta}^i_{\text{min}}$, initial joint angles $\theta(0)$, desired tracking trajectory $r_d(t)$, $\dot{r}_d(t)$, and contact force $F_d$. Feedback of actual coordinate $x(t)$, contact force $F(t)$, and joint angle $\theta$, task duration $T$.

Output: To achieve position-force control of the redundant manipulator and optimize joint torque

1: Initialize $\lambda_1(0), \lambda_2(0)$

2: Repeat

3: On-line feedback of $F$, $\theta$, $x$ ← from sensors

4: Calculate the weight matrix $A$

5: Calculate the reference command $r_r$ ← Eq. (6.19)

6: Update joint velocity command $\omega$ by $\dot{\omega}$ ← Eq. (6.23a)

7: Update state variable $\lambda_1$ by $\dot{\lambda}_1$ ← Eq. (6.23b)

8: Update state variable $\lambda_2$ by $\dot{\lambda}_2$ ← Eq. (6.23c)

Until ($t > T$)

6.4.2 Stability Analysis

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

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

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

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

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

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

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

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

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

$$\epsilon \dot{\xi} = -\xi + P_{\bar{\Omega}}[\xi - F(\xi)], \quad (6.25)$$
where $F(\xi) = [F_1(\xi), F_2(\xi), F_3(\xi)]^T \in \mathbb{R}^{2n+3n}$, in which

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

Let $\nabla F(\xi) = \frac{\partial F}{\partial \xi}$, we have

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

(6.26)

It is remarkable that

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

(6.27)

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

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

6.5 Illustrative Examples

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

6.5.1 Simulation Setup

As shown in Fig. 6.2, a contact surface in the workspace can be described as $y = 0$, the end-effector can move freely along the horizontal axis, and the desired contact force $F_d$ is aligned with the vertical direction. The stiffness coefficient $k_f$ is
set to be 1000N/mm. Control positive control gains are set as $\alpha = 10$, $\beta = 10$, $k = 8$, $\varepsilon = 0.005$, respectively. Physical constraints of joint angles, velocities and torque are defined as $\theta_{\text{min}} = [-2, -2, -2, -2]^T \text{rad}$, $\theta_{\text{max}} = [2, 2, 2, 2]^T \text{rad}$, $\dot{\theta}_{\text{min}} = [-2, -2, -2, -2]^T \text{rad/s}$, $\dot{\theta}_{\text{max}} = [2, 2, 2, 2]^T \text{rad/s}$, $\tau_{\text{min}} = [-10, -10, -10, -10]^T \text{Nm}$, $\tau_{\text{max}} = [10, 10, 10, 10]^T \text{Nm}$, respectively.

### 6.5.2 Position-Force Control Without Optimization

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

6.5.3 Position-Force Control with Optimization

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

(1) Position-Force Control on A Fixed Point

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

(2) Position-Force Control Along A Straight Line

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

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

(3) Position-Force Control Along An Arc Surface

In this part, the end-effector is controlled to track a quarter-circular surface, which is centered at $[0.3, 0.3]^T$ m with radius 0.2 m, and is provided a constant force of 10 N in the vertical direction. The initial values of joint angles are selected as $\theta_0 = [1.5708, -0.9851, -1.1714, 0]^T$ rad. Numerical results are shown in Fig. 6.6. The trajectory of end-effector is shown in Fig. 6.6a, while in Fig. 6.6b, optimization is not introduced. The proposed controller enables the robot to achieve precision control of both position and force, and at the same time, by adjusting its joint angles, the
Fig. 6.4 Numerical results when the robot is controlled to offer constant force at a fixed point. a Profiles of the end-effector (black dashed line) and the corresponding joint configurations. b Profiles of position error. c Profiles of contact force. d Comparison of Euclidean norm of joint torque with and without optimization. e Profiles of joint angles. f Profiles of joint velocities.

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

(4) Adaptability to Different Initial Settings

To further illustrate the joint optimization scheme, another fixed-point control is presented. Desired signals are set as $x_d = [0.3, 0]^T$ and $F_d = [0, -10]^T$. The initial
values of joint angles is selected as $\theta_0 = [1.8850, -1.8850, -1.2566, 0]^T$ rad, consequently, the corresponding position of the end-effector is exactly the same as $x_d$. As shown in Fig. 6.7a, the robot adjusts its posture and stops in final state, while keeping its end-effector on the fixed point. This phenomenon is similar to the null-space movement based on pseudo-inverse methods. However, different with pseudo-inverse
Fig. 6.6  Numerical results when tracking a time varying force command along an arc surface with optimization. a Time history of the end-effector (black curve) and the corresponding joint configurations with optimization. b Time history of the end-effector (black curve) and the corresponding joint configurations without optimization. c Profiles of contact force. d Profiles of position error. e Comparison of Euclidean norm of joint torque with and without optimization. f Profiles of joint angles

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

Q1: “What’s the complexity of the proposed RNN?”

Answer: The network is organized in a one-layer architecture, which consists of $2m + 3n$ neurons, namely $\omega \in \mathbb{R}^n$, $\lambda_1 \in \mathbb{R}^{2m}$ and $\lambda_2 \in \mathbb{R}^{2n}$. Despite the difference between the proposed neural network with traditional recurrent neural networks, from both the mathematical description Eq. (23) and the architecture, one characteristic of the established neural network can be found that the neural network uses its historical information to calculate the output at current moment, which is also a typical feature of recurrent neural networks.
Fig. 6.8 Time history of robot’s joint configurations in fixed point control from different initial joint angle $\theta_0$. a $\theta_0 = [0.9, -0.75, -1.5, -1.6]^T$ rad. b $\theta_0 = [1.8, -0.3, -1.6, 0.6]^T$ rad. c $\theta_0 = [1.9, -1.5, -1.6, -0.6]^T$ rad. d $\theta_0 = [0.5, -0.5, -1.6, -0.6]^T$ rad. e $\theta_0 = [0.7, -0.3, -2, 0.2]^T$ rad. f $\theta_0 = [0.3, -1.5, -1.6, -0.6]^T$ rad

Q2: “As described in Section II, the matrices $\Sigma_f$ and $\bar{\Sigma}_f$ are crucial in the controller design, however, the authors didn’t show the details. How to obtain those matrices in actual applications requires detailed description.”

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

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

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

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

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

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

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

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

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

6.7 Summary

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

References

1. C. Yang, Y. Jiang, J. Na, Z. Li, L. Cheng and C. Su, “Finite-time Convergence Adaptive Fuzzy Control for Dual-Arm Robot with Unknown Kinematics and Dynamics,” IEEE Trans. Fuzzy Syst., to be published, https://doi.org/10.1109/tafuzz.2018.2864940, 2018
2. H. Wu, Y. Guan, J. Rojas, A latent state-based multimodal execution monitor with anomaly detection and classification for robot introspection. Applied Sciences. 9(6), 1072 (2019)
3. H. Wu, Z. Xu, W. Yan, Q. Su, S. Li, T. Cheng, X. Zhou, Incremental Learning Introspective Movement Primitives From Multimodal Unstructured Demonstrations. IEEE Access. 15(7), 159022–36 (2019)

4. Wu H, Guan Y, Rojas J. “Analysis of multimodal Bayesian nonparametric autoregressive hidden Markov models for process monitoring in robotic contact tasks.” International Journal of Advanced Robotic Systems. 2019 26(12):1729881419834840

5. L. Cheng, Z. Hou, M. Tan, Adaptive Neural Network Tracking Control for Manipulators with Uncertain Kinematics, Dynamics and Actuator Model. Automatica 45(10), 2312–2318 (2009)

6. L. Cheng, W. Liu, Z. Hou, Z. Li, M. Tan, Neural Network Based Nonlinear Model Predictive Control for Piezoelectric Actuators. IEEE Trans. Ind. Electron. 62(12), 7717–7727 (2015)

7. H. Wang, W. Liu, J. Qiu, P. Liu, Adaptive Fuzzy Decentralized Control for A Class of Strong Interconnected Nonlinear Systems with Unmodeled Dynamics. IEEE Trans. Fuzzy Syst. 26(2), 836–846 (2018)

8. D. Guo, Y. Zhang, Acceleration-Level Inequality-Based MAN Scheme for Obstacle Avoidance of Redundant Robot Manipulators. IEEE Trans. Ind. Electron. 61(12), 6903–6914 (2014)

9. J. Ren, B. Wang, M. Cai and J. Yu, “Adaptive Fast Finite-Time Consensus for Second-Order Uncertain Nonlinear Multi-Agent Systems With Unknown Dead-Zone,” IEEE Access, vol. 8, No. 1, pp. 25557-25569, 2020

10. J. Na, X. Ren, D. Zheng, Adaptive Control for Nonlinear Pure-feedback Systems with High-order Sliding Mode Observer. IEEE Trans. Neur. Net. Lear. 24(3), 370–382 (2013)

11. C. Yang, Y. Jiang, W. He, J. Na, Z. Li, B. Xu, Adaptive Parameter Estimation and Control Design for Robot Manipulators with Finite-Time Convergence. IEEE Trans. Ind. Electron. 65(10), 8112–8123 (2018)

12. Y. Liu, S. Tong, Barrier Lyapunov Functions-based Adaptive Control for A Class of Nonlinear Pure-feedback systems with Full State Constraints. Automatica 64(1), 70–75 (2016)

13. D. Guo and Y. Zhang, “A New Inequality-Based Obstacle-Avoidance MVN Scheme and Its Application to Redundant Robot Manipulators,” IEEE Trans. Syst., Man, Cybern. C, Appl. Rev., vol. 42, no. 6, pp. 1326-1340, 2012

14. W. Xu, J. Zhang, B. Liang, B. Li, Singularity Analysis and Avoidance for Robot Manipulators with Non-Spherical Wrists. IEEE Trans. Ind. Electron. 63(1), 277–290 (2016)

15. N. Hogan, On the Stability of Manipulators Performing Contact Tasks. IEEE J. Robot. Automat. 4(6), 677–686 (1988)

16. O. Khatib, A Unified Approach for Motion and Force Control of Robot Manipulators: The Operational Space Formulation. IEEE J. Robot. Automat. 3(1), 43–53 (2003)

17. K. Kiguchi, T. Fukuda, Position/force Control of Robot Manipulators for Geometrically Unknown Objects Using Fuzzy Neural Networks. IEEE Trans. Ind. Electron. 47(3), 641–649 (2000)

18. N. Hogan, “Impedance Control: An Approach to Manipulation, Parts I, II, III,” in J. Dyn. Syst., Meas. Control., vol. 107, no. 1, pp. 1-23, 1985

19. J. Craig, P. Hsu and S. Sastry “Adaptive Control of Mechanical Manipulators,” in Proc. IEEE Int. Conf. Robotics and Automation., San Francisco, Ca, USA., 1986, pp. 190-195

20. Z. Li, J. Liu, Z. Huang, Y. Peng, H. Pu, L. Ding. Adaptive Impedance Control of HumanCRobot Cooperation Using Reinforcement Learning. IEEE Trans. Ind. Electron. 64(10), 8013–8022 (2017)

21. D. Chen, S. Li, Q. Wu, X. Luo, Super-twisting ZNN for coordinated motion control of multiple robot manipulators with external disturbances suppression. Neurocomputing 371(1), 78–90 (2020)

22. Dechao Chen, Shuai Li, “A recurrent neural network applied to optimal motion control of mobile robots with physical constraints,” Applied Soft Computing, to be published, 2019, https://doi.org/10.1016/j.asoc.2019.105880

23. B. Nemec, L. Zlajpah, Force Control of Redundant Robots in Unstructured Environment. IEEE Trans. Ind. Electron. 49(1), 233–240 (2002)

24. R. Patel, H. Talebi, J. Jayender, F. Shadpey, A Robust Position and Force Control Strategy for 7-DOF Redundant Manipulators. IEEE/ASME Trans. Mechatron. 14(5), 575–589 (2009)
25. R. Patel, F. Shadpey, “Control of Redundant Robot Manipulators,” Lecture Notes in Control (Springer, Information Sciences., 2005)
26. W. He, Z. Yan, C. Sun, Adaptive Neural Network Control of A Marine Vessel with Constraints Using The Symmetric Barrier Lyapunov Function. IEEE Trans. Cybern. 47(7), 1641–1651 (2017)
27. Y. Zhang, J. Wang, Y. Xia, A Dual Neural Network For Redundancy Resolution of Kinematically Redundant Manipulators Subject to Joint Limits and Joint Velocity Limits. IEEE Trans. Neural Networks 14(4), 658–667 (2003)
28. L. Jin, Y. Zhang, G2-type SRMPC Scheme For Synchronous Manipulation of Two Redundant Robot Arms. IEEE Trans. Cybern. 45(2), 153–164 (2015)
29. B. Cai, Y. Zhang, Different-Level Redundancy-Resolution and Its Equivalent Relationship Analysis for Robot Manipulators Using Gradient-Descent and Zhang’s Neural-Dynamic Methods. IEEE Trans. Ind. Electron. 59(8), 3146–3155 (2012)
30. Y. Zhang, S. Li, J. Gui, X. Luo, Velocity-Level Control with Compliance to Acceleration-Level Constraints: A Novel Scheme for Manipulator Redundancy Resolution. IEEE Trans Ind. Informat. 14(3), 921–930 (2018)
31. S. Li, Y. Zhang, L. Jin, Kinematic Control of Redundant Manipulators Using Neural Networks. IEEE Trans. Neur. Net. Lear. 28(10), 2243–2254 (2017)
32. L. Jin, S. Li, H. La, X. Luo, Manipulability Optimization of Redundant Manipulators Using Dynamic Neural Networks. IEEE Trans. Ind. Electron. 64(6), 4710–4720 (2017)
33. S. Li, S. Chen, B. Liu, Y. Li, Y. Liang, Decentralized Kinematic Control of A Class of Collaborative Redundant Manipulators via Recurrent Neural Networks. Neurocomputing. 91(1), 1–10 (2012)
34. Z. Xu, S. Li, X. Zhou, Y. Wu, T. Cheng and D. Huang, “Dynamic Neural Networks Based Kinematic Control for Redundant Manipulators with Model Uncertainties,” Neurocomputing., to be published, https://doi.org/10.1016/j.neucom.2018.11.001
35. Z. Xu, S. Li, X. Zhou, T. Cheng, Dynamic Neural Networks Based Adaptive Admittance Control for Redundant Manipulators with Model Uncertainties. Neurocomputing 357(1), 271–281 (2019)
36. Z. Xu, S. Li, X. Zhou, W. Yan, T. Cheng, H. Dan, “Dynamic Neural Networks for Motion-Force Control of Redundant Manipulators: An Optimization Perspective”, IEEE transactions on industrial electronics. Early access (2020). https://doi.org/10.1109/TIE.2020.2970635
37. X. Li, Z. Xu, S. Li, H. Wu, X. Zhou, “Cooperative Kinematic Control For Multiple Redundant Manipulators Under Partially Known Information Using Recurrent Neural Network”. IEEE Access 8(1), 40029–40038 (2020)
38. X. Gao, Exponential stability of globally projected dynamic systems. IEEE Trans. Neural Netw. 14(2), 426–431 (2003)

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.