Chapter 4
Deep RNN Based Obstacle Avoidance Control for Redundant Manipulators

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

4.1 Introduction

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

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

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

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

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

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

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

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

4.2 Problem Formulation

4.2.1 Basic Description

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

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

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

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

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

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

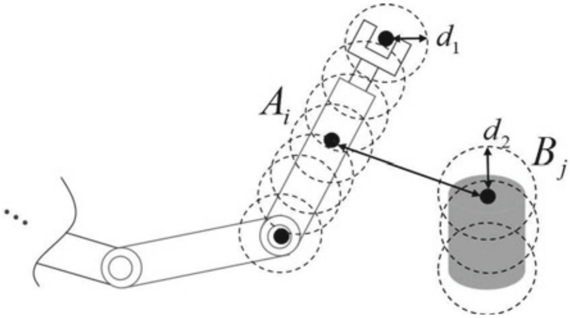


Fig. 4.1 The basic idea of obstacle avoidance in this chapter

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

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

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

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

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

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

4.2.2 *Reformulation of Inequality in Speed Level*

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

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

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

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

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

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

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

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

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

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

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

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

4.2.3 QP Type Problem Description

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

$$\min \quad \dot{\theta}^{\mathrm{T}}\dot{\theta}/2, \tag{4.10a}$$

$$s.t. \quad x = x_{\mathrm{d}}, \tag{4.10b}$$

$$\theta^{-} \le \theta \le \theta^{+}, \tag{4.10c}$$

$$\dot{\theta}^{-} \le \dot{\theta} \le \dot{\theta}^{+}, \tag{4.10d}$$

$$J_o\dot{\theta} \le B. \tag{4.10e}$$

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

$$\min \quad \dot{\theta}^{\mathrm{T}}\dot{\theta}/2, \tag{4.11a}$$

$$s.t. \quad J(\theta)\dot{\theta} = \dot{x}_{\mathrm{d}} + k(x_{\mathrm{d}} - x), \tag{4.11b}$$

$$\max(\alpha(\theta^{-} - \theta), \dot{\theta}^{-}) \le \dot{\theta} \le \min(\dot{\theta}^{+}, \alpha(\theta^{+} - \theta)), \tag{4.11c}$$

$$J_o\dot{\theta} \le B. \tag{4.11d}$$

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

4.3 Deep RNN Based Solver Design

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

4.3.1 Deep RNN Design

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

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

λ_1 and λ_2 are the dual variables corresponding to the constraints (4.11b) and (4.11d). According to Karush-Kuhn-Tucker conditions, the optimal solution of the optimization problem (4.12) can be equivalently formulated as

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

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

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

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

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

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

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

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

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

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

with ε is a positive constant scaling the convergence of (4.15).

Remark 4.2 As to the established deep RNN (4.15), the first dynamic equation is also the output dynamics, which combines the dynamics of state variables λ_1 and λ_2, as well as the physical limitations including joint angles and velocities. State variable λ_1 is used to ensure the equality constraint (4.11b), as shown in (4.15b), λ_1 is updated according to the difference between reference speed $\dot{x}_{\mathrm{d}} + k(x_{\mathrm{d}} - x)$ and actually speed $J(\theta)\dot{\theta}$. Similarly, λ_2 is used to ensure the inequality constraint (4.11d), which will be further discussed later. It is remarkable that ε plays an important role in the convergence of the deep RNN. The smaller ε, the faster the deep RNN converges.

Remark 4.3 By introducing the model information such as J, J_o, etc., the established deep RNN consists of three class of nodes, namely $\dot{\theta}$, λ_1 and λ_2, and the total number of nodes is $n + m + ab$. Comparing to traditional neural networks in [19], the complexity of neural networks is greatly reduced.

4.3.2 Stability Analysis

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

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

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

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

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

Lemma 4.2 *[37] Let $V : [0, \infty) \times B_d \to \mathbb{R}$ be a C^1 function, α_1, α_2 be class-K functions defined on $[0, d)$ which satisfy $\alpha_1(||x||) \leq V(t, x) \leq \alpha_2(||x||)$, $\forall (t, x) \in [0, d) \times B_d$, then $x = 0$ is a uniformly asymptotically stable equilibrium point if there exist some class-K function g defined on $[0, d)$, to make the following inequality hold*

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

According to the definition of monotone function, we can readily obtain that $F(\xi)$ is monotone. From the description of (4.24), the projection operator P_S can be formulated as $P_S = [P_\Omega; P_R; P_\Lambda]$, in which P_Ω is defined in (4.13), P_R can be regarded as a projection operator of λ_1 to R, with the upper and lower bounds being $\pm\infty$, and $P_\Lambda = (\bullet)^+$ is a special projection operator to closed set $\mathbb{R}_+^{ab}$. Therefore, P_S is a projection operator to closed set $[\Omega; \mathbb{R}^m; \mathbb{R}_+^{ab}]$. Based on Lemma 4.1, the proposed neural network (4.15) is stable and will globally converge to the optimal solution of (4.10). The proof is completed. □

4.4 Numerical Results

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

4.4.1 Simulation Setup

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

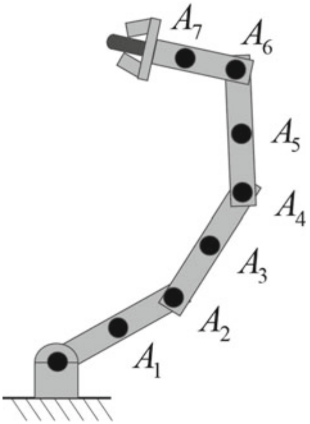


Fig. 4.2 The planar robot to be simulated in this chapter

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

4.4.2 Single Obstacle Avoidance

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

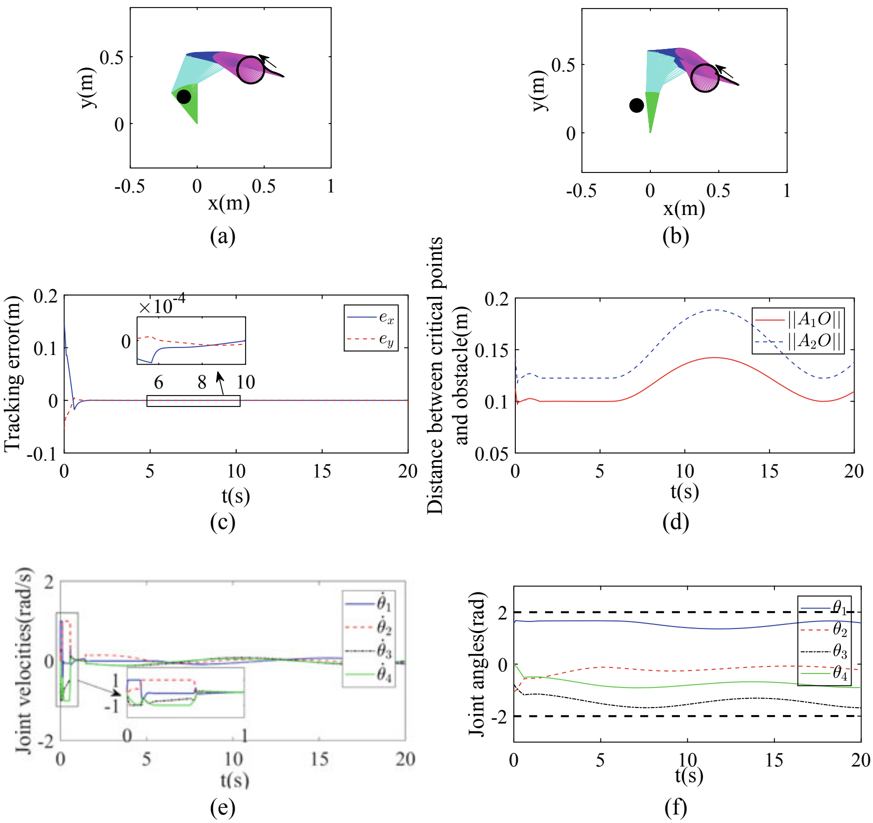


Fig. 4.3 Numerical results of single obstacle avoidance. **a** is the motion trajectories when ignoring obstacle avoidance scheme, **b** is the motion trajectories when considering obstacle avoidance scheme, **c** is the profile of tracking errors when considering obstacle avoidance scheme, **d** is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

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

4.4.3 Discussion on Class-K Functions

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

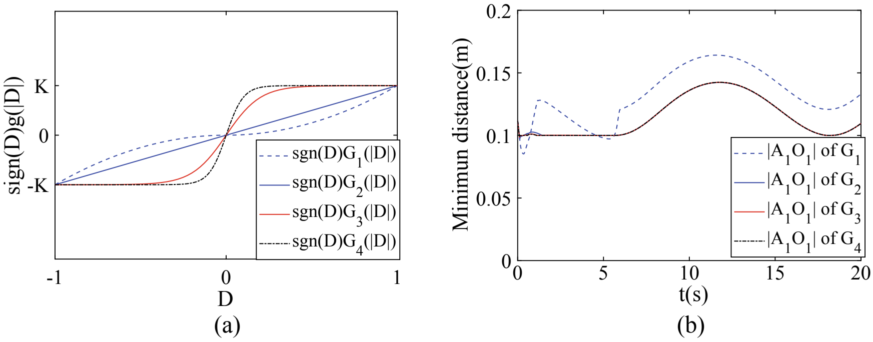
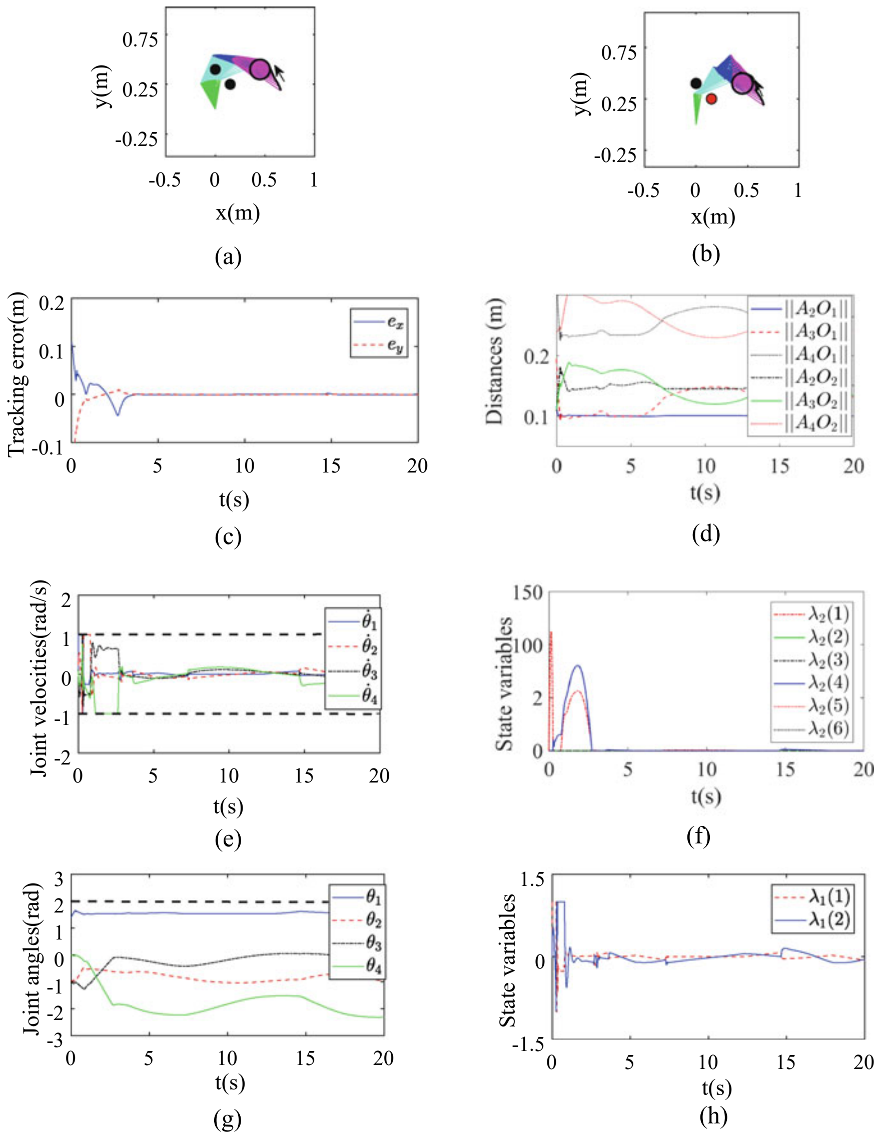


Fig. 4.4 Discussions on different obstacle avoidance functions. **a** is the comparative curves of different obstacle avoidance functions. **b** is the profile of minimum distance of the robot and obstacle using different obstacle avoidance functions

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

4.4.4 Multiple Obstacles Avoidance

In this part, we consider the case where there are two obstacles in the workspace. The obstacles are set at $[0.1, 0.25]^{\mathrm{T}}$m and $[0, 0.4]^{\mathrm{T}}$m, respectively. Simulation results are shown in Fig. 4.5. The desired path is defined as $x_{\mathrm{d}} = [0.45 + 0.1cos(0.5t), 0.4 + 0.1sin(0.5t)]^{\mathrm{T}}$. The initial joint angle of the robot is selected as $\theta_0 = [1.5, -1 - 1, 0]^{\mathrm{T}}$. To further show the effectiveness of the proposed obstacle avoidance strategy 4.5, $g|D|$ is selected as $g|D| = K_1/(1 + e^{-|D|}) - K_1/2$ with $K_1 = 200$. When λ_2 is set to 0, as shown in Fig. 4.5a, the inequality constraint (4.11d) will not work, in other words, only kinematic tracking problem is considered rather than obstacle avoidance, in this case, the robot would collide with the obstacles. After introducing online training of λ_2, the simulation results are given in Fig. 4.5b–h. The tracking errors are shown in Fig. 4.5c, with the transient time being about 4s, and steady state error less than 1×10^{-3}m. Correspondingly, the robot moves fast in the transient stage, ensuring the quick convergence of the tracking errors. It is remarkable that the distances between the critical points and obstacle points are kept larger than 0.1m at all times, showing the effectiveness of the proposed method. At $t = 14$s,



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

Fig. 4.5 Numerical results of multiple obstacle avoidance. **a** is the motion trajectories when ignoring obstacle avoidance scheme. **b** is the motion trajectories when considering obstacle avoidance scheme. **c** is the profile of tracking errors when considering obstacle avoidance scheme. **d** is the profile of distances between critical points and obstacles. **e** is the profile of joint velocities. **f** is the profile of λ_2. **g** is the profile of joint angles. **h** is the profile of λ_1

from Fig. 4.5d and g, when the distance between A_3 and O_1 is close to 0.1m, the corresponding dual variable λ_2 becomes positive, making the inequality constraint (4.11d) hold, and the boundary between the robot and obstacle is thus guaranteed. After $t = 18$s, $||A_3O_1||$ becomes greater, then λ_2 converges to aero. Notable that although λ_1 and λ_2 do not converge to certain values, the dynamic change of λ_1 and λ_2 ensures the regulation of the proposed deep RNN.

4.4.5 Enveloping Shape Obstacles

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

4.4.6 Comparisons

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

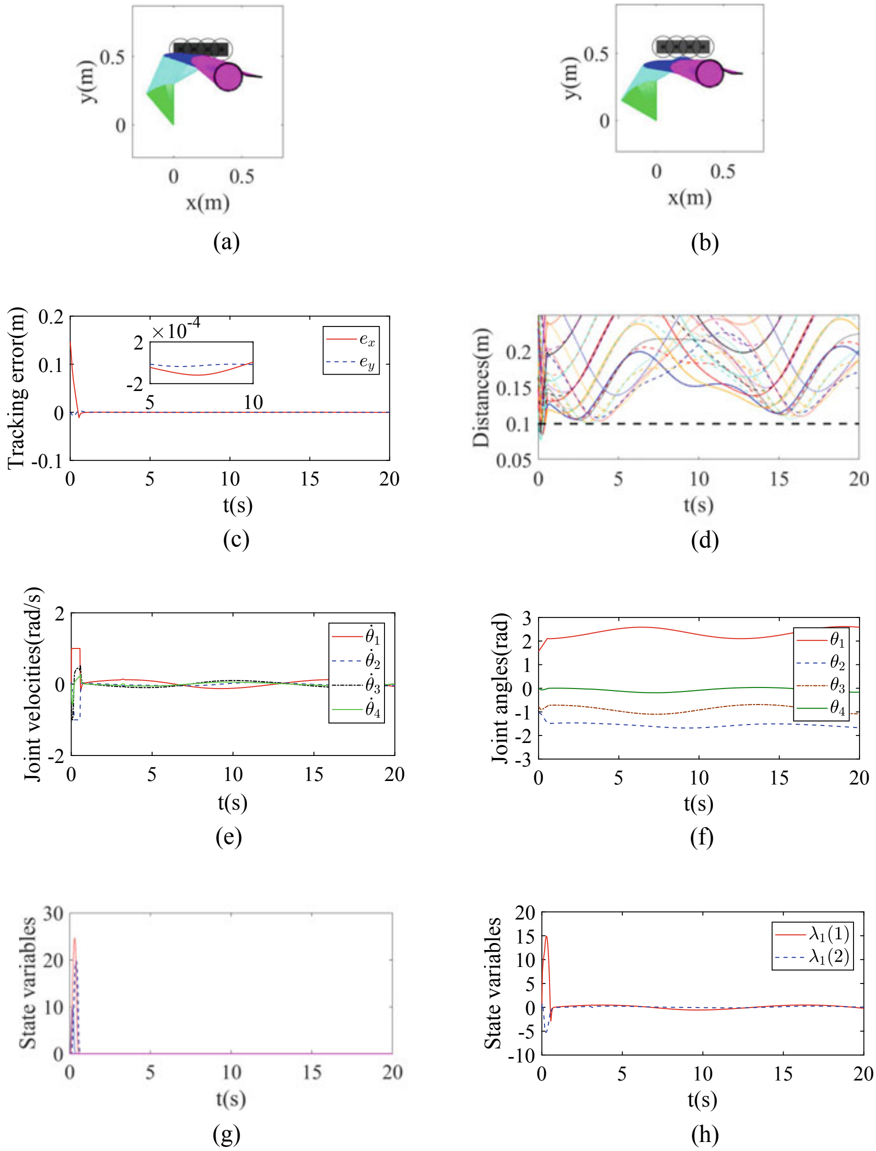


Fig. 4.6 Numerical results of enveloping shape obstacles. **a** is the motion trajectories when ignoring obstacle avoidance scheme. **b** is the motion trajectories when considering obstacle avoidance scheme. **c** is the profile of tracking errors when considering obstacle avoidance scheme. **d** is the profile of distances between critical points and obstacles. **e** is the profile of joint velocities. **f** is the profile of joint angles. **g** is the profile of λ_2. **h** is the profile of λ_1

Table 4.1 Comparisons among different obstacle avoidance controllers on manipulators

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

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

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

4.5 Summary

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

References

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

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

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