Cronfa - Swansea University Open Access Repository

_____

This is an author produced version of a paper published in:
Frontiers in Neurorobotics

_____

Cronfa URL for this paper:

http://cronfa.swan.ac.uk/Record/cronfa52000

_____

Paper:

Xu, Z., Zhou, X. & Li, S. (2019). Deep Recurrent Neural Networks Based Obstacle Avoidance Control for Redundant Manipulators. _Frontiers in Neurorobotics, 13_

http://dx.doi.org/10.3389/fnbot.2019.00047

_____

http://www.swansea.ac.uk/library/researchsupport/ris-support/

Check for updates

Deep Recurrent Neural Networks Based Obstacle Avoidance Control for Redundant Manipulators

Zhihao Xu[1], Xuefeng Zhou[1] and Shuai Li[2]*

[1] *Guangdong Key Laboratory of Modern Control Technology, Guangdong Institute of Intelligence Manufacturing, Guangzhou, China,* [2] *School of Engineering, Swansea University, Swansea, United Kingdom*

Obstacle avoidance is an important subject in the control of robot manipulators, but is remains challenging for robots with redundant degrees of freedom, especially when there exist complex physical constraints. In this paper, we propose a novel controller based on deep recurrent neural networks. By abstracting robots and obstacles into critical point sets respectively, the distance between the robot and obstacles can be described in a simpler way, then the obstacle avoidance strategy is established in form of inequality constraints by general class-K functions. Using minimal-velocity-norm (MVN) scheme, the control problem is formulated as a quadratic-programming case under multiple constraints. Then a deep recurrent neural network considering system models is established to solve the QP problem online. Theoretical conduction and numerical simulations show that the controller is capable of avoiding static or dynamic obstacles, while tracking the predefined trajectories under physical constraints.

Keywords: recurrent neural network, redundant manipulator, obstacle avoidance, zeroing neural network, motion plan

1. INTRODUCTION

As industrial automation develops, robot manipulators have been used in a wide range of applications such as painting, welding, assembly, etc., (Cheng et al., 2009; Yang et al., 2018a). With the evolution of intelligent manufacturing, the way robot works is also changing. In order to fulfill more difficult tasks in complex environment, the robot is required to have better execution capabilities (Pan et al., 2018). Therefore, robots with redundant DOFs have attracted much attention in the field of robotic control since its wonderful flexibility (Chan and Dubey, 1995; Zhang, 2015).

Obstacle avoidance is a core problem in the control of redundant manipulators, in order to realize human-machine collaboration and integration, robots no longer work in a separate environment that is completely isolated (Ge and Cui, 2000; Sugie et al., 2003; Lee and Buss, 2007). Instead, collaboration is required between human or other robots, as a result, the obstacle avoidance control is becoming a matter of urgency: robots need to achieve real-time avoidance of static or dynamic obstacles while completing given motion tasks.

Many obstacle avoidance methods for robot manipulators haven been reported, which are designed online or off-line. Based on stochastic sampling algorithm, a series of obstacle avoidance methods are proposed, these methods could obtain effective solutions even in ultra-redundant systems. In Wei and Ren (2018), Wei et al. propose a modified RRT based method, namely Smoothly RRT, in which a maximum curvature constraint is built to obtain a smooth curve when avoiding obstacles, simulation results also show that the method achieves faster convergence than

traditional RRT based ones. In Hsu et al. (2006), Hsu discusses the probabilistic foundations of PRM based methods, a conclusion is drew that the visibility properties rather than dimensionality of C has a greater impact on the probability, and the convergence would be faster if extract partial knowledge could be introduced. However, due to the large computational costs, those methods can be hardly used online.

Different from stochastic results obtained by above mentioned methods, artificial potential field methods plan the same path each time in the same environment, which is important in industrial applications (Khatib, 1986). The basic idea of artificial potential field methods is that the target bears as an attractive pole while the obstacle creates repulsion on the robot, then the robot will be controlled to converge to the target without colliding with obstacles. At the same time, artificial potential field methods have shown great ability in tracking dynamic targets as well as avoiding dynamic obstacles. In Csiszar et al. (2011), a modified method is proposed, which describes the obstacles by different geometrical forms, both theoretical conduction and experimental tests validate the proposed method. Considering the local minimum problem that may be caused by multi-link structures, in Badawy (2016), a two minima is introduced to construct potential field, such that a dual attraction between links enables faster maneuvers comparing with traditional methods. Other improvements to artificial potential field method can be found in Tsai et al. (2001); Tsuji et al. (2002); Wen et al. (2017). Taking advantage of redundant DOFs, obstacles can be avoided by the self-motion in the null space, by calculating pseudo-inverse of Jacobian matrix, the solution can be formulated as the sum of a minimum-norm particular solution and homogeneous solutions (Cao et al., 1999; Moosavian and Papadopoulos, 2007; Krzysztof and Joanna, 2016).

The application of artificial intelligence algorithms based on neural networks provide a new idea for robotic control, these methods are considered to be very promising since its excellent learning ability (Jung and Kim, 2007). For instance, in Pan et al. (2017), the authors propose a command-filtered back-stepping method, in which a neural network based learning scheme is introduced to deal with functional uncertainties. In Pan and Yu (2017), a biomimetic hybrid controller is established, in which the control strategy consist of a feed-forward predictive machine based on a RBF Neural Network and a feedback servo machine based on a proportional-derivative controller. In Fu et al. (2018), 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 Fu et al. (2019), an online learning mechanism is built for visual tracking systems. The controller uses both positive and negative sample importances as input, and it is shown that the proposed weighted multiple instance learning scheme achieves wonderful tracking performance in challenging environments. Typically, the structure of a neural network may be complex in order to achieve better performance. Although the model of robot manipulator is highly nonlinear, by introducing the priori information of the system model, the neural network can be optimized, i.e., the number of nodes in neural networks can

be reduced effectively while maintaining the learning efficiency (Fontaine and Germain, 2001). Inspired by this, a series dynamic neural networks are proposed to realize robotic control in realtime (Zhang et al., 2004; Li et al., 2017; Yang et al., 2018b). Based on the idea of constraint-optimization, quadratic-programming approaches haven been introduced for kinematic control of redundant manipulators. The designed outer-loop controller is described as equality constraints, and objective functions are established to describe certain performance of the system. Using the learning and parallel calculation ability, dynamic neural networks are established to solve the quadratic-programming problem online. The kinematic control is thus achieved by ensuring the equality constraints, and the flexibility is used by optimizing the objective functions. On the other hand, these methods is capable of handling inequality constraints and model uncertainties (Zhang et al., 2018; Li et al., 2019; Xu et al., 2019b). In Cheng et al. (1993), the obstacle avoidance strategy is described as equality constraints, but the parameters of escape velocity is difficult to obtain. In Zhang and Wang (2004), Zhang et al. propose an inequality based method, in which 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 Guo and Zhang (2012), which is capable of suppressing undesirable discontinuity in the original solutions.

Motivated by the above observations, in this paper, we proposed a novel obstacle avoidance strategy based on deep recurrent neural networks. By abstracting robot and obstacles as a set of critical points, the distances between the robot and obstacles are approximately described by a group of point-to-point distances. And the obstacle avoidance is realized by inequality constraint described by class-K functions. Then the obstacle avoidance problem is reformulated as a QP problem in the speed level, and a deep recurrent neural network is designed to solve the QP online. Numerical results show that the robot is capable of avoiding the obstacles while tracking the predefined trajectories. Before ending this section, the main contributions of this paper are summarized as below

- The proposed deep RNN based controller is able to achieve both path tracking and obstacle avoidance, at the same time, physical constraints such as angular joints and velocities are satisfied.
- In this paper, we propose a class-K function based obstacle avoidance strategy, which has a more general form of description than traditional linear escape velocity methods.
- By abstracting robots and obstacles into critical point sets respectively, the distance between the robot and the obstacle can be described in a simpler way. Besides, numerical results show that the control algorithm can realize the avoidance of static and dynamic obstacles.

2. PROBLEM FORMULATION
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:

$$x = f(\theta), \qquad (1)$$

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:

$$\dot{x} = J(\theta)\dot{\theta}, \qquad (2)$$

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 isolated robots from humans or other robots. This problem could be even more acute in tasks which requires 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

$$|O_j A_i| \geq d, \qquad \forall A_i \in C_1, \forall O_i \in C_2. \qquad (3)$$

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 (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, Equation (3) can be approximately described as below:

$$|O_j A_i| \geq d, \qquad (4)$$

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 Equation (4) in shown in **Figure 1**.

Remark. 1 In real implementations, there are many ways to measure $|O_j A_i|$. For instance, since physical structure of
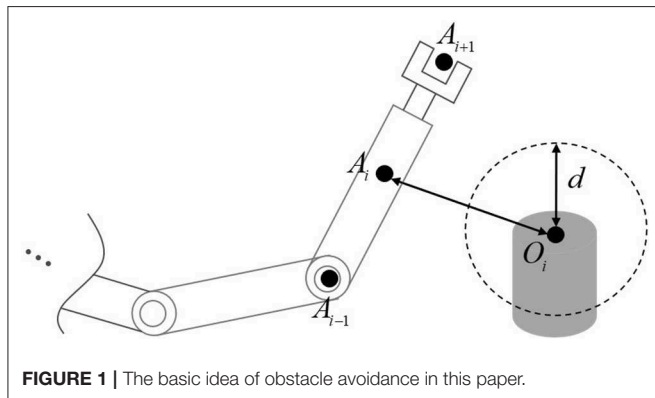

FIGURE 1 | The basic idea of obstacle avoidance in this paper.

the a manipulator is known, the key points A_i are available in advance, both positions and velocities of those points can be calculated directly using the feedback of robot joints. The real-time measurement of obstacles can be achieved through industrial cameras. Therefore, the information of A_i and B_j are all available. As to measurement noise, by introducing a bigger safety distance d, e.g., $d = 1.5(d_1 + d_2)$, the safety can be ensured.

2.2. Reformulation of Inequality in Speed Level

In order to guarantee the inequality (4), by defining $D = |O_j A_i| - d$, an inequality is rebuilt in speed level as:

$$\mathrm{d}(|O_j A_i|)/\mathrm{d}t \geq -\mathrm{sgn}(D)g(|D|), \qquad (5)$$

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

$$\dot{A}_i = J_{ai}(\theta)\dot{\theta}, \qquad (6)$$

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 Equation (5) can be reformulated as:

$$\begin{aligned}
\frac{\mathrm{d}}{\mathrm{d}t}(|O_j A_i|) &= \frac{\mathrm{d}}{\mathrm{d}t}\left(\sqrt{(A_i - O_j)^{\mathrm{T}}(A_i - O_j)}\right) \\
&= \frac{1}{|O_j A_i|}(A_i - O_j)^{\mathrm{T}}(\dot{A}_i - \dot{O}_j) \\
&= \overrightarrow{|O_j A_i|}^{\mathrm{T}} J_{ai}(\theta)\dot{\theta} - \overrightarrow{|O_j A_i|}^{\mathrm{T}}\dot{O}_j, \qquad (7)
\end{aligned}$$

where $\overrightarrow{|O_j A_i|} = (A_i - O_j)^{\mathrm{T}}/|O_j A_i| \in \mathbb{R}^{1 \times m}$ 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:

$$J_{oi}\dot{\theta} \leq \mathrm{sgn}(D)g(|D|) - \overrightarrow{|O_j A_i|}^{\mathrm{T}}\dot{O}_j, \qquad (8)$$

where $J_{oi} = -\overrightarrow{|O_j A_i|}^{\mathrm{T}} J_{ai} \in \mathbb{R}^{1 \times n}$. Based on the inequality description (8), the collision between robot and obstacle can be avoided by ensuring:

$$J_o\dot{\theta} \leq B, \qquad (9)$$

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 $B_{ij} = \mathrm{sgn}(D)g(|D|) - \overrightarrow{|O_j A_i|}^{\mathrm{T}}\dot{O}_j$.

Remark 2: According to 5 the definition of class-K functions, the original escape velocity based obstacle avoidance methods in Zhang and Wang (2004); Guo and Zhang (2012) can be regarded as a special case of 5, in which $G(|D|)$ is selected as $G(|D|) = k|D|$. Therefore, in this paper, the proposed obstacle avoidance strategy is more general than traditional methods.

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 paper, 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 paper, 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's sake, the robot is wished to move at a low speed, on the other hand, 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:

$$\min \quad \dot{\theta}^{\mathrm{T}}\dot{\theta}/2, \tag{10a}$$
$$s.t. \quad x = x_{\mathrm{d}}, \tag{10b}$$
$$\theta^{-} \leq \theta \leq \theta^{+}, \tag{10c}$$
$$\dot{\theta}^{-} \leq \dot{\theta} \leq \dot{\theta}^{+}, \tag{10d}$$
$$J_o\dot{\theta} \leq B. \tag{10e}$$

It is remarkable that the constraints 10b–10e and the objective function 10a to be optimized are built in different levels, which is very difficult to solve directly. Therefore, we will transform the original QP problem (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 (10c), according to escape velocity method, it can be obtained by limiting joint speed as $\alpha(\theta^{-} - \theta) \leq \dot{\theta} \leq \alpha(\theta^{+} - \theta)$, where α is a positive constant. Combing the kinematic property (2), the reformulated QP problem is:

$$\min \quad \dot{\theta}^{\mathrm{T}}\dot{\theta}/2, \tag{11a}$$
$$s.t. \quad J(\theta)\dot{\theta} = \dot{x}_{\mathrm{d}} + k(x_{\mathrm{d}} - x), \tag{11b}$$
$$\max(\alpha(\theta^{-} - \theta), \dot{\theta}^{-}) \leq \dot{\theta} \leq \min(\dot{\theta}^{+}, \alpha(\theta^{+} - \theta)), \tag{11c}$$
$$J_o\dot{\theta} \leq B. \tag{11d}$$

It is noteworthy that both the formula (11a) and (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.

3. DEEP RNN BASED SOLVER DESIGN

In this section, a deep RNN is proposed to solve the obstacle avoidance problem (11) online. To ensure the constraints (11b), (11c), and (11d), a group of state variables are introduced in the deep RNN. The stability is also proved by Lyapunov theory.

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:

$$L = \dot{\theta}^{\mathrm{T}}\dot{\theta}/2 + \lambda_1^{\mathrm{T}}(\dot{x}_{\mathrm{d}} + k(x_{\mathrm{d}} - x) - J(\theta)\dot{\theta}) + \lambda_2^{\mathrm{T}}(J_o\dot{\theta} - B), \tag{12}$$

λ_1 and λ_2 are the dual variables corresponding to the constraints (11b) and (11d). According to Karush-Kuhn-Tucker conditions, the optimal solution of the optimization problem (12) can be equivalently formulated as:

$$\dot{\theta} = P_\Omega(\dot{\theta} - \frac{\partial L}{\partial \dot{\theta}}), \tag{13a}$$
$$J(\theta)\dot{\theta} = \dot{x}_{\mathrm{d}} + k(x_{\mathrm{d}} - x), \tag{13b}$$
$$\begin{cases} \lambda_2 > 0 & \text{if } J_o\dot{\theta} = B, \\ \lambda_2 = 0 & \text{if } J_o\dot{\theta} \leq B, \end{cases} \tag{13c}$$

where $P_\Omega(x) = \operatorname{argmin}_{y \in \Omega}||y - x||$ is a projection operator to a restricted interval Ω, which is defined as $\Omega = \{\dot{\theta} \in \mathbb{R}^n|\max(\alpha(\theta^{-} - \theta), \dot{\theta}^{-}) \leq \dot{\theta} \leq \min(\dot{\theta}^{+}, \alpha(\theta^{+} - \theta))\}$. Notable that Equation (13c) can be simply described as:

$$\lambda_2 = (\lambda_2 + J_o\dot{\theta} - B)^{+}, \tag{14}$$

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

Although the solution of (13) is exact the optimal solution of the constrained-optimization problem (11), it is still a challenging issue to solve (13) online since its inherent nonlinearity. In this paper, in order to solve (13), a deep recurrent neural network is designed as:

$$\epsilon\ddot{\theta} = -\dot{\theta} + P_\Omega(J^{\mathrm{T}}\lambda_1 - J_o^{\mathrm{T}}\lambda_2), \tag{15a}$$
$$\epsilon\dot{\lambda}_1 = \dot{x}_{\mathrm{d}} + k(x_{\mathrm{d}} - x) - J(\theta)\dot{\theta}, \tag{15b}$$
$$\epsilon\dot{\lambda}_2 = -\lambda_2 + (\lambda_2 + J_o\dot{\theta} - B)^{+}, \tag{15c}$$

with ϵ is a positive constant scaling the convergence of (15).

Remark. 3 As to the established deep RNN (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 (11b), as shown in (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 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 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 Jung and Kim (2007), the complexity of neural networks is greatly reduced.

3.2. Stability Analysis

In this subsection, 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 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 1: A dynamic neural network is said to converge to the equilibrium point if it satisfies:

$$\kappa \dot{x} = -x + P_S(x - \varrho F(x)), \qquad (16)$$

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

Lemma 2: (Slotine and Li, 2004) 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 $\alpha_1(||x||) \le V(t, x) \le \alpha_2(||x||), \forall (t, x) \in [0, d) \times B_d$, then $x = 0$ is a uniformly asymptotically stable equilibrium point if there exist some class-K function g defined on $[0, d)$, to make the following inequality hold:

$$\frac{\partial V}{\partial t} + \frac{\partial V}{\partial x} f(t, x) \le -\alpha(||x||), \forall (t, x) \in [0, \infty) \times B_d, \qquad (17)$$

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

Theorem 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 (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 (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 (15), let $(\dot{\theta}^*, \lambda_1^*, \lambda_2^*)$ be the equilibrium of the deep RNN, then $(\dot{\theta}^*, \lambda_1^*, \lambda_2^*)$ satisfies:

$$-\dot{\theta}^* + P_\Omega(J^{\mathrm{T}}\lambda_1^* - J_o^{\mathrm{T}}\lambda_2^*) = 0, \qquad (18a)$$

$$\dot{x}_{\mathrm{d}} + k(x_{\mathrm{d}} - x) - J(\theta)\dot{\theta}^* = 0, \qquad (18b)$$

$$-\lambda_2^* + (\lambda_2^* + J_o\dot{\theta}^* - B)^+ = 0, \qquad (18c)$$

with $\dot{\theta}^*$ be the output of deep RNN. By comparing Equation (18) and (13), we can readily obtain that they are identical, i.e., the equilibrium point satisfies the KKT condition of (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 (11b), we have:

$$\dot{V} = e^{\mathrm{T}}\dot{e} = e^{\mathrm{T}}(\dot{x}_{\mathrm{d}} - J(\theta)\dot{\theta}^*)$$
$$= -ke^{\mathrm{T}}e \le 0, \qquad (19)$$

in which the dynamic Equation 18b is also used. It can readily obtained that the tracking error would eventually converge to 0.

It is notable that the dynamic (Equation 18c) satisfies:

$$\lambda_2^* + J_o\dot{\theta}^* - B - (\lambda_2^* + J_o\dot{\theta}^* - B)^+ = J_o\dot{\theta}^* - B. \qquad (20)$$

According to the property of projection operator $(\bullet)^+$, $y - (y)^+ \le 0$ holds for any y, then we have $J_o\dot{\theta}^* - B \le 0$, together with (5), the inequality (5) is satisfied. Notable that (5) can be rewritten as:

$$\dot{D} \ge -\mathrm{sgn}(D)g(|D|). \qquad (21)$$

As to (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 $\alpha_1(|D|) = \alpha_2(|D|)$ belong to class-K, and the following inequality will always hold:

$$\alpha_1(|D|) \le V_2(t, D) \le \alpha_2(|D|). \qquad (22)$$

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

$$\frac{\partial V_2}{\partial \mathrm{t}} + \frac{\partial V}{\partial D}\dot{D} = -|D|g(|D|) \le 0. \qquad (23)$$

According to Lemma 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 $|O_jA_i| \ge d$ when $\mathrm{d}(|O_jA_i|)/\mathrm{d}t \ge -\mathrm{sgn}(D)g(|D|)$.

Part II. Then we will show the stability of the deep RNN (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 (15) can be rewritten as:

$$\epsilon \dot{\xi} = -\xi + P_{\bar{\Omega}}[\xi - F(\xi)], \qquad (24)$$

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:

$$\nabla F(\xi) = \begin{bmatrix} I & -J^{\mathrm{T}} & J_o^{\mathrm{T}} \\ J & 0 & 0 \\ -J_o^{\mathrm{T}} & 0 & 0 \end{bmatrix}. \qquad (25)$$

According to the definition of monotone function, we can readily obtain that $F(\xi)$ is monotone. From the description of (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 (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 1, the proposed neural network (15) is stable and will globally converge to the optimal solution of (10). The proof is completed.

4. NUMERICAL RESULTS

In this section, 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.

4.1. Simulation Setup

The physical structure of the 4-link planar robot to be simulated in shown in **Figure 2**, in which the critical points of the robot are also marked. As shown in **Figure 2A**, 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. The D-H parameters are given in **Figure 2B**. 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 Equation 8, J_{o1} can be obtained. A_i and J_{oi} can be calculated similarly. The control parameters are set as $\epsilon = 0.001$, $\alpha = 8$, $k = 8$. 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 $i = 1 \dots 4$. The safety distance d is set to be 0.1m.
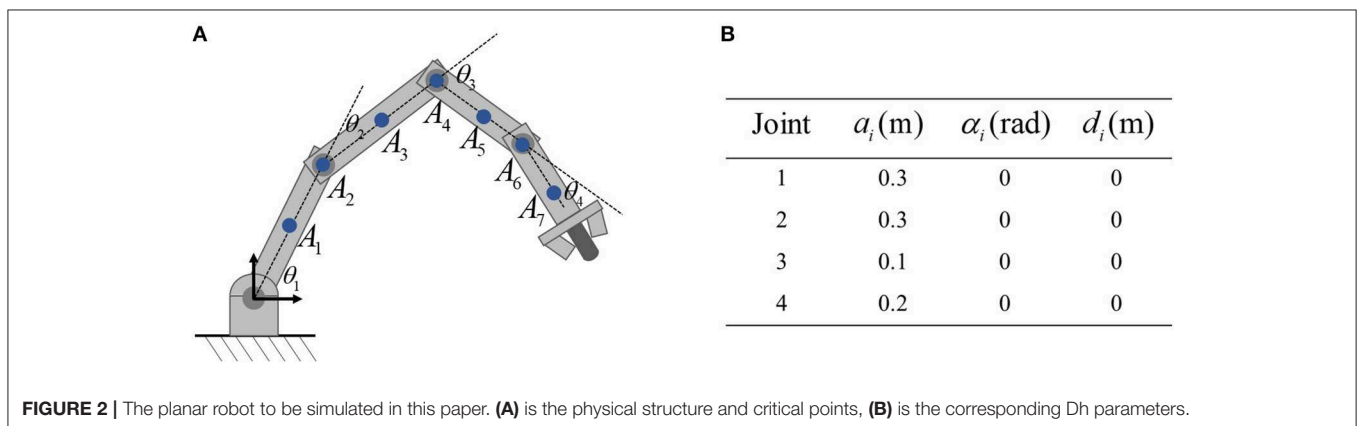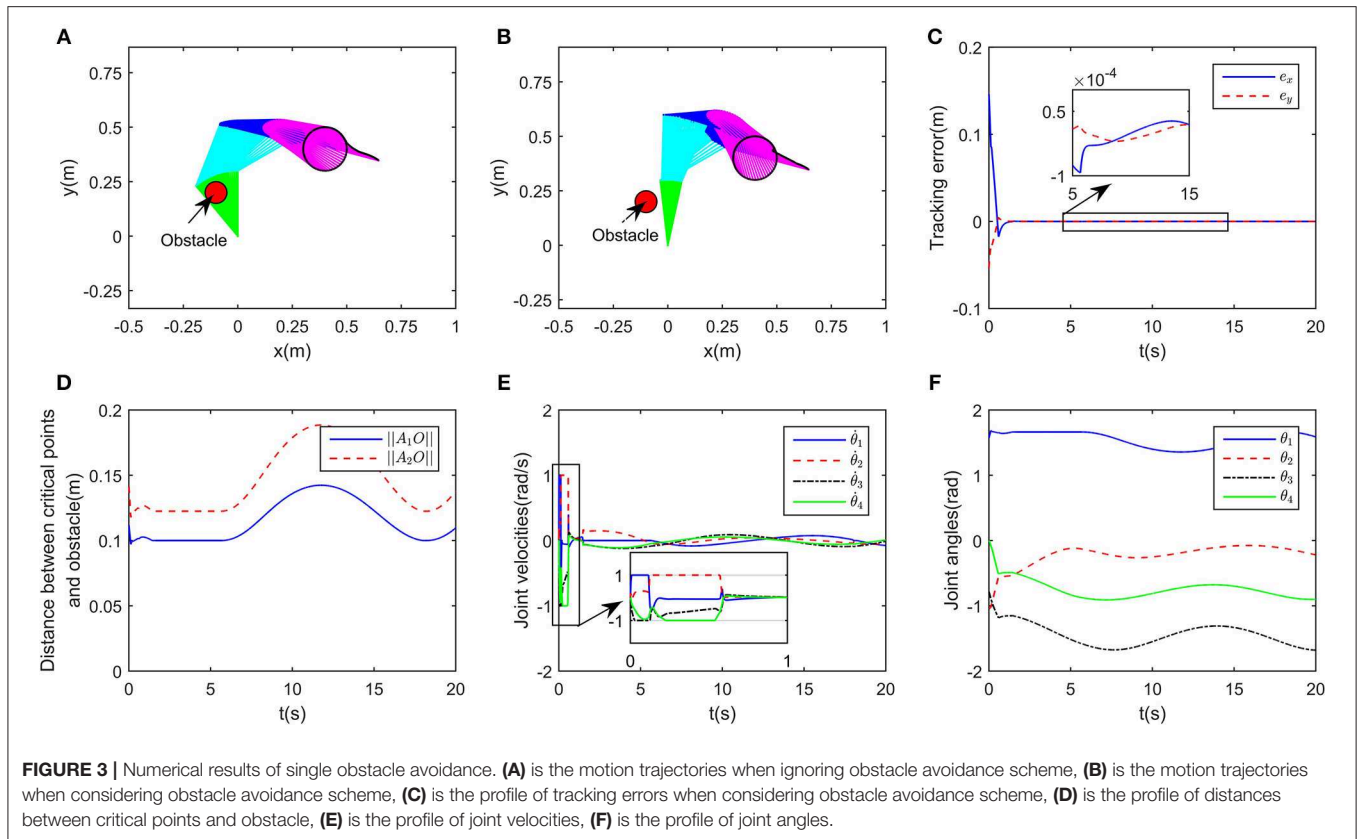
4.2. Single Obstacle Avoidance

In this simulation, the obstacle is assumed to be centered at $[-0.1, 0.2]^T$m, the desired path is set as $x_d = [0.4 + 0.1cos(0.5t), 0.4 + 0.1sin(0.5t)]^T$m, and the initial joint angles are set to be $\theta_0 = [\pi/2, -\pi/3, -\pi/4, 0]^T$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 (10e) are conducted. Simulation results are shown in **Figure 3**. When ignoring the obstacle, the end-effector trajectories and the corresponding incremental configurations are shown in **Figure 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 (**Figure 3B**). Simultaneously, the tracking errors when tracking the given circle are shown in **Figure 3C**. From the initial state, the end-effector moves toward the circle quickly and smoothly, after that, the tracking error in stable state keeps $< 1 \times 10^{-4}$m, showing that the robot could achieve kinematic control as well as obstacle avoidance tasks. To show more details of the proposed deep RNN based method, some important process data is given. As the obstacle is close to the first joint, critical points A_1 and A_2 are more likely to collide with obstacle, therefore, as a result, the distances between the obstacle O_1 and A_1, A_2 are shown in **Figure 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 avoidance method, the robot maintains minimum distance from the obstacle. The profile of joint velocities are shown in **Figure 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 **Figure 3F**.

4.3. Discussion on Class-K Functions

In this part, we will discuss the influence of different class-K functions in the avoidance scheme (5). Four functions are selected as $G_1(|D|) = K|D|^2$, $G_2(|D|) = K|D|$, $G_3(|D|) = K\tanh(5|D|)$, $G_4(|D|) = K\tanh(10|D|)$, **Figure 4A** shows the comparative curves the these functions. Other simulation settings are the same as the previous one. Simulation results are shown in **Figure 4B**. When selecting the same positive gain K, the minimum distance is about 0.08m, which shows the robot can avoid colliding with the obstacle using the avoidance scheme (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 0.

FIGURE 2 | The planar robot to be simulated in this paper. **(A)** is the physical structure and critical points, **(B)** is the corresponding Dh parameters.

Joint	a_i (m)	α_i (rad)	d_i (m)
1	0.3	0	0
2	0.3	0	0
3	0.1	0	0
4	0.2	0	0

FIGURE 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.

Therefore, one conclusion can be drawn that the function can be more similar with Sign function, to achieve better obstacle avoidance.

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 $[0.1, 0.25]^T$m and $[0, 0.4]^T$m, respectively. Simulation results are shown in **Figure 5**. The desired path is defined as $x_d = [0.45 + 0.1cos(0.5t), 0.4 + 0.1sin(0.5t)]^T$. The initial joint angle of the robot is selected as $\theta_0 = [1.5, -1 - 1, 0]^T$. To further show the effectiveness of the proposed obstacle avoidance strategy 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 **Figure 5A**, the inequality constraint (11d) will not work, in other words, only kinematic tracking problem in 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 **Figures 5B–H**. The tracking errors are shown in **Figure 5C**, with the transient time being about 4s, and steady state error $< 1 \times 10^{-3}$m. Correspondingly, the robot moves fast in the transient stage, ensuring the quick convergence of the tracking errors. It is remarkable that the distances between the critical points and obstacle points are kept larger than 0.1m at all times, showing the effectiveness of the proposed method. At $t = 14$s, from **Figures 5D,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 (11d) hold, and the

boundedness between the robot and obstacle is thus guaranteed. After $t = 18$s, $||A_3O_1||$ becomes greater, then λ_2 converges to 0. 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.

4.5. 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 $[0, 0.5]^T$, $[0.4, 0.5]^T$, $[0.4, 0.6]^T$ and $[0.5, 0.6]^T$, respectively. By selecting the safety distance $d = 0.1$m, and obstacle points as $O_1 = [0.05, 0.55]^T$, $O_2 = [0.15, 0.55]^T$, $O_3 = [0.25, 0.55]^T$ and $O_4 = [0.35, 0.55]^T$. 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 **Figure 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. While in **Figure 6A**, when obstacle avoidance is ignored, the collision emerges. **Figures 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.
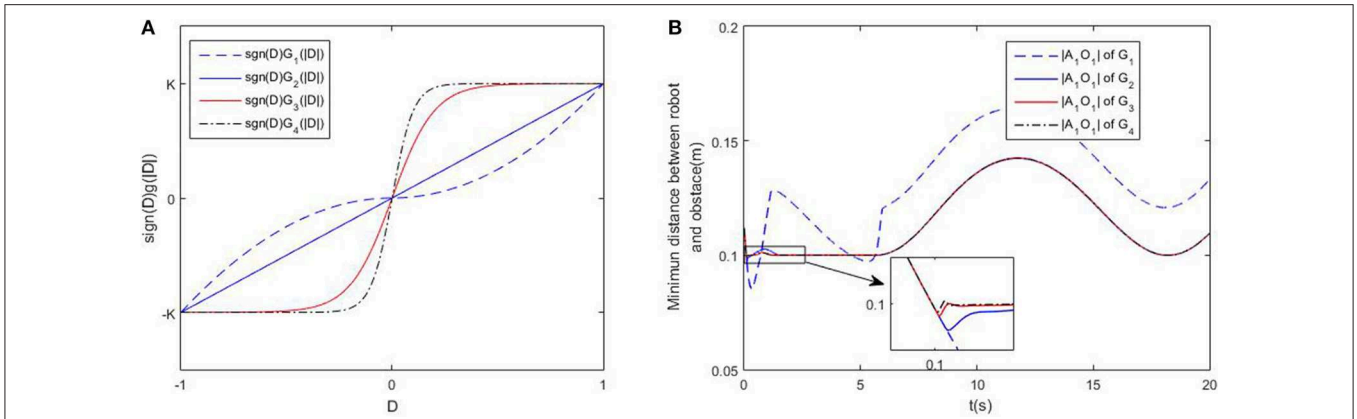
FIGURE 4 | Discussions on different obstacle avoidance functions. **(A)** is the comparative curves of different obstacle avoidance functions. **(B)** is the profile of minimum distance of the robot and obstacle using different obstacle avoidance functions.
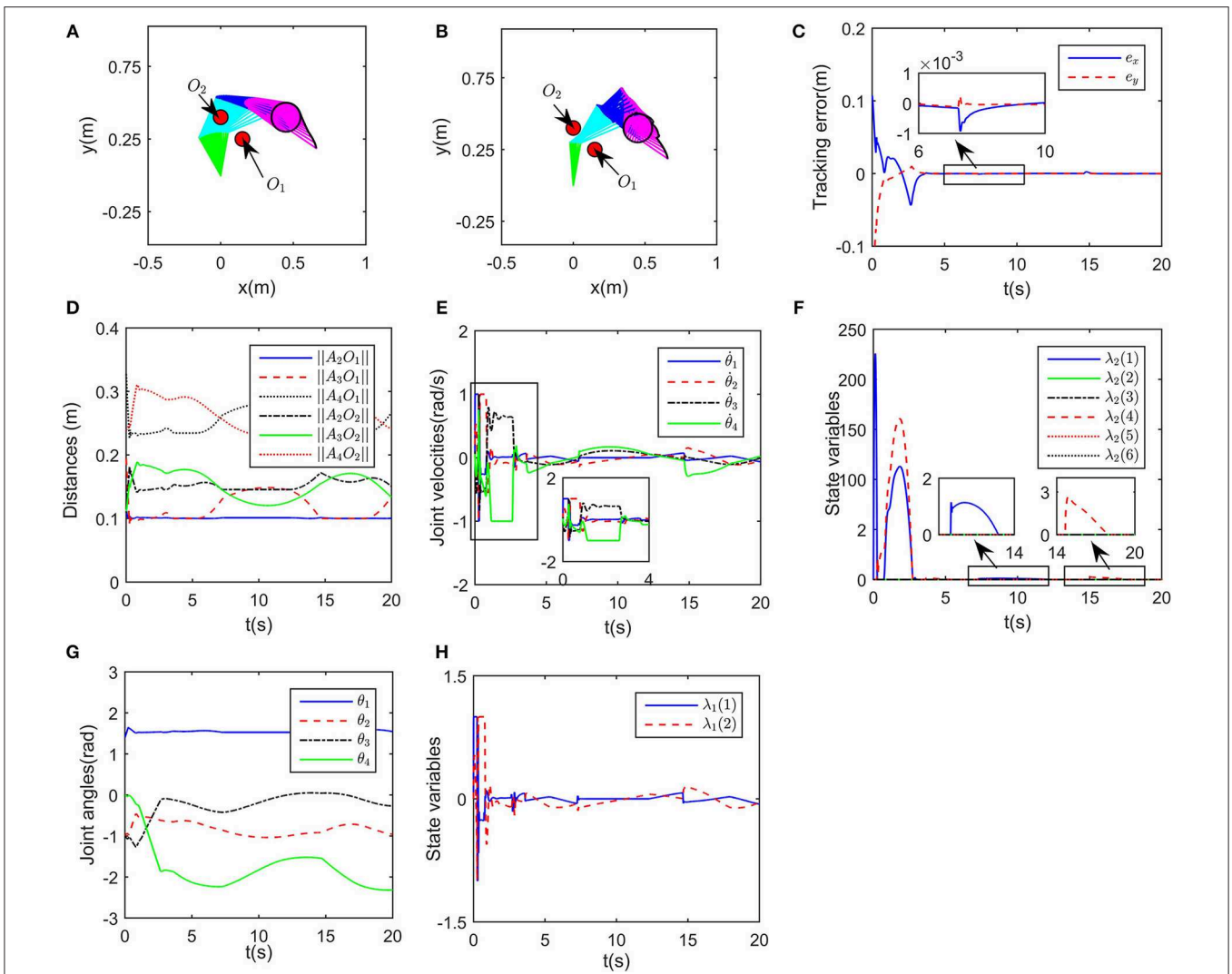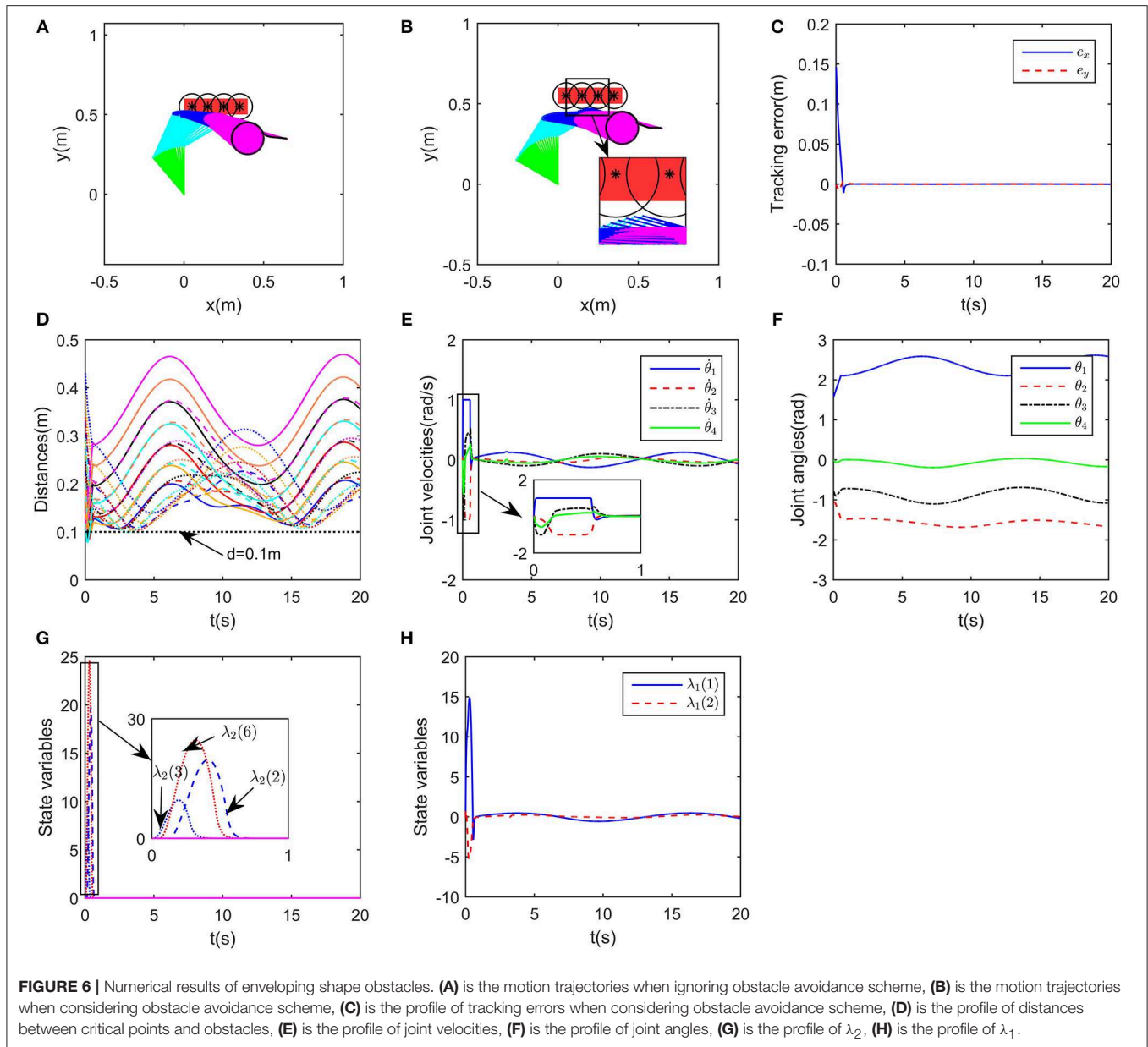

FIGURE 5 | Numerical results of multiple obstacle avoidance. **(A)** is the motion trajectories when ignoring obstacle avoidance scheme, **(B)** is the motion trajectories when considering obstacle avoidance scheme, **(C)** is the profile of tracking errors when considering obstacle avoidance scheme, **(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.

FIGURE 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.

4.6. Dynamic Obstacles

In this part, we consider dynamic obstacles moving in the workspace. In real applications, pedestrian or other obstacles always tend to be mobile. Obstacle avoidance for dynamic obstacles is of more general significance. In real time, static obstacles can be considered a special case. In this simulation, the simulation duration is selected as 20s, and the trajectory of a dynamic obstacle is defined as $x_d = [-0.1 + 0.01t, 0.3]^T$. The snapshots in the control process are shown in **Figure 8**. While ensuring effective tracking of the defined path, the robot is able to use its self-motion to avoid the dynamic obstacle effectively, and maintain a safe distance. The distances between critical points and the dynamic O is shown in **Figure 7B**. At the beginning of simulation, the tracking error is big, in order to ensure the convergence of tracking error, the joints move a big range except $J1$. It is worth noting that since the critical point A_2 is next to O, joint 1 rotates in the direction which conforms to the movement of O. In the stable state, tracking error is $< 5 \times 10^{-4}$m (**Figure 7A**). At about $t = 14$s, A_2O decreases to 0.1m, accordingly, the joint velocities change obviously (as shown the significant change of joint velocities in **Figure 7C**, the tracking error changes to 10^{-3}m, and then converges quickly. At $t = 18$s, although A_2 and A_3 are near O, the robot is still capable of avoiding the dynamic obstacle. During the control process, joint angles are ensured not to exceed the limits, as shown in **Figure 7D**.
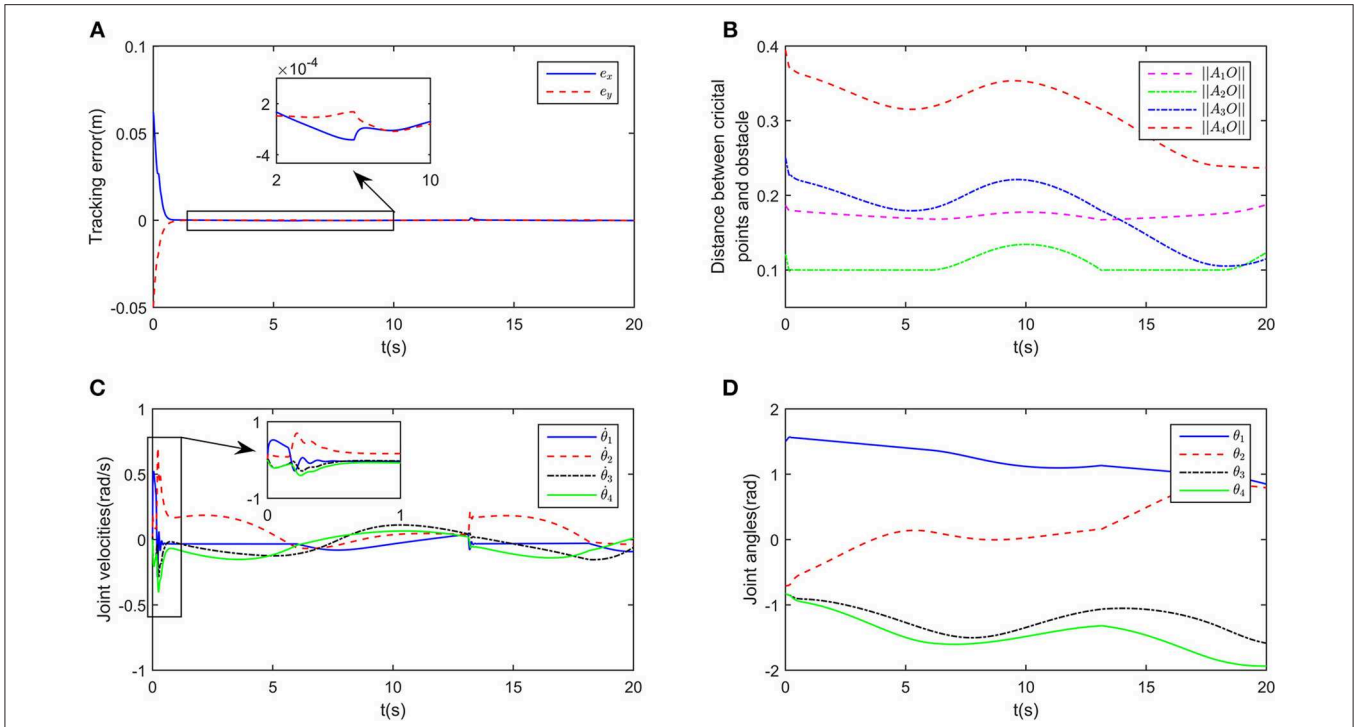
FIGURE 7 | Numerical results of enveloping shape obstacles. **(A)** is the profile of tracking errors when considering obstacle avoidance scheme, **(B)** is the profile of distances between critical points and obstacles, **(C)** is the profile of joint velocities, **(D)** is the profile of joint angles.
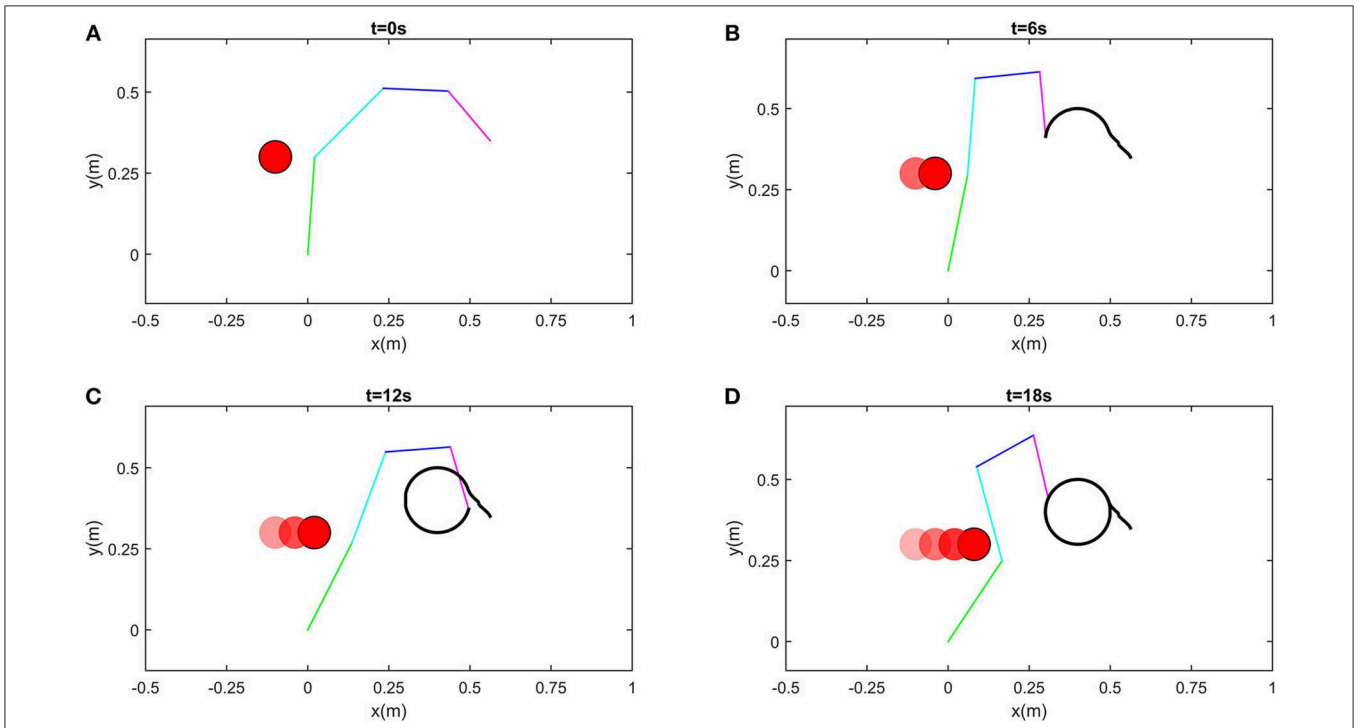

FIGURE 8 | Snapshots when robot avoiding a dynamic obstacle. **(A)** is the snapshot when $t = 0$s, **(B)** is the snapshot when $t = 6$s, **(C)** is the snapshot when $t = 12$s, **(D)** is the snapshot when $t = 18$s.
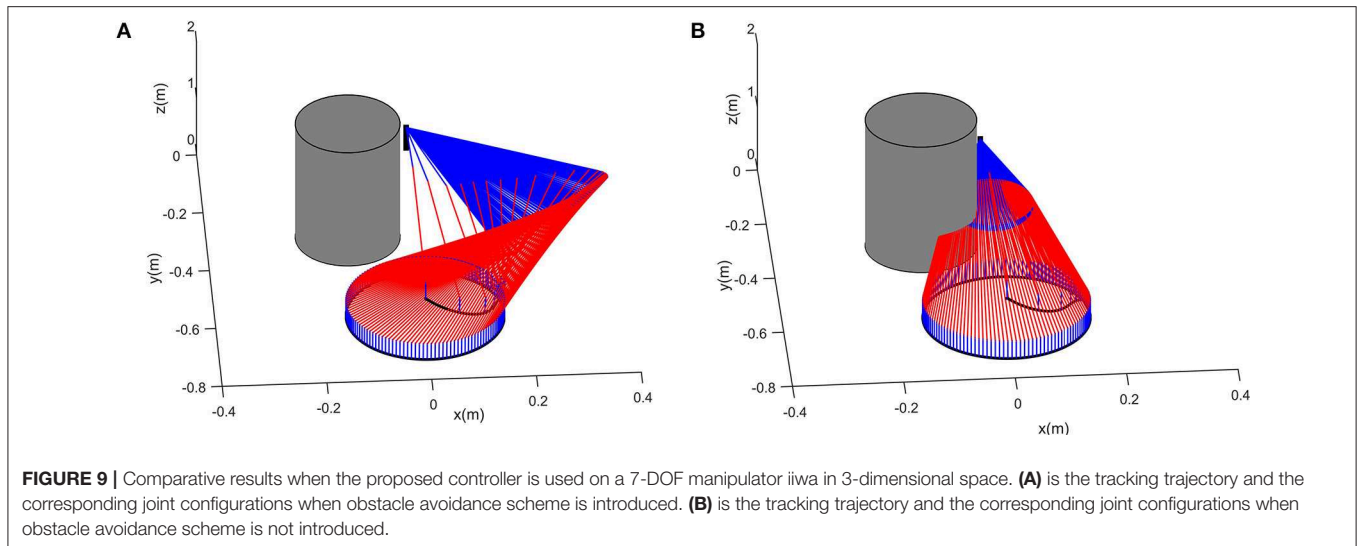
FIGURE 9 | Comparative results when the proposed controller is used on a 7-DOF manipulator iiwa in 3-dimensional space. **(A)** is the tracking trajectory and the corresponding joint configurations when obstacle avoidance scheme is introduced. **(B)** is the tracking trajectory and the corresponding joint configurations when obstacle avoidance scheme is not introduced.

4.7. Obstacle Performance on 7-DOF Manipulator in 3-Dimensional Space

To further verify the effectiveness of the control scheme, another simulation on a 7DOF manipulator iiwa is carried out. The desired path to be tracked is also a planar circular, which is centered at $[0, -0.6, 0.1]^T$m with radius being 0.15m. The physical parameters can be found in Xu et al. (2019a). Suppose that there exist a cylinder obstacle in the workspace, the obstacle is centered as $[-0.13, -0.3, 0]^T$m, with the radius and height being 0.15m and 2m, respectively. Simulation results are shown in **Figure 9**. It can be readily found that the proposed schemes can obtain satisfying performance in 3-dimensional spaces.

4.8. Comparisons

To illustrate the priority of the proposed scheme, a group of comparisons are carried out. As shown in **Table 1**, all the controllers in Zhang and Wang (2004); Csiszar et al. (2011); Guo and Zhang (2012); Krzysztof and Joanna (2016) achieve the avoidance of obstacles. Comparing to APF method in Csiszar et al. (2011); Krzysztof and Joanna (2016) of JP based method in Csiszar et al. (2011); Krzysztof and Joanna (2016), 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 paper, both dynamic trajectories and obstacles are considered. The comparisons above also highlight the main contributions of this paper.

5. CONCLUSIONS

In this paper, a novel obstacle avoidance strategy is proposed based on a deep recurrent neural network. The robots are obstacles are presented by sets of critical points, then the distance between the robot and obstacle can be approximately describes as point-to-points distances. By understanding the nature escape velocity methods, a more general description

TABLE 1 | Comparisons among different obstacle avoidance controllers on manipulators.

Method	Convergence	Secondary task	Handling physical constraints	Dynamic obstacles	obstacle avoidance description
This paper	Yes	Yes	Yes	Yes	Inequalities
Guo and Zhang, 2012	Yes	Yes	Yes	*	Inequalities**
Zhang and Wang, 2004	Yes	Yes	Yes	*	Inequalities**
Csiszar et al., 2011	Yes	No	No	Yes	Repulsion
Krzysztof and Joanna, 2016	Yes	No	No	*	Null space

*In Zhang and Wang (2004); Guo and Zhang (2012); Krzysztof and Joanna (2016), dynamic obstacles are not considered.
**Regular escape velocity method is used, which is only a special case of 5.

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.

DATA AVAILABILITY

All datasets analyzed for this study are included in the manuscript and the supplementary files.

AUTHOR CONTRIBUTIONS

Theoretical derivation was done by ZX. Simulation research was performed by ZX, with important advice from XZ. The paper was revised by XZ and SL. All authors approved the manuscript.

FUNDING

ACKNOWLEDGMENTS

REFERENCES

Badawy, A. (2016). Dynamic and interactive path planning and collision avoidance for an industrial robot using artificial potential field based method. *Alexandria Eng. J.* 55, 1235–1241. doi: 10.1016/j.aej.2016.03.042

Cao, B., Dodds, G., and Irwin, G. (1999). Redundancy resolution and obstacle avoidance for cooperative industrial robots. *J. Robot. Syst.* 16, 405–417.

Chan, T., and Dubey, R. (1995). A weighted least-norm solution based scheme for avoiding joint limits for redundant joint manipulators. *IEEE Trans. Robot. Automat.* 11, 286–292.

Cheng, F., Chen, T., Wang, Y., and Sun, Y. (1993). "Obstacle avoidance for redundant manipulators using the compact qp method," in *IEEE International Conference on Robotics and Automation* (Atlanta, GA: IEEE), 41–50. doi: 10.1109/ROBOT.1993.292186

Cheng, L., Hou, Z. G., and Tan, M. (2009). Adaptive parameter estimation and control design for robot manipulators with finite-time convergence. *Automatica* 45, 2312–2318. doi: 10.1016/j.automatica.2009.06.007

Csiszar, A., Drust, M., Dietz, T., Verl, A., and Brisan, C. (2011). Dynamic and interactive path planning and collision avoidance for an industrial robot using artificial potential field based method. *Mechatronics* 1, 413–421. doi: 10.1007/978-3-642-23244-2_50

Fontaine, J., and Germain, A. (2001). Model-based neural networks. *Comput. Chem. Eng.* 25, 1045–1054. doi: 10.1016/S0098-1354(01)00679-2

Fu, C., Duan, R., and Kayacan, E. (2019). Visual tracking with online structural similarity-based weighted multiple instance learning. *Informat. Sci.* 481, 292–310. doi: 10.1016/j.ins.2018.12.080

Fu, C., Sarabakha, A., Kayacan, E., Wagner, C., John, R., and Garibaldi, J. (2018). Input uncertainty sensitivity enhanced non-singleton fuzzy logic controllers for long-term navigation of quadrotor uavs. *IEEE/ASME Trans. Mech.* 23, 725–734. doi: 10.1109/TMECH.2018.2810947

Ge, S., and Cui, Y. (2000). New potential functions for mobile robot path planning. *IEEE Trans. Robot. Automat.* 16, 615–620. doi: 10.1109/70.880813

Guo, D., and Zhang, Y. (2012). A new inequality-based obstacle-avoidance mvn scheme and its application to redundant robot manipulators. *IEEE Trans. Syst. Man Cybernet. Part C* 42, 1326–1340. doi: 10.1109/TSMCC.2012.2183868

Hsu, D., Latombe, J., and Kurniawati, H. (2006). On the probabilistic foundations of probabilistic roadmap planning. *Int. J. Robot. Res.* 25, 627–643. doi: 10.1177/0278364906067174

Jung, S., and Kim, S. (2007). Hardware implementation of a real-time neural network controller with a dsp and an fpga for nonlinear systems. *IEEE Trans. Indust. Electr.* 54, 265–271. doi: 10.1109/TIE.2006.888791

Khatib, O. (1986). Real-time obstacle avoidance system for manipulators and mobile robots. *Int. J. Robot. Res.* 5, 90–98.

Krzysztof, T., and Joanna, R. (2016). Dynamically consistent jacobian inverse for non-holonomic robotic systems. *Nonlinear Dynam.* 85, 107–122. doi: 10.1007/s11071-016-2672-x

Lee, K. K., and Buss, M. (2007). "Obstacle avoidance for redundant robots using jacobian transpose method," *2007 IEEE/RSJ International Conference on Intelligent Robots and Systems* (San Diego, CA: IEEE), 3509–3514.

Li, S., Zhang, Y., and Jin, L. (2017). Kinematic control of redundant manipulators using neural networks. *IEEE Trans. Neural Netw. Learn. Syst.* 28, 2243–2254. doi: 10.1109/TNNLS.2016.2574363

Li, Y., Li, S., and Hannaford, B. (2019). A model based recurrent neural network with randomness for efficient control with applications. *IEEE Trans. Indust. Informat.* 15, 2054–2063. doi: 10.1109/TII.2018.2869588

Moosavian, S. A. A., and Papadopoulos, E. (2007). Modified transpose jacobian control of robotic systems. *Automatica* 43, 1226–1233. doi: 10.1016/j.automatica.2006.12.029

Pan, Y., Sun, T., Liu, Y., and Yu, H. (2017). Composite learning from adaptive backstepping neural network control. *Neural Netw.* 95, 134–142. doi: 10.1016/j.neunet.2017.08.005

Pan, Y., Yang, C., Pan, L., and Yu, H. (2018). Integral sliding mode control: Performance, modification, and improvement. *IEEE Trans. Indust. Informat.* 14, 3087–3096. doi: 10.1109/TII.2017.2761389

Pan, Y., and Yu, H. (2017). Biomimetic hybrid feedback feedforward neural-network learning control. *IEEE Trans. Neural Netw. Learn. Syst.* 28, 1481–1487. doi: 10.1109/TNNLS.2016.2527501

Slotine, J., and Li, W. (2004). *Applied Nonlinear Control*. Beijing: China Machine Press.

Sugie, T., Kito, Y., and Fujimoto, K. (2003). Obstacle avoidance of manipulators with rate constraints. *IEEE Trans. Robot. Automat.* 19, 168–174. doi: 10.1109/TRA.2002.807554

Tsai, C., Lee, J., and Chuang, J. (2001). Path planning of 3-d objects using a new workspace model. *IEEE Trans. Syst. Man. Cybernet. Part C* 31, 405–410. doi: 10.1109/5326.971669

Tsuji, T., Tanaka, Y., Morasso, P., Sanguineti, V., and Kaneko, M. (2002). Bio-mimetic trajectory generation of robots via artificial potential field with time base generator. *IEEE Trans. Syst. Man. Cybernet. Part C* 32, 426–439. doi: 10.1109/TSMCC.2002.807273

Wei, K., and Ren, B. (2018). A method on dynamic path planning for robotic manipulator autonomous obstacle avoidance based on an improved rrt algorithm. *Sensors* 18, 571–578. doi: 10.3390/s18020571

Wen, G., Ge, S., Tu, F., and Choo, Y. (2017). Artificial potential based adaptive h∞ synchronized tracking control for accommodation vessel. *IEEE Trans. Indust. Elect.* 64, 5640–5647. doi: 10.1109/TIE.2017.2677330

Xu, Z., Li, S., Zhou, X., and Cheng, T. (2019a). Dynamic neural networks based adaptive admittance control for redundant manipulators with model uncertainties. *Neurocomputing* 1, 1–22. doi: 10.1016/j.neucom.2019.04.069

Xu, Z., Li, S., Zhou, X., Yan, W., Cheng, T., and Huang, D. (2019b). Dynamic neural networks based kinematic control for redundant manipulators with model uncertainties. *Neurocomputing* 329, 255–266. doi: 10.1016/j.neucom.2018.11.001

Yang, C., Jiang, Y., He, W., Na, J., Li, Z., and Xu, B. (2018a). Adaptive parameter estimation and control design for robot manipulators with finite-time convergence. *IEEE Trans. Indust. Electr.* 65, 8112–8123. doi: 10.1109/TIE.2018.2803773

Yang, C., Peng, G., Li, Y., Cui, R., Cheng, L., and Li, Z. (2018b). Neural networks enhanced adaptive admittance control of optimized robot-environment interaction. *IEEE Trans. Cybern.* 49, 2568–2579. doi: 10.1109/TCYB.2018.2828654

Zhang, Y. (2015). Singularity-conquering tracking control of a class of chaotic systems using zhang-gradient dynamics. *IET Control Theory Appl.* 9, 871–881. doi: 10.1049/iet-cta.2014.0931

Zhang, Y., Chen, S., Li, S., and Zhang, Z. (2018). Adaptive projection neural network for kinematic control of redundant manipulators with

unknown physical parameters. *IEEE Trans. Indust. Electr.* 65, 4909–4920. doi: 10.1109/TIE.2017.2774720

Zhang, Y., Ge, S., and Lee, T. (2004). A unified quadratic-programming-based dynamical system approach to joint torque optimization of physically constrained redundant manipulators. *IEEE Trans. Syst. Man Cybernet. Part B* 34, 2126–2132. doi: 10.1109/TSMCB.2004.830347

Zhang, Y. and Wang, J. (2004). Obstacle avoidance for kinematically redundant manipulators using a dual neural network. *IEEE Trans. Syst. Man Cybernet. Part B* 34, 752–759. doi: 10.1109/TSMCB.2003.8 11519