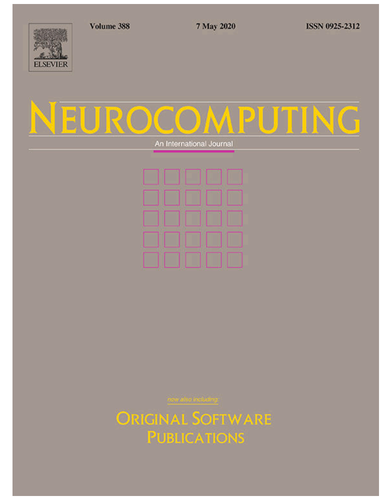
Journal Pre-proofs



Collaboration of Multiple SCARA Robots with Guaranteed Safety Using Recurrent Neural Networks

Yuhong He, Xiaoxiao Li, Zhihao Xu, Xuefeng Zhou, Shuai Li

PII: S0925-2312(21)00802-X
DOI: https://doi.org/10.1016/j.neucom.2021.05.049
Reference: NEUCOM 23879

To appear in: *Neurocomputing*

Received Date: 7 January 2021
Revised Date: 17 March 2021
Accepted Date: 16 May 2021

Please cite this article as: Y. He, X. Li, Z. Xu, X. Zhou, S. Li, Collaboration of Multiple SCARA Robots with Guaranteed Safety Using Recurrent Neural Networks, *Neurocomputing* (2021), doi: https://doi.org/10.1016/j.neucom.2021.05.049

Collaboration of Multiple SCARA Robots with Guaranteed Safety Using Recurrent Neural Networks

Yuhong He[a], Xiaoxiao Li[b], Zhihao Xu[b], Xuefeng Zhou[b,*], Shuai Li[a,*]

[a]*School of Engineering, Swansea University, Swansea SA2 8PP, U.K.*
[b]*Guangdong Key Laboratory of Modern Control Technology, Institute of Intelligence Manufacturing, Guangdong Academy of Sciences, Guangzhou, Guangdong 510070, P.R. China*

Abstract

SCARA robot is one of the most popularly used robots in industry. The obstacle avoidance feature of multiple SCARA robot collaboration is essential and prominent, which can be used to support multiple robots to accomplish not only more sophisticated tasks but also more efficient than individual robot. This paper mainly focuses on studying the problem of simultaneous multi-robot coordination and obstacle avoidance. A cooperative kinematic control problem of multiple robot manipulators, collision avoidance is taken into account to be the primary task as an inequality constraint and trajectory planning task is considered to be the secondary objective as to ensure the priority of safety, is described as a quadratic programming(QP) problem. Then, a recurrent neural network (RNN) based dynamic controller is designed to solve the formulated QP problem recursively. The convergence of the designed neural network is proved through Lyapunov analysis. With three SCARA planar robots, the effectiveness of the proposed controller is validated through numerical simulations. As observed in the results, when the minimal distance between robots is less than the setting safety distance, the collision avoidance strategy reacts to impel robots to avoid collision, which achieves the primary objective for obstacle avoidance; otherwise, the robot performs the desired trajectory tracking task.

⋆Y. He, X. Li and Z. Xu are jointly first authors. X. Zhou and S. Li are jointly corresponding authors.

*Corresponding author (xf.zhou@gia.ac.cn; shuai.li@swansea.ac.uk)

Keywords: Multi-robot collaboration, obstacle avoidence, kinematic control, constrained optimization, recurrent neural network(RNN), safety

1. Introduction

In recent years, robot technology has continuously improved the capabilities of robots, and has constantly expanded the scope of robot applications[1, 2, 3, 4]. While society often expects robots can accomplish more sophisticated tasks, these challenges can hardly achieve by an individual primitive robot[5]. The superior advantages of multi-robot collaboration are reflected mainly in the wide application field, good fault tolerance, high efficiency and good scalability, which leads for the need of Multi-Robot Coordination. Having said that, path planning, synchronized control, and team formation [6, 7, 8]are crucial fields of diverse robot application features. Thanks to the enhanced reliability of platforms and sensors, as well as the low-cost of technology, robotic platforms are now used widely for sophisticated tasks such as domestic services or industrial assembly lines.

Multi-robot systems have been broadly applied to various applications to perform a given task collaboratively and cooperatively[9]. In a multi-robot environment, path-planning or collision avoidance is an important feature[10]. In multi-robot collaborations, direct collisions between robots and any collateral damage for technicians will be prevented. Thus, Real-Time obstacle avoiding is the main target. But for non-redundant robots, due to the limitation of the degree of freedom, the robot cannot physically achieve the motion control task while realizing obstacle avoidance. In view of the above contradiction, we proposed a safety precaution mechanism, i.e., the priority of obstacle avoidance is superior to kinematic control.

Obstacle avoidance is a fundamental yet profound design factor of rigorous industrial manipulators. It has been a long journey for scholars studying various methods aiming to solve different scenarios of real applications. According to [11][12], Zhang *et al.* used online quadratic-programming(QP) method to solve for inverse-kinematic obstacle avoidance control problem. The simulation was based on the PAlO robot(redundant) manipulator in the presence of obstacles. In [13], similarly, Xu *et al.* described the avoidance strategy as QP with general class-K functions to solve QP problem online. The simulation was done on a 4-link planar redundant robot. Both static and dynamic obstacles were there to test for its performance. Khan *et al.* [14] announced an "beetle antennae olfactor" RNN which integrates path tracking control with obstacle avoidance into a optimization problem. The proposed algorithm was proved via simulations done by KUKA LBR 7-DOF industrial manipulator. All of these studies aimed to find solutions via online programming planning, and realized by single robot manipulator as the main object in actual trials. Obstacles are either static or dynamic objects. In factory assembly lines, manipulators often encounter collision risks from other robots or machines. Our study is focusing on multi-robot collaboration scenarios.

On the contrary, the following studies examined multi-robot collaboration control with obstacle avoidance. Both [15] and [16] implemented kinematic controls to solve for optimal trajectory problem with affective algorithms where multi-robot cooperation can be achieved. In [16, 17, 18, 19], authors examined dynamic trajectory planning method of multiple robot manipulators scenario as an optimal control problem. Analogously,[20] researched collaboration of multiple wheeled mobile manipulators by a collision avoidance technique. [21] used another optimization algorithm to solve for trajecotory planning problem of multi-robot formation. [15, 16, 20, 21] negelected to explore an immportant feature in depth when desgining our multi-robot coordination control system, that is the obstacle avoidance whose function is to ensure safety while in operation with other machines. [22] asserted that by exploiting redundancy and potential function method to avoid collisions when manipulating two cooperative robots. In pursuit of fastness of completion of certain kinematic tasks, we cannot afford to sacrifice safety criterion. Safety shall always be the top priority, and everything else builds upon this foundation. Our research goal is to fulfill secure operation while adding tracking control for multi-robot collaboration on top of this foundation.

Neural networks, combining with artificial intelligence techniques, provide a fresh perspective for robotic formation control.[23]. Planning problem is an important topic for robotics control. It can be catagorized as online planning and offline planning[24, 25]: Offline planning lacks of the ability for real-time controlling of manipulators. It is ideally suited for simplex trajectories in static known environment. In virtue of its rapid response when environments are constantly changing, online planning can handle challenges in unknown dynamic by continually modifying strategies and models[26, 27]. Berkeley Artificial Intelligence Research Lab announced a full-fledged discovery that a dynamic neural network model-based reinforcement learning algorithm can produce sound and creditable gaits that accomplish various complex locomotion tasks[28]. [29] is an example of an offline path planning problem. This problem pertains to combinatorial optimization problems. In [30], authors developed a neural network confined by kinematic constraints for mobile robots; The dynamics is identified online by the neural network estimators. Without measuring joint velocities, a robust online learning neural network output feedback scheme is offered to control motion control of robot manipulators[31].

Neural networks have various computational methodologies that can be realized to increase processing speed[32, 33]. In studies of [34] and [35], both objectives were to solve for unknown dynamics of manipulators; while the former used adaptive neural network and the later modified the conventional backpropagation algorithm. In an unknown environment, [36] provides an effective RNN model for collision-free path planning with limited information of the obstacle positions. Nevertheless, [37] mentions a Lagrangian network(RNN) can be used to deal with obstacle avoidance and trajectory tracking simultaneously knowing obstacle positions. [38] presents two neural networks of velocity inverse kinematics problem for redundant robots. In each proposed neural network approach, two cooperating recurrent neural networks are used.

Enlightened from the above awareness, we describe the problem of robot kinematic control as an optimization problem, which describes obstacle avoidance as an inequality constraint. As stated before, the foundation of multi-robot collaboration is based upon safety ensurance. The design intellection is to keep the moving error of manipulators running with respect to desired trajectory. Following this concept, we took a step forward by designing a dynamic(recurrent) neural network to solve this problem. Through assigning robot manipulators as a set of critical points, the distances between the manipulators are approximately described by a group of point-to-point distances. The obstacle avoidance problem is then reformulated as a QP problem in the velocity level, and a dynamic(recurrent) neural network is designed to solve the QP online. In numerical results, we show the experiment results from simulation of 3 Scara robots to prove the reliability of our method proposed in this paper.

The remainder of this paper is arranged as below. In Section 2, fundamental robot kinematics are given, and illustration of obstacle avoidance implication are demonstrated with the control objective. In Section 3, original QP formulation is introduced first, then an optimization control problem is shown. With further in-depth derivations, we summarize a sophisticated constrained optimization problem. Followed by Section 4 an RNN algorithm is implemented for solutions to the inequality constraint and analysis of the tracking error in Cartesian space is also discussed. In Section 5, numerical and experimental results and comparisons are conducted on three 2-link Scara planar robot manipulators. Lastly, a comprehensive conclusions are included to summarize our overall effort devoted in this work. Before ending the introductory section, we highlight the main contributions of this paper as below:

- By the proposed RNN based control scheme, both trajectory tracking and collision avoidance can be realized concurrently. Meanwhile, Manipulators' kinematic and dynamic constraints are satisfied. We are able to keep manipulators running with pre-defined trajectory while complying with a safe distance between one another.
- An innovative method of obstacle avoidance between multiple SCARA manipulators in the form of multi-robot-collaboration system is introduced.

The RNN algorithm(QP-based optimization) we designed for this research is capable of simultaneously guaranteeing controlled quantity in real time, and maintaining the stability of the control system.

- The distinction from others is that our method first considers planning collision-free path of multiple planar Scara manipulators, and then takes further exploration on physical simulation held by these 2-link robots, particularly to test robot's self-correcting competency when pre-defined path is impacted and minimized the production loss(i.e. machine damages).

2. Problem Formulation

In this section, basic knowledges of robot kinematics and obstacle avoidence are presented to lay a foundation for latter illustrations.

2.1. Robot Kinematics

The forward kinematics of a non-manipulator involves a nonlinear transformation from a joint space to a Cartesian workspace, as described by

$$r(t) = f(\theta(t)). \tag{1}$$

where $r(t) \in \mathbb{R}^m$ is an m-dimensional vector in the workspace that describes the position and orientation of the end effector at time t, $\theta(t) \in \mathbb{R}^n$ is an n-dimensional vector in the joint space, each element of which describes a joint angle, and $f(\cdot) : \mathbb{R}^n \to \mathbb{R}^m$ denotes the nonlinear mapping from the joint space to the task space of the manipulator. Because of the nonlinearity of the mapping $f(\cdot)$, it is usually difficult to directly obtain the corresponding $\theta(t)$ for a desired $r(t) = r_d(t)$, $r_d(t) : \mathbb{R} \to \mathbb{R}^m$ is a smooth function defining a desired path to be tracked by the end-effector. By contrast, the mapping from the joint space to the workspace at the velocity level is an affine mapping and thus can be used to significantly simplify the problem, which can be illustrated as follows. Computing the time derivative on both sides of (1) yields

$$\dot{r}(t) = J\dot{\theta}(t). \tag{2}$$

where $J = \partial(f)/\partial(\theta) \in \mathbb{R}^{m \times n}$ is the Jacobian matrix of $f(\cdot)$, and $\dot{r}(t)$ and $\dot{\theta}(t)$ are the Cartesian velocity and the joint velocity, respectively.

2.2. Obstacle Avoidance

Generally speaking, obstacle avoidance problem usually contains how to track the desired end-effector trajectory while simultaneously ensuring that there are no collisions with any obstacle in the workspace of the manipulator.

Similar to game theory phenomenon, each robot can be treated as a individual player, and consider all other robots as obstacles during kinematic operation. Between the robot i and the specific obstacles, there exists the closest distance with two fix points. While satisfying kinematic condition, robot i also has to maintain a safe distance that is further than the closest distance with obstacles. Every robot i cares only about those two objectives, such as the player only acting on his/her behalf.

Let Z_1 be the set of all points on robot i, and Z_2 be the set of points on obstacle(s), then the goal of obstacle avoidance of a robot manipulator is to constantly satisfy $Z_1 \cap Z_2 = \emptyset$. By presenting a safety distance d_0 between the robot and obstacle(s). The obstacle avoidance is formulated as

$$||p_i^* - p_j^*|| \geq d_0, \ p_i^* \forall Z_1, \ p_j^* \forall Z_2 \tag{3}$$

while $p_i^*, i = 1, \ldots, a$ and $p_j^*, j = 1, \ldots, b$ being the vertexes of a specific robot and another vertex of the obstacle(s) within testing environment, respectively. To avoid any possible obstacles the manipulator has to maintain that $d_{min} \geq d_0$ as shown in Fig. 1, where $d_{min} = ||p_i^* - p_j^*||$ denotes the Euclidean distance of $p_i^* - p_j^*$. d_{min} is the minimum distance between all robot manipulators for obstacle avoidence. The safety distance d_0 is a positive constant resolved by the developer.

Conventionally, the basic strategy for obstacle avoidance is to identify the points on the robotic arm that are near obstacles and then assign to them the motion component that moves those points away from the obstacle, as shown in Fig. 1. The robot motion (configuration) is changed if at least one part of the

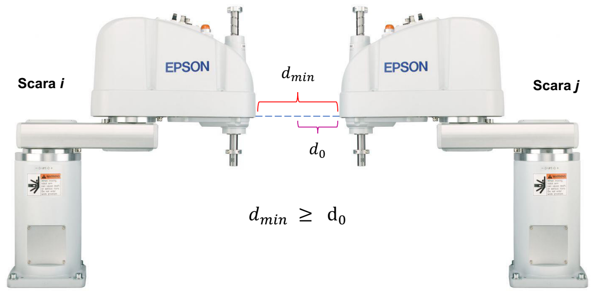


Figure 1: Two SCARA robots moving towards each other while avoiding collision. d_{min} is the minimum distance between the two robots. For safety consideration, $d_{min} \geq d_0$, where d_0 is the critical distance.

robot is at a critical distance from an obstacle. We denote the obstacles that are closer to the critical distance as the active obstacles and the corresponding closest points on the body of the manipulator as the critical points.

Remark 1. For industrial robots it is usually assumed that the motion of the end-effector is not disturbed by any obstacle. If such a situation occurs, either the task execution has to be interrupted and the higher-level path planning has to recalculate the desired motion of the end-effector.

2.3. Control Objective

In this paper, we consider obsctacle avoidance problem of multi-robot collaboration for multiple planar non-redundant robot manipulators, where precise values of kinematic parameters are available. Based on this target, we proposed the safety precaution mechanism. In order for us to achieve this goal, we adjust this kinematic control problem to an optimization problem in which obstacle avoidence condition turns into an inequality constraint of the opimization problem. More specifically, we keep this avoidence constraint always fulfilled, while trying to make end-effector's kinematic error as small as possible.

The control objective, theoretically speaking, is to express d_{min} in the inequality constraint, and with the aid of our RNN algorithm as the main tool for solution, we can manage to find the minimum value of the optimization problem which is down to the joint velocity level to find the norm distance of critical points of joints' actual speed $\dot{p}_i^*$ to the end-effectors' desired speed $\dot{r}_{id}$. As mentioned earlier, d_{min} is the minimum distance between all robot manipulators for obstacle avoidence. However, how to find d_{min} and keep the condition $d_{min} \geq d_0$ remain the biggest challenge for this research experiment.

3. Quadratic Problem Formulation

3.1. Original QP Formulation

Based on above mentioned, the kinematic control of a redundant manipulator while avoiding obstacles can be prudently described as:

$$\min \sum_{i=1}^{N} ||r_i(t) - r_{id}(t)||^2/2, \tag{4a}$$

$$s.t. \;\; r_i(t) = f(\theta_i(t)), \;\; i = 1, \cdots, N, \tag{4b}$$

where $r_i(t)$ and $r_{id}(t)$ are the actual position and desired position of end-effector respectively; p_i^*, p_j^* are the critical points resulting minimum distance d_{min}, i.e. $d_{min} = ||p_i^* - p_j^*||$.

As seen from (4), the end-effector of the manipulator is expected to track a desired path defined by $r_d(t)$, which is often referred to as the primary task of a manipulator. Specifically, during the tracking process, the position norm $r_i(t) - r_{id}(t)$ needs to be minimized.

3.2. QP Reformulation

Directly finding the direct solution for (4) is quite difficult, we need to make some adjustments if we want to receive an expressive answer. The main reason behind this phenomenon is that all three formulas were made in the displacement(position) level, which is not quite related to our control quality $\dot{\theta}$. Therefore, to bypass this barrier, the best way to go about it is to reformulate these equations to which can link to $\dot{\theta}$ directly, so that they can facilitate the progress of our research.

The derivation process can be presented as to focus on transforming the original optimization function, and from there branch out new twigs of original inequality constraint, and arrive at the ultimate version of our constrained optimization problem formulation.

For a non-redundant manipulator described by Equation(2) that is subject to the joint velocity $\dot{\theta}(t)$, we wish to find a control variable $\dot{\theta}(t)$, such that the tracking error $e_i(t) = r_i(t) - r_{id}(t)$ for a given reference trajectory $r_{id}(t)$ converges over time. Generating joint velocity $\dot{\theta}(t)$ command in real-time to ensure the difference between $r_i(t) - r_{id}(t)$ coverges to zero.

By using the formula $\dot{e}(t) = -C_0 e(t)$, which guarantees that $e(t)$ exponentially converges to zero with constant $C_0 > 0 \in \mathbb{R}$ being the parameter to scale the convergence rate, the following equation is obtained:

$$\dot{r}_{iref}(t) = \dot{r}_{id}(t) - C_0(r_i(t) - r_{id}(t)), \tag{5}$$

From (4) and (5), the objective function can be rewritten as follows:

$$\min \sum_{i=1}^{N} ||r_i(t) - r_{iRef}(t)||^2/2. \tag{6}$$

Based on $d_{min} = ||p_i^* - p_j^*||$, we can consider p_i^*, p_j^* at the velocity level, which equal $\dot{p}_i^*, \dot{p}_j^*$. We can further rewrite the formula for the inequality constraint as following:

$$\begin{aligned} &d_{min} \geq d_0 <=> ||p_i^* - p_j^*||^2/2 \geq d_0^2/2, \\ &(p_i^* - p_j^*)^{\mathrm{T}}(\dot{p}_i^* - \dot{p}_j^*) \geq -k_1[d_{min}^2/2 - d_0^2/2] \end{aligned} \tag{7}$$

By algebraically rewritting, we have

$$(p_i^* - p_j^*)^{\mathrm{T}}(J_i^* \dot{\theta}_i^* - J_j^* \dot{\theta}_j^*) \geq -k_1[d_{min}^2/2 - d_0^2/2], \tag{8}$$

with $k_1 > 0$ is used to adjust the tracking accurracy of the manipulator to the desired trajectory, and then we define a_1, a_2, a_3 to be the value of:

$$a_1 \equiv -(p_i^* - p_j^*)^{\mathrm{T}} J_i^*,$$
$$a_2 \equiv (p_i^* - p_j^*)^{\mathrm{T}} J_j^*,$$
$$a_3 \equiv -k_1[d_{min}^2/2 - d_0^2/2].$$

From Equation (8), we have the following format:

$$a_1\dot{\theta}_i^* + a_2\dot{\theta}_j^* + a_3 \leq 0. \tag{9}$$

Refer back to (2), $\dot{\theta}_i$ is the angular speed of a specific joint of the manipulator, J_i is the Jacobian matrix from the critical point to joint space. Then the velocities of critical points can be described as $\dot{r_i} = J_i\dot{\theta}_i^*$.

Hence, we have the new form of constrained optimization problem:

$$\min ||J_i\dot{\theta}_i^* - \dot{r}_{iRef}||^2/2, \tag{10a}$$
$$s.t.\ a_1\dot{\theta}_i^* + a_2\dot{\theta}_j^* + a_3 \leq 0. \tag{10b}$$

4. Design of Recurrent Neural Network

As stated in Section I, the kinematic control of non-redundant manipulators using RNNs has been extensively studied in recent decades. Although the existing methods of this type differ in the objective functions or neural dynamics used, most of them follow similar design principles. The redundant manipulator control problem is typically formulated as a constrained quadratic optimization problem, which can be equivalently converted into a set of implicit equations. Then, a convergent RNN model, the equilibrium of which is identical to the solution of this implicit equation set, is devised to solve the problem recursively.

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. 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.

4.1. Setup Lagrange Function

Consider a Lagrange function as

$$L = \sum_{i=1}^{n} (J_i \dot{\theta}_i - \dot{r}_{iRef})^{\mathrm{T}} (J_i \dot{\theta}_i - \dot{r}_{iRef})/2 + u(a_1 \dot{\theta}_i^* + a_2 \dot{\theta}_j^* + a_3), u \in \mathbb{R}^m \tag{11}$$

with u serves as the Lagrange multiplier corresponding to the inequality constraint (10).

4.2. KKT Condition

According to Karush-Kuhn-Tucker conditions, the optimal solution of the optimization problem (11) can be equivalently formulated as:

$$\frac{\partial L}{\partial \theta_i(t)} = 0 \tag{12a}$$

$$(u + a_1 \dot{\theta}_i^* + a_2 \dot{\theta}_j^* + a_3)^+ = u \tag{12b}$$

Which can be elaborated as:

$$J_i^{\mathrm{T}} (J_i \dot{\theta}_i - \dot{r}_{iRef}) + a_2^{\mathrm{T}} u = 0, \ for\ robot\ i \tag{13a}$$

$$J_j^{\mathrm{T}} (J_j \dot{\theta}_j - \dot{r}_{jRef}) + a_1^{\mathrm{T}} u = 0, \ for\ other\ robot \tag{13b}$$

$$(u + a_1 \dot{\theta}_i^* + a_2 \dot{\theta}_j^* + a_3)^+ = u \tag{13c}$$

In Equation (12b) and (13c), the opertation function $(\cdot)^+$ is defined as a mapping to the non-negative space. For instance, $x^+ = max(x, 0)$ takes the positive part of the real x, denoted by x^+.

4.3. RNN design

The solution of (13) is exact optimal solution of the constrained-optimization problem (10), it is still a challenging issue to solve (13) online since its inherent nonlinearity. In this paper, in order to solve (13), a recurrent neural network is designed as:

$$\ddot{\theta}(t) = -k_0[J_i^{\mathrm{T}}(J_i\dot{\theta}_i - \dot{r}_{iRef}) + a_1^{\mathrm{T}}u], \tag{14a}$$

$$\ddot{\theta}(t) = -k_0[J_j^{\mathrm{T}}(J_j\dot{\theta}_j - \dot{r}_{jRef}) + a_2^{\mathrm{T}}u], \tag{14b}$$

$$\dot{u} = k_0[(u + a_1\dot{\theta}_i^* + a_2\dot{\theta}_j^* + a_3)^+ - u]. \tag{14c}$$

where $k_0 > 0$ is a constant which is used to scale the convergence rate of neural network. $\lambda = 0$ will always hold for $d_{min} = ||p_i^* - p_j^*|| \geq d_0$, except when the collision detection mechanism is met, (14c) comes into play. Now, $\lambda > 0$.

Algorithm 1 Multiple SCARAs cooperation incorporated with obstacle avoidance based on RNN

Input: Control parameters ε, k_1, number of SCARAs n; SCARA parameters r_d, L, d_0; task duration and internal t_s, d_t, safety distance d_0; initial parameters $\theta_i(0)$, $\dot{\theta}_i(0)$, base; desired trajectory information r_{id} and $\dot{r}_{id}$; feedback information r_{iRef} and $\dot{r}_{iRef}$.

Output: To simultaneously achieve obstacle avoidance and global cooperation between multiple SCARAs.

```
while t < t_s do
    Reading ṙ_iRef, θ(t) by sensor.
    Calculate ṙ_iRef and error e = r_id − r_iRef.
    Calculate A, B.
    Update joint velocities of every SCARA using Eq.(26a).
    Update state variable λ using Eq.(26b).
    Update θ and r_iRef.
end while
```

4.4. Optimality and Convergence Analysis

In this subsection, we provide stability and convergence analysis of the obstacle avoidance algorithm based on sophiticated RNN method to show our controller can reach an optimal solution through iterations of feedback distance tracking. The theoretical deriviation process relies on important definitions and lemmas.

Definition 1 (Projection Operator). The projection operator for a set $S \subset \mathbb{R}^m$ and $x \in S$ is defined by:

$$P_S(x) = argmin_{y \in S} \parallel y - x \parallel^2, \tag{15}$$

where $\parallel \cdot \parallel$ denotes the Euclidean norm.

Definition 2 (Monotone Mapping). A mapping $F(\cdot)$ is called monotone if for each pair of points (x,y), there is:

$$(x - y)^{\mathrm{T}}(F(x) - F(y)) \geq 0. \tag{16}$$

This property can be extended to multi-variable mappings. For a continuously differentiable mapping $F(\cdot)$, it is a monotone mapping if

$$\nabla F + \nabla^{\mathrm{T}} F \geq 0, \tag{17}$$

where '≥ 0' means the left side of this operator is positive semi-definite, knowing ∇F is the gradient of $F(\cdot)$.

Lemma 1 (Convergence Of Dynamic Neural Networks). Assume that $F(x)$ is monontone and continuously differentiable. The dynamic system (18) is said to converge to its equilibrium point correspond to:

$$k\dot{x} = -x + P_s(x - \varrho F(x)), \tag{18}$$

where $k > 0$ and $\varrho > 0$ are both positive constants. (15) is a projection operator to closed set S.

Proof: There are two parts of analysis: Part I is to reformulate the designed RNN (14) as the form of (18). Part II is to compute the value of the expression shown in (17).

Part I: First provide value ranges of three parameters, assuming there are N robots in the environment, let

$$r_i = \begin{bmatrix} r_1 \\ r_2 \\ \vdots \\ r_N \end{bmatrix} \in \mathbb{R}^{2N}, r_{id} = \begin{bmatrix} r_{1d} \\ r_{2d} \\ \vdots \\ r_{Nd} \end{bmatrix} \in \mathbb{R}^{2N} \tag{19}$$

$$\theta_i = \begin{bmatrix} \theta_1 \\ \theta_2 \\ \vdots \\ \theta_N \end{bmatrix} \in \mathbb{R}^{2N}. \tag{20}$$

by rewritting (8) as $H\dot{\theta} \leq H_R$, where H is $-(p_i^* - p_j^*)^{\mathrm{T}}(J_i - J_j)$ and H_R is $k(||r_i - r_j||^2/2 - d_0^2/2)$. From (10) We have:

$$\min ||r_i - r_{iRef}||^2/2, \tag{21a}$$

$$s.t.\ H\dot{\theta} \leq H_R. \tag{21b}$$

To elaborate on 21b, $H = -ABC$. Matrix $A = diag(M) \in \mathbb{R}^{(N^2-N)/2\times(N^2-N)}$ with $M = [m_{12}, m_{13}, \cdots,$
$m_{ij}, \cdots, m_{(N-1)N} \in \mathbb{R}^{1\times 2N}]$. $diag(\bullet)$ denotes a diagonal matrix and m_{ij} is represented as $m_{ij} = (p_i^* - p_j^*)^T, i = 1, \cdots, N-1; j = i+1, \cdots, N$. The complete form of A is displayed bellow:

$$A = \begin{bmatrix} (p_1^* - p_2^*)^T & & & \\ & (p_1^* - p_3^*)^T & & \\ & & \ddots & \\ & & & (p_{N-1}^* - p_N^*)^T \end{bmatrix} \in \mathbb{R}^{(N^2-N)/2\times(N^2-N)}, \tag{22}$$

Additionally, B and C are expressed in the following:

$$B = \begin{bmatrix} I & -I & 0 \\ I & 0 & -I \\ 0 & I & -I \end{bmatrix} \in \mathbb{R}^{(N^2-N)\times 2N}, \tag{23}$$

$$C = \begin{bmatrix} J_1 & & & \\ & J_1 & & \\ & & \ddots & \\ & & & J_N \end{bmatrix} \in \mathbb{R}^{2N \times 2N}. \tag{24}$$

The Lagrange function and its derivative are compute as followed:

$$L = ||r_i - r_{iRef}||^2/2 + \lambda^{\mathrm{T}}(H\dot{\theta} - H_R), \tag{25a}$$

$$\frac{\partial L}{\partial \dot{\theta}} = J^{\mathrm{T}}(\dot{r} - \dot{r}_{iRef}) + H^{\mathrm{T}}\lambda. \tag{25b}$$

The designed RNN(14) becomes:

$$\begin{aligned} k\ddot{\theta} &= -\dot{\theta} + P_{\Omega}(\dot{\theta} - \frac{\partial L}{\partial \dot{\theta}}) \\ &= -\dot{\theta} + P_{\Omega}(\dot{\theta} - J^{\mathrm{T}}(J\dot{\theta} - \dot{r}_{iRef}) - H^{\mathrm{T}}\lambda), \end{aligned} \tag{26a}$$

$$k\dot{\lambda} = -\lambda + (\lambda + H\dot{\theta} - H_R)^{+}. \tag{26b}$$

Part II: Define $x = [\dot{\theta}\ ; \lambda]$, so that (26) can be converted to:

$$F(x) = \begin{bmatrix} J^{\mathrm{T}}(J\dot{\theta} - \dot{r}_{iRef}) + H^{\mathrm{T}}\lambda \\ -H\dot{\theta} + H_R \end{bmatrix}.$$

From (17), we have:

$$\nabla F(x) = \begin{bmatrix} J^{\mathrm{T}}J & H^{\mathrm{T}} \\ -H & 0 \end{bmatrix},$$

and

$$(\nabla)^{\mathrm{T}} F(x) = \begin{bmatrix} J^{\mathrm{T}}J & -H \\ H^{\mathrm{T}} & 0 \end{bmatrix}.$$

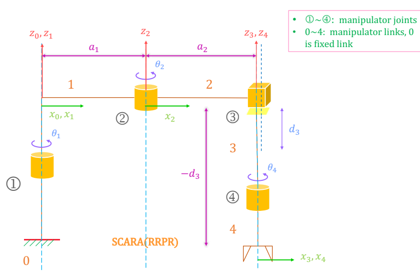
Thus, F(x) is countinuously differentiable in light of the existence of $\nabla F(x)$; furthermore, we can sum them together to produce:

$$\nabla F(x) + (\nabla)^{\mathrm{T}} F(x) = \begin{bmatrix} 2J^{\mathrm{T}}J & 0 \\ 0 & 0 \end{bmatrix}.$$

According to Definition 2, we can conclude that $\nabla F(x) + (\nabla)^{\mathrm{T}} F(x)$ is indeed positive semi-definite, and $F(x)$ is monotone function.

From Lemma 1, it can be summarized from the above analyses with generalized description that the proposed dynamical neural network(14) is stable and is globally convergent to the optimal solution of (10). The proof is completed.

5. Numerical Results



(a)

i	α_{i-1}	a_{i-1}	d_i	θ_i
1	0	0	0	θ_1
2	0	a_1	0	θ_2
3	0	a_2	$-d_3$	0
4	0	0	0	θ_4

(b)

Figure 2: Scara manipulator kinematic properties. (a) Analysis of Scara's physical structure. (b) D-H parameters of Scara manipulator.

In this section, the proposed RNN based controller is applied to three planar 2-link robots. Fig. 2 is a detailed self-explainatory supplement to our agenda. Complying to the principle of our main task, a simplified yet sufficient version of the Scara robot—2-link planar non-redundant type of Scara robot is going to be the main object that we studied in this experiment. Firstly, a successful case where manipulators are allowed to draw circles by pre-defined trajectory planning is discussed, and then the controller is tested to run with the case that manipulators follow the safety protocal which cannot finish the task because of obstacles avoidance mechanism.

5.1. Simulation Setup

To verify the effectiveness of our controller, we decide to implement our simulation on three Scara manipulators for validation. The general physical structure of the 2-link Scara manipulator can be dissected as shown in Figure. 2(a). The Kinematic characteristics are also displayed in Figure. 2(b). It is noteworthy to point out that Jacobian matrix J_2, critical points coordinates P_i, manipulators' angular speed and end-effectors' coordinates are the key variables in the proposed control scheme.

For our simulation, the 3D model of Scara manipulator in Figure. 3. marked with critical points P_1, P_2 *and* P_3 is more intuitive for inspection and comprehension. As shown in Figure. 3, critical points P_1, P_2 *and* P_3 are selected at the center of maniputors' joints.

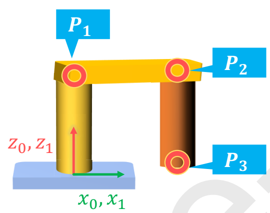


Figure 3: 3D model of Scara manipulator

Based on the above description of Scara manipulators, for the sake of this research experiment, we establish the following parameters: The values of the number of manipulators n, manipulators' link length L, manipulators' base coordinates $base$, manipulators' initial joints' angle θ, manipulators' initial joints' angular speed $\dot{\theta}$, and manipulators' end-effectors' initial coordinates r_d are all known at the start of this experiment.

Since the goal of controlled simulation is to the express d_{min} such that $d_{min} \geq d_0$ is fulfilled, our focus next is to express this inequality constraint as the relationship between critical points of the manipulator. Therefore, to find the norm of the distance between the critical point of a manipulator p_1^* and another critical point of second manipulator p_2^* is the next step, i.e. $d_{min} = ||p_1^* - p_2^*||$. After the long inequality constraint is formed as Equation(12) shown, we use a_1, a_2, a_3 to denote three polynomials as parts of the Equation(13), and we realize these formulas in the core body of the RNN algorithm code. Finally, the optimization index *mu* is achieved and updated as simulation proceed.

5.2. Simulation Results

The RNN controller is comprised of two parts: motion tracking and collision avoidance. When robots run in a collision-free environment, motion tracking plays the dominant role; however, as collision is detected, the obstacle avoidance mechanism takes control.

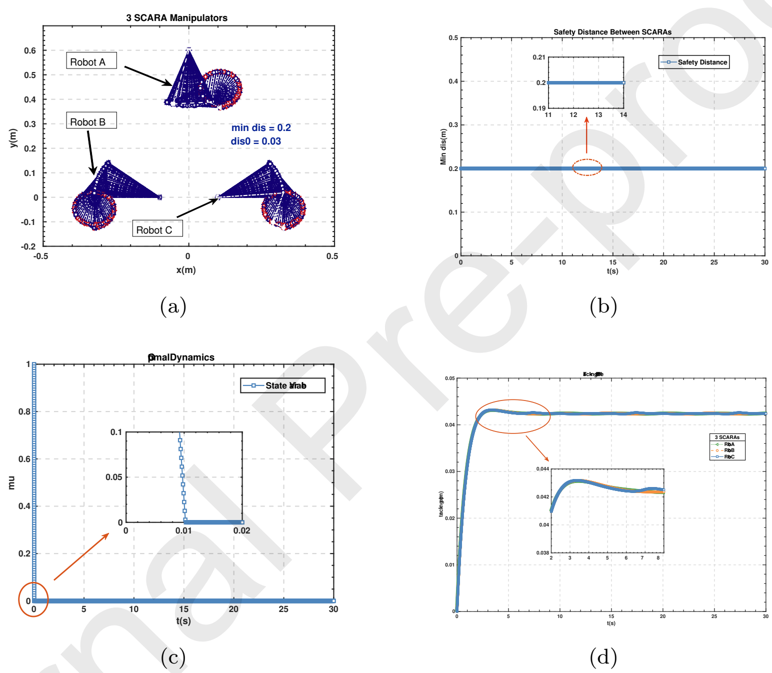
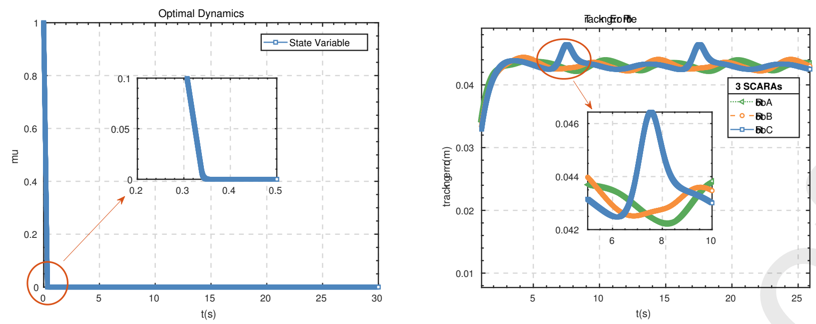


Figure 4: Simulation results for the kinematic control of end effector of three 2-link Scara robots where each of them is placed to be interference-free from other robots along a circular path. (a) End-effector trajectory (blue curver) with respect to the reference position. (b) Minimum distance determined by RNN algorithm to maintain safety between robots. (c) Ensuring the inequality constraint (10b), Optimized index(mu) is mesured at $k_0 = 10^5$. (d) Tracking errors with respect to time history at $k_0 = 10^5$.

5.2.1. Obstacle Free

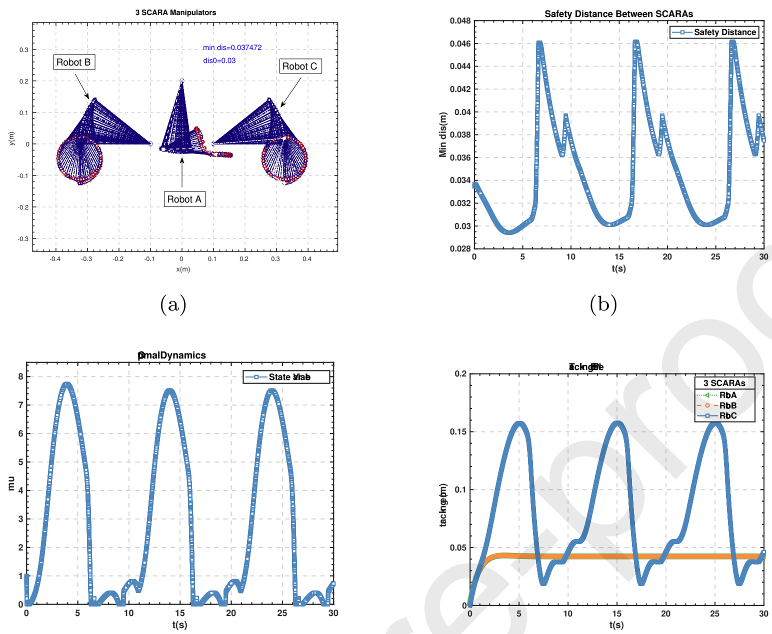
As indicated in Fig. 4, from plot (a) can readily see the harmonious working environment between the three 2-link planar Scara robots. One robot is located



(a) (b)

Figure 5: Simulation results for the kinematic control of end effector of three 2-link Scara robots where each of them is placed to be interference-free from other robots along a circular path. It is remarkable to note that $k_0 = 10^3$ achieves slower convergence of mu, thus The larger k_0, the faster the RNN converges. (a) Optimized index(smaller k_0). (b) Obstacle-free tracking errors

on top of the figure, and the other two robots are based on the middle part of figure, all of them are placed on the same platform. In this case, we select $dis0 = 0.03$ to be the value for safety distance, and the instant value for $min\ dis$ after the simulation finished is 0.2. The state variable also named as the optimization index u is put into the control scheme so that the inequality constraint (10b) is ensured, as shown in (13c) and also in the code where the kernal of RNN is placed. The Index u is updated according to the difference between actual speed $J_i\dot{\theta}_i^*$ and reference speed $\dot{r}_{iRef}$. During the experiment, it is engaging to find out that k_0 plays an significant role in the convergence of the RNN control system. The larger k_0, the faster the RNN converges. From Fig. 4(c) and (d), the simulation performance result can verify our initial statement which is this simulation is successfully ran. The zoom-in figure within Fig. 4(d) exhibits the process is carried smoothly and congruently, indicating no collision risk. From Fig. 5(a) and (b), an illustration of the optimized controller converges in a slower response.



(a) (b) (c) (d)

Figure 6: Simulation results for the kinematic control of end effector of three 2-link Scara robots where each of them is placed to have interference(obstacle detected) from other robots along a circular path. (a) End-effector trajectory (blue curver) with respect to the reference position. (b) Minimum distance determined by RNN algorithm to maintain safety between robots. (c) Ensuring the inequality constraint (10b), Optimized index(mu) is mesured at $k_0 = 10^5$. (d) Tracking errors with respect to time history at $k_0 = 10^5$.

5.2.2. Obstacle Encountered

In this scenario, an illustration of the case when the middle Scara robots strived for trajectory tracking as its original kinematic motion object, but as displayed that the middle robot is contiuously moving in its best to approach the pre-defined curve, this is under the influence of the safety precaution implanted to prevent robots from running too close to collide with each other. We can also comprehend this as Fig. 6(a) and 6(d) shown, both plots indicate that the middle robot tried to avoid running into the robot on the right during simulation. In Fig. 6(d), both the left and right robots are running ordinarily with planned trajectories, however the middle robot is following safety protocal to avoid collision, this is the reason that the purple line deviates from other two tracking errors. Based on this phenomenon, an conclusion can be summed up: 1) Avoid placing robots too close to one another. 2) The safety distance can affect the simulation as it can pose a hedge while robots are in operation. 3) Slow down the angular speed of end-effectors can support the movement realization as to give more time for robots to react with incoming obstacles.

6. Conclusion

In allusion to obstacles avoidance between operational robots among multi-robot collaboration, this paper represents the knowledge of Recurrent Neural Network method specialized designed for preventing collisions between robot manipulators, and proposes a corresponding mechanism of safety assurance based on a unique control constraint that can trigger the precautions when certain parts of machines are determined to be within the safe distance. Utilizing geometric features of manipulators, the manipulators can be denoted by sets of critical points, thereby the distance between the robots is approximately described as point-to-points distances. Therefore, the collision avoidance strategy can be formulated as an inequality constraints. By keeping the minimal distance between robots, collision-free environment is ensured. With three Scara robots, we perform simulative experiments on Matlab, indicating that when the minimal distance between robots is less than the setting safety distance, the collision avoidance strategy come in the control command, the robots successfully avoid collision with other manipulators. If forthcoming path is free of obstacles, the robot performs the desired trajectory tracking task with a promising tracking error. In the future, this proposed method can be further upgraded to apply for multiple redundant industrial manipulators in three-dimensional space. Each robot act as a player in the game theory, and by having an inequality control constraint which treats all other robots as obstacles, a congruous industrial multi-robot collaboration platform can be achieved.

References

[1] T. Arai, E. Pagello, L. E. Parker, et al., Advances in multi-robot systems, IEEE Transactions on robotics and automation 18 (5) (2002) 655–661.

[2] W. Burgard, M. Moors, C. Stachniss, F. E. Schneider, Coordinated multi-robot exploration, IEEE Transactions on robotics 21 (3) (2005) 376–386.

[3] B. Yamauchi, Frontier-based exploration using multiple robots, in: Proceedings of the second international conference on Autonomous agents, 1998, pp. 47–53.

[4] R. Kurazume, S. Nagata, S. Hirose, Cooperative positioning with multiple robots, in: Proceedings of the 1994 IEEE International Conference on Robotics and Automation, IEEE, 1994, pp. 1250–1257.

[5] T. Petrič, A. Gams, N. Likar, L. Žlajpah, Obstacle avoidance with industrial robots, in: Motion and Operation Planning of Robotic Systems, Springer, 2015, pp. 113–145.

[6] H. Lee, H. Kim, H. J. Kim, Path planning and control of multiple aerial manipulators for a cooperative transportation, in: 2015 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), IEEE, 2015, pp. 2386–2391.

[7] W. F. Carriker, P. K. Khosla, B. H. Krogh, Path planning for mobile manipulators for multiple task execution, IEEE Transactions on Robotics and Automation 7 (3) (1991) 403–408.

[8] X. Li, J. Xiao, Z. Cai, Backstepping based multiple mobile robots formation control, in: 2005 IEEE/RSJ International Conference on Intelligent Robots and Systems, IEEE, 2005, pp. 887–892.

[9] C. Cai, C. Yang, Q. Zhu, Y. Liang, Collision avoidance in multi-robot systems, in: 2007 International Conference on Mechatronics and Automation, IEEE, 2007, pp. 2795–2800.

[10] D. P. Garg, M. Kumar, Optimization techniques applied to multiple manipulators for path planning and torque minimization, Engineering applications of artificial intelligence 15 (3-4) (2002) 241–252.

[11] Y. Zhang, J. Wang, Obstacle avoidance of redundant manipulators using a dual neural network, in: 2003 IEEE International Conference on Robotics and Automation (Cat. No. 03CH37422), Vol. 2, IEEE, 2003, pp. 2747–2752.

[12] 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) (2004) 752–759.

[13] Z. Xu, X. Zhou, S. Li, Deep recurrent neural networks based obstacle avoidance control for redundant manipulators, Frontiers in neurorobotics 13 (2019) 47.

[14] A. H. Khan, S. Li, X. Luo, Obstacle avoidance and tracking control of redundant robotic manipulator: An rnn based metaheuristic approach, IEEE Transactions on Industrial Informatics.

[15] G. Li, Q. Jia, Cooperative receding horizon path planning of multiple robots by genetic algorithm, in: 2008 IEEE International Conference on Automation and Logistics, IEEE, 2008, pp. 2449–2453.

[16] S. Sun, A. S. Morris, A. M. Zalzala, Trajectory planning of multiple coordinating robots using genetic algorithms, Robotica 14 (2) (1996) 227–234.

[17] G. Peng, C. Yang, W. He, C. P. Chen, Force sensorless admittance control with neural learning for robots with actuator saturation, IEEE Transactions on Industrial Electronics 67 (4) (2020) 3138–3148.

[18] H. Huang, T. Zhang, C. Yang, C. P. Chen, Motor learning and generalization using broad learning adaptive neural control, IEEE Transactions on Industrial Electronics 67 (10) (2020) 8608–8617.

[19] C. Yang, C. Chen, W. He, R. Cui, Z. Li, Robot learning system based on adaptive neural control and dynamic movement primitives, IEEE transactions on neural networks and learning systems 30 (3) (2019) 777–787.

[20] Y. Yamamoto, S. Fukuda, Trajectory planning of multiple mobile manipulators with collision avoidance capability, in: Proceedings 2002 IEEE International Conference on Robotics and Automation (Cat. No. 02CH37292), Vol. 4, IEEE, 2002, pp. 3565–3570.

[21] J. Wang, X. Ren, J. Liu, Trajectory planning for multi-robot formation by one hybrid particle swarm optimization algorithm, in: 2013 5th International Conference on Intelligent Human-Machine Systems and Cybernetics, Vol. 1, IEEE, 2013, pp. 344–348.

[22] A. Mohri, M. Yamamoto, G. Hirano, Cooperative path planning for two manipulators, in: Proceedings of IEEE International Conference on Robotics and Automation, Vol. 3, IEEE, 1996, pp. 2853–2858.

[23] F. Lewis, S. Jagannathan, A. Yesildirak, Neural network control of robot manipulators and non-linear systems, CRC Press, 1998.

[24] G. Wittenberg, Developments in offline programming: an overview, Industrial Robot: An International Journal 22 (3) (1995) 21–23.

[25] Z. Pan, J. Polden, N. Larkin, S. Van Duin, J. Norrish, Recent progress on programming methods for industrial robots, in: ISR 2010 (41st International Symposium on Robotics) and ROBOTIK 2010 (6th German Conference on Robotics), VDE, 2010, pp. 1–8.

[26] C. Kohrt, R. Stamp, A. Pipe, J. Kiely, G. Schiedermeier, An online robot trajectory planning and programming support system for industrial use, Robotics and Computer-Integrated Manufacturing 29 (1) (2013) 71–79.

[27] F. L. Lewis, D. Liu, Reinforcement learning and approximate dynamic programming for feedback control, Vol. 17, John Wiley & Sons, 2013.

[28] A. Nagabandi, G. Kahn, R. S. Fearing, S. Levine, Neural network dynamics for model-based deep reinforcement learning with model-free fine-tuning, in: 2018 IEEE International Conference on Robotics and Automation (ICRA), IEEE, 2018, pp. 7559–7566.

[29] J. Botzheim, Y. Toda, N. Kubota, Bacterial memetic algorithm for offline path planning of mobile robots, Memetic Computing 4 (1) (2012) 73–86.

[30] S. Lin, A. A. Goldenberg, Neural-network control of mobile manipulators, IEEE Transactions on Neural networks 12 (5) (2001) 1121–1133.

[31] Y. H. Kim, F. L. Lewis, Neural network output feedback control of robot manipulators, IEEE Transactions on robotics and automation 15 (2) (1999) 301–309.

[32] R. W. Parks, D. L. Long, D. S. Levine, D. J. Crockett, E. G. McGeer, P. L. McGeer, I. E. Dalton, R. F. Zec, R. E. Becker, K. L. Coburn, et al., Parallel distributed processing and neural networks: Origins, methodology and cognitive functions, International journal of neuroscience 60 (3-4) (1991) 195–214.

[33] L. Jin, S. Li, J. Yu, J. He, Robot manipulator control using neural networks: A survey, Neurocomputing 285 (2018) 23–34.

[34] W. He, Y. Chen, Z. Yin, Adaptive neural network control of an uncertain robot with full-state constraints, IEEE transactions on cybernetics 46 (3) (2015) 620–629.

[35] B. Jin, Robotic manipulator trajectory control using neural networks, in: Proceedings of 1993 International Conference on Neural Networks (IJCNN-93-Nagoya, Japan), Vol. 2, IEEE, 1993, pp. 1793–1796.

[36] N. Bin, C. Xiong, Z. Liming, X. Wendong, Recurrent neural network for robot path planning, in: International Conference on Parallel and Distributed Computing: Applications and Technologies, Springer, 2004, pp. 188–191.

[37] W. S. Tang, C. M. L. Lam, J. Wang, Kinematic control and obstacle avoidance for redundant manipulators using a recurrent neural network, in: International Conference on Artificial Neural Networks, Springer, 2001, pp. 922–929.

[38] H. Ding, J. Wang, Recurrent neural networks for minimum infinity-norm kinematic control of redundant manipulators, IEEE Transactions on Systems, Man, and Cybernetics-Part A: Systems and Humans 29 (3) (1999) 269–276.