

ORIGINAL RESEARCH

Recursive recurrent neural network: A novel model for manipulator control with different levels of physical constraints

Zhan Li¹  | Shuai Li²¹Department of Computer Science, Swansea University, Swansea, UK²College of Engineering, Swansea University, Swansea, UK**Correspondence**Zhan Li, Department of Computer Science, Swansea University, Swansea, UK.
Email: zhan.li@swansea.ac.uk**Funding information**

IMPACT Fund of Swansea University

Abstract

Manipulators actuate joints to let end effectors to perform precise path tracking tasks. Recurrent neural network which is described by dynamic models with parallel processing capability, is a powerful tool for kinematic control of manipulators. Due to physical limitations and actuation saturation of manipulator joints, the involvement of joint constraints for kinematic control of manipulators is essential and critical. However, current existing manipulator control methods based on recurrent neural networks mainly handle with limited levels of joint angular constraints, and to the best of our knowledge, methods for kinematic control of manipulators with higher order joint constraints based on recurrent neural networks are not yet reported. In this study, for the first time, a novel recursive recurrent network model is proposed to solve the kinematic control issue for manipulators with different levels of physical constraints, and the proposed recursive recurrent neural network can be formulated as a new manifold system to ensure control solution within all of the joint constraints in different orders. The theoretical analysis shows the stability and the purposed recursive recurrent neural network and its convergence to solution. Simulation results further demonstrate the effectiveness of the proposed method in end-effector path tracking control under different levels of joint constraints based on the Kuka manipulator system. Comparisons with other methods such as the pseudoinverse-based method and conventional recurrent neural network method substantiate the superiority of the proposed method.

1 | INTRODUCTION

Manipulators, which can greatly reduce the heavy burden on labour forces of workers, nowadays have been widely applied in many industrial fields such as welding, painting, and assembling areas. By using extra degrees of freedom (DOFs), manipulators are able to make use of inherent redundancy to perform flexible operations and fulfil complicated tasks [1, 2]. Kinematic control of manipulators by taking advantage of such inherent redundancy in various circumstances have drawn intensive interests and concerns in recent years [3].

Generally, kinematic control of manipulators is to seek a suitable control action in the joint space that produces a desired motion for the end effector in the workspace.

However, strong coupled non-linearity exists in the mapping between a joint space and a Cartesian workspace of a redundant manipulator when it executes manipulation tasks. It is difficult to tackle this problem for analytical solutions in the joint space level through directly solving the coupled non-linear equations that are used to describe the kinematic characteristics of manipulators. Therefore, the kinematic control problem in the joint space is converted into a problem depicted by the velocity kinematic equations. Some early research studies found the control solutions directly by solving the pseudoinverse of the Jacobian matrix of a manipulator [4, 5], and such a way of processing may increase local instability and even need more computational costs which were not expected.

This is an open access article under the terms of the Creative Commons Attribution-NonCommercial-NoDerivs License, which permits use and distribution in any medium, provided the original work is properly cited, the use is non-commercial and no modifications or adaptations are made.

© 2022 The Authors. *CAAI Transactions on Intelligence Technology* published by John Wiley & Sons Ltd on behalf of The Institution of Engineering and Technology and Chongqing University of Technology.

To overcome the shortcomings of pseudoinverse-based approaches, optimisation-based (e.g. quadratic programming) methods have been proposed to find the optimal control solutions [6, 7]. Such optimisation-based methods can involve physical constraints into the optimisation formulation [8] and solutions to such constrained optimisation problems cannot be obtained in an analytical manner as well. Thus, numerical methods in a way of serial processing were applied, but they are still resulting in low computational efficiency. In order to remedy this weakness, recurrent neural networks with parallel processing ability have been proposed for manipulator control [9–18]. Many researchers have developed kinematics control paradigms with the involvement of one or two of the three levels of constraints: joint angle and joint velocity, and joint acceleration constraints. In ref. [19], joint limits and joint velocity limits are taken into account for the kinematic resolution based on a dual neural network. In ref. [20], automatic handling of kinematic constraints is proposed in a real-time operational space motion planner. In ref. [21], repetitive motion planning is performed with acceleration-level constraints. In ref. [22], kinematic control of manipulators with obstacle avoidance in the joint-acceleration level is proposed with a minimum acceleration norm as the optimisation objective. In ref. [23], kinematic control of the manipulators with joint velocity constraints is solved by a passivity-based approach from an energy perspective. In ref. [24], the velocity-level and acceleration-level redundancy resolution are combined to form a unified quadratic programming for joint torque optimisation. In ref. [25], visual serving control of manipulators with joint physical limits to ensure safety. In ref. [26], both velocity-level and acceleration-level constraints are integrated into the kinematic control of manipulators. However, for joint constraints in a higher level such as more than the fourth-order joint constraints, kinematic control methods based on recurrent neural networks have not yet been proposed and reported.

Motivated by the aforementioned points, as current kinematic control methods based on recurrent neural networks can only deal with joint angular velocity and joint acceleration constraints, in this study, we are making breakthroughs to propose a novel recursive recurrent neural network for kinematic control of manipulators with different levels of physical constraints, with high-order joint constraints considered. The contributions of this study are summarised as follows:

- (1) To the best of our knowledge, this article is the first work to propose the kinematics control method for manipulators with different levels of physical constraints simultaneously.
- (2) The proposed method is described by a novel recursive recurrent neural network model, and the theoretical results on the purposed recursive recurrent neural network show the stability in a manifold system perspective and its convergence to kinematic control solutions with aforementioned physical constraints.
- (3) Simulation results with comparisons to the pseudoinverse-based method and conventional recurrent neural network method on the Kuka manipulator system demonstrate the

efficiency and superiority of the proposed method in kinematic control with different levels of physical constraints.

2 | PROBLEM FORMULATION

Let us consider the following velocity kinematics equation for manipulator control

$$\dot{r} = J\dot{\theta} \quad (1)$$

where $J \in R^{m \times n}$ denotes the Jacobian matrix of the manipulator, $r \in R^m$ denotes trajectory in the Cartesian space of the end effector, and $\theta \in R^n$ denotes the joint space variable. The end effector has to follow the desired path r_d to make the tracking error $e = r - r_d$ as small as possible with suitable control actions from the joint space of the manipulator. Due to the physical restrictions of joint actuation of the manipulator, generally, the joint variable usually has to follow the limits in the joint angle level, joint velocity level, joint acceleration level, and higher order levels as follows

$$\begin{aligned} \theta_0^- &\leq \theta \leq \theta_0^+ \\ \theta_1^- &\leq \dot{\theta} \leq \theta_1^+ \\ \theta_2^- &\leq \ddot{\theta} \leq \theta_2^+ \\ &\vdots \\ \theta_{n-1}^- &\leq \theta^{(n-1)} \leq \theta_{n-1}^+ \\ \theta_n^- &\leq \theta^{(n)} \leq \theta_n^+ \end{aligned}$$

where θ_0^- and θ_0^+ denote the lower and upper bounds of the joint angle, respectively, θ_1^- and θ_1^+ denote the lower and upper bounds of the joint angular velocity, respectively, θ_2^- and θ_2^+ denote the lower and upper bounds of the joint angular acceleration, respectively, and θ_i^- and θ_i^+ ($i \geq 3$) denote the lower and upper limits of the higher order joint variables, respectively. As manipulators are usually actuated by electrical motors, the high-order time derivative term $\theta^{(n)}$ ($n \geq 2$) may reflect the physical constraints.

2.1 | Conventional kinematic control of manipulator with joint constraints of lower levels

Note that, current existing works mainly focus on solving the kinematic control issue with the first two low levels of constraints for joint variables as follows:

$$\begin{aligned} r &\rightarrow r_d \\ \theta_0^- &\leq \theta \leq \theta_0^+ \\ \theta_1^- &\leq \dot{\theta} \leq \theta_1^+ \end{aligned}$$

The kinematic control solution for the manipulator with the joint angle and joint velocity constraints is depicted by the following constrained optimization problem

$$\begin{aligned} \min \quad & \dot{\theta}^T \dot{\theta} / 2 \\ \text{s.t.} \quad & \dot{r}_d = J \dot{\theta} \\ & \theta_0^- \leq \theta \leq \theta_0^+ \\ & \theta_1^- \leq \dot{\theta} \leq \theta_1^+ \end{aligned}$$

It can be further transformed to the following optimization formulation

$$\begin{aligned} \min \quad & \dot{\theta}^T \dot{\theta} / 2 \\ \text{s.t.} \quad & \dot{r}_d = J \dot{\theta} \\ & -\kappa(\theta - \theta_0^-) \leq \dot{\theta} \leq -\kappa(\theta - \theta_0^+) \\ & \theta_1^- \leq \dot{\theta} \leq \theta_1^+ \end{aligned}$$

where $\kappa > 0$ denotes the scaling parameter for constructing joint limits from the joint angle level to the joint acceleration level. Here, we define the solution set for $\dot{\theta}$ as $\Omega_0 = \{\dot{\theta} \mid -\kappa(\theta - \theta_0^-) \leq \dot{\theta} \leq -\kappa(\theta - \theta_0^+)\} \cap \{\dot{\theta} \mid \theta_1^- \leq \dot{\theta} \leq \theta_1^+\}$. Then the optimization for manipulator control issue becomes

$$\begin{aligned} \min \quad & \dot{\theta}^T \dot{\theta} / 2 \\ \text{s.t.} \quad & \dot{r}_d = J \dot{\theta} \\ & \dot{\theta} \in \Omega_0 \end{aligned}$$

For solving the optimization problem above, according to the Karush-Kuhn-Tucker condition [27], one can construct a recurrent neural network solver as follows:

$$\begin{aligned} \epsilon \dot{w} &= -w + P_{\Omega_0} \left(-\kappa J^T \int_0^t (J \dot{\theta} - \dot{r}_d) dt \right) \\ &= -w + P_{\Omega_0} (-\kappa J^T (r - r_d)) \end{aligned} \quad (2)$$

where $w \in R^n$ denotes the state variable of the recurrent neural network, $\epsilon > 0$ denotes the convergence parameter, and $P_{\Omega_0}(\cdot)$ denotes the piecewise linear projection function array.

2.2 | Kinematic control of manipulator with joint constraints of higher levels

However, as we may know, the following kinematic control issue with additional higher levels of joint variable constraints is still not considered and investigated

Tracking control of end - effector $r \rightarrow r_d$

$$\begin{aligned} \text{Lower - order constraints} & \begin{cases} \theta_0^- \leq \theta \leq \theta_0^+ \\ \theta_1^- \leq \dot{\theta} \leq \theta_1^+ \end{cases} \\ \text{Higher - order constraints} & \begin{cases} \theta_2^- \leq \ddot{\theta} \leq \theta_2^+ \\ \theta_3^- \leq \theta^{(3)} \leq \theta_3^+ \\ \vdots \\ \theta_{n-1}^- \leq \theta^{(n-1)} \leq \theta_{n-1}^+ \\ \theta_n^- \leq \dot{\theta}^{(n)} \leq \theta_n^+ \end{cases} \end{aligned}$$

This situation can be frequently encountered. For instance, in jerk operations, the manipulator needs to compute the time derivatives of the joint acceleration, which leads to the third-order term for the joint angular variable. To avoid unnecessary dynamic vibrating that is affected by large variations from joint acceleration, it would be very useful to limit the range of the fourth-order term for the joint angular variable.

Therefore the aforementioned original solver for the kinematic control of manipulator with lower order joint constraints

$$\epsilon \dot{w} = -w + P_{\Omega_0} (-\kappa J^T (r - r_d)) \quad (3)$$

is not able to handle with additional higher order constraints and simultaneously fulfil the tracking control task for the end effector, because the higher order constraint information cannot be involved in current recurrent neural network model.

As the lower order time derivative of joint variables can be obtained through the integration of the higher order time derivative of joint variables, in this paper, we try to solve the kinematic control issue by letting the higher order joint target firstly tracked and then lower order joint target tracked by the recurrent neural network solver. Under these considerations, we let the higher order time derivative of the state variable w follow

$$\begin{aligned} w^{(n-2)} &\rightarrow w_d^{(n-2)} \\ \theta_n^- &\leq w^{(n-1)} \leq \theta_n^+ \end{aligned}$$

and we get $w^{(n-1)} \rightarrow \theta^{(n)}$. And this procedure repeats for the next lower order situation, that is,

$$\begin{aligned} w^{(n-3)} &\rightarrow w_d^{(n-3)} \\ \theta_{n-1}^- &\leq w^{(n-2)} \leq \theta_{n-1}^+ \end{aligned}$$

The recursive procedures repeat until

$$\begin{aligned} \dot{w} &\rightarrow \dot{w}_d \\ \theta_3^- &\leq \ddot{w} \leq \theta_3^+ \end{aligned}$$

and we get $\ddot{w} \rightarrow \ddot{w}_d$. Thus the equilibrium point for the tracking task of the end effector is

$$w_d = P_{\Omega_0}(-\kappa J^T(r - r_d)) \quad (4)$$

then we have

$$\begin{aligned} w &\rightarrow w_d \\ \theta_2^- &\leq \dot{w} \leq \theta_2^+ \end{aligned}$$

and we get $\dot{w} = \dot{w}_d$ finally by the recursive updating of the corresponding recurrent neural network solver.

3 | CASCADED RECURSIVE RECURRENT NEURAL NETWORK

In order to achieve kinematic control of the manipulator with all levels of joint constraints, based on the aforementioned discussions, we have the equilibrium points in a different order as follows:

$$\begin{cases} w_d^{(1)} = P_{\Omega_1}(-k_1 J^T(r - r_d)) \\ w_d^{(2)} = P_{\Omega_2}(-k_2(w^{(1)} - w_d^{(1)})) \\ w_d^{(3)} = P_{\Omega_3}(-k_3(w^{(2)} - w_d^{(2)})) \\ \vdots \\ w_d^{(i)} = P_{\Omega_i}(-k_i(w^{(i-1)} - w_d^{(i-1)})) \\ \vdots \\ w_d^{(n-1)} = P_{\Omega_{n-1}}(-k_{n-1}(w^{(n-2)} - w_d^{(n-2)})) \\ w_d^{(n)} = P_{\Omega_n}(-k_n(w^{(n-1)} - w_d^{(n-1)})) \end{cases} \quad (5)$$

where $P_{\Omega_1}(\cdot)$, $P_{\Omega_2}(\cdot)$, $P_{\Omega_3}(\cdot)$, ..., $P_{\Omega_i}(\cdot)$, ..., $P_{\Omega_{n-1}}(\cdot)$, $P_{\Omega_n}(\cdot)$ denote the piecewise linear projection function arrays for the solution sets Ω_1 , Ω_2 , Ω_3 , ..., Ω_i , ..., Ω_{n-1} , Ω_n in different levels of time derivatives for joint variables, and k_1 , k_2 , k_3 , ..., k_i , ..., k_{n-1} , k_n denote the convergence scaling parameters for desired equilibrium points in different levels of time derivatives for joint variables, which are all positive scalars.

In order to track the aforementioned equilibrium points in different levels, we first define the following state variables for constructing a state-space dynamic model which can describe the time derivatives of different orders for the state variables of the recurrent neural network solver

$$\begin{cases} x_0 = w^{(0)} \\ x_1 = w^{(1)} \\ x_2 = w^{(2)} \\ \vdots \\ x_{n-1} = w^{(n-1)} \end{cases} \quad (6)$$

The state-space dynamic model is constructed as follows:

$$\begin{cases} \dot{x}_0 = x_1 \\ \dot{x}_1 = x_2 \\ \dot{x}_2 = x_3 \\ \vdots \\ \dot{x}_{n-2} = x_{n-1} \\ \dot{x}_{n-1} = P_{\Omega_n}[-k_n(x_{n-1} - w_d^{(n-1)})] \end{cases} \quad (7)$$

Therefore, the equilibrium points can be described by the state variables as follows:

$$\begin{cases} w_d^{(1)} = P_{\Omega_1}(-k_1 J^T(r - r_d)) \\ w_d^{(2)} = P_{\Omega_2}(-k_2(x_1 - w_d^{(1)})) \\ w_d^{(3)} = P_{\Omega_3}(-k_3(x_2 - w_d^{(2)})) \\ \vdots \\ w_d^{(i)} = P_{\Omega_i}(-k_i(x_{i-1} - w_d^{(i-1)})) \\ \vdots \\ w_d^{(n-1)} = P_{\Omega_{n-1}}(-k_{n-1}(x_{n-2} - w_d^{(n-2)})) \\ w_d^{(n)} = P_{\Omega_n}(-k_n(x_{n-1} - w_d^{(n-1)})) \end{cases} \quad (8)$$

Next, we define the congregation state variable vector as the following form

$$\bar{x}_{n-2} = \begin{bmatrix} x_0 \\ x_1 \\ x_2 \\ \vdots \\ x_{n-2} \end{bmatrix}. \quad (9)$$

Thus, we can get the following state-space dynamic equations

$$\begin{cases} \dot{\bar{x}}_{n-2} = f(\bar{x}_{n-2}, x_{n-1}) \\ \dot{x}_{n-1} = P_{\Omega_n}(-k_n(x_{n-1} - w_d^{(n-1)})) \end{cases} \quad (10)$$

where $f(\cdot)$ denotes the mapping function array which is composed of the linear projection function arrays $P_{\Omega_1}(\cdot)$, $P_{\Omega_2}(\cdot)$, ..., $P_{\Omega_{n-1}}(\cdot)$. The first equation is a fast-manifold relative to the second equation when $k_n \gg 1$. Applying a singular perturbation analysis [28], we conclude the following slow manifold $\bar{\tilde{x}}_{n-2}$ with the faster manifold \dot{x}_{n-1} as follows:

$$\begin{cases} \dot{\bar{\tilde{x}}}_{n-2} = f(\bar{\tilde{x}}_{n-2}, x_{n-1}, 1/k_n) \\ \dot{x}_{n-1} = P_{\Omega_n}(-k_n(x_{n-1} - w_d^{(n-1)})) \end{cases} \quad (11)$$

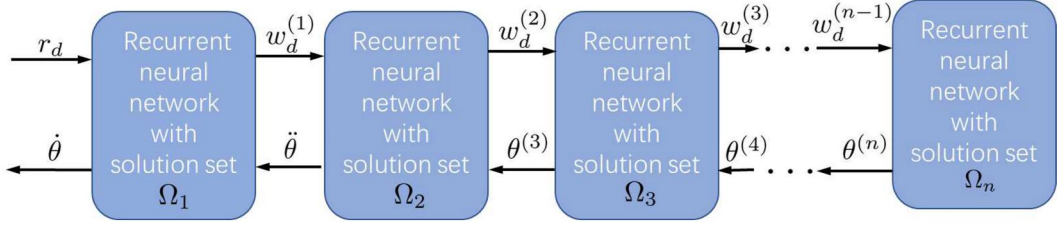


FIGURE 1 Flow chart of the proposed recursive recurrent neural network model for kinematic control with different level of physical constraints satisfied

where $0 < 1/k_n \leq 1$. We expect the manifold system, which is also a recursive recurrent neural network solver, can converge to the equilibrium points when dealing with the kinematic control of the manipulator system with high-level joint constraints involved as shown in Figure 1. In the recurrent neural networks with different solution sets, the higher order desired target variables are gradually generated as the inputs until the highest desired one is satisfied, and then the resolved joint angular variables in descending orders are generated until the resolved joint velocity variable is obtained. In this manner, lower order desired end-effector variable is acting as the input to the recurrent neural network model with a corresponding solution set to generate the output as the higher order desired end-effector variable, and higher order desired joint angular variable is acting as the input to the recurrent neural network model with the corresponding solution set to generate the output as the lower order joint angular variable. The overall computational complexity depends on the calling times of the primal-dual neural networks, but the dimension of the network variables is not increased which does not increase the complexity of each single constrained optimization. In the ensuing sections, we present the stability analysis and show the convergence properties.

4 | THEORETICAL ANALYSIS

In this section, we present a theoretical analysis on the solution of the control actions by the aforementioned recursive neural network based on the manifold system with stability and convergence results provided.

4.1 | Constraint compliance analysis

For the manifold system equations

$$\begin{cases} \dot{\bar{x}}_{n-2} = f(\bar{x}_{n-2}, x_{n-1}, 1/k_n) \\ \dot{x}_{n-1} = P_{\Omega_n}(-k_n(x_{n-1} - w_d^{(n-1)})) \end{cases} \quad (12)$$

It can be rewritten as

$$\begin{cases} \dot{\bar{x}}_{n-2} = f(\bar{x}_{n-2}, x_{n-1}, 1/k_n) \\ \dot{x}_{n-1} = k_n P_{\Omega_n}(-k_n(x_{n-1} - w_d^{(n-1)})) \end{cases} \quad (13)$$

where solution set cone Ω'_n is scaled by the parameter k_n from the original solution set cone Ω_n .

Therefore, the equations above can be equivalently

$$\begin{cases} \dot{\bar{x}}_{n-2} = f(\bar{x}_{n-2}, x_{n-1}, 1/k_n) \\ \frac{1}{k_n} \dot{x}_{n-1} = P_{\Omega'_n}(-k_n(x_{n-1} - w_d^{(n-1)})) \end{cases} \quad (14)$$

Recall the stability properties of singular perturbation system, if the second equation $\frac{1}{k_n} \dot{x}_{n-1} = P_{\Omega'_n}(-k_n(x_{n-1} - w_d^{(n-1)}))$ is asymptotically stable with $0 < 1/k_n \leq 1$, then the first equation $\dot{\bar{x}}_{n-2} = f(\bar{x}_{n-2}, x_{n-1}, 1/k_n)$ can be exponentially stable. It means that

$$\begin{cases} \dot{\bar{x}}_{n-2} = f(\bar{x}_{n-2}, x_{n-1}, 1/k_n) \\ \dot{x}_{n-1} = P_{\Omega_n}(-k_n(x_{n-1} - w_d^{(n-1)})) \end{cases} \quad (15)$$

is stable at least $\dot{x}_{n-1} = P_{\Omega_n}(-k_n(x_{n-1} - w_d^{(n-1)}))$ is asymptotically stable, that is, $\dot{x}_{n-1} \rightarrow 0$ and $x_{n-1} \rightarrow w_d^{(n-1)}$ make $\dot{\bar{x}}_{n-2} \rightarrow 0$. So such a recursive procedure repeats until the final equilibrium state is achieved, that is, $\dot{x}_1 \rightarrow 0$ and $x_1 \rightarrow w_d^{(1)}$ make $\dot{x}_0 \rightarrow 0$, and the desired path r_d can be tracked by the manipulator system.

4.2 | Stability and convergence analysis of intermediate variables for high-order joint constraints

Review the second equation of the manifold system

$$\dot{x}_{n-1} = P_{\Omega_n}(-k_n(x_{n-1} - w_d^{(n-1)})). \quad (16)$$

Now we analyse its stability and convergence for the solution. Define a Lyapunov function $V = \|x_{n-1} - w_d^{(n-1)}\|^2/2$, and its time derivative is

$$\dot{V} = [x_{n-1} - w_d^{(n-1)}]^T \dot{x}_{n-1}. \quad (17)$$

According to the definition of the piecewise linear projection function $P_{\Omega_n}(\cdot)$, we have

$$\begin{aligned} & \|P_{\Omega_n}(-k_n(x_{n-1} - w_d^{(n-1)})) + k_n(x_{n-1} - w_d^{(n-1)})\|^2 \\ & \leq k_n^2 \|x_{n-1} - w_d^{(n-1)}\|^2 \end{aligned} \quad (18)$$

That is

$$\begin{aligned} & \left\| P_{\Omega_n}(-k_n(x_{n-1} - w_d^{(n-1)})) \right\|^2 + k_n^2 \|x_{n-1} - w_d^{(n-1)}\|^2 \\ & + 2k_n \left[P_{\Omega_n}(-k_n(x_{n-1} - w_d^{(n-1)})) \right]^T (x_{n-1} - w_d^{(n-1)}) \\ & \leq k_n^2 \|x_{n-1} - w_d^{(n-1)}\|^2 \end{aligned} \quad (19)$$

Thus, we have

$$\begin{aligned} & \left\| P_{\Omega_n}(-k_n(x_{n-1} - w_d^{(n-1)})) \right\|^2 \\ & + 2k_n \left[P_{\Omega_n}(-k_n(x_{n-1} - w_d^{(n-1)})) \right]^T (x_{n-1} - w_d^{(n-1)}) \leq 0 \end{aligned} \quad (20)$$

As the following inequality always holds

$$\left\| P_{\Omega_n}(-k_n(x_{n-1} - w_d^{(n-1)})) \right\|^2 \geq 0$$

For the residual term, we have

$$\left[P_{\Omega_n}(-k_n(x_{n-1} - w_d^{(n-1)})) \right]^T (x_{n-1} - w_d^{(n-1)}) \leq 0 \quad (21)$$

that is,

$$\dot{V} \leq 0 \quad (22)$$

which is a negative definite. According to Lyapunov theory, as the Lyapunov function V is a positive definite, the second equation of the manifold system is asymptotically stable.

According to LaSalle's invariant set principle [28] and letting $\dot{V} = 0$, it yields

$$\left[P_{\Omega_n}(-k_n(x_{n-1} - w_d^{(n-1)})) \right]^T (x_{n-1} - w_d^{(n-1)}) = 0 \quad (23)$$

Since the following inequality holds

$$\begin{aligned} & \left\| P_{\Omega_n}(-k_n(x_{n-1} - w_d^{(n-1)})) \right\|^2 \\ & + 2k_n \left[P_{\Omega_n}(-k_n(x_{n-1} - w_d^{(n-1)})) \right]^T (x_{n-1} - w_d^{(n-1)}) \leq 0 \end{aligned} \quad (24)$$

we can obtain

$$P_{\Omega_n}(-k_n(x_{n-1} - w_d^{(n-1)})) = 0 \quad (25)$$

Review the definition of the linear projection operator P_{Ω_n} , the equation above is equivalent to

$$\arg \min_{z \in \Omega_n} \|z + k_n(x_{n-1} - w_d^{(n-1)})\| = 0 \quad (26)$$

Therefore, we have

$$\left\| k_n(x_{n-1} - w_d^{(n-1)}) \right\|^2 \leq \|z + k_n(x_{n-1} - w_d^{(n-1)})\|^2, \forall z \in \Omega_n \quad (27)$$

That is

$$\|z\|^2 + 2k_n z^T (x_{n-1} - w_d^{(n-1)}) \geq 0, \forall z \in \Omega_n \quad (28)$$

For the inequality above, there exists a value $\rho > 0$ such that

$$\|z\| = \frac{\left\| -\rho(x_{n-1} - w_d^{(n-1)}) \right\|}{2 \left\| x_{n-1} - w_d^{(n-1)} \right\|} \leq \rho \quad (29)$$

which yields

$$\frac{\left\| \rho(x_{n-1} - w_d^{(n-1)}) \right\|^2}{4 \left\| x_{n-1} - w_d^{(n-1)} \right\|^2} \quad (30)$$

$$-k_n \left(\frac{\rho(x_{n-1} - w_d^{(n-1)})}{\|x_{n-1} - w_d^{(n-1)}\|^2} \right)^T (x_{n-1} - w_d^{(n-1)}) \geq 0$$

Thus, we can get the following inequality

$$\left\| x_{n-1} - w_d^{(n-1)} \right\| \leq \frac{\rho}{4k_n} \quad (31)$$

Recall that the value of ρ can be chosen to be as small as possible and parameter k_n is large enough, and thus the inequality holds for any $\rho > 0$. This implies that

$$\left\| x_{n-1} - w_d^{(n-1)} \right\| \rightarrow 0 \quad (32)$$

that is,

$$x_{n-1} \rightarrow w_d^{(n-1)} \quad (33)$$

In addition to the stability analysis, now we show that x_{n-1} can converge to $w_d^{(n-1)}$, that is, the equilibrium point can be got as $x_{n-1} = w_d^{(n-1)}$. In this case, we can substitute $x_{n-1} = w_d^{(n-1)}$ into

$$\dot{\bar{x}}_{n-2} = f(\bar{x}_{n-2}, x_{n-1}, 1/k_{n-1}) \quad (34)$$

and then we obtain

$$\begin{cases} \dot{\bar{x}}_{n-2} = f(\bar{x}_{n-2}, w_d^{(n-1)}, 1/k_{n-1}) \\ \dot{x}_{n-1} = P_{\Omega_n}(-k_n(x_{n-1} - w_d^{(n-1)})) \end{cases} \quad (35)$$

Recursively, we have the following equations for the manifold system in the similar form

$$\begin{cases} \dot{\bar{x}}_{n-3} = f(\bar{x}_{n-3}, x_{n-2}, 1/k_{n-1}) \\ \dot{x}_{n-2} = P_{\Omega_{n-1}}(-k_{n-1}(x_{n-2} - w_d^{(n-2)})) \end{cases} \quad (36)$$

and the convergence relationship

$$\|x_{n-2} - w_d^{(n-2)}\| \leq \frac{\rho}{4k_{n-1}} \quad (37)$$

which indicates $x_{n-2} \rightarrow w_d^{(n-2)}$ with a small enough value ρ and a large enough value k_{n-1} .

Repeat the similar derivation procedure as for $\dot{x}_{n-3}, \dot{x}_{n-4}, \dots, \dot{x}_2$, we can get

$$\begin{aligned} \|x_{n-3} - w_d^{(n-3)}\| &\leq \frac{\rho}{4k_{n-2}} \\ \|x_{n-3} - w_d^{(n-3)}\| &\leq \frac{\rho}{4k_{n-2}} \\ &\vdots \\ \|x_2 - w_d^{(2)}\| &\leq \frac{\rho}{4k_3} \end{aligned}$$

and the convergence solution $x_{n-3} \rightarrow w_d^{(n-3)}, x_{n-4} \rightarrow w_d^{(n-4)}, \dots, x_2 \rightarrow w_d^{(2)}$.

Repeat the same procedure until concerning \dot{x}_1 , we can finally get

$$\begin{cases} \dot{\bar{x}}_0 = f(\bar{x}_0, x_1, 1/k_2) \\ \dot{x}_1 = P_{\Omega_2}(-k_2(x_1 - w_d^{(1)})) \end{cases} \quad (38)$$

with the convergence relationship

$$\|x_1 - w_d^{(1)}\| \leq \frac{\rho}{4k_2} \quad (39)$$

showing that $x_1 \rightarrow w_d^{(1)}$.

Generally, based on the aforementioned derivations, we can summarise that for the manifold system equation

$$\begin{cases} \dot{\bar{x}}_{n-i} = f(\bar{x}_{n-i}, x_{n-i+1}, 1/k_{n-i+2}) \\ \dot{x}_{n-i+1} = P_{\Omega_{n-i+2}}(-k_{n-i+2}(x_{n-i+1} - w_d^{(n-i+1)})) \end{cases} \quad (40)$$

it can produce

$$\|x_{n-i+1} - w_d^{(n-i+1)}\| \leq \frac{\rho}{4k_{n-i+2}} \quad (41)$$

and we can conclude that $x_{n-i+1} = w_d^{(n-i+1)}$ where $i = \{1, 2, \dots, n\}$.

4.3 | Stability and convergence analysis for the end effector in tracking path

After showing the stability and convergence of the state-space dynamic systems with intermediate variables for high-order constraints, we still need to analyse the stability and convergence properties for the end effector of the manipulator system to track the desired path. According to the velocity kinematics equation and the proposed recursive recurrent neural network solver, we define the term $e = r - r_d$ as the tracking error of the end effector of the manipulator, and we have

$$\dot{r} = JP_{\Omega_1}(-k_1 J^T(r - r_d)) = JP_{\Omega_1}(-k_1 J^T e) \quad (42)$$

As $\dot{e} = \dot{r} - \dot{r}_d = \dot{r}$ provided that the \dot{r}_d is constant, thus the equation above further becomes

$$\dot{e} = JP_{\Omega_0}(-k_1 J^T e) \quad (43)$$

Define a Lyapunov function $V = \|e\|^2/2$, and its time derivative is

$$\dot{V} = \dot{e}^T e = [JP_{\Omega_1}(-k_1 J^T e)]^T e \quad (44)$$

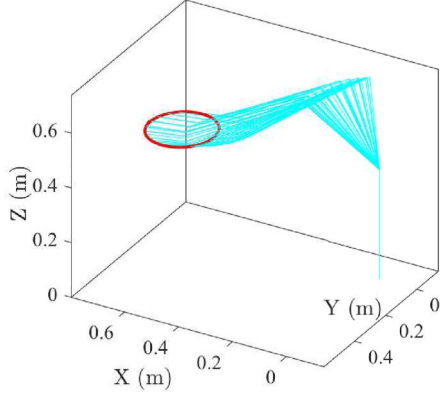
According to the definition of the projection operation, we have

$$\|P_{\Omega_1}(-k_1 J^T e) + k_1 J^T e\|^2 \leq \|k_1 J^T e\|^2 \quad (45)$$

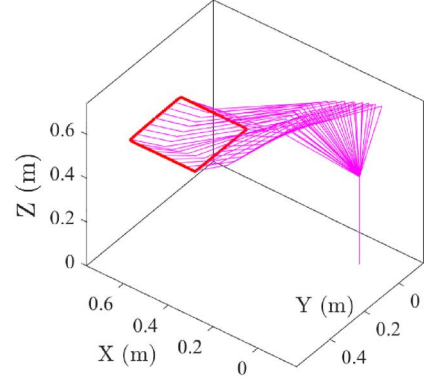
As the following inequality holds

$$\begin{aligned} &\|P_{\Omega_1}(-k_1 J^T e) + k_1 J^T e\|^2 \\ &= \|P_{\Omega_1}(-k_1 J^T e)\|^2 + \|k_1 J^T e\|^2 \\ &\quad + 2[P_{\Omega_1}(-k_1 J^T e)]^T (k_1 J^T e) \leq \|k_1 J^T e\|^2 \end{aligned} \quad (46)$$

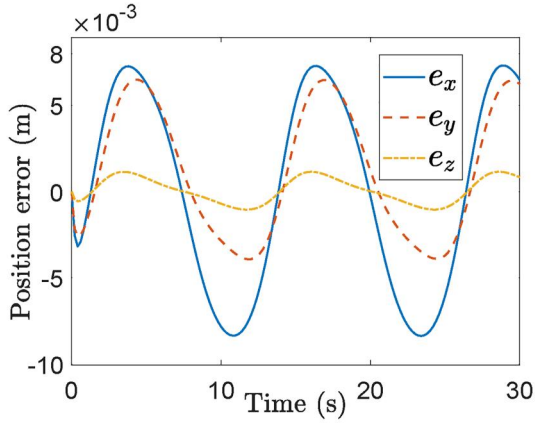
Thus, we have



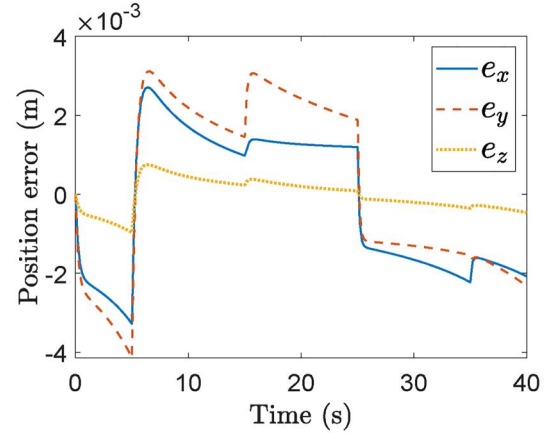
(a) End-effector trajectory with respect to the desired circle path position.



(b) End-effector trajectory with respect to the desired square path position.



(c) Position error for circle path



(d) Position error for square path

FIGURE 2 Tracking performance of circle and square paths by the proposed method.

$$\|P_{\Omega_1}(-k_1 J^T e)\|^2 \leq -2[P_{\Omega_1}(-k_1 J^T e)]^T (k_1 J^T e) \quad (47)$$

we can get

which leads to

$$[JP_{\Omega_1}(-k_1 J^T e)]^T e \leq 0 \quad (48)$$

Therefore, we have

$$\dot{V} \leq 0 \quad (49)$$

which indicates that the tracking system is stable.

Next, we again use LaSalle's invariant set principle [28] and force $\dot{V} = 0$, and then it yields

$$[JP_{\Omega_1}(-k_1 J^T e)]^T e = 0 \quad (50)$$

As

$$\|P_{\Omega_1}(-k_1 J^T e)\|^2 \leq -2[P_{\Omega_1}(-k_1 J^T e)]^T (k_1 J^T e) \quad (51)$$

$$P_{\Omega_1}(-k_1 J^T e) = 0 \quad (52)$$

The equation above can be equivalent to

$$\arg \min_{y \in \Omega_1} \|y + k_1 J^T e\| = 0 \quad (53)$$

Therefore, we have the following inequality

$$\|k_1 J^T e\|^2 \leq \|y + k_1 J^T e\|^2, \forall y \in \Omega \quad (54)$$

That is

$$\|y\|^2 + 2k_1 y^T J^T e \geq 0, \forall y \in \Omega \quad (55)$$

There exists a value $\rho > 0$ such that

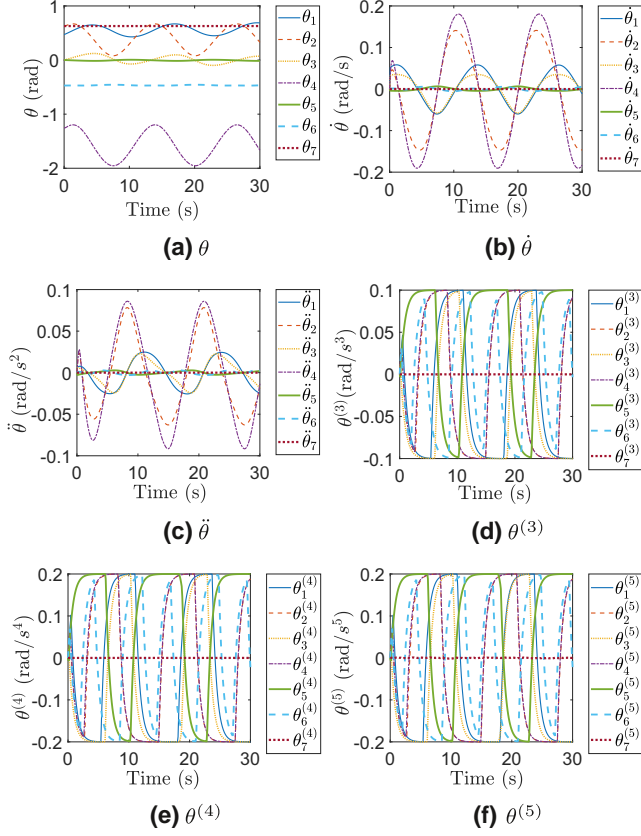


FIGURE 3 Joint variables of different orders in motion planning of the manipulator for circle path tracking by the proposed method.

$$\frac{\|-\rho J^T e\|}{2\|J^T e\|} \leq \rho \quad (56)$$

which yields

$$\frac{\|-\rho J^T e\|^2}{4\|J^T e\|^2} + 2k_1 \left(\frac{\rho J^T e}{2\|J^T e\|^2} \right)^T J^T e \geq 0 \quad (57)$$

Therefore, we have

$$\|J^T e\| \leq \frac{\rho}{4k_1} \quad (58)$$

Recall that the value of ρ can be chosen to be as small as possible and $k_1 > 1$ can be enlarged. This implies that

$$\|J^T e\| \rightarrow 0 \quad (59)$$

that is,

$$J^T e \rightarrow 0 \quad (60)$$

If the Jacobian matrix is full rank, which means the manipulator will not fall into the singularity situation during motion control process, then we have

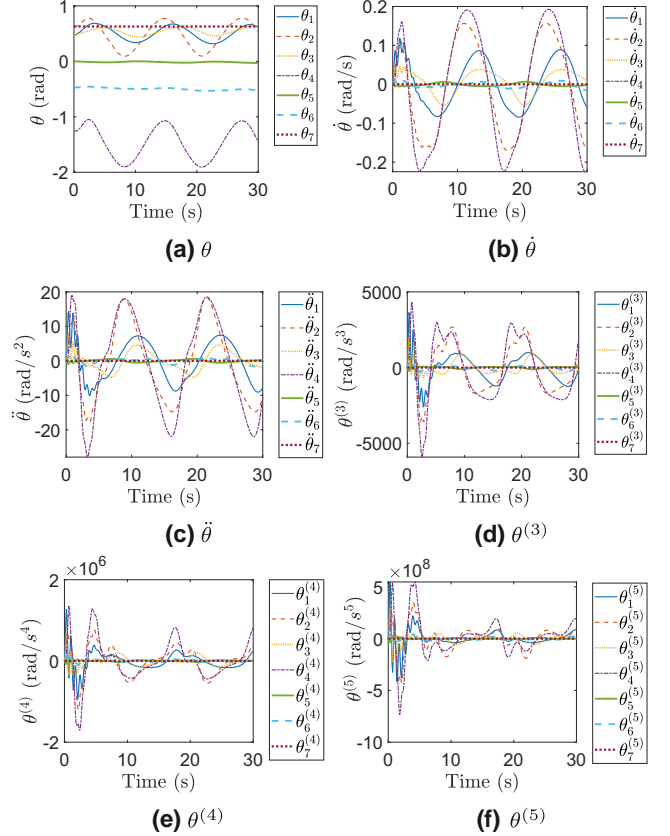


FIGURE 4 Joint variables of different orders in motion planning of the manipulator for circle path tracking by the pseudoinverse method.

$$e \rightarrow 0 \quad (61)$$

which indicates that the tracking error of the end effector of the manipulator can converge to zero.

5 | RESULTS

In this section, the proposed recursive recurrent neural network (10) based on the manifold system is utilised to control the manipulator with fifth-order joint variable constraints, that is, $\theta_1^- \leq \theta_1 \leq \theta_1^+$, $\theta_2^- \leq \dot{\theta}_2 \leq \theta_2^+$, $\theta_3^- \leq \theta_3^{(3)} \leq \theta_3^+$, $\theta_4^- \leq \theta_4^{(4)} \leq \theta_4^+$ and $\theta_5^- \leq \theta_5^{(5)} \leq \theta_5^+$. The desired targets for the manipulator to track are circle and square paths. In order to perform such kinematic control with these joint constraints, the resultant recursive recurrent neural network controller is

$$\begin{cases} \dot{\bar{x}}_3 = f(\bar{x}_3, x_4) \\ \dot{x}_4 = P_{\Omega_5}(-k_5(x_4 - w_d^{(4)})) \end{cases} \quad (62)$$

According to aforementioned analysis, controller (62) depicted by recurrent neural network recursively updates to produce (\bar{x}_2, \dot{x}_3) , (\bar{x}_1, \dot{x}_2) and (\bar{x}_0, \dot{x}_1) to perform the path tracking tasks and ensure the constraints of different orders are satisfied during control. The solution sets $\Omega_1, \Omega_2, \Omega_3, \Omega_4$ and

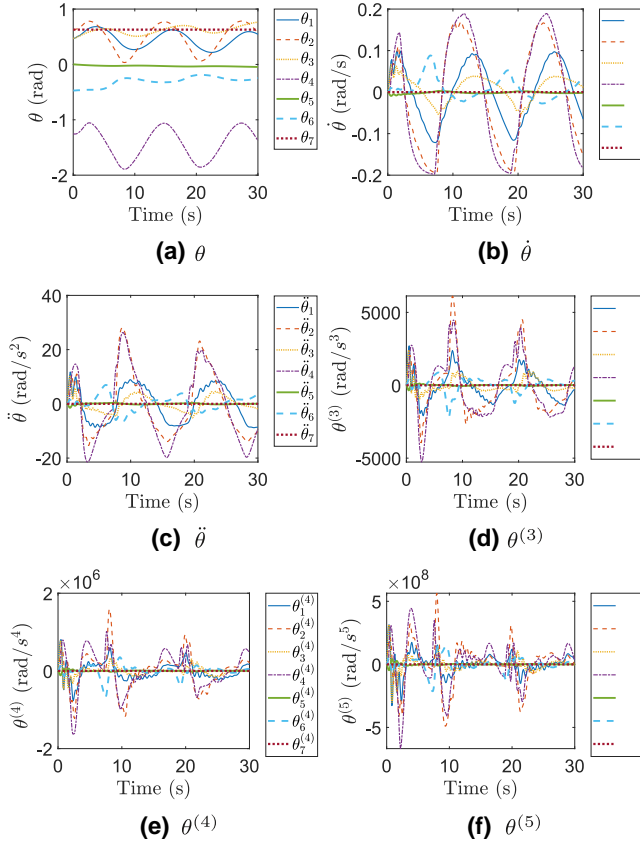


FIGURE 5 Joint variables of different orders in motion planning of the manipulator for circle path tracking by the conventional recurrent neural network method.

Ω_5 are also generated to correspond the joint variable constraints in different orders.

5.1 | Simulation setup and parameter configuration

In the simulation, we evaluate the performance for the Kuka manipulator by the proposed method. The forward kinematics modelling of the manipulator is established based on the D-H table in ref. [2].

For the circle path tracking task, the radius of the circle is 0.15 m, and the parameters for the proposed recursive recurrent neural network are configured as follows: $k_1 = 100$, $k_2 = 100k_1$, $k_3 = 200k_1$, $k_4 = 200k_1$, $k_5 = 200k_1$, $\epsilon_1 = 0.0001$, $\epsilon_2 = 0.9$, $\epsilon_3 = 0.8$, $\epsilon_4 = 0.7$, $\epsilon_5 = 0.7$. The joint constraints of different orders are set as $\theta \in [\theta_0^- \theta_0^+] = [-2 \ 2]$ rad, $\dot{\theta} \in [-0.2 \ 0.2]$ rad/s, $\ddot{\theta} \in [-0.1 \ 0.1]$ rad/s², $\theta^{(3)} \in [-0.1 \ 0.1]$ rad/s³, $\theta^{(4)} \in [-0.2 \ 0.2]$ rad/s⁴, $\theta^{(5)} \in [-0.2 \ 0.2]$ rad/s⁵. The motion duration is limited to 30 s. From the boundaries of the joint variable in different orders for the solution sets Ω_1 , Ω_2 , Ω_3 , Ω_4 and Ω_5 , we have $\theta_1^- = \max(-0.2, -(\theta - \theta_0^-))$ (rad/s) and $\theta_1^+ = \min(0.2, -(\theta - \theta_0^+))$ (rad/s), $\theta_2^- = -0.1$ and $\theta_2^+ = 0.1$,

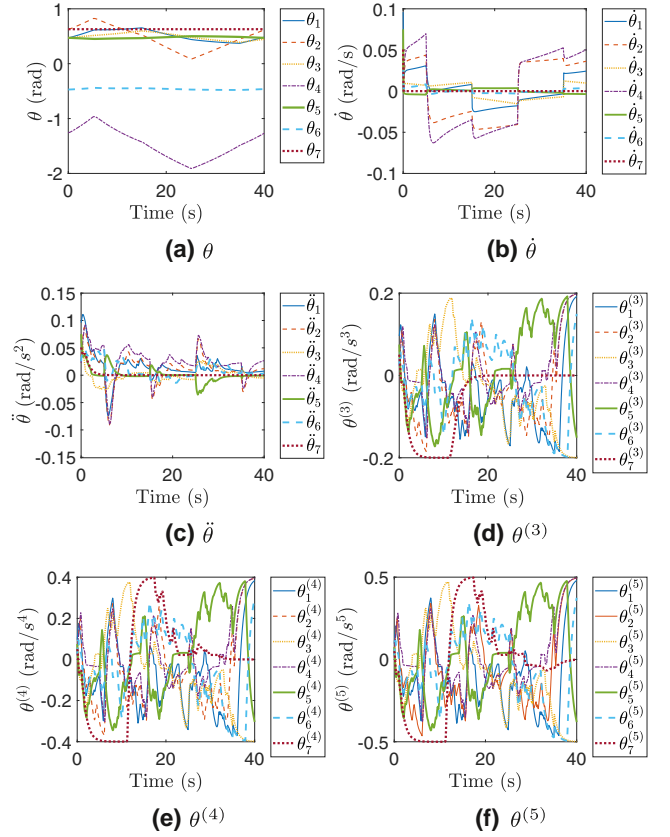


FIGURE 6 Joint variables of different orders in motion planning of the manipulator for square path tracking by the proposed method.

$\theta_3^- = -0.1$ and $\theta_3^+ = 0.1$, $\theta_4^- = -0.2$ and $\theta_4^+ = 0.2$, $\theta_5^- = -0.2$ and $\theta_5^+ = 0.2$.

For the square path tracking task, the length of the square is 0.20 m, and the parameters for the proposed recursive recurrent neural network are configured as follows: $k_1 = 100$, $k_2 = 1000k_1$, $k_3 = 1500k_1$, $k_4 = 2000k_1$, $k_5 = 2500k_1$, $k_6 = 3000k_1$, $\epsilon_1 = 0.0001$, $\epsilon_2 = 1$, $\epsilon_3 = 1$, $\epsilon_4 = 1$, $\epsilon_5 = 1$. The joint constraints of different orders are set as $\theta \in [\theta_0^- \theta_0^+] = [-2 \ 2]$ rad, $\dot{\theta} \in [-0.1 \ 0.1]$ rad/s, $\ddot{\theta} \in [-0.15 \ 0.15]$ rad/s², $\theta^{(3)} \in [-0.2 \ 0.2]$ rad/s³, $\theta^{(4)} \in [-0.4 \ 0.4]$ rad/s⁴, $\theta^{(5)} \in [-0.5 \ 0.5]$ rad/s⁵. The motion duration is limited to 40 s. From the boundaries of the joint variable in different orders for the solution sets Ω_1 , Ω_2 , Ω_3 , Ω_4 and Ω_5 , we have $\theta_1^- = \max(-0.1, -(\theta - \theta_0^-))$ (rad/s) and $\theta_1^+ = \min(0.1, -(\theta - \theta_0^+))$ (rad/s), $\theta_2^- = -0.15$ and $\theta_2^+ = 0.15$, $\theta_3^- = -0.2$ and $\theta_3^+ = 0.2$, $\theta_4^- = -0.4$ and $\theta_4^+ = 0.4$, $\theta_5^- = -0.5$ and $\theta_5^+ = 0.5$.

5.2 | Tracking performance

Figure 2 shows the tracking performance of the manipulator for circle and square paths by the recursive recurrent neural network solver (62) with the aforementioned parameters

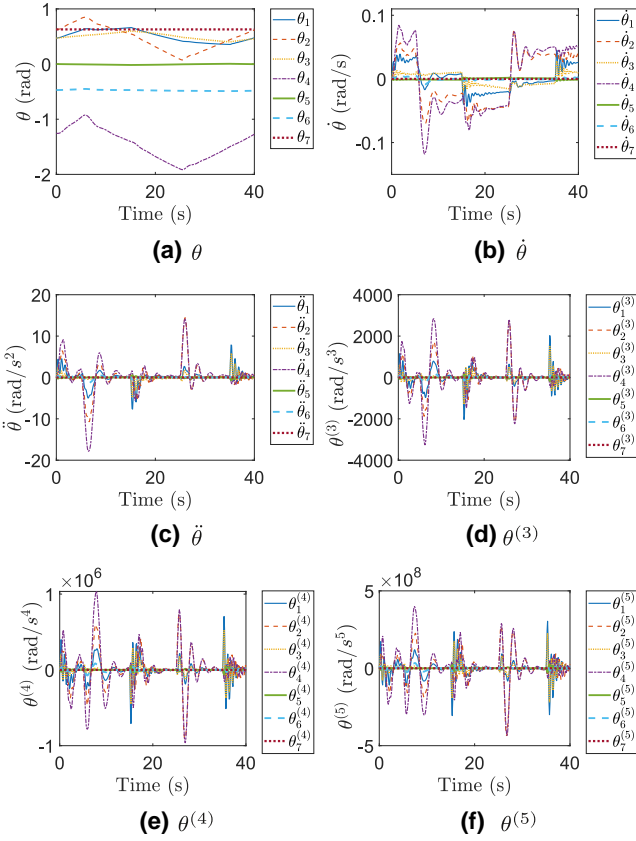


FIGURE 7 Joint variables of different orders in motion planning of the manipulator for square path tracking by the pseudoinverse method.

configured. From Figure 2a and 2b, we could observe that, the manipulator can fulfil the path tracking tasks well with its end effector promisingly following the positions of the paths. As seen from Figure 2c and 2d, the position errors $e = [e_x, e_y, e_z]$ between the end effector and the desired paths can be rather small, with the range being from less than 10×10^{-3} m for the circle path and less than 8×10^{-3} m for the square path. These results demonstrate the efficiency of the proposed recurrent neural network solver based on the manifold system for kinematic control of the manipulator in the aspect of tracking accuracy.

5.3 | Joint constraints

In addition to fulfilling path tracking control, the manipulator system has to simultaneously follow the joint constraints in different orders. Figures 3 and 6 show the joint variables of different orders for circle and square path tracking tasks by the proposed method, that is, joint angles, joint angular velocity, joint angular acceleration, joint variable of third order, joint variable of fourth order, and joint variable of fifth order.

For the circle path tracking task, Figure 3a shows the revolved joint angle, and Figure 3b-f show other joint variables of different orders which have to be within the aforementioned

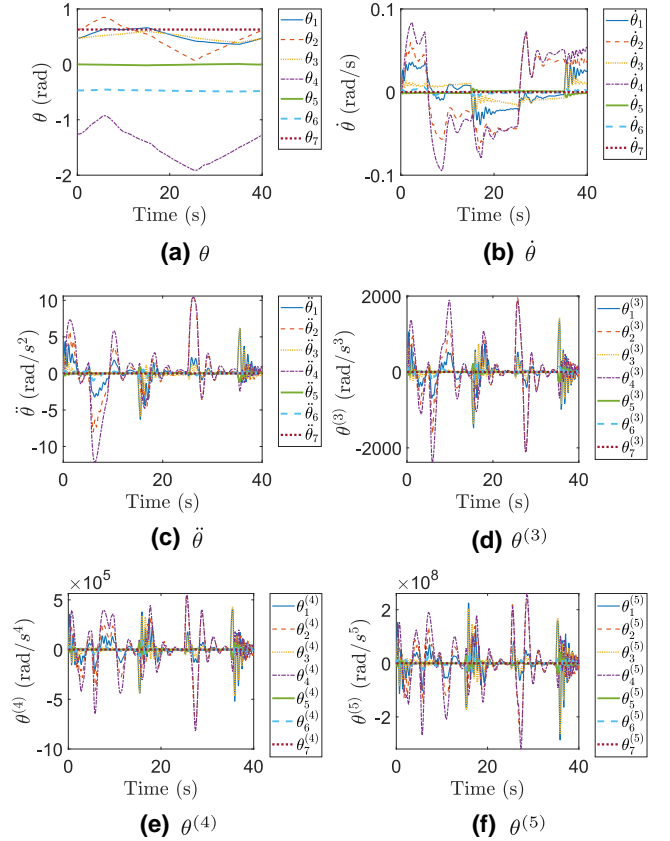


FIGURE 8 Joint variables of different orders in motion planning of the manipulator for square path tracking by the conventional recurrent neural network method.

constraints. Clearly seen from Figure 3, the joint angular velocity $\dot{\theta}$ successfully falls into $[-0.2, 0.2]$ rad/s, the joint angular acceleration $\ddot{\theta}$ falls into $[-0.1, 0.1]$ rad/s², the joint variable of the third order $\theta^{(3)}$ falls into $[-0.1, 0.1]$ rad/s³, the joint variable of the fourth order $\theta^{(4)}$ falls into $[-0.2, 0.2]$ rad/s⁴, and the joint variable of the fifth order $\theta^{(5)}$ falls into $[-0.2, 0.2]$ rad/s⁵. Together with the tracking results of the end effector in Section V-B for circle path tracking, the proposed method is demonstrated as an efficient way to deal with joint constraints in high order when fulfilling kinematic control. For comparisons, Figures 4 and 5 show the joint variables of different orders based on the pseudoinverse method and the conventional recurrent neural network method in the circle path tracking tasks, and it can be observed that, when encountering the higher order (≥ 2) joint constraints (shown in Figures 4c-f and 5c-f), these two methods cannot work well as the higher order joint variables obviously exceed the corresponding joint limits.

For the square path tracking task, Figure 6a shows the revolved joint angle, and Figure 6b-f show other joint variables of different orders which have to be within the aforementioned constraints. Clearly seen from Figure 6, the joint angular velocity $\dot{\theta}$ successfully falls into $[-0.1, 0.1]$ rad/s, the joint angular acceleration $\ddot{\theta}$ falls into $[-0.15, 0.15]$ rad/s², the joint variable of the third order $\theta^{(3)}$ falls into $[-0.2, 0.2]$ rad/s³, the

joint variable of the fourth order $\theta^{(4)}$ falls into $[-0.4, 0.4]$ rad/s⁴, and the joint variable of the fifth order $\theta^{(5)}$ falls into $[-0.5, 0.5]$ rad/s⁵. Together with the tracking results of the end effector in Section V-B for square path tracking, the proposed method is demonstrated as an efficient way to deal with the joint constraints in high order when fulfilling kinematic control. Furthermore, Figures 7 and 8 show the joint variables of different orders based on the pseudoinverse method and the conventional recurrent neural network method in the square path tracking tasks, respectively and it can be observed that, when encountering the higher order (≥ 2) joint constraints (shown in Figure 7c-f and Figure 8c-f), these two methods cannot work well as the higher order joint variables evidently exceed the corresponding joint limits.

To summarise, from all these results including comparisons, we can conclude that the proposed recursive neural network method is efficient and superior for the manipulator control with high-order joint constraints.

6 | CONCLUSION

Due to physical limitations and actuation saturation of manipulator joints, the involvement of joint constraints for kinematic control of manipulators is essential and critical. In this work, for the first time, a novel recursive recurrent network model is proposed to solve the kinematic control issue for manipulators with different levels of physical constraints, and the proposed recursive recurrent neural network model can be formulated as a new manifold system to ensure a control solution within all of the joint constraints in different orders. Theoretical analysis shows the stability and convergence of the proposed recursive recurrent neural network and its convergence of solution. Simulation results further demonstrate the effectiveness and superiority of the proposed method in end-effector path tracking control under different levels of joint constraints based on the Kuka manipulator system.

ACKNOWLEDGMENT

This work is partially supported by the IMPACT Fund of Swansea University.

CONFLICT OF INTEREST

The authors declare that there is no conflict of interest.

DATA AVAILABILITY STATEMENT

The running data/code that support the findings of this study are available from the corresponding author upon reasonable request.

ORCID

Zhan Li  <https://orcid.org/0000-0002-3928-1642>

REFERENCES

- Kanoun, O., Lamiraux, F., Wieber, P.: Kinematic control of redundant manipulators: generalizing the task-priority framework to inequality task. *IEEE Trans. Robot.* 27(4), 785–792 (2011)
- Li, Z., et al.: A fault-tolerant method for motion planning of industrial redundant manipulator. *IEEE Trans. Ind. Inf.* 16(1–1), 7469–7478 (2020). <https://doi.org/10.1109/tii.2019.2957186>
- Shibata, T., Murakami, T.: Null space motion control by pid control considering passivity in redundant manipulator. *IEEE Trans. Ind. Inf.* 4(4), 261–270 (2008). <https://doi.org/10.1109/tii.2008.2007847>
- Klein, C.A., Huang, C.: Review of pseudoinverse control for use with kinematically redundant manipulators, *IEEE Transactions on Systems, Man, and Cybernetics.* SMC- 13(2), 245–250 (1983). <https://doi.org/10.1109/tsmc.1983.6313123>
- Maciejewski, A.A.: Kinetic limitations on the use of redundancy in robotic manipulators. *IEEE Trans. Robot. Autom.* 7(2), 205–210 (1991). <https://doi.org/10.1109/70.75903>
- Ma, S.: A new formulation technique for local torque optimization of redundant manipulators. *IEEE Trans. Ind. Electron.* 43(4), 462–468 (1996). <https://doi.org/10.1109/41.510637>
- Chen, D., Zhang, Y.: A hybrid multi-objective scheme applied to redundant robot manipulators. *IEEE Trans. Autom. Sci. Eng.* 14(3), 1337–1350 (2017). <https://doi.org/10.1109/tase.2015.2474157>
- Flacco, F., De Luca, A., Khatib, O.: Control of redundant robots under hard joint constraints: saturation in the null space. *IEEE Trans. Robot.* 31(3), 637–654 (2015). <https://doi.org/10.1109/tro.2015.2418582>
- Li, S., Zhang, Y., Jin, L.: Kinematic control of redundant manipulators using neural networks. *IEEE Transact. Neural Networks Learn. Syst.* 28(10), 2243–2254 (2017). <https://doi.org/10.1109/tnnls.2016.2574363>
- He, W., et al.: Neural-learning-based control for a constrained robotic manipulator with flexible joints. *IEEE Transact. Neural Networks Learn. Syst.* 29(12), 5993–6003 (2018). <https://doi.org/10.1109/tnnls.2018.2803167>
- Wai, R., Yang, Z.: Adaptive fuzzy neural network control design via a t-s fuzzy model for a robot manipulator including actuator dynamics. *IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics)* 38(5), 1326–1346 (2008). <https://doi.org/10.1109/tsmcb.2008.925749>
- Li, S., et al.: Distributed recurrent neural networks for cooperative control of manipulators: a game-theoretic perspective. *IEEE Transact. Neural Networks Learn. Syst.* 28(2), 415–426 (2017). <https://doi.org/10.1109/tnnls.2016.2516565>
- Li, S., Shao, Z., Guan, Y.: A dynamic neural network approach for efficient control of manipulators. *IEEE Trans. Syst., Man Cybernet. Syst.* 49(5), 932–941 (2019). <https://doi.org/10.1109/tsmc.2017.2690460>
- Li, S., Zhou, M., Luo, X.: Modified primal-dual neural networks for motion control of redundant manipulators with dynamic rejection of harmonic noises. *IEEE Transact. Neural Networks Learn. Syst.* 29(10), 4791–4801 (2018). <https://doi.org/10.1109/tnnls.2017.2770172>
- He, W., et al.: Disturbance observer-based neural network control of cooperative multiple manipulators with input saturation. *IEEE Transact. Neural Networks Learn. Syst.* 31(5), 1735–1746 (2020). <https://doi.org/10.1109/tnnls.2019.2923241>
- Zhang, Y., et al.: Adaptive projection neural network for kinematic control of redundant manipulators with unknown physical parameters. *IEEE Trans. Ind. Electron.* 65(6), 4909–4920 (2018). <https://doi.org/10.1109/tie.2017.2774720>
- Zhang, Y., et al.: Recurrent neural network for kinematic control of redundant manipulators with periodic input disturbance and physical constraints. *IEEE Trans. Cybern.* 49(12), 4194–4205 (2019). <https://doi.org/10.1109/tcyb.2018.2859751>
- Chen, D., Zhang, Y.: Robust zeroing neural-dynamics and its time-varying disturbances suppression model applied to mobile robot manipulators. *IEEE Transact. Neural Networks Learn. Syst.* 29(9), 4385–4397 (2018). <https://doi.org/10.1109/tnnls.2017.2764529>
- Zhang, Y., Wang, J., Xia, Y.: A dual neural network for redundancy resolution of kinematically redundant manipulators subject to joint limits and joint velocity limits. *IEEE Trans. Neural Network.* 14(3), 658–667 (2003). <https://doi.org/10.1109/tnn.2003.810607>
- Guarino Lo Bianco, C., Ghilardelli, F.: Real-time planner in the operational space for the automatic handling of kinematic constraints. *IEEE Trans. Autom. Sci. Eng.* 11(3), 730–739 (2014). <https://doi.org/10.1109/tase.2014.2310813>

21. Xiao, L., Zhang, Y.: Acceleration-level repetitive motion planning and its experimental verification on a six-link planar robot manipulator. *IEEE Trans. Control Syst. Technol.* 21(3), 906–914 (2013). <https://doi.org/10.1109/tcst.2012.2190142>
22. Guo, D., Zhang, Y.: Acceleration-level inequality-based man scheme for obstacle avoidance of redundant robot manipulators. *IEEE Trans. Ind. Electron.* 61(12), 6903–6914 (2014). <https://doi.org/10.1109/tie.2014.2331036>
23. Zhang, Y., et al.: A passivity-based approach for kinematic control of manipulators with constraints. *IEEE Trans. Ind. Inf.* 16(5), 3029–3038 (2020). <https://doi.org/10.1109/tii.2019.2908442>
24. Zhang, Y., Ge, S.S., Lee, T.H.: A unified quadratic-programming-based dynamical system approach to joint torque optimization of physically constrained redundant manipulators. *IEEE Trans. Syst. Man Cybernet. Part B (Cybernetics)* 34(5), 2126–2132 (2004). <https://doi.org/10.1109/tsmcb.2004.830347>
25. Zhang, Y., Li, S.: A neural controller for image-based visual serving of manipulators with physical constraints. *IEEE Transact. Neural Networks Learn. Syst.* 29(11), 5419–5429 (2018). <https://doi.org/10.1109/tnnls.2018.2802650>
26. Zhang, Y., et al.: Velocity-level control with compliance to acceleration-level constraints: a novel scheme for manipulator redundancy resolution. *IEEE Trans. Ind. Inf.* 14(3), 921–930 (2018). <https://doi.org/10.1109/tii.2017.2737363>
27. Boyd, S., Vandenberghe, L.: *Convex Optimization*. Cambridge University Press (2004)
28. Khalil, H.K.: *Nonlinear Systems*. Prentice Hall (1996)

How to cite this article: Li, Z., Li, S.: Recursive recurrent neural network: a novel model for manipulator control with different levels of physical constraints. *CAAI Trans. Intell. Technol.* 1–13 (2022). <https://doi.org/10.1049/cit2.12125>