A Novel Recurrent Neural Network for Manipulator Control With Improved Noise Tolerance

Shuai Li, Huanqing Wang, and Muhammad Usman Rafique

Abstract— **In this paper, we propose a novel recurrent neural network to resolve the redundancy of manipulators for efficient kinematic control in the presence of noises in a polynomial type. Leveraging the high-order derivative properties of polynomial noises, a deliberately devised neural network is proposed to eliminate the impact of noises and recover the accurate tracking of desired trajectories in workspace. Rigorous analysis shows that the proposed neural law stabilizes the system dynamics and the position tracking error converges to zero in the presence of noises. Extensive simulations verify the theoretical results. Numerical comparisons show that existing dual neural solutions lose stability when exposed to large constant noises or time-varying noises. In contrast, the proposed approach works well and has a low tracking error comparable to noise-free situations.**

Index Terms— **Kinematic control, noise, recurrent neural network, redundant manipulator.**

I. Introduction

NOWADAYS, robotic manipulators are widely employed in industry for labor-intensive and high-accuracy operations. Current shortage of skilled labor and aging population pose urgent demands for robotic manipulation, e.g., payload transport, welding, painting, and assembly. Among the broad categories of manipulators, redundant manipulators, which have more control degrees-of-freedom (DOFs) than desired, are advantageous for dextrous manipulations due to its redundancy in DOFs. In comparison with traditional manipulators without extra DOFs, redundant manipulators usually have more than one control solutions to a specific task, and thus allow designers to exploit this feature to fulfill additional requirements, such as obstacle avoidance and control optimization, and thus have received intensive research in past years.

The introduction of extra DOFs in redundant manipulators increases the capability and flexibility of robotic manipulation, but also sets challenges to the control design for efficient redundancy resolution in real time. Analytically, the joint velocity of a nonredundant manipulator can be expressed in terms of the inverse of its Jacobian matrix. This result extends to redundant manipulators by replacing the Jacobian matrix inverse with its pseudoinverse, as the Jacobian matrix in this situation is nonsquare [1]. From an optimization perspective, this solution is equivalent to the one that minimizes joint speeds. This type of methods paves the foundation of manipulator kinematic redundancy but suffers from several drawbacks, e.g., intensive computation due to the continuous computation for pseudoinverse, outbound of computed joint velocities due to the lack of velocity range constraints, and local instability in some situations due to the singularity of the Jacobian matrix [2]. Later works employ various strategies to deal with the drawbacks of pseudoinverse based-solutions, including the usage of fast matrix operation algorithms for numerical approximation [3] and iterative methods to approach the solution [4]. However, time efficiency still remains a challenging problem for redundancy resolution, since recomputation has to be conducted for every time step.

The parallel processing capability of neural networks inspires researchers for the exploration of using neural networks, including feedforward neural networks [5]–[8] and recurrent neural networks [9]–[14], to control redundant manipulators. Among them, recurrent neural network-based solutions receive great success by exploiting historical information for control update to save computation. In [9], the redundancy resolution problem is modeled as convex optimization under an equation constraint, and solved using a Lagrange neural network. However, this model can only deal with equation constraints. For inequality constraints, e.g., the boundedness of joint velocities cannot be directly considered by Lagrange neural networks. Dual neural networks are proposed to address this problem. Zhang *et al.* [11] model redundancy resolution problem as a quadratic program, with kinematic energy as the objective function, the workspace velocity requirement as an equation constraint, and the joint velocity constraint as an inequality. A set of nonlinear equations are obtained by considering the optimization problem in its dual space. The derived dual neural network converges to its equilibrium point, which is identical to the solution of a nonlinear equation set. In [15], this approach is extended to optimize the joint torque when resolving the redundancy. Xia and Wang [12] propose a single layered dual neural network for the control of kinematically redundant manipulators for the reduction of network spatial complexity and the increase of computational efficiency. Novel dual neural networks are constructed in [16] based on different objective functions. As shown in [17], there exists an equivalence for redundancy resolutions between the velocity level and the acceleration level. By formulating the problem as

Manuscript received February 28, 2016; revised July 12, 2016, November 16, 2016, and January 24, 2017; accepted February 20, 2017. Date of publication April 12, 2017; date of current version April 16, 2018. This work was supported in part by the Hong Kong Research Grants Council Early Career Scheme under Grant 25214015, in part by the Departmental General Research Fund of The Hong Kong Polytechnic University under Grant G.UA7L, and in part by the National Natural Science Foundation of China under Grant 61401385. *(Corresponding Author: Shuai Li.)*

S. Li and M. U. Rafique are with the Department of Computing, The Hong Kong Polytechnic University, Hong Kong (email: lishuai8@gmail.com).

H. Wang is with the Department of Systems and Computer Engineering, Carleton University, Ottawa, ON K1S 5B6, Canada.

Color versions of one or more of the figures in this paper are available online at http://ieeexplore.ieee.org.

Digital Object Identifier 10.1109/TNNLS.2017.2672989

a constrained quadratic programming at the acceleration level, various dynamic neural solutions can be obtained to reach similar control performances to velocity level resolution results [18]–[21]. Not confined to the redundancy resolution of a single serial manipulator, recurrent neural network approach is applied to kinematic redundancy resolution of parallel Stewart platforms in [22], the coordination of multi-manipulators [23], [24], and motion control of mobile manipulators [25], [26]. Actually, all recurrent neural networks for solving general constrained optimization problems in principle can be applied to address redundancy resolution of manipulators [27]–[33].

Although great success has been achieved and various models have been proposed for the redundancy resolution of manipulators using recurrent neural networks, most existing works mainly focus on the nominal design without considering the appearance of noises explicitly. Actually, in the implementation of neural controllers for manipulator control, it is inevitable for the presence of noises in the forms of truncation error, rounding error, model uncertainty, and external disturbance. The robustness against additive noises is an important factor in designing reliable neural controllers. However, it remains unexplored for the design of dual neural networks inherently tolerant to noises. This paper makes progress by proposing a novel recurrent neural network design that is immune to any time-varying noises in the form of polynomials. Based on the feature that polynomial noises become zero after passing through time derivation operations for enough number of times, a neural dynamics is constructed to learn the noise and counter its impact. Accordingly, we are able to build an improved dual neural network with guaranteed convergence in noisy environments. The contributions of this paper are summarized as follows.

1) Most existing works on dual neural network design for manipulator control investigate the problem from an optimization perspective. This paper extends the horizon of neural dynamics design from purely optimization-based solution to optimization and robustness codesign.
2) To the best of our knowledge, this is the first recurrent neural network that can completely eliminate the impact of polynomial noises for manipulator control.
3) It is proved in this paper that the presented neural controllers are guaranteed to be globally convergent.
4) As shown in simulations, the performance of existing solutions degrades significantly when exposed to noisy environments and the system demonstrates oscillations when big additive noises present, while in the same environment, the proposed scheme converges well.

The remainder of this paper is organized as follows. In Section II, the redundancy resolution problem is formulated as a control one. The design of dual neural networks for redundancy resolution of manipulators in nominal situation without noises is presented in Section III. Based on this result, Section IV extends the nominal neural network to a general one that is able to deal with any polynomial noises. The convergence of this dynamic neural network in the presence of noises is also rigorously proved in this section. Section V provides numerical verifications of the proposed neural networks for real-time redundancy resolution of a PUMA 560 manipulator. It also compares the proposed solution with existing dual neural networks to show the superiority of the proposed one in robustness. Section VI concludes this paper.

II. Problem Formulation

In this section, we present the kinematic model of manipulators and formulate the redundancy resolution problem as a control one.

For a k-DOF redundant manipulator with the joint angle $\theta = [\theta_1, \theta_2, \ldots, \theta_k] \in \mathbb{R}^k$, its Cartesian coordinate $r \in \mathbb{R}^m$ in the workspace can be described as a nonlinear mapping

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

For a redundant manipulator, the joint space dimension m is greater than the workspace dimension k, i.e., $m > k$. Equation (1) becomes as follows after computing the time derivative on both sides:

$$v = \frac{\partial f}{\partial \theta} w = J w \tag{2}$$

where $J = (\partial f / \partial \theta) \in \mathbb{R}^{m \times k}$ is called the Jacobian matrix of f, $v = \dot{r}$ is the end-effector velocity in the workspace, and $w = \dot{\theta}$ is the manipulator joint angular velocity in the joint space. Equation (2) describes the kinematics of the manipulator in the velocity level. In comparison with the position level description (1), the velocity level descriptions are easier to deal with, since $\dot{r}$ is affine to $\dot{\theta}$ in (2). In applications, the joint angle usually is driven by acceleration or equivalently by force generated by motors, described as

$$\dot{w} = u + n \tag{3}$$

where u is the acceleration as the control input, n is an additive noise in the control channel, e.g., the disturbance from electrical and mechanical vibrations. Due to physical constraints, the angular velocity of a manipulator cannot exceed certain limits. Let Ω denotes the feasible angular velocity set, then the physical constraint can be expressed as

$$w = \dot{\theta} \in \Omega. \tag{4}$$

With (4) and (2), we are ready to define the kinematic redundancy resolution problem of a manipulator as follows.

Problem 1 (Kinematic Redundancy Resolution): For a redundant manipulator with kinematics (2) and a given desired workspace velocity v_d, find a real-time control law $u = h(w, v_d)$ such that the workspace velocity error $v - v_d$ converges to zero, and the joint angular velocity converges to the convex constraint set Ω as in (4).

III. Dual Neural Networks for the Nominal System

In this section, we present the design of a dual neural network for the control of a nominal manipulator without the presence of noises. This will pave the foundation for our further consideration with noises taken into account.

A. Neural Network Design

Due to the redundancy of the manipulator, there exist more than one control designs that can solve the redundancy resolution. To exploit the redundancy in an optimal fashion, we first define the problem as an optimization one and then convert it into a control law to solve the problem in a recurrent way.

To exploit the extra design freedom, we find a control action to minimize the following quadratic cost function by following conventions for dual neural network design:

$$\min_{w} \frac{1}{2} w^T W w + b^T w \tag{5}$$

where $W \in \mathbb{R}^{k\times k}$ is a symmetric positive semidefinite matrix $W = W^T \succeq 0$, $b \in \mathbb{R}^k$ is a vector. This cost function can be utilized to characterize kinematic energy expenses for angular velocity w. For example, choosing $W = I$ as an identity matrix and $b = 0$, this cost function is the kinematic energy. Generally, this cost function describes biased and weighted kinematic energy. The optimal solution to (5) also needs to meet the kinematic model (2) of the manipulator such that the workspace velocity v equals the desired velocity v_d ultimately, and also the feasibility constraint (4). Overall, we formulate it as the following constrained optimization:

$$\min_{w} \frac{1}{2} w^T W w + b^T w \tag{6a}$$

$$v_d = Jw \tag{6b}$$

$$w \in \Omega. \tag{6c}$$

Conventionally, a dual neural network is designed based on a Lagrange function as $L = (1/2)w^T W w + b^T w + \lambda^T (Jw - v_d)$. In this paper, we augment such a Lagrange function with the equality constraint, which benefits the convergence of the resulting neural dynamics, and define it as follows:

$$L(w \in \Omega, \lambda) = \frac{1}{2} w^T W w + \frac{1}{2} (Jw - v_d)^T W' (Jw - v_d) + b^T w + \lambda^T (Jw - v_d) \tag{7}$$

where W' is a symmetric positive semidefinite weighting matrix $W' = W'^T \succeq 0$. The optimal solution to (6) is identical to the saddle point of the following problem:

$$\min_{w\in\Omega} \max_{\lambda} L(w, \lambda). \tag{8}$$

According to Karush–Kuhn–Tucker condition [34], the solution of (8) satisfies

$$0 \in \frac{\partial L}{\partial w} + N_\Omega(w)$$

$$0 = \frac{\partial L}{\partial \lambda} \tag{9}$$

where $N_\Omega(w)$ denotes the normal cone of set Ω at w. According to the definition of normal cone, we have

$$s \in N_\Omega(w) \leftrightarrow s^T (x - w) \le 0 \quad \forall x \in \Omega \tag{10}$$

with which, the first expression in (9) can be written as

$$-\left(\frac{\partial L}{\partial w}\right)^T (x - w) \le 0 \quad \forall x \in \Omega. \tag{11}$$

With (11), we further conclude

$$\left\| w - \frac{\partial L}{\partial w} - x \right\|^2 - \left\| \frac{\partial L}{\partial w} \right\|^2 = \|w - x\|^2 + 2\left(\frac{\partial L}{\partial w}\right)^T (x - w) \ge 0 \quad \forall x \in \Omega. \tag{12}$$

This implies

$$w = P_\Omega\left(w - \frac{\partial L}{\partial w}\right) \tag{13}$$

where $P_\Omega(\cdot)$ is the projection operator defined as

$$P_\Omega(y) = \text{argmin}_{z\in\Omega} \|z - y\|^2 \tag{14}$$

or equivalently, $P_\Omega(y) = \alpha$ is the one satisfying

$$\alpha \in \Omega, \text{ and } \|z - y\|^2 - \|\alpha - y\|^2 \ge 0 \quad \forall z \in \Omega. \tag{15}$$

The derivation of (13) from (12) directly follows by setting new variables $z = x$, $y = w - (\partial L/\partial w)$, and $\alpha = w$ in (12). Overall, (9) can be finally expressed as

$$w = P_\Omega\left(w - \frac{\partial L}{\partial w}\right) = P_\Omega(w - Ww - J^T W'(Jw - v_d) - b - J^T \lambda)$$

$$0 = \frac{\partial L}{\partial \lambda} = Jw - v_d. \tag{16}$$

This is a nonlinear equation set and usually is impossible to solve analytically. To solve it numerically, we propose the following recurrent neural network in the form of:

$$\epsilon \dot{w} = -w + P_\Omega(w - Ww - J^T W'(Jw - v_d) - b - J^T \lambda)$$

$$\epsilon \dot{\lambda} = Jw - v_d. \tag{17}$$

This amounts to the following control law:

$$\epsilon u = -w + P_\Omega(w - Ww - J^T W'(Jw - v_d) - b - J^T \lambda)$$

$$\epsilon \dot{\lambda} = Jw - v_d \tag{18}$$

in the absence of noises, i.e., $n = 0$.

B. Convergence Analysis

About the presented neural network model, we have the following theoretical conclusion.

Theorem 1: Suppose there is no noise in the control, i.e., $n = 0$ in (3). Neural control law (18) solves the kinematic resolution problem of a redundant manipulator modeled by (2) and (3), and stabilizes the system to the optimal solution of (6).

Proof: For the convenience of proof, we define a new variable

$$\eta = \begin{bmatrix} w \\ \lambda \end{bmatrix} \tag{19}$$

and a function

$$F(\eta) = \begin{bmatrix} Ww + J^T W'(Jw - v_d) + b + J^T \lambda \\ -Jw + v_d \end{bmatrix} \tag{20}$$

and an expanded set of Ω as

$$\bar{\Omega} = \left\{ x = \left[x_1^T, x_2^T\right]^T, x_1 \in \Omega \subset \mathbb{R}^k, x_2 \in \mathbb{R}^m \right\}. \tag{21}$$

With the assistance of the new variable η, the function $F(\cdot)$, and the set $\bar{\Omega}$, the neural dynamics can be expressed in a compact way as

$$\epsilon\dot{\eta} = -\eta + P_{\bar{\Omega}}(\eta - F(\eta)). \tag{22}$$

This falls into the canonical form of projected dynamic systems investigated in [35] and [36]. For the function $F(\cdot)$, its gradient is expressed as

$$\nabla F = \begin{bmatrix} W + J^T W' J & J^T \\ -J & 0 \end{bmatrix}$$

which satisfies

$$\nabla F + \nabla^T F = \begin{bmatrix} 2W + 2J^T W' J & 0 \\ 0 & 0 \end{bmatrix} \succeq 0.$$

Due to this property, we conclude that $F(\cdot)$ is monotone by the following reasoning:

$$\begin{aligned} &(\eta_1 - \eta_2)^T (F(\eta_1) - F(\eta_2)) \\ &= (\eta_1 - \eta_2)^T \nabla F(\zeta)(\eta_1 - \eta_2) \\ &= \frac{1}{2}(\eta_1 - \eta_2)^T (\nabla F(\zeta) + \nabla^T F(\zeta))(\eta_1 - \eta_2) \geq 0 \quad \forall \eta_1 \ \forall \eta_2. \end{aligned} \tag{23}$$

According to [36, Theorem 1], the projected dynamics in (22) is Lyapunov stable and globally converges to $\eta^* = (w^*, \lambda^*)$ satisfying

$$(\eta - \eta^*)^T F(\eta^*) \geq 0 \quad \forall \eta \in \bar{\Omega}. \tag{24}$$

Expression (24) is a variational inequality, and can be equivalently written in the projection form as

$$P_{\bar{\Omega}}(\eta^* - F(\eta^*)) = \eta^* \tag{25}$$

which is the equilibrium point of (22). Recall the definition of η, we have

$$\begin{aligned} P_{\Omega}(w^* - Ww^* - J^T W'(Jw^* - v_d) - b - J^T\lambda^*) = w^* \\ Jw^* - v_d = 0. \end{aligned} \tag{26}$$

This is the case when choosing $w = w^*$ in (16). The equivalence among (16), (9), and (8) implies that w^* and λ^* are the optimal solution to

$$\min_{w\in\Omega} \max_{\lambda} L(w, \lambda) \tag{27}$$

for $L(w, \lambda)$ defined in (7). Since $L(w, \lambda)$ is the Lagrange function of (6), we thus conclude the optimality of w^* to the constrained programming (6), which completes the proof. ∎

IV. Neural Design in the Presence of Noises

In the last section, we have designed a neural network to deal with kinematic redundancy resolution in nominal situation without noises, i.e., $n = 0$ in (3). In real applications, noise is inevitable, and it is important to design a noise tolerant controller for effective control. In this section, we present the design of neural dynamics that takes noises into account.

A. Polynomial Noises

In this part, we consider the situation with polynomial noises. According to Taylor expansion, it is possible to represent any time-varying noises into a polynomial for a desired accuracy. In this sense, a polynomial noise can be regarded as a general representation of noises.

1) Neural Dynamics: We first present the neural dynamics with the capability to deal with polynomial noises. For an l-order polynomial noise $n = \sum_{i=0}^{l} n_i$ with $n_i = k_i t^i$ for $i = 0, 1, \ldots, l$ and k_i unknown, we present the following neural controller:

$$\begin{aligned} \epsilon u &= -(c_0\epsilon + 1)w + P_{\Omega}(w - Ww - J^T W'(Jw - v_d) \\ &\qquad - b - J^T\lambda) - \epsilon\sum_{i=1}^{l} c_i\eta_i - \sum_{i=0}^{l} c_i\mu_i \\ \epsilon\dot{\lambda} &= Jw - v_d \\ \dot{\mu}_0 &= w - P_{\Omega}(w - Ww - J^T W'(Jw - v_d) - b - J^T\lambda) \\ \dot{\mu}_i &= \mu_{i-1} \quad \forall i = 1, 2, \ldots, l \\ \dot{\eta}_1 &= w \\ \dot{\eta}_i &= \eta_{i-1} \quad \forall i = 2, 3, \ldots, l. \end{aligned} \tag{28}$$

About this neural controller, we have the following theoretical results.

Theorem 2: Suppose that the noise in (3) is an l-order polynomial relative to time, i.e., $n = \sum_{i=0}^{l} n_i$ with $n_i = k_i t^i$ for $i = 0, 1, \ldots, l$, where k_i is an unknown constant and t represents time. Neural control law (28) solves the kinematic resolution problem of a redundant manipulator modeled by (2) and (3), and stabilizes the system to the optimal solution of (6), provided the coefficients c_i for $i = 0, 1, 2, \ldots, l$ are chosen such that all the roots of the polynomial $s^{l+1} + c_0 s^l + c_1 s^{l-1} + \cdots + c_l = 0$ are in the left half plane, no matter how large the constant k_i is for $i = 1, 2, \ldots, l$.

Proof: We first define the control error as $e = w - P_{\Omega}\big(w - Ww - J^T W'(Jw - v_d) - b - J^T\lambda\big)$. Then, after substituting the expression of u in (18) for n, the neural dynamics (28) becomes

$$\begin{aligned} \epsilon\dot{w} &= -e - c_0\epsilon w - \epsilon\sum_{i=1}^{l} c_i\eta_i - \sum_{i=0}^{l} c_i\mu_i + \epsilon\sum_{i=0}^{l} n_i \\ \epsilon\dot{\lambda} &= Jw - v_d \\ \dot{\mu}_0 &= e \\ \dot{\mu}_i &= \mu_{i-1} \quad \forall i = 1, 2, \ldots, l \\ \dot{\eta}_1 &= w \\ \dot{\eta}_i &= \eta_{i-1} \quad \forall i = 2, 3, \ldots, l. \end{aligned} \tag{29}$$

Define a new variable $p_i = \epsilon\eta_i + \mu_i$ for $i = 1, 2, \ldots, l$, and $p_0 = \epsilon w + \int_0^t e dt$. Then, $\dot{p}_i = \epsilon\dot{\eta}_i + \dot{\mu}_i = \epsilon\eta_{i-1} + \mu_{i-1} = p_{i-1}$ for $i = 1, 2, \ldots, l$. With the new variable p_i, (29) can be further simplified to

$$\begin{aligned} \dot{p}_0 &= -\sum_{i=0}^{l} c_i p_i + \epsilon\sum_{i=0}^{l} n_i, \\ \dot{p}_i &= p_{i-1} \quad \forall i = 1, 2, 3, \ldots, l. \end{aligned} \tag{30}$$

It is noteworthy that the above-mentioned dynamics, in terms of the state variables p_i for $i = 0, 1, 2, \ldots, l$, represents a linear system with the noise $\epsilon \sum_{i=0}^{l} n_i$ as input. We apply Laplace transform to explore the stability and steady-state values of such a linear system with polynomial inputs. After transformation, the frequency domain equivalence to (30) is obtained as

$$\begin{aligned} sP_0(s) - p_0(0) &= -\sum_{i=0}^{l} c_i P_i(s) + \epsilon \sum_{i=0}^{l} N_i(s) \\ sP_i(s) - p_i(0) &= P_{i-1}(s) \quad \forall i = 1, 2, 3, \ldots, l \end{aligned} \tag{31}$$

where $P_i(s) = \mathcal{L}(p_i(t))$ represents the Laplace transform of p_i for $i = 0, 1, 2, \ldots, l$. $N_i(s) = \mathcal{L}(n_i(t)) = \mathcal{L}(k_i t^i) = i!/s^{i+1}$, where $i! = \Pi_{j=1}^{i} j$ represents the Laplace transform of $n_i = k_i t^i$. From the second equation in (31), it is obtained that

$$P_i(s) = \frac{p_i(0)}{s} + \frac{p_{i-1}(0)}{s^2} + \frac{p_{i-2}(0)}{s^3} + \cdots + \frac{p_1(0)}{s^i} + \frac{P_0(s)}{s^i}. \tag{32}$$

Submitting (32) into the first equation of (31) results in

$$\begin{aligned} sP_0(s) - p_0(0) = &-P_0(s)\left(c_0 + \frac{c_1}{s} + \frac{c_2}{s^2} + \cdots + + \frac{c_l}{s^l}\right) \\ &- \frac{1}{s}(c_1 p_1(0) + c_2 p_2(0) + \cdots + c_i p_i(0)) \\ &- \frac{1}{s^2}(c_2 p_1(0) + c_3 p_2(0) + \cdots + c_i p_{i-1}(0)) \\ &\cdots \\ &- \frac{1}{s^l} c_l p_1(0) + \epsilon \sum_{i=0}^{l} N_i(s). \end{aligned} \tag{33}$$

After reorganization, the above-mentioned equation becomes

$$\begin{aligned} P_0(s) = &\frac{1}{s^{l+1} + c_0 s^l + c_1 s^{l-1} + \cdots + c_l} \\ &\times (p_0(0)s^l - s^{l-1}(c_1 p_1(0) + c_2 p_2(0) + \cdots + c_l p_l(0)) \\ &\quad - s^{l-2}(c_2 p_1(0) + c_3 p_2(0) + \cdots + c_l p_{l-1}(0)) \ldots \\ &\quad - c_l p_1(0)) \\ &+ \frac{\epsilon s^l \sum_{i=0}^{l} N_i(s)}{s^{l+1} + c_0 s^l + c_1 s^{l-1} + \cdots + c_l}. \end{aligned} \tag{34}$$

Now we define $y = \dot{p}_0$. In frequency domain $Y(s) = \mathcal{L}(y(t)) = sP_0(s) - p_0(0)$. It can be further expressed as

$$\begin{aligned} Y(s) = &\frac{s}{s^{l+1} + c_0 s^l + c_1 s^{l-1} + \cdots + c_l} \\ &\times (p_0(0)s^l - s^{l-1}(c_1 p_1(0) + c_2 p_2(0) + \cdots + c_l p_l(0)) \\ &\quad - s^{l-2}(c_2 p_1(0) + c_3 p_2(0) + \cdots + c_l p_{l-1}(0)) \ldots \\ &\quad - c_l p_1(0) - p_0(0)) \\ &+ \frac{\epsilon s^{l+1} \sum_{i=0}^{l} N_i(s)}{s^{l+1} + c_0 s^l + c_1 s^{l-1} + \cdots + c_l}. \end{aligned} \tag{35}$$

Consider the system with $\sum_{i=0}^{l} N_i(s)$ as input and $Y(s)$ as output. Its transfer function can be defined as $G(s) = Y(s)(\sum_{i=0}^{l} N_i(s))^{-1}$, which characterizes the mapping from the noise $\sum_{i=0}^{l} N_i(s)$ to $Y(s)$ in frequency domain. From (35), it is clear that the transfer function $G(s)$ can be written as

$$G(s) = \frac{\epsilon s^{l+1}}{s^{l+1} + c_0 s^l + c_1 s^{l-1} + \cdots + c_l}. \tag{36}$$

The system characterized by $G(s)$ is stable because all roots of its characteristic polynomial $s^{l+1} + c_0 s^l + c_1 s^{l-1} + \cdots + c_l = 0$ are in the left half plane according to Routh–Hurwitz criterion [37]. In (34), the term $s\big(p_0(0)s^l - s^{l-1}(c_1 p_1(0) + c_2 p_2(0) + \cdots + c_l p_l(0)) - s^{l-2}(c_2 p_1(0) + c_3 p_2(0) + \cdots + c_l p_{l-1}(0)) \ldots - c_l p_1(0)\big)/(s^{l+1} + c_0 s^l + c_1 s^{l-1} + \cdots + c_l)$ corresponds to the zero input response and the term $\epsilon s^{l+1} \sum_{i=0}^{l} N_i(s)/(s^{l+1} + c_0 s^l + c_1 s^{l-1} + \cdots + c_l)$ to the zero state response. The employment of final value theorem to (34) yields

$$\begin{aligned} &\lim_{t\to\infty} y(t) \\ &= \lim_{s\to 0} sY(s) \\ &= \lim_{s\to 0} \frac{s^2}{s^{l+1} + c_0 s^l + c_1 s^{l-1} + \cdots + c_l} \\ &\quad \times \Bigg(p_0(0)s^l - s^{l-1}(c_1 p_1(0) + c_2 p_2(0) + \cdots + c_l p_l(0)) \\ &\qquad - s^{l-2}(c_2 p_1(0) + c_3 p_2(0) + \cdots + c_l p_{l-1}(0)) \ldots \\ &\qquad - c_l p_1(0) - p_0(0) + \epsilon \sum_{i=0}^{l} \frac{s^l i!}{s^{i+1}} \Bigg) = 0. \end{aligned} \tag{37}$$

Recall that $y = \dot{p}_0 = \epsilon \dot{w} + d(\int_0^t e dt)/dt = \epsilon \dot{w} + e = \epsilon \dot{w} + w - P_\Omega\big(w - Ww - J^T W'(Jw - v_d) - b - J^T \lambda\big) = 0$, which implies that the system dynamics ultimately reduces to the nominal system controlled by the dual neural network without the perturbation of any noises. According to Theorem 1, w solves the kinematic redundancy resolution problem and stabilizes to the optimal solution of (6), which completes the proof. ■

About the choice of parameters c_i for $i = 0, 1, 2, \ldots, l$ in the neural controller, we have the following remark.

Remark 1: The neural coefficients are required to be chosen such that all roots of the polynomial $s^{l+1} + c_0 s^l + c_1 s^{l-1} + \cdots + c_l = 0$ are in the left half plane. As an easy way for their choice, we may set c_i as follows:

$$\begin{aligned} c_0 &= (l+1)s_0 \\ c_1 &= \frac{(l+1)l}{2} s_0^2 \\ c_2 &= \frac{(l+1)l(l-1)}{6} s_0^3 \\ &\cdots \\ c_i &= \frac{(l+1)!}{(l-i)!(i+1)!} s_0^{i+1} \\ &\cdots \\ c_l &= s_0^{l+1} \end{aligned} \tag{38}$$

which are coefficients of the binomial $(s + s_0)^{l+1} = \sum_{i=0}^{l+1}((l+1)!/(i!(l+1-i)!))s_0^{l+1-i} s^i = 0$ for $s_0 \in \mathbb{R}$ and $s_0 > 0$. Note that the roots of this binomial are all $-s_0$ and the root condition is satisfied automatically.

About the generality of the proposed neural controller in coping with any type of time-varying noises, we have the following remark.

Remark 2: Taylor's theorem [38] claims that it can be reached for any desired accuracy to approximate a nonlinear function with polynomial Taylor series under some

mild conditions. In other words, we can always find an l to represent the noise as an l-order polynomial noise $n = \sum_{i=0}^{l} n_i$ with $n_i = k_i t^i$ with an arbitrarily small approximation error. This implies the generality of the proposed neural controller (28) to deal with arbitrary noises.

The proposed method in this paper is applicable to various real problems. On this point, we have the following remark.

Remark 3: In real applications, especially industrial ones, manipulators are inevitably affected by additive noises and this often results in the deviation of the real trajectory of the end-effector from the desired one. The proposed method in this paper significantly improves the tolerance of neural controllers to noise and, thus, is applicable to various engineering applications using robotic manipulators, e.g., painting robots, welding robots, quadlegged robots, and humanoid robots. The model considered in this paper is a general one, and it is suitable for any type of serial redundant-manipulators with any number of DOFs. This paper mainly focuses on the theoretical part, i.e., theoretical derivation of the neural controllers with high tolerance to additive noise, with its verification using numerical simulations based on PUMA 560 robotic arms. The verification on real robotic arms is out of the scope of the current paper and will be covered in our future research that emphasizes the practical implementation on real robotic hardware.

2) Practical Considerations: In real applications, it is necessary to determine a proper order for the polynomial noise before implementing the neural controller. For a particular problem, a trial-and-error procedure can be used to determine its value. In specific, we can first assume that the polynomial order is $l = 0$, i.e., a constant noise, and design neural controllers accordingly. With the devised neural controller, we can measure the control performance in terms of the tracking error. If it is not satisfactory, we can increase the polynomial order to $l = 1$, and check the tracking error again. Otherwise, we just stop iteration and use the current neural controller for control. If $l = 1$ cannot return with a satisfactory performance, we proceed to try $l = 2$. Recursively, we can find a proper order of the polynomial, and the corresponding neural controller, with which the control error is satisfactory. Note that we can always improve the tracking accuracy by increasing the order l as implied by the Taylor's theorem on the approximation power of polynomials.

In addition, we present the architecture of the proposed neural network (see Fig. 1) to show its structure from a connectionist perspective. It is clear from (28) and Fig. 1 that the neural model contains hidden states λ, μ_0, μ_1, $\mu_2, \ldots, \mu_l$, η_1, $\eta_2, \ldots, \eta_l$, and an explicit output u. The hidden states interact with each other in a dynamic way and form the recurrent layer of this neural network. The recurrent layer then impacts the output u with a static mapping and forms the output layer. The dependence of variables and the relationship between them as described by (28) is shown in Fig. 1. From Fig. 1, it can also be observed that there are $2l+2$ variables in the recurrent layer, i.e., λ, μ_0, μ_1, $\mu_2, \ldots, \mu_l$, η_1, $\eta_2, \ldots, \eta_l$. As to η, w impacts η_1, η_1 impacts η_2, ..., and η_{l-1} impacts η_l, in a cascaded way. The hidden variable λ dynamically depends on itself and the two inputs w and v_d. As to μ, the dynamics of μ_0 depends on λ, w and v_d, the dynamics of μ_1 only depends on μ_0, the dynamics of μ_2 only depends on $\mu_1, \ldots$, the dynamics of μ_l only depends on μ_{l-1}. All variables in the recurrent layer then impact the output u in a nonlinear way.

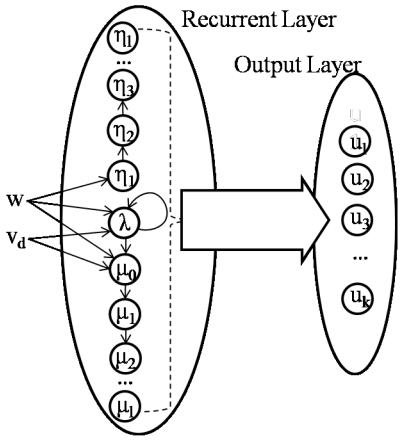


Fig. 1. Architecture of the proposed recurrent neural networks.

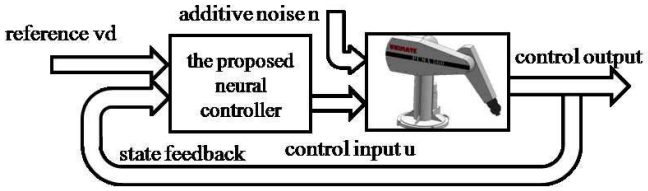


Fig. 2. Control diagram of the system using the proposed neural controller to steer the motion of a redundant manipulator.

Furthermore, the implementation of the interaction between the neural controller and the manipulator is also very important in real implementations. The manipulator is driven by the acceleration u of joint motors. With the actuation of all individual joints, the end-effector moves in its workspace and the control objective is to reach the desired velocity v_d in the workspace. For the neural controller in the presence of polynomial noises, the input to this neural controller includes two quantities: one is w, which is the angular velocity of joints and comes from the feedback provided by sensors mounted on the manipulator, and the other one is the reference velocity v_d, which comes from the control command. As a summary, we provide the control diagram shown in Fig. 2 to describe the interconnection between the proposed neural controller and the manipulator.

B. Special Cases

In this part, we examine two special cases of polynomial noises, namely constant noises and linearly time-varying noises, to show the specific presentation of the neural controllers.

1) Constant Noises: We first consider the situation with a constant noise. In practice, this category of noises can be employed to describe uncompensated weights, manipulator parameter errors, and so on.

Choosing $l = 0$ in (28), the neural controller falls into the special case with a constant noise, as represented in the

following form:

$$\begin{aligned}\epsilon u &= -(c_0\epsilon+1)w + P_\Omega\big(w - Ww - J^T W'(Jw-v_d) \\ &\qquad - b - J^T\lambda\big) - c_0\mu \\ \epsilon\dot{\lambda} &= Jw - v_d \\ \dot{\mu} &= w - P_\Omega(w - Ww - J^T W'(Jw-v_d) - b - J^T\lambda).\end{aligned} \tag{39}$$

In comparison with the neural controller (18) in the absence of noises, two additional terms are inserted into the control law: the term $-c_0\mu$, where μ is the integration of the $w - P_\Omega(w - Ww - J^T(Jw - v_d) - b - J^T\lambda)$ that measures the violation of the solution's optimality in quantity, and the term $-c_0\epsilon w$, which provides an extra damping to stabilize the system.

About the neural controller (39), we have the following theorem.

Corollary 1: Suppose that the noise in (3) is a constant. Neural control law (39) solves the kinematic resolution problem of a redundant manipulator modeled by (2) and (3), and stabilizes the system to the optimal solution of (6), no matter how large the constant noise is.

Proof: This result directly follows the proof of Theorem 2 by setting $l = 0$ in it. ■

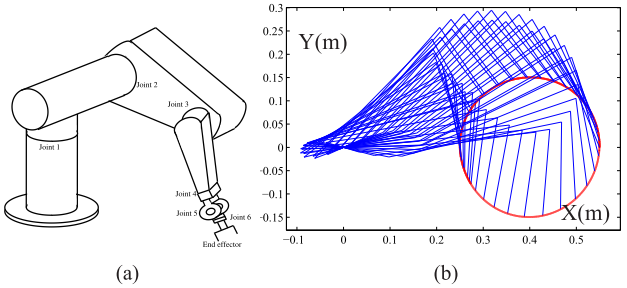
2) Linear Noises: Another important special case of polynomial noises is the one linearly varying with time. It models various physical processes, e.g., aging and fatigue. The linearly varying noise n can be represented as $n = k_1 t + k_0$ where k_1 and k_0 are unknown constants and t represents time. The linear noise n carries two parts: the constant part k_0 and the time proportional part $k_1 t$. Choosing $l = 1$ in (28), the neural controller falls into this special case with its neural controller expressed as

$$\begin{aligned}\epsilon u &= -(c_0\epsilon+1)w + P_\Omega(w - Ww - J^T W'(Jw-v_d) \\ &\qquad - b - J^T\lambda) - c_0\mu_0 - c_1\epsilon\theta - c_1\mu_1 \\ \epsilon\dot{\lambda} &= Jw - v_d \\ \dot{\mu}_0 &= w - P_\Omega(w - Ww - J^T W'(Jw-v_d) - b - J^T\lambda) \\ \dot{\mu}_1 &= \mu_0\end{aligned} \tag{40}$$

where $c_0 > 0$ and $c_1 > 0$ are constants. As stated in Corollary 1, the neural model (39) is able to deal with the constant noise part k_0. For the additional part $k_1 t$, two extra terms are introduced to form the new neural controller: $-c_1\mu_1$ and $-c_1\epsilon\theta$. The first term is the twice integration of the control error $w - P_\Omega\big(w - Ww - J^T W'(Jw-v_d) - b - J^T\lambda\big)$ introduced to penalize the control error resulting from the additive noise $k_1 t$. The second term is the integration of w introduced as a damping term for stabilization consideration. Notice that the newly introduced terms are both in integration format, due to the fact that the time-varying noise $n = k_1 t + k_0$ approaches infinity with the increase of time t and an increased amount of control action is desired to conquer its impact. About the proposed neural controller (40), we have the following convergence result in the presence of linear noises.

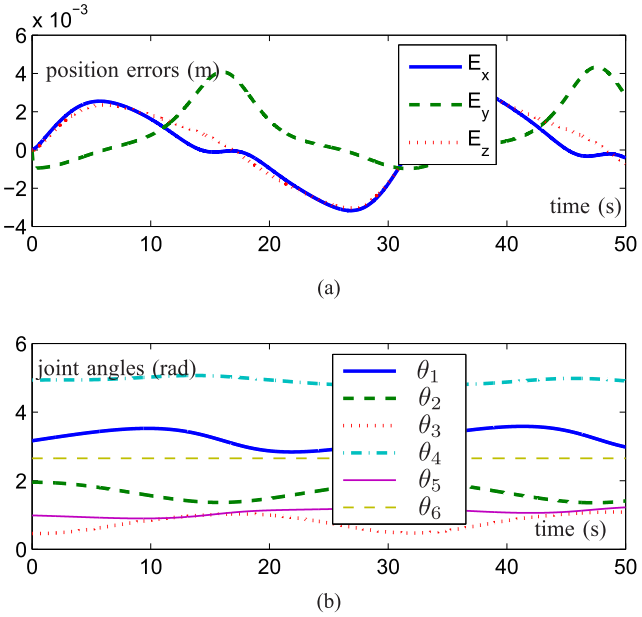
Corollary 2: Suppose that the noise in (3) is linear in time, i.e., $n = k_1 t + k_0$ where k_1 and k_0 are unknown constants, t represents time. Neural control law (40) solves the kinematic resolution problem of a redundant manipulator modeled by (2) and (3), and stabilizes the system to the optimal solution of (6), no matter how large the constants k_1 and k_0 are.

Proof: This result directly follows the proof of Theorem 2 by setting $l = 1$ in it. ■



(a) (b)

Fig. 3. Schematic of a 6-DOF PUMA 560 manipulator considered in the simulation and the generated trajectory of PUMA 560 controlled by the proposed neural network in the absence of noises. Right: red curve is the generated trajectory and the piecewise straight line in blue represents the links of the manipulator at different time. (a) PUMA 560 manipulator. (b) End-effector trajectory.



(a)

(b)

Fig. 4. Time profile of control parameters for the proposed neural network in the absence of noises. (a) Position tracking error. (b) Joint angles.

V. Simulations

In this section, we apply the proposed neural controller to the redundancy resolution of a PUMA 560 robot arm for the tracking of a circular motion. We compare the proposed method with existing dynamic neural solutions in the presence of various noises and show its robustness in dealing with additive noises.

A. Simulation Setup

In the simulation, we consider a PUMA 560 robot arm, with its D-H parameters summarized in Table I, as the manipulator platform for the tracking of a circular motion with its end-effector. PUMA 560 is a serial manipulator with six joints and thus have six DOFs for control. As the control task only

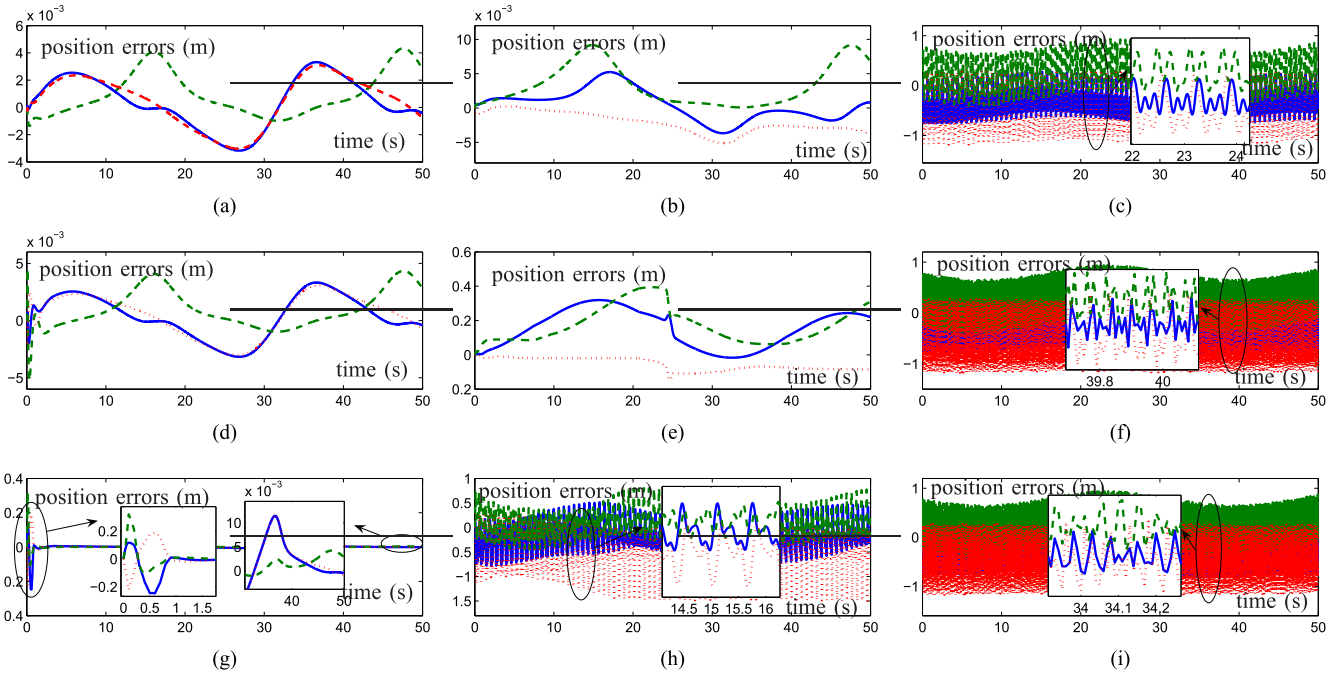


Fig. 5. Comparisons of the proposed approach with existing ones in the presence of different levels of noises. (a) This paper with noise $n = 0.1/\epsilon$. (b) Paper [39] with noise $n = 0.1/\epsilon$. (c) Paper [11] with noise $n = 0.1/\epsilon$. (d) This paper with noise $n = 1/\epsilon$. (e) Paper [39] with noise $n = 1/\epsilon$. (f) Paper [11] with noise $n = 1/\epsilon$. (g) This paper with noise $n = 10/\epsilon$. (h) Paper [39] with noise $n = 10/\epsilon$. (i) Paper [11] with noise $n = 10/\epsilon$.

TABLE I
SUMMARY OF THE D-H PARAMETERS OF THE PUMA 560 MANIPULATOR USED IN THE SIMULATION

link	$a\,(m)$	$\alpha\,(rad)$	$d\,(m)$
1	0	$\pi/2$	0
2	0.43180	0	0
3	0.02030	$-\pi/2$	0.15005
4	0	$\pi/2$	0.43180
5	0	$-\pi/2$	0
6	0	0	0.20000

refers to the position tracking, it forms a 3-D workspace. For this task, PUMA 560 serves as a redundant manipulator. The desired motion is to track a circle in xy plane with the radius 0.15 m at an angular speed 0.2 rad/s. In the simulation, we consider noises up to fourth order of polynomials with different amplitude. For parameters in the proposed neural network, we choose $\epsilon = 10^{-2}$, $W = I$, $W' = 0$, and $b = 0$. As to the parameter c_i, it is determined by the equation given in remark 1 by choosing $s_0 = 2$. The simulation results are compared with existing neural solutions proposed in [11] and [39] by setting the same parameters.

B. Nominal Situation

In this section, we present simulation results in the nominal situation without the presence of noises to show the convergence of the proposed neural controller. Without losing generality, we consider the situation with $l = 5$. Recall that we have chosen $s_0 = 2$ and we have $c_0 = 12$, $c_1 = 60, c_2 = 160, c_3 = 240, c_4 = 192, c_5 = 64$ from (38). Figs. 3 and 4 show the result of a typical simulation run for 50 s. As shown in Fig. 3, the generated trajectory for the PUMA 560 manipulator matches the desired circular motion. Quantitatively, the position tracking error in the workspace along x-, y-, and z-directions are drawn in Fig. 4(a). As observed from Fig. 4(a), the position tracking error remains at a low level (lower than 5×10^{-3} in absolute values for all dimensions). The generated joint angles at different time are shown in Fig. 4(b).

C. Constant Noises

In this simulation, we set the noise as a constant, and compare the performance obtained by using different neural controllers. Note that the noise goes into the control channel in the form of (3) and the control input u is amplified by $1/\epsilon$. Therefore, we consider a set of noise comparable to the control input as $n = 0.1/\epsilon$, $n = 1/\epsilon$, and $n = 10/\epsilon$. We compare the results generated by the proposed algorithm in this paper with two existing neural controllers proposed in [39] and [11] with the same parameter setup. As shown in Fig. 5, with the presence of the constant noise $n = 0.1/\epsilon$, the position tracking error for the proposed method is restricted to be lower than 5×10^{-3} in the absolute value across the entire simulation run. In contrast, the dual neural method proposed in [39] has a larger tracking error with a peak value around 10×10^{-3} at time $t = 15$ s and $t = 48$ s [see Fig. 5(e)]. As to the dual neural method proposed in [11], the system loses stability and fluctuates severely with time [see Fig. 5(f)]. With the increase of the noise to $n = 1/\epsilon$, the position tracking error of the method proposed in [39] increases a lot and reaches a peak value 0.4 at time $t = 25$ s, and the system controlled by the neural law proposed in [11] demonstrates enhanced fluctuation with an increased frequency. Note that due to the restriction introduced by the physical size of the

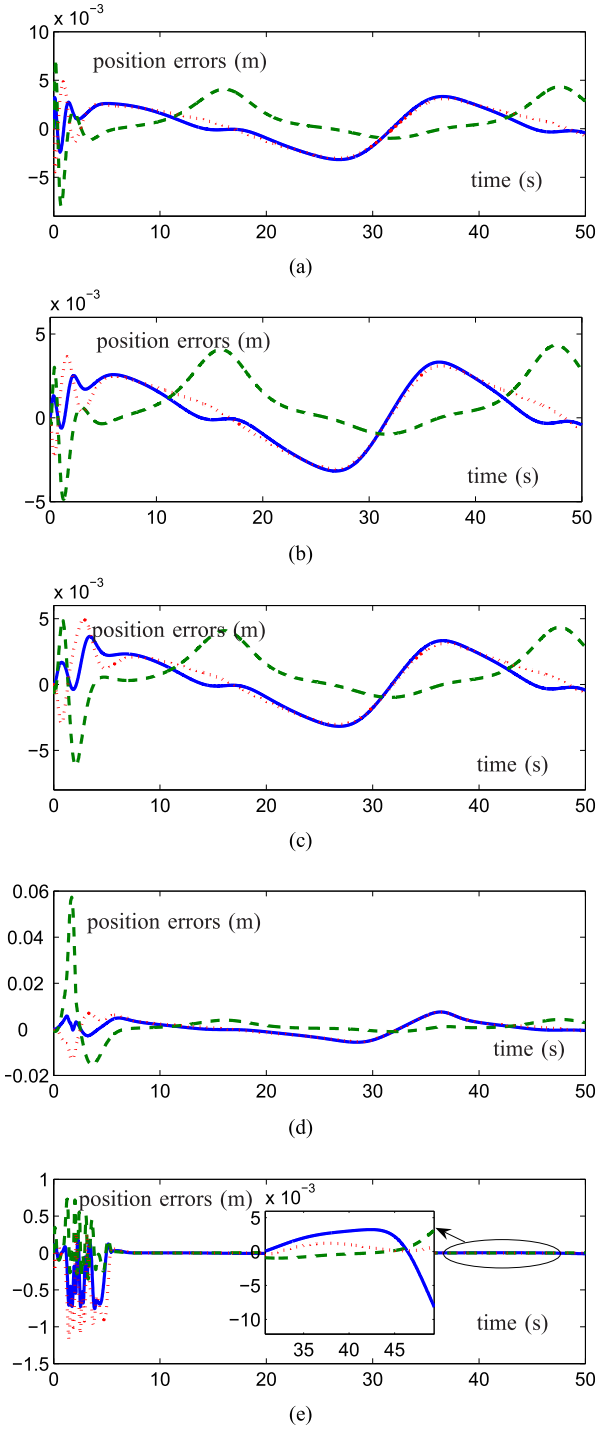


Fig. 6. Performance comparisons of the proposed approach for the tracking of circular motions in the presence of various noises. (a) Linear noise $n = 10t/\epsilon$. (b) Quadratic noise $n = 10t^2/\epsilon$. (c) Cubic noise $n = 10t^3/\epsilon$. (d) Fourth-order noise $n = 10t^4/\epsilon$. (e) Noise $n = 10t^4/\epsilon + 10t^3/\epsilon + 10t^2/\epsilon + 10t/\epsilon + 10/\epsilon$.

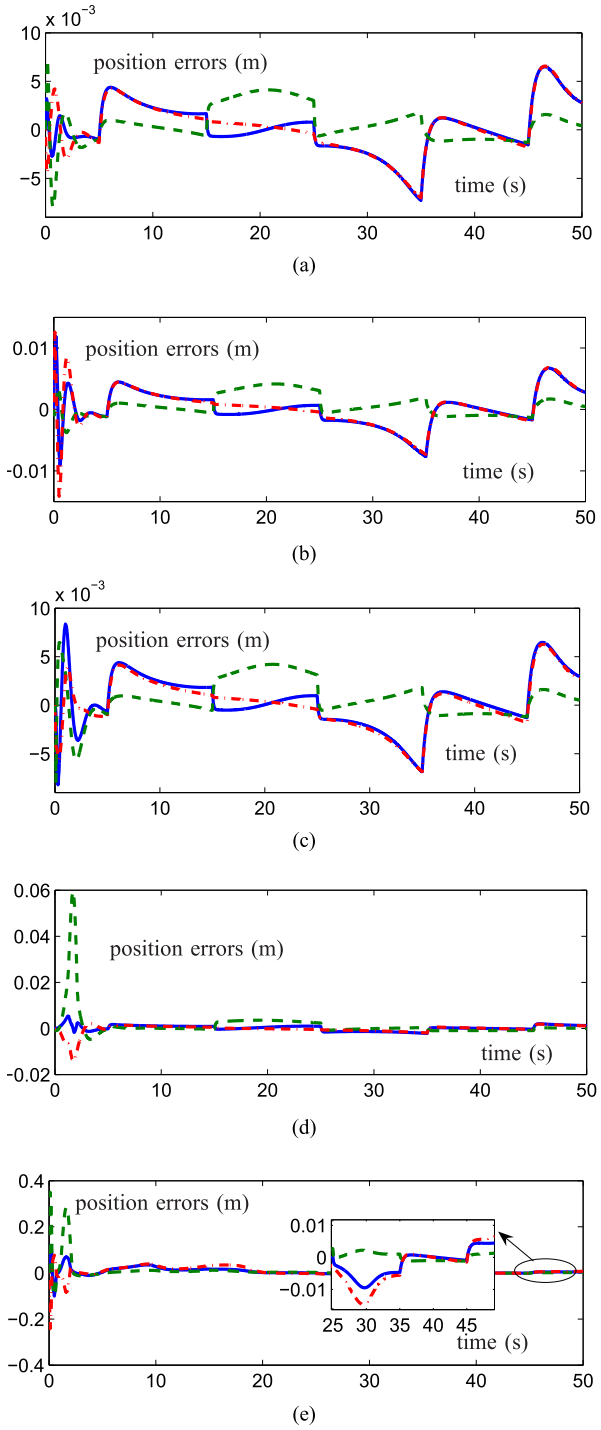


Fig. 7. Performance comparisons of the proposed approach for the tracking of square motions in the presence of various noises. (a) Linear noise $n = 10t/\epsilon$. (b) Quadratic noise $n = 10t^2/\epsilon$. (c) Cubic noise $n = 10t^3/\epsilon$. (d) Fourth-order noise $n = 10t^4/\epsilon$. (e) Noise $n = 10t^4/\epsilon + 10t^3/\epsilon + 10t^2/\epsilon + 10t/\epsilon + 10/\epsilon$.

manipulator, the position tracking error is always bounded. For the case with noise $n = 1/\epsilon$, in contrast to the severe performance degradation of existing methods [11], [39], the proposed one still performs well with a similar noise level as in the situation without the presence of noise [see Fig. 5(a)]. For even larger noise $n = 10/\epsilon$, the method proposed in [39] also loses stability and starts to oscillate with a big error amplitude as observed in Fig. 5(h). As to the proposed method shown in Fig. 5(g), after a short period for transition, the position tracking error returns to a small value below 10×10^{-3} very soon. Note that the short transition shown in Fig. 5(g) (left) is due to the fact that the large constant noise $n = 10/\epsilon$ acts on the system at the beginning of this simulation run, while it takes a short transient time for the proposed control to

adapt to this noise. Overall, the simulation reveals that existing solutions designed in nominal situations are not robust against additive noises and the proposed neural controller with an explicit consideration of noise tolerance receives a significant performance improvement over existing ones in dealing with unknown constant noises.

D. Time-Varying Polynomial Noises

In this simulation, we consider a set of polynomial noises to verify the effectiveness of the proposed neural controller. Particularly, we consider a linear time-varying noise $n = 10t/\epsilon$, a quadratically time-varying one $n = 10t^2/\epsilon$, a cubically time-varying one $n = 10t^3/\epsilon$, a fourth-order time-varying one $n = 10t^4/\epsilon$, and one with all the above ingredients plus a constant, i.e., $n = 10t^4/\epsilon + 10t^3/\epsilon + 10t^2/\epsilon + 10t/\epsilon + 10/\epsilon$. Notice that the noise signals considered in this section are all time-varying and their values could be very large. For example, at the end of the simulation $t = 50$ s, the linear time-varying noise becomes $n = 500/\epsilon$, which is 50 times larger than the largest constant noise considered in the last part. For existing methods [11], [39], as demonstrated in Section V-C, they cannot handle such large noises. In the presence of the above-mentioned noises, the position tracking error remains at a low level for the linear noise, the quadratic noise, and the cubic noise as shown in Fig. 6(a)–(c). Although it takes a short transition for neural adaption with the increase of noises, as seen in Fig. 6(d) and (e), the proposed neural network is able to reach a small tracking error no greater than 10×10^{-3} ultimately, which is comparable to the situation without the presence of noises. To further show the improved performance with the proposed neural network in the presence of polynomial noises, we additionally consider the track of a square path in the horizontal plane with the size of each edge being 0.3 m. With the same setup of neural controller parameters as the situation to track a circular motion, we run one more set of simulation. As shown in Fig. 7, a small tracking error is reached for the position tracking of a square path with a PUMA 560 manipulator using the proposed neural network. Simulation results verify the theoretical conclusions on the convergence of the proposed neural networks for the redundancy resolution of manipulators.

VI. Conclusion

In this paper, we consider the recurrent neural network design for redundancy resolution of manipulators in the presence of polynomial noises. A novel neural control law is proposed to tackle this problem. Theorems are established to show the convergence of the neural controller with noisy inputs. Extensive simulations verify the theoretical conclusions. Numerical comparisons with existing neural controllers show significant performance improvements in control accuracy. Before ending this paper, it is worth mentioning that to the best of our knowledge, this is the first neural controller with the capability to completely eliminate the impact of time-varying noises in the format of polynomials for efficient redundancy resolution of manipulators.

References

[1] J. Hollerbach and K. Suh, "Redundancy resolution of manipulators through torque optimization," *IEEE J. Robot. Automat.*, vol. 3, no. 4, pp. 308–316, Aug. 1987.

[2] S. Cocuzza, I. Pretto, and S. Debei, "Novel reaction control techniques for redundant space manipulators: Theory and simulated microgravity tests," *Acta Astron.*, vol. 68, nos. 11–12, pp. 1712–1721, Jun./Jul. 2011.

[3] A. Escande, N. Mansard, and P.-B. Wieber, "Hierarchical quadratic programming: Fast online humanoid-robot motion generation," *Int. J. Robot. Res.*, vol. 33, no. 7, pp. 346–355, 2014.

[4] K. C. Park, P. H. Chang, and S. H. Kim, "The enhanced compact QP method for redundant manipulators using practical inequality constraints," in *Proc. IEEE Int. Conf. Robot. Autom.*, vol. 1. May 1998, pp. 107–114.

[5] H. Sadjadian, H. D. Taghirad, and A. Fatehi, "Neural networks approaches for computing the forward kinematics of a redundant parallel manipulator," *Int. J. Comput. Intell.*, vol. 2, no. 1, pp. 40–47, Jan. 2005.

[6] N. Kumar, V. Panwar, N. Sukavanam, S. P. Sharma, and J. H. Borm, "Neural network-based nonlinear tracking control of kinematically redundant robot manipulators," *Math. Comput. Model.*, vol. 53, nos. 9–10, pp. 1889–1901, May 2011.

[7] L. Zhang, K. Li, H. He, and G. W. Irwin, "A new discrete-continuous algorithm for radial basis function networks construction," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 24, no. 11, pp. 1785–1798, Nov. 2013.

[8] P. K. Patchaikani, L. Behera, and G. Prasad, "A single network adaptive critic-based redundancy resolution scheme for robot manipulators," *IEEE Trans. Ind. Electron.*, vol. 59, no. 8, pp. 3241–3253, Aug. 2012.

[9] S. Zang and A. G. Constantinides, "Lagrange programming neural networks," *IEEE Trans. Circuits Syst. II, Analog Digit. Signal Process.*, vol. 39, no. 7, pp. 441–452, Jul. 1992.

[10] S. Li, Y. Zhang, and L. Jin. "Kinematic control of redundant manipulators using neural networks," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 2, no. 12, pp. 528–538, 2016.

[11] Y. Zhang, J. Wang, and Y. Xia, "A dual neural network for redundancy resolution of kinematically redundant manipulators subject to joint limits and joint velocity limits," *IEEE Trans. Neural Netw.*, vol. 14, no. 3, pp. 658–667, May 2003.

[12] Y. Xia and J. Wang, "A dual neural network for kinematic control of redundant robot manipulators," *IEEE Trans. Syst., Man, Cybern. B, Cybern.*, vol. 31, no. 1, pp. 147–154, Feb. 2001.

[13] S. Li, J. He, Y. Li, and M. U. Rafique, "Distributed recurrent neural networks for cooperative control of manipulators: A game-theoretic perspective," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 28, no. 2, pp. 415–426, Feb. 2016.

[14] Y. Zhang, *Recurrent Neural Networks*. Saarbrücken, Germany: Lambert Academic Publishing, 2009.

[15] Y. Zhang, S. S. Ge, and T. H. Lee, "A unified quadratic-programming-based dynamical system approach to joint torque optimization of physically constrained redundant manipulators," *IEEE Trans. Syst., Man, Cybern. B, Cybern.*, vol. 34, no. 5, pp. 2126–2132, Oct. 2004.

[16] Y. Zhang, J. Wang, and Y. Xu, "A dual neural network for bi-criteria kinematic control of redundant manipulators," *IEEE Trans. Robot. Autom.*, vol. 18, no. 6, pp. 923–931, Dec. 2002.

[17] B. Cai and Y. Zhang, "Different-level redundancy-resolution and its equivalent relationship analysis for robot manipulators using gradient-descent and Zhang's neural-dynamic methods," *IEEE Trans. Ind. Electron.*, vol. 59, no. 8, pp. 3146–3155, Aug. 2012.

[18] Y. Zhang and J. Wang, "A dual neural network for constrained joint torque optimization of kinematically redundant manipulators," *IEEE Trans. Syst., Man, B, Cybern.*, vol. 32, no. 5, pp. 654–662, Oct. 2002.

[19] S. Li, Z.-H. You, H. Guo, X. Luo, and Z.-Q. Zhao, "Inverse-free extreme learning machine with optimal information updating," *IEEE Trans. Cybern.*, vol. 46, no. 5, pp. 1229–1241, May 2016.

[20] L. Jin, S. Li, H. La, and X. Luo, "Manipulability optimization of redundant manipulators using dynamic neural networks," *IEEE Trans. Ind. Electron.*, vol. 12, no. 15, pp. 312–323, 2016.

[21] L. Jin and S. Li, "Distributed task allocation of multiple robots: A control perspective," *IEEE Trans. Syst., Man, Cybern., Syst.*, vol. 21, no. 18, pp. 12–23, 2017.

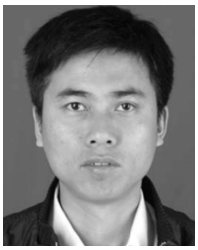
[22] A. M. Mohammed and S. Li, "Dynamic neural networks for kinematic redundancy resolution of parallel Stewart platforms," *IEEE Trans. Cybern.*, vol. 17, no. 3, pp. 1400–1410, Jul. 2015.

[23] S. Li, S. Chen, B. Liu, Y. Li, and Y. Liang, "Decentralized kinematic control of a class of collaborative redundant manipulators via recurrent neural networks," *Neurocomputing*, vol. 91, pp. 1–10, Aug. 2012.

[24] L. Jin and Y. Zhang, "G2-type SRMPC scheme for synchronous manipulation of two redundant robot arms," *IEEE Trans. Cybern.*, vol. 45, no. 2, pp. 153–164, Feb. 2015.
[25] Y. Zhang, W. Li, and Z. Zhang, "Physical-limits-constrained minimum velocity norm coordinating scheme for wheeled mobile redundant manipulators," *Robotica*, vol. 33, no. 6, pp. 1325–1350, Jul. 2015.
[26] L. Xiao and Y. Zhang, "A new performance index for the repetitive motion of mobile manipulators," *IEEE Trans. Cybern.*, vol. 44, no. 2, pp. 280–292, Feb. 2014.
[27] Y. Zhang, Yinyan, and S. Li, "Predictive suboptimal consensus of multiagent systems with nonlinear dynamics," *IEEE Trans. Syst., Man, Cybern., Syst.*, vol. 10, no. 23, pp. 231–242, 2017.
[28] S. Li, M. Zhou, X. Luo, and Z.-H. You, "Distributed winner-take-all in dynamic networks," *IEEE Trans. Autom. Control*, vol. 62, no. 2, pp. 577–589, Feb. 2016.
[29] S. Li, B. Liu, and Y. Li, "Selective positive–negative feedback produces the winner-take-all competition in recurrent neural networks," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 24, no. 2, pp. 301–309, Feb. 2013.
[30] S. Li, Y. Lou, and B. Liu, "Bluetooth aided mobile phone localization: A nonlinear neural circuit approach," *ACM Trans. Embedded Comput. Syst.*, vol. 13, no. 4, p. 78, 2014.
[31] S. Li and Y. Li, "Nonlinearly activated neural network for solving time-varying complex Sylvester equation," *IEEE Trans. Cybern.*, vol. 44, no. 8, pp. 1397–1407, Aug. 2014.
[32] H. Zhang, Z. Wang, and D. Liu, "A comprehensive review of stability analysis of continuous-time recurrent neural networks," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 25, no. 7, pp. 1229–1262, Jul. 2014.
[33] S. Li *et al.*, "SP-NN: A novel neural network approach for path planning," in *Proc. IEEE Int. Conf. Robot. Biomimetics (ROBIO)*, Dec. 2007, pp. 1355–1360.
[34] S. Boyd and L. Vandenberghe, *Convex Optimization*. New York, NY, USA: Cambridge Univ. Press, 2004.
[35] Y. S. Xia and J. Wang, "On the stability of globally projected dynamical systems," *J. Optim. Theory Appl.*, vol. 106, no. 1, pp. 129–150, Jul. 2000.
[36] X. B. Gao, "Exponential stability of globally projected dynamic systems," *IEEE Trans. Neural Netw.*, vol. 14, no. 2, pp. 426–431, Mar. 2003.
[37] G. F. Franklin, J. D. Powell, and A. Emami-Naeini, *Feedback Control of Dynamic Systems*, 4th ed. Upper Saddle River, NJ, USA: Prentice-Hall, 2001.
[38] G. Strang, *Calculus*, 2nd ed. Wellesley, MA, USA: Wellesley-Cambridge, 2010.
[39] Y. Xia, G. Feng, and J. Wang, "A primal-dual neural network for online resolving constrained kinematic redundancy in robot motion control," *IEEE Trans. Syst., Man, Cybern. B, Cybern.*, vol. 35, no. 1, pp. 54–64, Feb. 2005.

Shuai Li received the B.E. degree in precision mechanical engineering from the Hefei University of Technology, Hefei, China, in 2005, the M.E. degree in automatic control engineering from the University of Science and Technology of China, Hefei, in 2008, and the Ph.D. degree in electrical and computer engineering from the Stevens Institute of Technology, Hoboken, NJ, USA, in 2014.

He is currently an Assistant Professor with the Department of Computing, The Hong Kong Polytechnic University, Hong Kong. His current research interests include control of dynamic networks, distributed estimation and control, and coordination of multiple robots.



Huanqing Wang received the B.Sc. degree in mathematics from Bohai University, Jinzhou, China, in 2003, the M.Sc. degree in mathematics from Inner Mongolia University, Hohhot, China, in 2006, and the Ph.D. degree from the Institute of Complexity Science, Qingdao University, Qingdao, China, in 2013.

He was a Post-Doctoral Fellow with the Department of Electrical Engineering, Lakehead University, Thunder Bay, ON, Canada, in 2014. He is currently a Post-Doctoral Fellow with the Department of Systems and Computer Engineering, Carleton University, Ottawa, ON, Canada. He has authored or co-authored over 30 papers in top international journals. His current research interests include adaptive backstepping control, fuzzy control, neural networks control, stochastic nonlinear systems, and teleoperation.

Dr. Wang serves as an Associate Editor for several journals, including *Neural Computing and Applications*, the *International Journal of Control, Automation, and Systems*, and the IEEE ACCESS.

Muhammad Usman Rafique received the bachelor's and master's degrees in mechatronics engineering from the National University of Sciences and Technology, Karachi, Pakistan, in 2007 and 2010, respectively. He is currently pursuing the Ph.D. degree with the Department of Computing, The Hong Kong Polytechnic University, Hong Kong.