This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TII.2019.2899909, IEEE Transactions on Industrial Informatics



# On Generalized RMP Scheme for Redundant Robot Manipulators Aided with Dynamic Neural Networks and Nonconvex Bound Constraints

Zhengtai Xie, Long Jin, *Member, IEEE*, Xiujuan Du, Xiuchun Xiao, Hongxin Li, and Shuai Li

***Abstract*—In this paper, in order to analyze the existing repetitive motion planning (RMP) schemes for kinematic control of redundant robot manipulators, a generalized RMP scheme, which systematizes the existing RMP schemes, is presented. Then, the corresponding dynamic neural networks are derived, which leverage the gradient descent method with the velocity compensation with the feasibility proven theoretically. Given that the position errors of the end-effector should be tiny enough in the applications of redundant robot manipulators when executing a given task, especially for a precision instrument, the performance analyses on the control schemes are urgently desirable. In this paper, the upper bound of the position error on the existing RMP schemes is deduced theoretically and verified by computer simulations, with the relationship between the position error and the manipulability derived. In addition, dynamic neural networks are constructed to solve the generalized RMP schemes, with the joint velocity limits in RMP schemes extended to the nonconvex constraint. Finally, computer simulations based on different redundant robot manipulators and comparisons based on different controllers are conducted to verify the feasibility of the generalized RMP scheme and the proposed dynamic neural networks.**

***Index Terms*—Dynamic neural network, gradient descent, repetitive motion planning, kinematic control, nonconvex constraint, manipulability, simulative results.**

## I. INTRODUCTION

This work is supported by the National Natural Science Foundation of China (with number 61703189), by the Fund of Key Laboratory of IoT of Qinghai Province (No. 2017-ZJ-Y21), by the International Science and Technology Cooperation Program of China (No. 2017YFE0118900), by the Natural Science Foundation of Gansu Province, China (No. 18JR3RA264), by the Sichuan Science and Technology Program (No. 19YYJC1656), by the Natural Science Foundation of Hunan Province, (No. 2017JJ3257), by the Key Lab of Digital Signal and Image Processing of Guangdong Province (No. 2016GDDSIPL-02), by the Doctoral Initiating Project of Guangdong Ocean University (No. E13428), by the Innovation and Strength Project of Guangdong Ocean University (No. Q15090), and by the Research Foundation of Education Bureau of Hunan Province, China (No. 17C1299). *(Corresponding author: Long Jin).* 

Z. Xie, L. Jin and H. Li are with the School of Information Science and Engineering, Lanzhou University, Lanzhou, China (e-mail: jinlongsysu@foxmail).

X. Du is with the Key Laboratory of IoT of Qinghai Province, Qinghai Normal University, Xining, China.

X. Xiao is with the College of Electronic and Information Engineering, Guangdong Ocean University, Zhanjiang, China.

S. Li is with the Department of Computing, The Hong Kong Polytechnic University, Hong Kong. Kindly note that Z. Xie, L. Jin, X. Du, and X. Xiao are jointly of the first authorship.

WHEN it comes to redundant robot manipulators, the key characteristic is that they have more degrees of freedom (DOFs) than required, which is different from non-redundant ones. Therefore, redundant robot manipulators have the capacity to complete complex tasks with secondary constraints and performance indices considered [1]–[6]. In this sense, the DOFs, to some extent, determine the number of additional tasks or indices that the manipulators can perform. Increasing number of schemes for redundant robot manipulators have been investigated for decades, such as minimum kinetic energy consumption [7], haptic identification [8], avoidance of singularities [9] and parameter estimation [10]. In addition, redundant manipulators are able to equip with the noise tolerance [11] and a powerful adaptability to environment [12]. Moreover, redundant robot manipulators are used to establish human-robot interaction systems in [13], in which the involved manipulators could be controlled directly and accurately by human mind [14]. Given that redundant manipulators have great potential advantages in dealing with knotty tasks in scientific and engineering fields, much effort on their high performance control with various emphases is still being devoted by scholars. Kinematic control of redundant robot manipulators involves the transformation from Cartesian space to joint space, which carries nonlinear mapping information. To solve this nonlinear mapping problem, a great number of schemes and control laws at different levels have been presented and investigated [15]–[19]. In [15], a pseudo-inverse method is employed to solve the robot redundancy problem, which is of the limitations in handling the joint limits [16] and optimizing a specific performance index [17]. As an improvement, Li et al. present a dynamic neural network for kinematic control of redundant robot manipulators in [18], which is able to remedy the weakness of pseudo-inverse method with pseudo-inversion operation eliminated. In [19], Yang et al. present a neural control method for bimanual robots with the strong global stability and accuracy, which is of great significance for high-precision needed tasks.

It is worth pointing out that all of the existing repetitive motion planning (RMP) schemes for kinematic control of redundant robot manipulators impose the convex bound constraints for joint limits, which may fail to describe the actual situations in robotics applications with the nonconvex mapping in bound constraints. For example, for a motor performing the task in the $i$th joint of a redundant robot manipulator, its output joint velocity or joint acceleration may be discontinuous, e.g., the pulse per second (PPS) for controlling the motors

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TII.2019.2899909, IEEE Transactions on Industrial Informatics



in robots is depicted in discrete form, which is evidently a nonconvex mapping condition. A detailed explanation is given as follows. Suppose that the joint velocity outputted by a motor is discontinuous in the range $[\dot{q}^-, \dot{q}^+]$, where $\dot{q}^-$ and $\dot{q}^+$ denote the lower and upper bounds of the velocity outputted by the motor. Then, there exists a value $\dot{q}^0 \notin [\dot{q}^-, \dot{q}^+]$ with $\dot{q}^- < \dot{q}^0 < \dot{q}^+$. Besides, there must exist $\dot{q}^1 \in [\dot{q}^-, \dot{q}^+]$, $\dot{q}^2 \in [\dot{q}^-, \dot{q}^+]$ and $\dot{q}^0 = (\dot{q}^1 + \dot{q}^2)/2$. This indicates that the discontinuous range is of nonconvexity.

Intelligent computing techniques have effectively solved many knotty problems in reality [20]–[25]. The classification and positioning optimization algorithm based on differential evolution is introduced into the iterative self-marking process to construct the framework in [23], which achieves high performance. In addition, protein interaction networks are employed to detect protein complexes, which improves computational efficiency [24]. Moreover, neural networks and neural dynamics are frequently used in intelligent computing. As powerful tools, neural networks have relatively convenience and giant superiority in fault-tolerant capability [26], high robustness [27], etc [28]. Consequently, neural networks have been increasingly popular with many applications and investigated widely [29]–[34]. At present, neural networks are widely used in various spheres, for instance, artificial intelligence, robotics [35], biology and so on [36]. On account of the above advantages, plenty of schemes based on neural networks have been proposed to solve the kinematics control problem of redundant manipulators [18], [19], [37], [38]. For instance, Yang et al. put forward neural networks to enhance tele-operated control system [37] and to improve stability and accuracy of a system composed of dual robot manipulators [19]. However, neural-dynamic methods are not able to handle constrained optimization problems in many scenes [6], [40]. Moreover, one of the most common approaches for kinematics control of redundant manipulators is to use the Lagrangian multiplier method or Karush-Kuhn-Tucker conditions [39] in order to derive neural networks taking equality constraints and inequality constraints into account. In contrast, a novel method using gradient descent method [40], [41] with the velocity compensation considered to remedy the lagging errors is introduced in this paper, which does not exploit the conventional Lagrangian multiplier method or Karush-Kuhn-Tucker conditions and is able to solve the equality and inequality constraints simultaneously.

The redundancy resolution of redundant robot manipulators is often formulated as a constrained optimization problem and then solved by numerical algorithms or neural networks. For example, in [42], dynamic neural networks are constructed to solve a quadratic programming problem originated from the kinematic control of robots. A performance index in a quadratic programming selects a most suitable solution among infinite ones for achieving a given criterion. As one of the most common indices, the repetitive motion has been investigated in the past years [38], [43]–[47]. In [43], as a seminal work, five different RMP schemes based on pseudo-inverse method are proposed, which opens the door to the research on the repetitive motion in redundant robot manipulators. Following the inspiring work in [43], an RMP scheme is presented in [44], in which the preferable repeatable inverse could be obtained. These achievements in [43] and [44] are obtained by leveraging the pseudo-inverse method, which can not handle the joint limits. By deeply understanding the nature of redundancy resolution, optimization techniques are used to model the repetitive motion generation in redundant robot manipulators [45]–[47] and physical constraints in joint space are considered and imposed to the RMP scheme in [48] with the aid of recurrent neural networks. After that, the RMP schemes in single manipulators are extended to the repetitive motion generation in dual redundant manipulators [38] or even multiple manipulators with limited communications [49]. It is worth noting that, even for the simulations synthesized by these existing RMP schemes and conducted in ideal situation, position errors are always observed and a theoretical investigation on the origination of the position errors remains unclear in the existing RMP schemes.

For robotic tasks requiring high precision, position errors perform an extremely vital factor, which are closely related to the performance. For the computer simulations or experiments based on redundant robot manipulators aided with the existing RMP schemes, as observed in [38], [45]–[49], the position errors are unavoidable even in the ideal environment. The most popular treatment is to feedback the position error to achieve the high performance. However, the position error always fluctuates or even diverges even with large feedback gains in the existing RMP schemes. This paper is devoted to figuring out the quantitative formation of the position error in the existing RMP schemes with theoretical analyses, in order to complete high-precision task in obtaining accurate controlling for redundant robot manipulators. To this end, a generalized RMP scheme systematizing the existing RMP schemes is presented. Then, dynamic neural networks are constructed by using the gradient descent method with the velocity compensation. In addition, this paper breaks the limitation on the convexity of the bound constraints and proposes nonconvex-allowed constraints for a better modeling on the joint limits of the redundant robot manipulators.

The rest of this paper is organized as follows. In Section II, a generalized RMP scheme is presented to systematize the existing RMP schemes. Then the dynamic neural networks are conducted by using the gradient descent method with the velocity compensation considered and their feasibility are analyzed theoretically. The formulations describing the position error in the existing RMP schemes is deduced and analyzed theoretically, with the relationship between the position error and the manipulability derived. In section III, simulation results are provided and analyzed based on different redundant robot manipulators. The nonconvex projection problem in the bound constraints is solved and the analyses on position error bound are verified in the simulations. Section IV provides comparisons among the controller presented in this paper and the existing controllers [7], [18], [38], [43], [48], [50], [51], [54]. Section V presents conclusions. Before ending this section, the contributions of this paper are listed as follows:

- The gradient descent method is exploited to construct the dynamic neural networks with the velocity compensation considered to remedy the lagging errors for RMP schemes.

- Different from the existing RMP schemes, the generalized RMP scheme investigated in this paper removes the convex requirements on bound constraint with compliance to the nonconvex situation.
- For the first time, the relationship between the joint angle error in joint space and the position error in Cartesian space in the existing RMP schemes is observed and analyzed.
- For the first time, the relationship between the manipulability and the position error synthesized by the RMP schemes is theoretically derived and verified.
- Computer simulations based on different redundant robot manipulators and comparisons based on different controllers are conducted to verify the feasibility of the generalized RMP scheme, the effectiveness of the dynamic neural networks and the correctness of the theoretical analyses.

## II. Preliminaries, scheme construction, and solutions

In this section, preliminaries on the manipulator kinematics and the joint-drift-free problem existing in the kinematic control of redundant robot manipulators are revisited. Then, a generalized RMP scheme and novel dynamic neural networks are constructed to remedy the weaknesses of the existing RMP schemes. Theoretical analyses are presented to prove the feasibility and stability of the generalized RMP scheme aided with the novel neural networks.

### A. Manipulator kinematics

The forward kinematics equation of a redundant robot manipulator describes the nonlinear transformation from the joint space $\mathbf{q}(t) \in \mathbb{R}^n$ to the Cartesian space $\mathbf{r}(t) \in \mathbb{R}^m$ (with n>m), which can be written as

$$\mathbf{r}(t) = \Phi(\mathbf{q}(t)), \tag{1}$$

where $\Phi(\cdot)$ represents the nonlinear transformation from $\mathbf{q}(t)$ to $\mathbf{r}(t)$. Taking the time derivative on both sides of (1), we are able to get

$$\dot{\mathbf{r}}(t) = J(\mathbf{q}(t))\dot{\mathbf{q}}(t), \tag{2}$$

where $J(\mathbf{q}(t)) = \partial\Phi(\mathbf{q})/\partial\mathbf{q} \in \mathbb{R}^{m\times n}$ is the Jacobian matrix of $\Phi(\cdot)$, $\dot{\mathbf{r}}(t)$ and $\dot{\mathbf{q}}(t)$ are the end-effector velocity and joint velocity, respectively, with the desired trajectory of end-effector $\mathbf{r}(t)$ of robot manipulators denoted as $\mathbf{r_d}(t)$.

### B. Problem formulation and generalized RMP scheme

It can be observed from the illustrations on the computer simulations or experiments based on redundant robot manipulators aided with the existing RMP schemes [38], [45]–[49], the profiles of position errors always fluctuate or sometimes diverge even with large gains on the feedback of the position error. It can be readily imaged that, for a high-precision task, the controlling of position errors plays an indispensable role in obtaining accurate solutions. However, the theoretical analyses on the position errors of the existing RMP schemes remain unexplored. Besides, to the best of the authors' knowledge, the bound constraints for handling joint physical limits are assumed to be convex in the existing RMP schemes, which may fail to describe the actual situations in robotics applications with the nonconvex mapping in bound constraints.

To lay a basis for further research, a generalized RMP scheme is presented as follows, which systematizes the existing RMP schemes [4], [38], [43], [45], [47] with removing the convex requirements on the bound constraint with compliance to the nonconvex situation:

$$\text{minimize} \quad \dot{\mathbf{q}}^\intercal\dot{\mathbf{q}}/2 + \kappa(\mathbf{q}-\mathbf{q}_0)^\intercal\dot{\mathbf{q}} \tag{3}$$

$$\text{subject to} \quad \dot{\mathbf{r}}_\mathbf{d} = J\dot{\mathbf{q}} \tag{4}$$

$$\dot{\mathbf{q}} \in \omega, \tag{5}$$

where $\mathbf{q_0}$ denotes the desired initial state of joint angles; $^\intercal$ stands for the transpose of a vector or a matrix; $\kappa > 0$ represents a coefficient; $\omega$ is a set denoting the feasible region of $\dot{\mathbf{q}}$ in joint space, which is often designed as $\omega = \{x \in \mathbb{R}^n, \beta^- \leq x \leq \beta^+\}$ in the existing works [48], [50], [51], [54], [55], with $\beta^-$ and $\beta^+$ denoting the lower and upper bounds of the set, respectively. Besides, the position error can be described as $\epsilon = \mathbf{r} - \mathbf{r_d}$ similar to the joint drift error $\mathbf{e_q} = \mathbf{q} - \mathbf{q}_0$, which is able to demonstrate the accuracy of the repetitive motion in a different perspective.

### C. Dynamic neural networks

Different from the conventional approaches based on the Lagrangian multiplier method or Karush-Kuhn-Tucker conditions, we exploit a gradient descent method with the velocity compensation to design the dynamic neural networks in this part. An error function is constructed as follows to initialize the design of dynamic neural networks:

$$e = ||\mathbf{r_d} - \mathbf{r}||_2^2/2. \tag{6}$$

Employing the gradient descent method to find the evolution direction for minimizing the error function leads to

$$\dot{\mathbf{q}} = -\gamma\frac{\partial(e)}{\partial\mathbf{q}} = \gamma J^\intercal(\mathbf{r_d} - \mathbf{r}). \tag{7}$$

where $\gamma$> 0. Then, imposing the joint limits in (5) to the above equation results in

$$\dot{\mathbf{q}} = P_\omega(\gamma J^\intercal(\mathbf{r_d} - \mathbf{r})), \tag{8}$$

where $P_\omega(\cdot)$, defined as $P_\omega(x) = \text{argmin}_{y\in\omega}||y - x||$, which refers to [52], [53], is used to denote a projection operation to the set $\omega$. It is evident that the above equation lacks the velocity compensation, thereby leading to large lagging errors. To handle this issue, a new item $\mathbf{b}$ is added into (8):

$$\dot{\mathbf{q}} = P_\omega(\gamma J^\intercal(\mathbf{r_d} - \mathbf{r}) + \mathbf{b}), \tag{9}$$

where $\mathbf{b} \in \mathbb{R}^n$ is designed as a compensation for the joint velocity. In addition, $\mathbf{b}$ is satisfied with equation (2), i. e., $\dot{\mathbf{r}}_\mathbf{d} = J\mathbf{b}$. In this sense, we set $\mathbf{b}$ to be a parameter which can satisfy the minimum consumption of kinematic energy, thus resulting in $\mathbf{b} = J^\dagger\dot{\mathbf{r}}_\mathbf{d}$. In this equation, $J^\dagger$ is the pseudo

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TII.2019.2899909, IEEE Transactions on Industrial Informatics



inverse matrix of $J$, which is formulated as $J^\dagger = J^\intercal(JJ^\intercal)^{-1}$. A recursive approach is designed as follows:

$$\begin{aligned}\nu\dot{\mathbf{w}} &= \dot{\mathbf{r}}_\mathbf{d} - JJ^\intercal\mathbf{w},\\ \mathbf{b} &= J^\intercal\mathbf{w} - \kappa(\mathbf{q}-\mathbf{q}_0),\end{aligned} \tag{10}$$

where $\nu>0$, $\mathbf{w}$ is an auxiliary variable and $\kappa(\mathbf{q}-\mathbf{q}_0)$ is used to amend the obtained solution satisfying the requirements of the repetitive motion. Combining (9) and (10) generates

$$\dot{\mathbf{q}} = P_\omega(\gamma J^\intercal(\mathbf{r}_\mathbf{d}-\mathbf{r}) + J^\intercal\mathbf{w} - \kappa(\mathbf{q}-\mathbf{q}_0)), \tag{11}$$

$$\nu\dot{\mathbf{w}} = \dot{\mathbf{r}}_\mathbf{d} - JJ^\intercal\mathbf{w}. \tag{12}$$

We have the following remarks on the proposed dynamic neural networks (11) and (12).

*Remark 1:* The proposed dynamic neural networks (11) and (12) are designed to solve the generalized RMP scheme (3) through (5), which can be divided into three parts: a position error feedback part, made up of $\gamma J^\intercal(\mathbf{r}_\mathbf{d}-\mathbf{r})$, to control the difference between the desired trajectory $\mathbf{r}_\mathbf{d}$ and the actual path $\mathbf{r}$; a joint angle error feedback part, made up of $\kappa(\mathbf{q}-\mathbf{q}_0)$, to regulate the joint drift between the desired initial state $\mathbf{q}_0$ and the actual joint $\mathbf{q}$; a dynamic compensation for the joint velocity, made up of $J^\intercal\mathbf{w}$, to eliminate the lagging errors.

*Remark 2:* As investigated in [6], [47], the motion generation of mobile robots or multiple robots can be modeled as a quadratic programming problem. The solution in this paper is to solve the constrained quadratic programming problem. Therefore, the proposed control law is able to handle the motion generation of mobile robots or multiple robots with equality constraints and inequality constraints.

In addition, as reported in [56], [57], the obstacle avoidance in the motion control of robots can be modeled as an inequality constraint and further converted into a bound constraint by setting infinity as a bound and finally can be solved by quadratic programming. It is worth pointing out that the dynamic neural networks proposed in this paper are able to solve constrained quadratic programming problems. In this sense, this control method can be further extended to obstacle avoidance repetitive motion planning.

### D. Theoretical Analyses

For the controller proposed in this paper, i.e., dynamic neural networks (11) and (12), we offer the following theorem.

*Theorem:* The position error $\epsilon = \mathbf{r}-\mathbf{r}_\mathbf{d}$ synthesized by dynamic neural networks (11) and (12) for a redundant robot manipulator tracking a given closed path globally converges to $-(J^\intercal)^\dagger\kappa(\mathbf{q}-\mathbf{q}_0)/\gamma$, on the condition that $J^\intercal\mathbf{w}\in \mathrm{int}(\omega)$, with $\mathrm{int}(\omega)$ denoting the interior of $\omega$.

*proof:* The proof can be divided into two steps: the first step is to prove the stability of $\mathbf{w}$, which is involved in the implementation of the neural networks and stores the recursive state; the second step is to prove the convergence of $\epsilon$.

*The first step*: The stability of $\mathbf{w}$.

For the dynamic system depicted in (12), $\mathbf{w}$ is the decision variable. Given that $\nu>0$, the stability of $\mathbf{w}$ can be readily proven on the condition that $JJ^\intercal\succeq 0$. The singular value decomposition (SVD) method is used to prove $JJ^\intercal\succeq 0$, and we have

$$J = SAP, \tag{13}$$

where $J\in\mathbb{R}^{m\times n}$; $S\in\mathbb{R}^{m\times m}$; $A=[A_0 \quad B]\in\mathbb{R}^{m\times n}$ with $A_0\in\mathbb{R}^{m\times m}$ a diagonal matrix, and $B\in R^{m\times(n-m)}$ a null matrix; $P\in\mathbb{R}^{n\times n}$; $n>m$; $S^\intercal S=I$ and $P^\intercal P=I$. Then, $J^\intercal = P^\intercal A^\intercal S^\intercal$. We can get

$$JJ^\intercal = SAPP^\intercal A^\intercal S^\intercal = SAA^\intercal S^\intercal = S(A_0^2)S^\intercal \succeq 0. \tag{14}$$

From (14), we come to the conclusion that $\dot{\mathbf{w}}$ globally and exponentially converges to zero and $\mathbf{w}$ converges to $(JJ^\intercal)^{-1}\dot{\mathbf{r}}_\mathbf{d}$.

*The second step*: The convergence of position error $\epsilon$.

In view of the conclusion that $\mathbf{w}$ globally converges to $(JJ^\intercal)^{-1}\dot{\mathbf{r}}_\mathbf{d}$, employing LaSalle's invariance principle [58], we have $\mathbf{w}=(JJ^\intercal)^{-1}\dot{\mathbf{r}}_\mathbf{d}$. Thus, dynamical system (11) can be rewritten as

$$\begin{aligned}\dot{\mathbf{q}} =& P_\omega(\gamma J^\intercal(\mathbf{r}_\mathbf{d}-\mathbf{r}) + J^\intercal(JJ^\intercal)^{-1}\dot{\mathbf{r}}_\mathbf{d} - \kappa(\mathbf{q}-\mathbf{q}_0))\\ =& P_\omega(\gamma J^\intercal(\mathbf{r}_\mathbf{d}-\mathbf{r}) + J^\dagger\dot{\mathbf{r}}_\mathbf{d} - \kappa(\mathbf{q}-\mathbf{q}_0)).\end{aligned} \tag{15}$$

By combing (15) with $\dot{\mathbf{r}} = J\dot{\mathbf{q}}$, position error $\epsilon=\mathbf{r}-\mathbf{r}_\mathbf{d}$ can be transformed as

$$\dot{\epsilon} = JP_\omega(-\gamma J^\intercal\epsilon + J^\dagger\dot{\mathbf{r}}_\mathbf{d} - \kappa(\mathbf{q}-\mathbf{q}_0)) - \dot{\mathbf{r}}_\mathbf{d}. \tag{16}$$

In addition, noting that $\dot{\mathbf{r}}_\mathbf{d} = JJ^\dagger\dot{\mathbf{r}}_\mathbf{d}$, we have

$$\dot{\epsilon} = J(P_\omega(-\gamma J^\intercal\epsilon + J^\dagger\dot{\mathbf{r}}_\mathbf{d} - \kappa(\mathbf{q}-\mathbf{q}_0)) - J^\dagger\dot{\mathbf{r}}_\mathbf{d}). \tag{17}$$

Designing a Lyapunov function $V=\epsilon^\intercal\epsilon/2$ and calculating its time derivative leads to

$$\begin{aligned}\dot{V} =& \epsilon^\intercal\dot{\epsilon} = \epsilon^\intercal J(P_\omega(-\gamma J^\intercal\epsilon + J^\dagger\dot{\mathbf{r}}_\mathbf{d} - \kappa(\mathbf{q}-\mathbf{q}_0))\\ &-J^\dagger\dot{\mathbf{r}}_\mathbf{d}) = -\frac{1}{\gamma}((-\gamma J^\intercal\epsilon + J^\dagger\dot{\mathbf{r}}_\mathbf{d}) - J^\dagger\dot{\mathbf{r}}_\mathbf{d})^\intercal\\ &\times(P_\omega(-\gamma J^\intercal\epsilon + J^\dagger\dot{\mathbf{r}}_\mathbf{d} - \kappa(\mathbf{q}-\mathbf{q}_0)) - J^\dagger\dot{\mathbf{r}}_\mathbf{d}).\end{aligned} \tag{18}$$

Taking nonlinear mapping operation $P_\omega(x) = \mathrm{argmin}_{y\in\omega}||y-x||$ into consideration, we can naturally come to that $||P_\omega(x)-x||\leq||y-x||, \forall y\in\omega$. Designing $x=-\gamma J^\intercal\epsilon + J^\dagger\dot{\mathbf{r}}_\mathbf{d} - \kappa(\mathbf{q}-\mathbf{q}_0)$ and $y=J^\dagger\dot{\mathbf{r}}_\mathbf{d}$ yields

$$\begin{aligned}&||-\gamma J^\intercal\epsilon + J^\dagger\dot{\mathbf{r}}_\mathbf{d} - \kappa(\mathbf{q}-\mathbf{q}_0) - P_\omega(-\gamma J^\intercal\epsilon + J^\dagger\dot{\mathbf{r}}_\mathbf{d} -\\ &\kappa(\mathbf{q}-\mathbf{q}_0))||^2 \leq ||-\gamma J^\intercal\epsilon + J^\dagger\dot{\mathbf{r}}_\mathbf{d} - \kappa(\mathbf{q}-\mathbf{q}_0) - J^\dagger\dot{\mathbf{r}}_\mathbf{d}||^2.\end{aligned} \tag{19}$$

Expanding the left-hand side of (19), we obtain

$$\begin{aligned}&||-\gamma J^\intercal\epsilon + J^\dagger\dot{\mathbf{r}}_\mathbf{d} - \kappa(\mathbf{q}-\mathbf{q}_0) - P_\omega(-\gamma J^\intercal\epsilon + J^\dagger\dot{\mathbf{r}}_\mathbf{d}\\ &-\kappa(\mathbf{q}-\mathbf{q}_0))||^2 = ||-\gamma J^\intercal\epsilon + J^\dagger\dot{\mathbf{r}}_\mathbf{d} - \kappa(\mathbf{q}-\mathbf{q}_0)\\ &-J^\dagger\dot{\mathbf{r}}_\mathbf{d}||^2 + ||P_\omega(-\gamma J^\intercal\epsilon + J^\dagger\dot{\mathbf{r}}_\mathbf{d} - \kappa(\mathbf{q}-\mathbf{q}_0)) - J^\dagger\dot{\mathbf{r}}_\mathbf{d}||^2\\ &-2((-\gamma J^\intercal\epsilon + J^\dagger\dot{\mathbf{r}}_\mathbf{d} - \kappa(\mathbf{q}-\mathbf{q}_0)) - J^\dagger\dot{\mathbf{r}}_\mathbf{d})^\intercal\\ &(P_\omega(-\gamma J^\intercal\epsilon + J^\dagger\dot{\mathbf{r}}_\mathbf{d} - \kappa(\mathbf{q}-\mathbf{q}_0)) - J^\dagger\dot{\mathbf{r}}_\mathbf{d}).\end{aligned} \tag{20}$$

Combining with (19) and (20) results in

$$\begin{aligned}&||P_\omega(-\gamma J^\intercal\epsilon + J^\dagger\dot{\mathbf{r}}_\mathbf{d} - \kappa(\mathbf{q}-\mathbf{q}_0)) - J^\dagger\dot{\mathbf{r}}_\mathbf{d}||^2\\ \leq& 2((-\gamma J^\intercal\epsilon + J^\dagger\dot{\mathbf{r}}_\mathbf{d} - \kappa(\mathbf{q}-\mathbf{q}_0)) - J^\dagger\dot{\mathbf{r}}_\mathbf{d})^\intercal\\ &\times(P_\omega(-\gamma J^\intercal\epsilon + J^\dagger\dot{\mathbf{r}}_\mathbf{d} - \kappa(\mathbf{q}-\mathbf{q}_0)) - J^\dagger\dot{\mathbf{r}}_\mathbf{d}),\end{aligned} \tag{21}$$

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TII.2019.2899909, IEEE Transactions on Industrial Informatics



which is related to the previous formula (18). We find

$$\dot{V}+\frac{\kappa}{\gamma}(\mathbf{q}-\mathbf{q}_0)^\intercal(P_\omega(-\gamma J^\intercal\epsilon+J^\dagger\dot{\mathbf{r}}_\mathbf{d}-\kappa(\mathbf{q}-\mathbf{q}_0)) \\ -J^\dagger\dot{\mathbf{r}}_\mathbf{d})\leq-\frac{1}{2\gamma}||P_\omega(-\gamma J^\intercal\epsilon+J^\dagger\dot{\mathbf{r}}_\mathbf{d}-\kappa(\mathbf{q}-\mathbf{q}_0)) \\ -J^\dagger\dot{\mathbf{r}}_\mathbf{d}||^2\leq 0. \quad (22)$$

Simply letting $\epsilon=-(J^\intercal)^\dagger\kappa(\mathbf{q}-\mathbf{q}_0)/\gamma$ and considering $J^\intercal\mathbf{w}\in\text{int}(\omega)$, we have

$$P_\omega(-\gamma J^\intercal\epsilon+J^\dagger\dot{\mathbf{r}}_\mathbf{d}-\kappa(\mathbf{q}-\mathbf{q}_0))-J^\dagger\dot{\mathbf{r}}_\mathbf{d}=0. \quad (23)$$

Putting (22) and (23) together obtains $\dot{V}\leq 0$. Via LaSalle's invariance principle in [58], and selecting $\dot{V}=0$ in (18), we have the following two cases: Case I):

$$P_\omega(-\gamma J^\intercal\epsilon+J^\dagger\dot{\mathbf{r}}_\mathbf{d}-\kappa(\mathbf{q}-\mathbf{q}_0))=J^\dagger\dot{\mathbf{r}}_\mathbf{d}. \quad (24)$$

and Case II):

$$(-\gamma J^\intercal\epsilon+J^\dagger\dot{\mathbf{r}}_\mathbf{d})-J^\dagger\dot{\mathbf{r}}_\mathbf{d}=0.$$

Regarding case I, the following result can be derived by considering $J^\dagger\dot{\mathbf{r}}_\mathbf{d}\in\text{int}(\omega)$ and (24):

$$\gamma J^\intercal\epsilon+\kappa(\mathbf{q}-\mathbf{q}_0)=0.$$

Arranging the error $\epsilon$ to one side of the equation, we have

$$\epsilon=-(J^\intercal)^\dagger\frac{\kappa}{\gamma}(\mathbf{q}-\mathbf{q}_0). \quad (25)$$

For case II, we directly have $\epsilon=0$.

Considering that, at the time instant $t=T$ with $T$ denoting the task execution time for a closed path, $\mathbf{q}\rightarrow\mathbf{q}_0$, of which the detailed proof can be referred to [38]. Then, it can be readily obtained that, for case I, $\epsilon\rightarrow 0$. Therefore, the result $\epsilon=0$ can be cast into (25). In this sense, we come to the conclusion that $\epsilon=-(J^\intercal)^\dagger\kappa(\mathbf{q}-\mathbf{q}_0)/\gamma$.The proof is thus complete.

From (25), we would obtain Remark 3.

*Remark 3:* Through the above derivation, the conclusion described in (25) can be applied to all states, whether in the transient state or in the steady state for the motion generation and control of redundant robot manipulators synthesized by the existing RMP schemes.

In addition, we provide the following remark in regards to the error formulation depicted in (25).

*Remark 4:* It can be observed from (25) that the inversion operation of the transpose of the Jacobian matrix is involved, which is highly related to the manipulability of a robot [49]. Manipulability of a redundant robot manipulator is defined as

$$\mathcal{M}=\sqrt{\det(JJ^\mathrm{T})}=\sqrt{\mathcal{M}_1\mathcal{M}_2...\mathcal{M}_n} \quad (26)$$

where $\det(\cdot)$ denotes the determinant of a matrix; $\mathcal{M}_i\geq 0$ is the $i$th largest eigenvalue of $JJ^\mathrm{T}$ with $i=1,2,...,n$. For a redundant robot manipulator, we have $(J^\intercal)^\dagger=(J^\dagger)^\intercal$ and $J^\intercal(JJ^\intercal)^{-1}=J^\dagger$, and further have

$$(J^\dagger)^\intercal=(J^\intercal(JJ^\intercal)^{-1})^\intercal=((JJ^\intercal)^{-1})^\intercal J.$$

Then, we have

$$(J^\dagger)^\intercal(J^\dagger)=((JJ^\intercal)^{-1})^\intercal J(J^\dagger)=((JJ^\intercal)^{-1})^\intercal.$$

Given that $\det(((JJ^\intercal)^{-1})^\intercal)=\det((JJ^\intercal)^{-1})=1/\mathcal{M}^2$, and that $\det((J^\dagger)^\intercal)=\det(J^\dagger)$, we have

$$\det((J^\intercal)^\dagger)^2\mathcal{M}^2=1,$$

and finally have

$$\det((J^\intercal)^\dagger)\mathcal{M}=1.$$

The above analysis indicates that the position error depicted in (25) of a redundant robot manipulator synthesized by the generalized RMP scheme (3) through (5) is inversely proportional to its manipulability.

## III. Illustrative Examples

In this part, the presented generalized RMP scheme (3) through (5) is simulated based on a five-link planar robot manipulator and a PUMA 560 spatial robot manipulator with the aid of dynamic neural networks (11) and (12). The scheme can be employed to the repetitive motion generation of redundant robot manipulators in the situation that the initial position is not located on the desired path.

### *A. Five-link robot manipulator*

In this part, we simulate a triangle-path tracking task based on a five-link robot manipulator to verify the accuracy and the feasibility of the generalized RMP scheme (3) through (5) with the aid of dynamic neural networks (11) and (12) as well as the correctness of the corresponding theoretical analyses. In the simulations, task execution time $T=12$ s; the joint limit $-\beta^-=\beta^+=0.27$ rad/s is imposed to the scheme; $\gamma=1000$, $\kappa=0.42$, and $\nu=0.001$; initial angle state $\mathbf{q}_0=[\pi/2;-\pi/2;\pi/2;-\pi/2;-\pi/4]$ rad. It can be seen from Fig. 1(a) that the velocities of the end-effector start from zero and end at zero, which indicates that the robot stops accurately when the given task is finished. In addition, Fig. 1(b) plots the profiles of joint angle, of which the initial state at $t=0$ s is identical to the finial state at $t=12$ s, indicating that the repetitive motion is achieved. Moreover, as visualized in Fig. 1(c), the joint limit constraint is activated at around $t=2$ s, and the finial joint velocities all converge to zero. Besides, It can be observed from Fig. 1(d) that the position error $\epsilon$ achieves its maximal value at $t=4$ s. At the same time, as shown in Fig. 1(e), the manipulability achieves its minimal value. By comparatively observing Fig. 1(d) and Fig. 1(e), it is evident that the position error $\epsilon$ is generally inversely proportional to the manipulability, which is coincides with the theoretical analyses in Remark 4. The joint velocity error in Fig. 1(f) is less than $6\times10^{-3}$ m/s, which demonstrates the stability of the system. Fig. 1(g) illustrates that the simulated trajectory is quite closed to the desired path, which confirms the feasibility of the generalized RMP scheme (3) through (5) with the aid of dynamic neural networks (11) and (12). Fig. 1(h) shows the motion trajectories of each joint. Based on the above analyses, we come to the conclusion that the given task is fulfilled well with additional constraints and requirements satisfied.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TII.2019.2899909, IEEE Transactions on Industrial Informatics



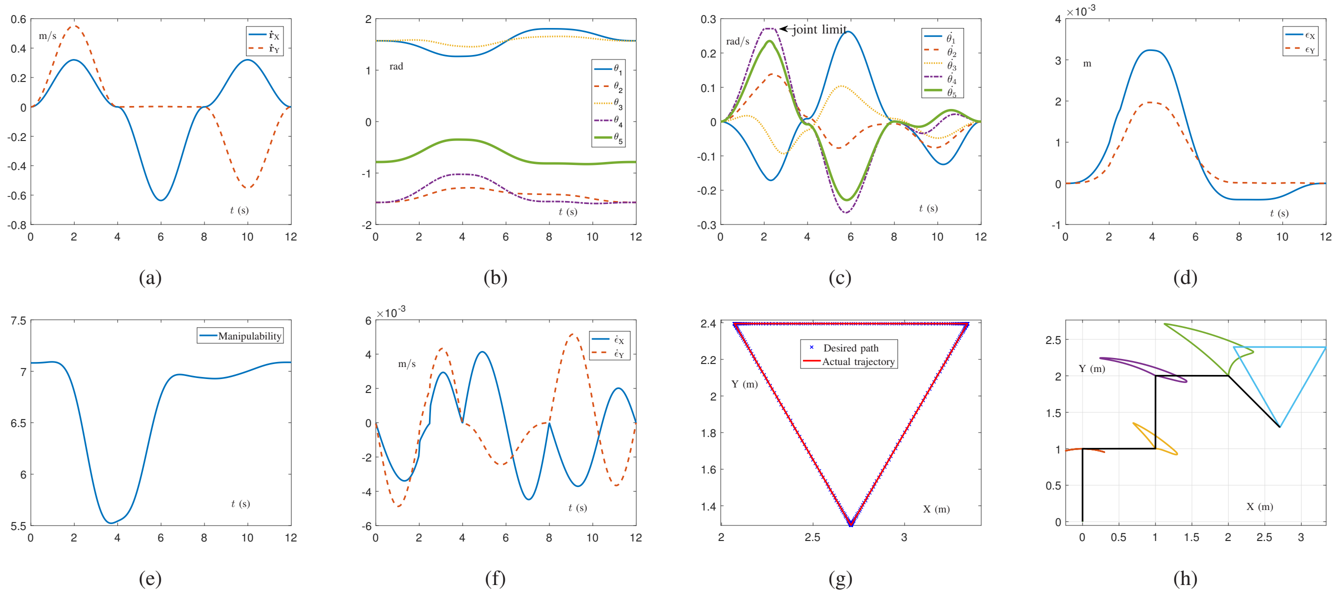


Fig. 1. Simulation results based on a five-link planar robot manipulator and synthesized by the generalized RMP scheme (3) through (5) with the aid of dynamic neural networks (11) and (12) for tracking a triangle path with execution time $T = 12$ s. (a) Profiles of end-effector velocity $\dot{\mathbf{r}}$. (b) Profiles of the joint angle. (c) Profiles of the joint velocity $\dot{\theta}$ with joint limits $\dot{\theta}^{+} = 0.27$ rad/s and $\dot{\theta}^{-} = -0.27$ rad/s. (d) Profiles of the position error. (e) Profiles of the manipulability measurement. (f) Profiles of the end-effector velocity error. (g) Profiles of the desired path and the actual trajectory. (h) Profiles of joint trajectories.

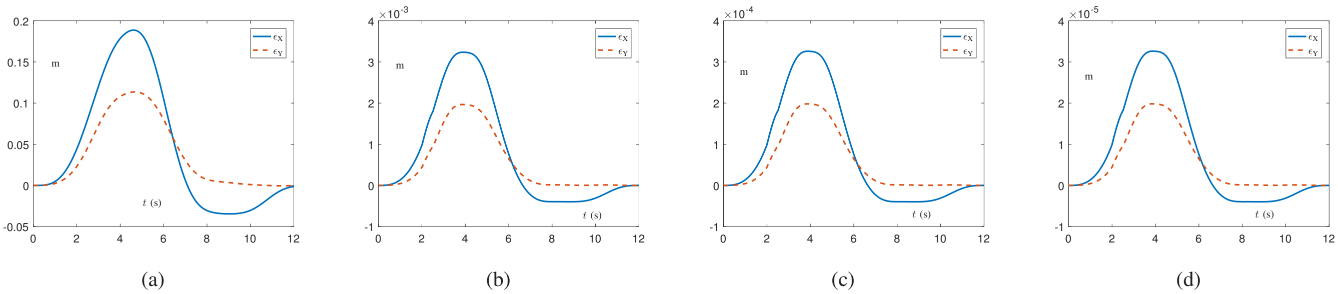


Fig. 2. Position errors based on a five-link planar robot manipulator and synthesized by the generalized RMP scheme (3) through (5) with the aid of dynamic neural networks (11) and (12) for tracking a triangle path with execution time $T = 12$ s, with $\kappa = 0.42$ and different $\gamma$. (a) $\gamma = 1$. (b) $\gamma = 100$. (c) $\gamma = 1000$. (d) $\gamma = 10000$.

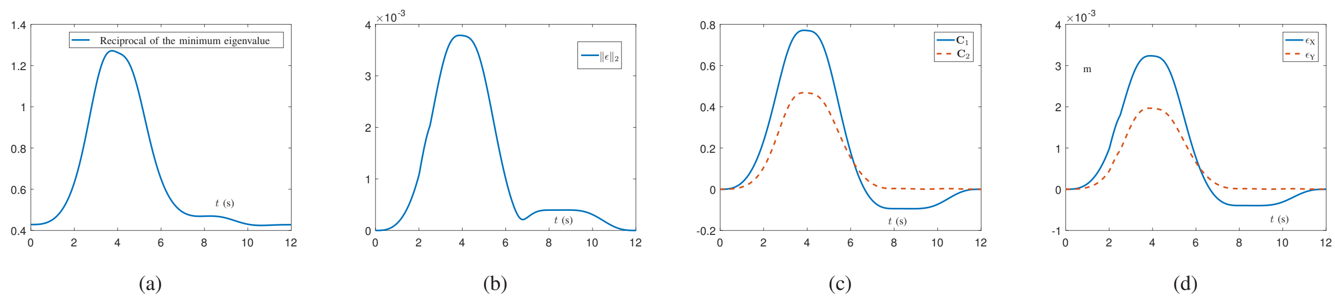


Fig. 3. Simulation results based on a five-link planar robot manipulator and synthesized by the generalized RMP scheme (3) through (5) with the aid of dynamic neural networks (11) and (12) for tracking a triangle path with execution time $T = 12$ s, $\gamma = 100$, and $\kappa = 0.42$. (a) Profiles of the reciprocal of the minimum eigenvalue of $JJ^{\mathsf{T}}$. (b) Profiles of the two norm of the position error. (c) Profiles of $\mathbf{C} = -(J^{\mathsf{T}})^{\dagger}(\mathbf{q} - \mathbf{q}_0)$ varying with time. (d) Profiles of the end-effector position error.

### *B. Verifications on theoretical analyses*

In this section, computer simulations are conducted to verify the correctness of the results depicted in (25) with different $\gamma$ with the rest parameters the same as those in Fig. 1. Specifically, the residual errors based on different values of $\gamma$ are plotted in Fig. 2, which clearly show that, at $t = 4$ s, position error $\epsilon$ in each subfigure in Fig. 1 achieves its maximal value, which verifies the correctness of the theoretical analyses conducted in this paper. In addition, we are able to get the conclusion that, the position error is inversely proportional to $\gamma$. In this sense, we can change the parameter $\gamma$ repeatedly to adjust the position error as (25) derives. However, it can

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TII.2019.2899909, IEEE Transactions on Industrial Informatics



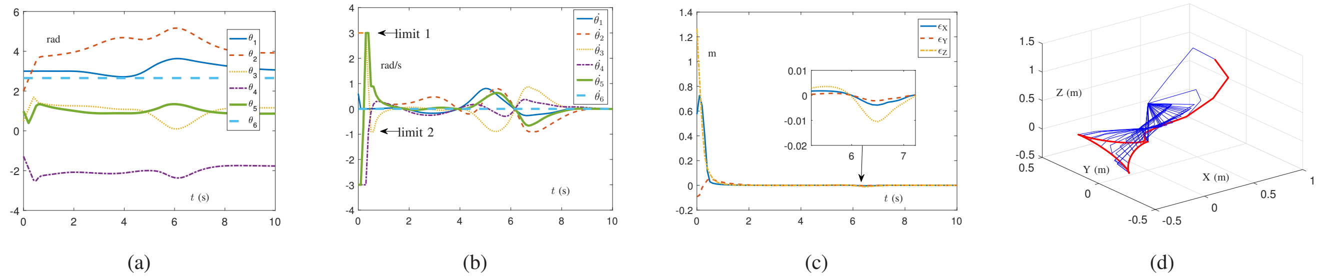


Fig. 4. Simulation results based on a PUMA 560 manipulator and synthesized by the generalized RMP scheme (3) through (5) with the aid of dynamic neural networks (11) and (12) for tracking a tricuspid valve path with execution time $T = 10$ s and undesired initial position. (a) Profiles of the joint angle. (b) Profiles of the joint velocity with limit 1 being 3 rad/s, limit 2 being 0.9 rad/s. (c) Profiles of the position error. (d) Profiles of the motion trajectory.

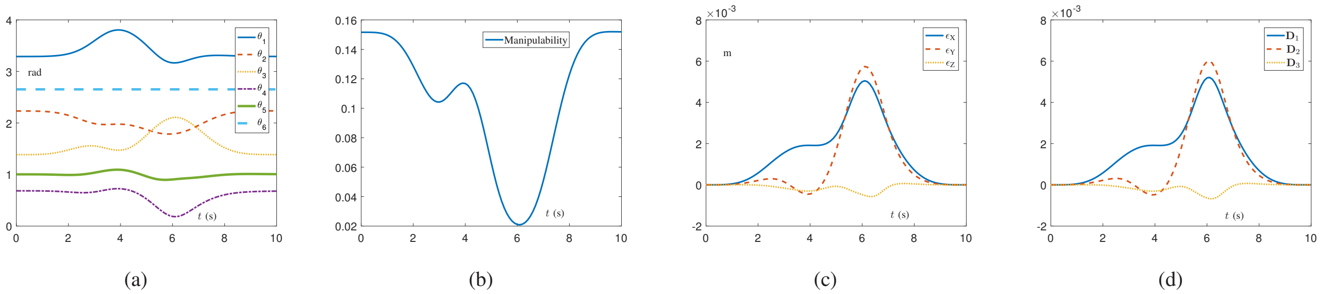


Fig. 5. Simulation results based on a PUMA 560 manipulator and synthesized by the generalized RMP scheme (3) through (5) with the aid of dynamic neural networks (11) and (12) for tracking a tricuspid valve path with execution time $T = 10$ s. (a) Profiles of the joint angle. (b) Profiles of the manipulability measurement. (c) Profiles of the position error. (d) Profiles of $\mathbf{D} = -(J^{\top})^{\dagger}\kappa(\mathbf{q} - \mathbf{q}_0)/\gamma$ varying with time.

be observed from Fig. 2(a) that, the maximal value of the residual error and its corresponding time instant are different from those in the rest subfigures. A detailed explanation can be provided as follows. With the maximal position error exceeding a certain value, the task execution can be deemed as a failure due to the fact that a large position error indicates a distinct deviation of the actual trajectory from the desired path. Then, the value of $(J^{\top})^{\dagger}$ changes greatly. Via (25), it can be highly expected that the profiles of the position error are different from the original ones.

To further verify the correctness of the theoretical analyses conducted in this paper, Fig. 3 is provided. It can be generalized from Remark 4 that the reciprocal of the minimum eigenvalue of $JJ^{\top}$ is in direct proportion to the determinant of $(J^{\top})^{\dagger}$. Thus, the profiles of the reciprocal of the minimum eigenvalue of $JJ^{\top}$ are plotted in Fig. 3(a), which are quite similar to the two norm of the position error shown in Fig. 3(b). In addition, the curve of $-(J^{\top})^{\dagger}(\mathbf{q}-\mathbf{q}_0)$ is drawn in Fig. 3(c), which is similar or even same to the position error shown in Fig. 3(d). Moreover, the maximal value of the profile in Fig. 3(c) is 0.0042 times to that in Fig. 3(d), which is exactly the same as the theory result obtained in (25). In other words, the theoretical results derived in this paper is completely verified by the simulation results.

*Remark 5:* We naturally conclude from the theoretical analyses and the computer simulations in this paper that, during the whole process of the repetitive motion, the joint drift error can not be eliminated completely with the existing RMP schemes. Theoretically, the position error can be regulated to converge to zero by adjusting the relevant parameters. However, as suggested by (25), the position error cannot be directly eliminated in the existing RMP schemes.

### *C. PUMA 560 manipulator with nonconvex projection*

In the existing RMP schemes, $\omega$ in (5) is required to be convex. To break this limitation, in this part, $\omega$ is designed as a nonconvex set:

$$\omega = \{x = [x_i] \in \mathbb{R}^6, \beta^- < x_i < \beta^+, \quad \text{or} \quad x_i = \pm\alpha\} \tag{27}$$

where $\beta^-$=-0.9 rad/s, $\beta^+$=0.9 rad/s and $\alpha$=3 rad/s. In what follows, $\alpha$ and $\beta$ are termed limit 1 and limit 2, respectively. It is worth noting that $\omega$ is nonconvex because 0 rad/s $\in \omega$ and 3 rad/s $\in \omega$ while (0+3)/2 rad/s $\notin \omega$. As for the other parameters, we design $\gamma = 1000$, $\kappa = 0.8$ and $\nu = 0.0001$. Fig. 4 illustrates the simulation results based on a PUMA 560 manipulator for tracking a tricuspid valve path, of which the initial state is not on the desired path. As demonstrated in Fig. 4(b), at the beginning, joint velocities maintain at limit 1 to find the right position. After about 0.4 s, the limit 2 is imposed to constrain the joint velocity. These results indicate that the nonconvex set $\omega$ works. In addition, profiles of joint angles are plotted in Fig. 4(a). Moreover, as shown in Fig. 4(c), starting from a large value, position errors converge to near zero within 0.4 s. Finally, the motion trajectories of the whole process are shown in Fig. 4(d). These results demonstrate the effectiveness of the generalized RMP scheme (3) through (5) with the aid of dynamic neural networks (11) and (12) with nonconvex cases.

To further show the effectiveness of the generalized RMP scheme (3) through (5) with the aid of dynamic neural networks (11) and (12), simulations are conducted with the

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TII.2019.2899909, IEEE Transactions on Industrial Informatics



TABLE I
COMPARISONS AMONG DIFFERENT CONTROLLERS FOR MOTION GENERATION AND CONTROL OF REDUNDANT ROBOT MANIPULATORS

| | Nonconvex $\omega$ | Repetitive movement | Original position* | Number of neurons | Relationship constructed** | Error accumulation elimination | Regulation error*** | Error boundary derivation |
|---|---|---|---|---|---|---|---|---|
| Controller (11) and (12) | ✓ | ✓ | any | n+m | ✓ | ✓ | Zero | ✓ |
| Controller in [7] | × | × | limited | n+m | × | × | Non-zero | × |
| Controller in [18] | ✓ | × | any | n+m | × | ✓ | Zero | × |
| Controller in [38] | × | ✓ | limited | n+m | × | × | Non-zero | × |
| Controller in [43] | × | ✓ | limited | n+m | × | × | Non-zero | × |
| Controller in [48] | × | × | limited | n | × | × | Non-zero | × |
| Controller in [50] | × | × | limited | 2n+m | × | × | Non-zero | × |
| Controller in [51] | × | × | limited | n+m | × | × | Non-zero | × |
| Controller in [54] | × | × | limited | n+m | × | × | Non-zero | × |

*Controllers in [7], [38], [43], [48], [50], [51], [54] share a common limitation that the initial position is required to be on the desired trajectory.
** "Relationship constructed" means that whether the relationship between the position error and the manipulability is constructed.
*** "Regulation error" means whether the manipulator is able to return to the desired position when its end-effector is not at the desired position aided with the corresponding controller.

initial position of the end-effector on the desired path, with the parameter setting $\gamma = 500$, $\kappa = 1$, $\nu = 0.001$ and initial angle state $\mathbf{q}_0 = [3.3, 2.2, 1.4, 0.7, 1.0, 2.7]$ rad. It can be observed from Fig. 5(a) that the joint angles go back to their initial state accurately. Moreover, the manipulability shown in Fig. 5(b) is inversely proportional to the position error in Fig. 5(c) as the theoretical analyses prove. Moreover, as plotted in Fig. 5(c), the position error is of the order $10^{-3}$ m. In addition, the curve of $\epsilon = -(J^{\top})^{\dagger}\kappa(\mathbf{q} - \mathbf{q}_0)/\gamma$ is drawn in Fig. 5(d), which is similar or even same to the position error shown in Fig. 5(c). These results further demonstrate the correctness of the theoretical analyses as well as the effectiveness of the generalized RMP scheme (3) through (5) with the aid of dynamic neural networks (11) and (12).

## IV. COMPARISONS

In this section, comparisons are conducted among the existing controllers [7], [18], [38], [43], [48], [50], [51], [54] as well as the novel controller proposed in this paper, i.e., the generalized RMP scheme (3) through (5) with the aid of dynamic neural networks (11) and (12) for the motion generation and control of redundant robot manipulators, which are listed in table I. As shown in the Table, the controller (11) and (12) proposed in this paper and the controller in [18] are able to handle the nonconvex constraints in dealing with the joint limits and regulate the error from a large value to zero while the rest ones are not. In addition, the proposed controller is the first one constructing the relationship between the position error and the manipulability, which provides a rigorous proof. Besides, due to the different structure of the neural networks, the number of neurons they require is different, which leads to inequable amounts of computation.

## V. CONCLUSIONS

This paper has presented a generalized RMP scheme as a systematization of the existing RMP schemes for the repetitive motion planning of redundant robot manipulators with the convexity limitation on bound constraint broken. To this end, the gradient descent method has been exploited to construct novel dynamic neural networks with the velocity compensation for eliminating the lagging errors. Then, the feasibility and accuracy of neural networks have been analyzed theoretically. In addition, the relationship between the position error and the manipulability synthesized by the the generalized RMP scheme (3) through (5) with the aid of dynamic neural networks (11) and (12) has been derived and the formula describing the position error has been provided. Finally, computer simulations based on different redundant robot manipulators and comparisons based on different controllers have been conducted to verify the feasibility and effectiveness of the generalized RMP scheme (3) through (5) with the aid of dynamic neural networks (11) and (12). As a summary, the advantages of this paper are emphasized here. First, the work is quite different from the existing researches on RMP, which breaks the existing preoccupations that the existing RMP schemes based on quadratic programming do not have theoretical errors. Second, the joint constraint extended to nonconvex case is different from the existing RMP schemes, which describes reality better. Third, different from the conventional approaches based on the Lagrangian multiplier method or Karush-Kuhn-Tucker conditions, the gradient descent method with the velocity compensation is designed to construct the dynamic neural networks to solve constrained quadratic programming problems. In the end, one of our future research directions is to propose a new RMP scheme with position error theoretically eliminated.

## REFERENCES

[1] L. Xiao, B. Liao, S. Li, Z. Zhang, L. Ding, and L. Jin, "Design and analysis of FTZNN applied to real-time solution of nonstationary lyapunov equation and tracking control of wheeled mobile manipulator," *IEEE Trans. Ind. Informat.*, vol. 14, no. 1, pp. 98–105, Jan. 2018.
[2] L. Jin, S. Li, X. Lin, R. Lu, and B. Liao, "Cooperative motion generation in a distributed network of redundant robot manipulators with noises," *IEEE Trans. Syst., Man, Cybern., Syst.*, vol. 48, no. 10, pp. 1715–1724, Oct. 2018.
[3] Y. Huang, J. Na, X. Wu, and G. Gao, "Approximation-free control for vehicle active suspensions with hydraulic actuator," *IEEE Trans. IEEE Trans. Ind. Electron.*, vol. 65, no. 9, pp. 7258–7267, Sep. 2018.
[4] Y. Zhang, H. Wu, Z. Zhang, L. Xiao, and D. Guo, "Acceleration-level repetitive motion planning of redundant planar robots solved by a simplified LVI-based primal-dual neural network," *Robot. Comput. Integr. Manuf.*, vol. 29, no. 2, pp. 328–343, Apr. 2013.
[5] L. Jin, S. Li, H. M. La, X. Zhang, and B. Hu, "Dynamic task allocation in multi-robot coordination for moving target tracking: A distributed approach," *Automatica*, vol. 100, pp. 75–81, Jan. 2019.
[6] L. Jin, S. Li, X. Luo, Y. Li, and B. Qin, "Neural dynamics for cooperative control of redundant robot manipulators," *IEEE Trans. Ind. Informat.*, vol. 14, no. 9, pp. 3812–3821, Sep. 2018.

[7] D. Guo and Y. Zhang, "Simulation and experimental verification of weighted velocity and acceleration minimization for robotic redundancy resolution," *IEEE Trans. Autom. Sci. Eng.*, vol. 11, no. 4, pp. 1203–1217, Oct. 2014.

[8] C. Yang, K. Huang, H. Cheng, Y. Li, and C.-Y. Su, "Haptic identification by ELM-controlled uncertain manipulator," *IEEE Trans. Syst., Man, Cybern., Syst.*, vol. 47, no. 8, pp. 2398–2409, Aug. 2017.

[9] T. Yoshikawa, "Manipulability of robotic mechanisms," *Int. J. Robot Res.*, vol. 4, no. 2, pp. 3–9, Jun. 1985.

[10] C. Yang , Y. Jiang, W. He, J. Na, Z. Li, and B. Xu, "Adaptive parameter estimation and control design for robot manipulators with finite-time convergence," *IEEE Trans. Ind. Electron.*, vol. 65, no. 10, pp. 8112–8123, Oct. 2018.

[11] L. Jin, Y. Zhang, S. Li, and Y. Zhang, "Modified ZNN for time-varying quadratic programming with inherent tolerance to noises and its application to kinematic redundancy resolution of robot manipulators," *IEEE Trans. Ind. Electron.*, vol. 63, no. 11, pp. 6978–6988, Nov. 2016.

[12] C. Yang, G. Peng, Y. Li, R. Cui, L. Cheng, and Z. Li, "Neural networks enhanced adaptive admittance control of optimized robot-environment interaction," *IEEE Trans. Cyben.*, In Press with DOI 10.1109/TCYB.2018.2828654.

[13] C. Yang, C. Zeng, Y. Cong, N. Wang, and M. Wang, "A learning framework of adaptive manipulative skills from human to robot," *IEEE Trans. Ind. Informat.,* vol. 15, no. 2, pp. 1153–1161, Feb. 2019.

[14] C. Yang, H. Wu, Z. Li, W. He, N. Wang, and C.-Y. Su, "Mind control of a robotic manipulator with visual fusion technology," *IEEE Trans. Ind. Informat.,* vol. 14, no. 9, pp. 3822–3830, Sep. 2018.

[15] 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.

[16] A. A. Maciejewski, "Kinetic limitations on the use of redundancy in robotic manipulators," *IEEE Trans. Robot. Autom.*, vol. 7, no. 2, pp. 205–210, Apr. 1991.

[17] S. Ma, "A new formulation technique for local torque optimization of redundant manipulators," *IEEE Trans. Ind. Electron.*, vol. 43, no. 4, pp. 462–468, Aug. 1996.

[18] S. Li, Y. Zhang, and L. Jin, "Kinematic control of redundant manipulators using neural networks," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 28, no. 10, pp. 2243–2254, Oct. 2017.

[19] C. Yang, Y. Jiang, Z. Li, W. He, and C.-Y. Su, "Neural control of bimanual robots with guaranteed global stability and motion precision," *IEEE Trans. Ind. Informat.*, vol. 13, no. 3, pp. 1162–1171, Jun. 2017.

[20] L. Jin, S. Li, and B. Hu, "RNN models for dynamic matrix inversion: A control-theoretical perspective," *IEEE Trans. Ind. Informat.*, vol. 14, no. 1, pp. 98–105, Jan. 2018.

[21] L. Jin, S. Li, B. Hu, M. Liu, and J. Yu, "A noise-suppressing neural algorithm for solving the time-varying system of linear equations: A control-based approach," *IEEE Trans. Ind. Informat.*, vol. 15, no. 1, pp. 236–246, Jan. 2019.

[22] X. Luo, M. Zhou, Z. Wang, Y. Xia, and Q. Zhu, "An effective scheme for QoS estimation via alternating direction method-based matrix factorization," *IEEE Trans on Services Computing*, In Press with DOI: 10.1109/TSC.2016.2597829.

[23] D. Wu, X. Luo, G. Wang, M. Shang, Y. Yuan, and H. Yan, "A highly accurate framework for self-labeled semisupervised classification in industrial applications," *IEEE Trans. Ind. Informat.,* vol. 14, no. 3, pp. 909–920, Mar. 2018.

[24] L. Hu, X. Yuan, X. Liu, S. Xiong, and X. Luo, "Efficiently detecting protein complexes from protein interaction networks via alternating direction method of multipliers," *IEEE/ACM Trans. on Computational Biology and Bioinformatics*, In Press with DOI: 10.1109/TCBB.2018.2844256.

[25] X. Luo and M. Shang, "Symmetric non-negative latent factor models for undirected large networks," *In Proc. Int. Joint Conf. Artificial Intelligence,* Melbourne, Australia, 2017, pp. 2435-2442.

[26] L. Liu, Y.-J. Liu, and S. Tong, "Neural networks-based adaptive finite time fault-tolerant control for a class of strict-feedback switched nonlinear systems," *IEEE Trans. Cybern.,* In Press with DOI 10.1109/TCYB.2018.2828308.

[27] Y. Liu, S. Tong, C. L. P. Chen, and D. Li, "Neural controller design-based adaptive control for nonlinear MIMO systems with unknown hysteresis inputs," *IEEE Trans. Cybern.,* vol. 46, no. 1, pp. 9–19, Jan. 2016.

[28] Y.-J. Liu, S. Tong, D.-J. Li, and Y. Gao, "Fuzzy adaptive control with state observer for a class of nonlinear discrete-time systems with input constraint," *IEEE Trans. Fuzzy Syst.*, vol. 24, no. 5, pp. 1147–1158, Oct. 2016.

[29] L. Xiao, B. Liao, S. Li, and K. Chen, "Nonlinear recurrent neural networks for finite-time solution of general time-varying linear matrix equations," *Neural Networks*, vol. 98, pp. 102–113, Feb. 2018.

[30] P. S. Stanimirović, I. Živković, and Y. Wei, "Recurrent neural network approach based on the integral representation of the drazin inverse," *Neural Computation*, vol. 27, no. 10, pp. 2107-2131, Oct. 2015.

[31] P. S. Stanimirović, I. Živković, and Y. Wei, "Recurrent neural network for computing the drazin inverse," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 26, no. 11, pp. 2830–2843, Nov. 2015.

[32] I. Živković, P. S. Stanimirović, and Y. Wei, "Recurrent neural network for computing outer inverse," *Neural Computation*, vol. 28, no. 5, pp. 970–998, May 2016.

[33] Y. Zhang, S. Li, and X. Liu, "Neural network-based model-free adaptive near-optimal tracking control for a class of nonlinear systems," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 29, no. 12, pp. 6227–6241, Dec. 2018.

[34] L. Xiao, S. Li, F. Lin, Z. Tan, and A. H. Khan, "Zeroing neural dynamics for control design: comprehensive analysis on stability, robustness, and convergence speed," *IEEE Trans. Ind. Informat.*, In Press with DOI 10.1109/TII.2018.2867169.

[35] L. Jin, S. Li, H. M. La, and X. Luo, "Manipulability optimization of redundant manipulators using dynamic neural networks," *IEEE Trans. IEEE Trans. Ind. Electron.*, vol. 64, no. 6, pp. 4710–4720, Feb. 2017.

[36] Y. Zhang and S. Li, "A neural controller for image-based visual servoing of manipulators with physical constraints," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 29, no. 11, pp. 5419–5429, Nov. 2018.

[37] C. Yang, X. Wang, L. Cheng, and H. Ma, "Neural-learning-based telerobot control with guaranteed performance," *IEEE Trans. Cybern.*, vol. 47, no. 10, pp. 3148–3159, Oct. 2017.

[38] 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.

[39] S. Boyd and L. Vandenberghe, *Convex Optimization*. Cambridge, U.K.: Cambridge Univ. Press, 2004.

[40] 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.

[41] P. S. Stanimirović and M. D. Petković, "Gradient neural dynamics for solving matrix equations and their applications" *Neurocomputing*, vol. 306, pp. 200–212, Sep. 2018.

[42] 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.

[43] A. De Luca, L. Lanari, and G. Oriolo, "Control of redundant robots on cyclic trajectories," *IEEE Int. Conf. Robot. Autom.*, Nice, France, 1992, pp. 500–506.

[44] R. G. Roberts and A. A. Maciejewski, "Repeatable generalized inverse control strategies for kinematically redundant manipulators," *IEEE Trans. Autom. Control*, vol. 38, no. 5, pp. 689–698, May 1993.

[45] K. Tchon and J. Jakubiak, "A repeatable inverse kinematics algorithm with linear invariant subspaces for mobile manipulators," *IEEE Trans. Syst., Man, Cybern. B, Cybern.*, vol. 35, no. 5, pp. 1051–1057, Oct. 2005.

[46] Y. Zhang, X. Lv, Z. Li, Z. Yang, and K. Chen, "Repetitive motion planning of PA10 robot arm subject to joint physical limits and using LVI-based primal-dual neural network," *Mechatronics*, vol. 18, no. 9, pp. 475–485, Nov. 2008.

[47] 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.

[48] 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.

[49] L. Jin, S. Li, B. Hu, and C. Yi, "Dynamic neural networks aided distributed cooperative control of manipulators capable of different performance indices," *Neurocomputing*, vol. 291, pp. 50–58, May 2018.

[50] 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.

[51] 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.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TII.2019.2899909, IEEE Transactions on Industrial Informatics



[52] X. Luo, M. Zhou, Y. Xia, and Q. Zhu, "An efficient non-negative matrix-factorization-based approach to collaborative filtering for recommender systems," *IEEE Trans. Ind. Informat.,* vol. 10, no. 2, pp. 1273–1284, May. 2014.

[53] X. Luo, M. Zhou, H. Leung, Y. Xia, Q. Zhu, Z. You, and S. Li, "An incremental-and-static-combined scheme for matrix-factorization-based collaborative filtering," *IEEE Trans. Autom. Sci. Eng.*, vol. 13, no. 1, Jan. 2016.

[54] Y.S. 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.

[55] Y. Zhang and J. Wang, "Obstacle avoidance for kinematically redundant manipulators using a dual neural network," *IEEE Trans. Syst., Man, Cybern. B, Cybern.*, vol. 34, no. 1, pp. 752–759, Feb. 2004.

[56] D. Guo and Y. Zhang, "A new inequality-based obstacle-avoidance MVN scheme and its application to redundant robot manipulators," *IEEE Trans. Syst., Man, Cybern. C, Appl. Rev.*, vol. 42, no. 6, pp. 1326–1340, Nov. 2012.

[57] D. Guo and Y. Zhang, "Acceleration-level inequality-based MAN scheme for obstacle avoidance of redundant robot manipulators," *IEEE Trans. IEEE Trans. Ind. Electron.*, vol. 61, no. 12, pp. 6903–6914, Dec. 2014.

[58] H. K. Khalil, *Nonlinear Systems*, 3rd ed. Englewood Cliffs, NJ, USA: Prentice–Hall, 2001.

**Zhengtai Xie** is currently working toward the B.E. degree in Electronic Information Science and Technology from Lanzhou University, China, where he has obtained the postgraduate recommendation, and will be working toward the M. S Degree. His main research interests include robotics, neural networks, intelligent information processing, artificial intelligence and optimization theory.

**Long Jin** (M'17) received the B.E. degree in automation and the Ph.D. degree in Information and Communication Engineering from Sun Yat-sen University, Guangzhou, China, in 2011 and 2016, respectively. He is currently a full professor with the School of Information Science and Engineering, Lanzhou University, Lanzhou, China. Before joining Lanzhou University in 2017, he was a Postdoctoral Fellow with the Department of Computing, The Hong Kong Polytechnic University, Hung Hom, Kowloon, Hong Kong. His main research interests include neural networks, robotics, and intelligent information processing. He is a member of the IEEE.

**Xiujuan Du** received her M.S. degree in Radio Physics from Lanzhou University, and the Ph.D. degree in computer application technology from Tianjin University, Tianjin, China. She is currently a professor of the provincial key Laboratory of the Internet of things, Qinghai Normal University. She was awarded as a New Century Excellent Talent from education ministry, China in 2011. Her research interests cover network and information security, mobile ad hoc networks, and underwater sensor networks etc., including network modeling, network protocol design, performance evaluation, optimization algorithms, distributed computing and their applications.

**Xiuchun Xiao** received the Ph.D. degree in Communication and Information System in 2013 from Sun Yat-sen University, Guangzhou, China. He is currently an Associate Professor with Guangdong Ocean University. His current research interests include artificial neural network and computer vision.

**Hongxin Li** received the B.E. degree in Mechatronic Engineering in 1991 from Nanjing University of Science and Technology, the M.E. degree in Radio Electronics in 1998 from Lanzhou University, and the Ph.D. degree in Mechanical and Electronic Engineering from Shenyang Institute of Automation, Chinese Academy of Sciences, Shenyang, China. He is currently an Associate Professor with Lanzhou University. His current research interests include analysis of distributed parameter system, modeling and control, and networked control system.

**Shuai Li** (SM'17, M'14) received the B.E. degree in Precision Mechanical Engineering from Hefei University of Technology, China in 2005, the M.E. degree in Automatic Control Engineering from University of Science and Technology of China, China in 2008, and the Ph.D. degree in Electrical and Computer Engineering from Stevens Institute of Technology, USA in 2014. He is currently a research assistant professor with Department of Computing, The Hong Kong Polytechnic University, Hong Kong. His current research interests include dynamic neural networks, robotic networks, and other dynamic problems defined on a graph.