IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

Received February 3, 2021, accepted March 14, 2021, date of publication March 24, 2021, date of current version April 7, 2021.

Digital Object Identifier 10.1109/ACCESS.2021.3068731

A Varying-Parameter Recurrent Neural Network Combined With Penalty Function for Solving Constrained Multi-Criteria Optimization Scheme for Redundant Robot Manipulators

NAN ZHONG[1], QINGYU HUANG[1], SONG YANG[2], FAN OUYANG[1], AND ZHIJUN ZHANG[2], (Senior Member, IEEE)

[1]College of Engineering, South China Agricultural University, Guangzhou 510642, China
[2]School of Automation Science and Engineering, South China University of Technology, Guangzhou 510641, China

Corresponding author: Zhijun Zhang (auzjzhang@scut.edu.cn)

This work was supported in part by the National Key Research and Development Plan under Grant 2017YFD070100103 and in part by Science and Technology Plan of Guangzhou City under Grant 201903010063.

ABSTRACT To effectively solve the multi-objective motion planning problem for redundant robot manipulators, a penalty neural multi-criteria optimization (PNMCO) scheme is proposed and investigated. The scheme includes two parts: a constrained multi-criteria optimization (CMCO) subsystem, and a varying-parameter recurrent neural network combined with penalty function (VP-RNN-PF) subsystem. Specifically, the CMCO subsystem is made up of velocity two norm, repetitive motion, and infinity norm. With these criteria, it can achieve energy minimization, repetitive motion, and avoidance of speed peaks. In addition, the CMCO subsystem is then transformed into a standard quadratic programming (QP) problem, and the VP-RNN-PF subsystem is applied to solve the QP problem. Results of computer simulations based on the JACO^2 robot manipulator demonstrate that the proposed PNMCO scheme is effective and feasible to plan the multi-objective motion tasks. Comparison experiments of two complex paths tracking between VP-RNN-PF and the traditional neural networks (e.g., simplified linear-variational-inequality-based primal-dual neural network, S-LVI-PDNN) shows that the proposed scheme as well as the neural network is more accurate and more efficient for solving multi-objective motion planning problem.

INDEX TERMS Redundant robot manipulators, recurrent neural network (RNN), constrained multi-criteria optimization (CMCO), quadratic programming (QP), complex path tracking.

I. INTRODUCTION

Nowadays, robot technology has received widespread attention and robot manipulators have been widely applied in various fields, such as medical surgery [1], [2], domestic service [3] industrial production [4], [5]. If a robot manipulator has more DOFs than the minimum number of DOFs to complete the primary task of an end-effector, it is termed a redundant robot manipulator. The advantage of a redundant robot manipulator is that it has additional DOFs to complete other secondary subtasks in addition to the primary tasks of an end-effector. Some common secondary subtasks are joint limits avoidance [6], [7], repetitive motion [8], [9], energy minimum [10], [11], *et al.*

Since the number of joint variables for a redundant robot manipulator is larger than the number of robot kinematic equations, the solutions to the inverse kinematics problem are more than one. The traditional solution to the inverse kinematics problem was based on the pseudo-inverse method [12]. However, this method would take a lot of time to calculate the inverse of matrices. In addition, this method also cannot consider inequality problems (such as physical limit constraints) [13]. If these constraints are not considered, the moving joints may touch their physical limits, which may cause task failure, even robot damage. In recent years, the inverse kinematics problem is converted into an optimiza-

The associate editor coordinating the review of this manuscript and approving it for publication was Saeid Nahavandi.

tion problem, Cheng *et al.* transformed the redundant inverse kinematics problem into a quadratic programming (QP) problem, and considered the physical limits as the inequality constraint [14].

To solve the joint-angular deviation problem of redundant robot manipulators, optimization approaches were preferred by an increasing number of scholars, and many optimization criteria were designed in the past years. For instance, infinity-norm velocity minimization (INVM) [15], infinity-norm acceleration minimization (INAM) [16], minimum acceleration norm (MAN) [17], minimum kinetic energy (MKE) [18], minimum weighted velocity norm (MWVN) [19], repetitive motion planning (RMP) [8], [9], and minimum-velocity-norm (MVN) [11]. However, in the practical application of redundant robot manipulators, single-criterion optimization schemes cannot meet the demand. In order to solve the multi-criteria optimization problem in practical applications, some bi-criteria minimization schemes were presented [20], [21]. Zhang *et al.* presented a bi-criteria scheme that combined the MVN and INVM [20]. Guo and Zhang *et al.* studied bi-criteria minimization schemes at torque, velocity, and acceleration levels, such as the MAN-MKE bi-criteria minimization scheme [21] and MVN-MAN bi-criteria minimization scheme [22]. Guo and Zhang also unified and compared the bi-criteria torque, velocity, and acceleration schemes [23]. Tri-criteria minimization schemes were presented and exploited in recent years [24], [25]. These multi-criteria schemes are reformulated as QP problems and then these QP problems are solved.

The solutions to QP problems are divided into two types, i.e., numerical methods [26], [27] and neural networks [28]–[36]. Compared with numerical methods, neural network methods can process information in parallel and have higher computational efficiency [28]. Scholars designed many neural network methods in the past years. Leung *et al.* presented a new gradient-based neural network (GNN) method to solve QP problems [29]. However, This method could not track the theoretical solution of QP problems well [30]. Zhang *et al.* presented a zeroing neural network (ZNN) [31], ZNN could track the theoretical solution of QP problems. However, it tracked the theoretical optimal solution only when the time approaches to infinity [30]. Miao *et al.* and Zhang *et al.* [33] respectively presented a finite time ZNN (FT-ZNN) and a varying parameter RNN (VP-RNN) to improve convergence speed of ZNN [32], [33]. However, these solutions usually fail to consider the joint physical limits.

To solve the optimization schemes of redundant robot manipulators with joint physical limit constraints, a dual neural network (DNN) was presented to solve the corresponding QP problem with inequality constraints [34], but the structure of this DNN was too simple. A linear variational inequality-based primal-dual neural network (LVI-PDNN) was presented [35]. To avoid solving the inverse of the matrix, reduce the computational complexity, a simplified LVI-PDNN (S-LVI-PDNN) was presented and applied [36].

The multi-objective motion tasks of redundant robot manipulators with joint physical limits was usually solved by S-LVI-PDNN [8], [16]–[25]. However, the rate of S-LVI-PDNN could not always reach exponential convergence [37]. Based on the research of VP-RNN [38], a penalty neural multi-criteria optimization (PNMCO) scheme is proposed. The scheme can effectively and feasible plan the multi-objective motion tasks of redundant robot manipulators.

This paper consists of five sections. In Section II, a PNMCO scheme based on QP form for redundant robot manipulators is constructed. For comparisons, we give the traditional S-LVI-PDNN. In Section III, computer simulation results demonstrate the feasibility of the PNMCO scheme. Meanwhile, comparison experiments between VP-RNN-PF and S-LVI-PDNN verify that the proposed schemes more accurate and more efficient for solving the multi-objective motion planning problem. Section IV draws the conclusion of this paper. Specifically, the main contributions of this paper are summarized below.

- A novel PNMCO scheme is proposed to solve the multi-objective motion planning problem for redundant robot manipulators. The proposed PNMCO scheme includes two subsystems, i.e., VP-RNN-PF and CMCO.
- The PNMCO scheme considers both the joint velocity and joint angle limits, since this scheme is solved at the velocity level, the joint velocity and joint angle limits combine as the piecewise function is proposed in this paper.
- Computer simulations by two different path tracking task results demonstrate the effectiveness and feasibility of the proposed PNMCO scheme to solve multi-objective motion planning problem for the JACO^2 robot manipulator.
- Comparison experiments between the VP-RNN-PF and traditional S-LVI-PDNN indicate that the proposed PNMCO scheme is more accurate and more effective in solving the multi-objective motion planning problem for redundant robot manipulators.

II. CONSTRUCTION OF PNMCO SCHEME AND S-LVI-PDNN SOLVER

The forward kinematics of the redundant robot manipulators is to obtain the end effector r path through the joint angle θ change, and the equation is considered as

$$r = f(\theta) \tag{2.1}$$

Due to the $f(\cdot)$ is a differentiable nonlinear mapping function, the equation (2.1) is difficult to solve, then the above equation is rewritten to solve at the velocity level [8]–[25]

$$J(\theta)\dot{\theta} = \dot{r} \tag{2.2}$$

where $\dot{\theta} \in R^n$ and $\dot{r} \in R^m$ denote joint velocity vector and end-effector velocity vector, respectively. The $J(\theta) \in R^{m\times n}$ is Jacobian matrix of the robot manipulators. The n and m denote the DOFs of manipulators and the dimension of end-effector Cartesian space, respectively. In this paper, for the

six-DOF JACO^2 redundant robot manipulator used, the $n = 6$, $m = 3$.

A. CMCO SUBSYSTEM FORMULATION

The PNMCO scheme combines three different optimization criteria (i.e., MVN, RMP, and INVM), which can be expressed as

$$\text{min. } \eta\|\dot{\theta}\|_2^2/2 + \delta\|\dot{\theta} + c\|_2^2/2 + (1-\eta-\delta)\|\dot{\theta}\|_\infty^2/2 \tag{2.3}$$

$$\text{s.t. } J(\theta)\dot{\theta} = \dot{r} \tag{2.4}$$

where $\|\cdot\|_\infty$ denotes the infinite norm, $\|\cdot\|_2$ denotes the two-norm. η and δ are weighting parameters for assigning different optimization criteria (i.e.,MVN, RMP and INVM), c denotes $\tau(\theta - \theta(0))$, $\tau > 0$.

In practical applications, redundant robot manipulators are always subject to physical constraints, considering the joint velocity limits and joint angle limits of redundant robot manipulators, the PNMCO scheme can be formulated as

$$\text{min. } \eta\|\dot{\theta}\|_2^2/2 + \delta\|\dot{\theta} + c\|_2^2/2 + (1-\eta-\delta)\|\dot{\theta}\|_\infty^2/2 \tag{2.5}$$

$$\text{s.t. } J(\theta)\dot{\theta} = \dot{r} \tag{2.6}$$

$$\theta^- \leqslant \theta \leqslant \theta^+ \tag{2.7}$$

$$\dot{\theta}^- \leqslant \dot{\theta} \leqslant \dot{\theta}^+ \tag{2.8}$$

where $\theta^\pm$ denote the physical limits for joint angle vector and $\dot{\theta}^\pm$ denote the physical limits for joint velocity vector.

According to the mathematical formula, the MVN and RMP in equation (2.5) can be rewritten as

$$\eta\|\dot{\theta}\|_2^2/2 = \eta\dot{\theta}^{\mathrm{T}} I\dot{\theta}/2 \tag{2.9}$$

$$\delta\|\dot{\theta} + c\|_2^2/2 = \delta(\dot{\theta}^{\mathrm{T}} I\dot{\theta} + \Lambda\dot{\theta}) \tag{2.10}$$

where $I \in R^{n\times n}$ is an identity matrix, and $\Lambda = 2\tau$.

The INVM criterion $(1-\eta-\delta)\|\dot{\theta}\|_\infty^2/2$ in equation (2.5) can be formulated as

$$\text{min. } (1-\eta-\delta)k^2/2 \tag{2.11}$$

$$\text{s.t. } \begin{bmatrix} I & -I_v \\ -I & -I_v \end{bmatrix}\begin{bmatrix} \dot{\theta} \\ k \end{bmatrix} \leqslant \begin{bmatrix} 0 \\ 0 \end{bmatrix} \tag{2.12}$$

where $k = \|\dot{\theta}\|_\infty$, $I_v = [1, 1, \ldots, 1]^{\mathrm{T}} \in R^n$ is a vector of ones.

In order to solve the schemes (2.5)-(2.8) at the velocity level, constraints (2.7)-(2.8) need be unified as the constraint of velocity level, the joint limits (2.7) are rewritten as [10], [34]

$$\upsilon(\theta^- - \theta) \leq \dot{\theta} \leq \upsilon(\theta^+ - \theta) \tag{2.13}$$

$\upsilon > 0$, the function of this parameter is to scales the feasible regions of joint-velocity vector $\dot{\theta}$ in equation (2.13). The equation (2.13) can guarantee joint velocity accelerate to zero when the joint-angle approaches its limits. Therefore, the joint limits (2.7) and joint velocity limits (2.8) combined into a variable joint physical constraint as

$$\zeta^- = \begin{cases} \upsilon(\theta_i^- - \theta_i), & \text{if } \theta_i(t) \in [\theta_i^-, \xi_1] \\ \dot{\theta}_i^-, & \text{if } \theta_i(t) \in [\xi_1, \theta_i^+] \end{cases} \tag{2.14}$$

$$\zeta^+ = \begin{cases} \dot{\theta}_i^+, & \text{if } \theta_i(t) \in [\theta_i^-, \xi_2] \\ \upsilon(\theta_i^+ - \theta_i), & \text{if } \theta_i(t) \in [\xi_2, \theta_i^+] \end{cases} \tag{2.15}$$

where $\xi_1 = \theta_i^- - \dot{\theta}_i^-/\upsilon$, $\xi_2 = \theta_i^+ - \dot{\theta}_i^+/\upsilon$, the ζ^- and ζ^+ denote new joint physical constraints after the combined equation (2.7)-(2.8).

The PNMCO scheme of redundant robot manipulators is finally rewritten as a standard QP form

$$\text{min. } x^{\mathrm{T}}Qx/2 + g^{\mathrm{T}}x \tag{2.16}$$

$$\text{s.t. } Ax = b \tag{2.17}$$

$$Bx \leq h \tag{2.18}$$

where

$$Q = \begin{bmatrix} (\eta+\delta)I & 0 \\ 0 & (1-\eta-\delta) \end{bmatrix} \in R^{(n+1)\times(n+1)}$$

$$g = \begin{bmatrix} \delta\Lambda(\theta - \theta(0)) \\ 0 \end{bmatrix} \in R^{(n+1)}, A = \begin{bmatrix} J & 0 \end{bmatrix} \in R^{m\times(n+1)}$$

$$b = \dot{r} \in R^m, B = \begin{bmatrix} s \\ E \\ -E \end{bmatrix} \in R^{(4n+2)\times(n+1)}$$

$$s = \begin{bmatrix} I & -I_v \\ -I & -I_v \end{bmatrix} \in R^{2n\times(n+1)}$$

$$h = \begin{bmatrix} d & x^+ & x^- \end{bmatrix}^{\mathrm{T}} \in R^{(4n+2)}, d = 0 \in R^{2n}$$

$$x^- = \begin{bmatrix} \zeta^- \\ 0 \end{bmatrix} \in R^{n+1}, x^+ = \begin{bmatrix} \zeta^+ \\ \varpi \end{bmatrix} \in R^{n+1}$$

where $x = [\dot{\theta}^{\mathrm{T}}, k]^{\mathrm{T}} \in R^{n+1}$ is a variable vector, $E \in R^{n+1}$ is an identity matrix. ϖ is an infinite constant. The 0 and ϖ represent the constraint range of k.

B. VP-RNN-PF SOLVER

Firstly, the inequality constraint (2.18) is converted into a penalty term and which added to the objective function (2.16) of the QP problem [39]. The penalty function is designed as

$$P(x) = p\sum_{i=1}^{4n+2} M(N_i(t))$$

$$\text{with } M(t) = e^{-\sigma N_i(x)}, \quad p \geq 0, \quad \sigma > 0. \tag{2.19}$$

where p is the penalty factor, σ is a positive design parameter, and $N_i(t) = h_i(t) - B_i(t)x(t)$, $i = 1\ 2\ \ldots\ 4n+2$. Therefore, the QP problem (2.16)-(2.18) is transformed into a QP problem without inequality constraints as

$$\text{min. } x^{\mathrm{T}}(t)Q(t)x(t)/2 + g^{\mathrm{T}}(t)x(t) + P(x(t)) \tag{2.20}$$

$$\text{s.t. } A(t)x(t) = b(t) \tag{2.21}$$

Secondly, for the solution of the above QP problem (2.20)-(2.21), we can construct a Lagrange function as

$$L(x(t),\lambda(t),t) = x^{\mathrm{T}}(t)Q(t)x(t)/2 + g^{\mathrm{T}}(t)x(t) + P(x(t)) + \lambda^{\mathrm{T}}(t)(A(t)x(t) - b(t)) \quad (2.22)$$

where $\lambda(t) \in R^m$ is a vector of Lagrange multiplier. In order to solve the above equation (2.22), the equation (2.22) differentiates $x(t)$ and $\lambda(t)$ respectively, and then make them equal to 0, that is

$$\begin{cases} \dfrac{\partial L(x(t),\lambda(t),t)}{\partial x(t)} = Q(t)x(t) + g(t) + A^{\mathrm{T}}(t)\lambda(t) \\ +p\sigma \displaystyle\sum_{i=1}^{4n+2}(e^{-\sigma N_i(t)}\cdot B_i^{\mathrm{T}}(t)) = 0 \\ \dfrac{\partial L(x(t),\lambda(t),t)}{\partial \lambda(t)} = A(t)x(t) - b(t) = 0 \end{cases} \quad (2.23)$$

Equation (2.23) is reformulate to get the following matrix equation

$$D(t)y(t) = u(t) \quad (2.24)$$

where

$$D(t) := \begin{bmatrix} Q(t) & A^{\mathrm{T}}(t) \\ A(t) & 0_{m\times m} \end{bmatrix} \in R^{(n+m+1)\times(n+m+1)}$$

$$y(t) := \begin{bmatrix} x(t) \\ \lambda(t) \end{bmatrix} \in R^{n+m+1}$$

$$u(t) := \begin{bmatrix} -g(t) - p\sigma \displaystyle\sum_{i=1}^{4n+2}(e^{-\sigma N_i(t)}\cdot B_i^{\mathrm{T}}(t)) \\ b(t) \end{bmatrix} \in R^{n+m+1}$$

Through the above operations, solving QP problem (2.16)-(2.18) is transformed into solving matrix equation (2.24).

Thirdly, to find the optimal solution of (2.24), an error function is defined as

$$\varphi(t) = D(t)y(t) - u(t) \quad (2.25)$$

To solve equation (2.24), the above equation (2.25) needs approach to zero, design method based on neural dynamic [28], [31], [33], [38], [40], the derivative of the $\varphi(t)$ to time t is designed as

$$\frac{\mathrm{d}\varphi(t)}{\mathrm{d}t} = -\gamma e^t \Phi(\varphi(t)) \quad (2.26)$$

where $\gamma > 0$, the γe^t is used to adjust the convergence rate, $\Phi(\cdot) : R^{n+m+1} \to R^{n+m+1}$ denotes activation-function processing-array, every unit $\phi(\cdot)$ of $\Phi(\cdot)$ is a monotonically-increasing odd activation function. Three activation functions are used in the simulation experiments of this paper.

The linear activation function as

$$\phi(\varphi(t)) = \varphi(t) \quad (2.27)$$

The sinh activation function as

$$\phi(\varphi(t)) = \frac{e^{(\varphi(t))} - e^{(-\varphi(t))}}{2} \quad (2.28)$$

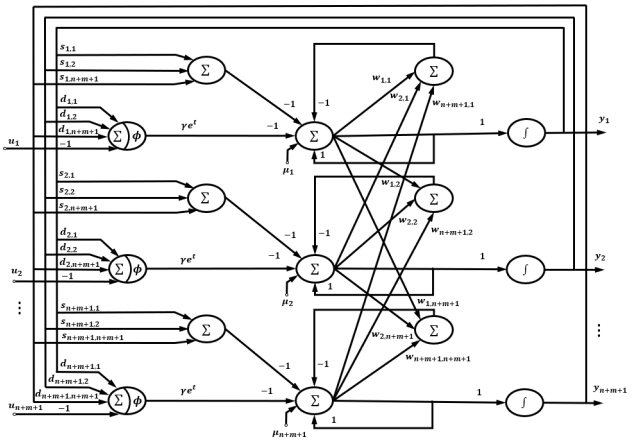

FIGURE 1. Neural topology structure graph of VP-RNN-PF.

The power-sigmoid activation function as

$$\phi(\varphi(t)) = \begin{cases} \dfrac{(1+e^{(-\alpha)})(1-e^{(-\alpha\varphi(t))})}{(1-e^{(-\alpha)})(1+e^{(-\alpha\varphi(t))})}, & \text{if } 0 \le |\varphi(t)| \le 1 \\ \varphi^{\chi}(t), & \text{if } |\varphi(t)| > 1 \end{cases} \quad (2.29)$$

with $\alpha \ge 2, \chi \ge 3$.

Finally, substituting equation (2.25) into equation (2.26), we can get obtain implicit-dynamic equation, then reconstruct this equation, we can obtain the following equation, that is

$$W(t)\dot{y}(t) = -S(t)y(t) - \gamma e^t \Phi(D(t)y(t) - u(t)) + \mu(t) \quad (2.30)$$

where

$$W(t) := \begin{bmatrix} Q(t) + p\sigma^2 \displaystyle\sum_{i=1}^{4n+2}(e^{-\sigma N_i(t)} B_i^{\mathrm{T}}(t)B_i(t)) & A^{\mathrm{T}}(t) \\ A(t) & 0_{m\times m} \end{bmatrix}$$

$$S(t) := \begin{bmatrix} \dot{Q}(t) + p\sigma^2 \displaystyle\sum_{i=1}^{4n+2}(e^{-\sigma N_i(t)} B_i^{\mathrm{T}}(t)\dot{B}_i(t)) & \dot{A}^{\mathrm{T}}(t) \\ \dot{A}(t) & 0_{m\times m} \end{bmatrix}$$

$$\mu(t) := \begin{bmatrix} -\dot{g}(t) - p\sigma \displaystyle\sum_{i=1}^{4n+2}(e^{-\sigma N_i(x)}(\dot{B}_i^{\mathrm{T}}(t) - \sigma B_i^{\mathrm{T}}(t)\dot{h}_i(t))) \\ \dot{b}(t) \end{bmatrix}$$

where $\dot{Q}(t) = \mathrm{d}Q(t)/\mathrm{d}t$, $\dot{B}_i = \mathrm{d}B_i(t)/\mathrm{d}t$, $\dot{A}(t) = \mathrm{d}A(t)/\mathrm{d}t$, $\dot{g}(t) = \mathrm{d}g(t)/\mathrm{d}t$, $\dot{h}_i(t) = \mathrm{d}h_i(t)/\mathrm{d}t$, and $\dot{b}(t) = \mathrm{d}b(t)/\mathrm{d}t$. The time complexity of the proposed neural network is $\mathcal{O}(n^3)$ when using the linear activation function. The structure of VP-RNN-PF is shown in Fig.1.

Theorem 1: The state variable $y(t) = [x^{\mathrm{T}}(t), \lambda^{\mathrm{T}}(t)]^{\mathrm{T}} \in R^{n+m+1}$ of VP-RNN-PF (2.30), starting from any initial state $y(0)$ globally converges to the unique theoretical solution

$y^*(t)$, i.e., $\lim_{t\to+\infty}(y(t) - y^*(t)) = 0$. We provide the following proof.

Proof: Firstly, A Lyapunov function is defined as

$$\Upsilon(t) = \|\varphi(t)\|_2^2/2 = \varphi^{\mathrm{T}}(t)\varphi(t)/2 \geq 0 \tag{2.31}$$

The time derivative of the above equation (2.31) is

$$\dot{\Upsilon}(t) = \frac{\mathrm{d}\Upsilon(t)}{\mathrm{d}t} = \varphi^{\mathrm{T}}(t)\frac{\mathrm{d}\varphi(t)}{\mathrm{d}t} \tag{2.32}$$

Then substituting equation (2.26) into equation (2.31), we can obtain the following equation

$$\dot{\Upsilon}(t) = -\gamma e^t \varphi^{\mathrm{T}}(t)\Phi(\varphi(t)) = -\gamma e^t \sum_{i=1}^{n+m+1} \varphi_i(t)\phi(\varphi_i(t)) \tag{2.33}$$

where $\varphi_i(t)$ denotes the ith element of vector $\varphi(t)$. The activation function $\phi(\varphi_i(t))$ is monotonically-increasing and odd, $\varphi_i(t)\phi(\varphi_i(t))$ meets the following conditions

$$\varphi_i(t)\phi(\varphi_i(t)) = \begin{cases} > 0, & \text{if } \varphi_i(t) > 0 \text{ or } \varphi_i(t) < 0 \\ = 0, & \text{if } \varphi_i(t) = 0 \end{cases} \tag{2.34}$$

Finally, combining equation (2.33) and equation (2.34), $\dot{\Upsilon}(t) \leq 0$ when $t \in [0, +\infty)$. According to the Lyapunov theory, $\lim_{t\to+\infty}(y(t) - y^*(t))$ globally converges to 0. The proof is completed.

C. S-LVI-PDNN SOLVER

To compare with the PNMCO scheme, S-LVI-PDNN is given here to solve the QP problem (2.16)-(2.18) [24], [25], [36], [41].

$$\dot{y} = \beta(P_\Omega(y - (My + q)) - y) \tag{2.35}$$

where

$$M = \begin{bmatrix} Q & -A^{\mathrm{T}} & E^{\mathrm{T}} \\ A & 0 & 0 \\ -E & 0 & 0 \end{bmatrix} \in R^{(3n+m+1)\times(3n+m+1)}$$

$$y = \begin{bmatrix} x \\ \psi \\ \iota \end{bmatrix} \in R^{2(3n+m+1)}, q = \begin{bmatrix} g \\ -b \\ d \end{bmatrix} \in R^{(3n+m+1)}$$

where $\beta > 0$ is used to adjust the convergence rate of the S-LVI-PDNN (2.35), the value of β cannot be infinite, otherwise will take a lot of time to calculate, this β is set as 10^7. $P_\Omega(\cdot)$ denote the projection operator $R^{2(3n+m+1)} \to \Omega$, $\psi \in R^m$ and $\iota \in R^{2n}$ are decision vector for (2.17) and (2.18), respectively.

III. COMPUTER SIMULATIONS

In this section, to illustrate the feasibility and effectiveness of the PNMCO scheme, computer simulations of the four-leaf-clover path and Chinese character "dai" path are performed on the Kinova JACO^2 robot manipulator. Besides, computer simulation results of VP-RNN-PF and traditional S-LVI-PDNN are compared. Joint physical limits of JACO^2 robot manipulator are shown in Table 1.

TABLE 1. Joint physical limits of JACO^2 robot manipulator.

Joint	θ^+(rad)	θ^-(rad)	$\dot{\theta}^+$(rad/s)	$\dot{\theta}^-$(rad/s)
1	174.44	-174.44	1.5000	-1.5000
2	3.89	-2.389	1.5000	-1.5000
3	1.238	-5.041	1.5000	-1.5000
4	174.44	-174.44	1.5000	-1.5000
5	174.44	-174.44	1.5000	-1.5000
6	174.44	-174.44	1.5000	-1.5000

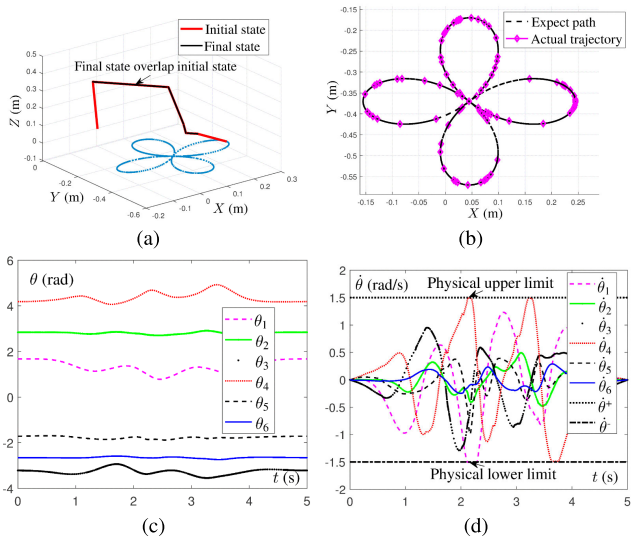

FIGURE 2. Four-leaf-clover path synthesized by PNMCO scheme. (a) Joint angle coincidence states. (b) End-effector motion trajectory. (c) Joint angle θ profiles. (d) Joint-velocity $\dot{\theta}$ profiles.

A. FOUR-LEAF-CLOVER PATH TRACKING

In this subsection, the Kinova JACO^2 robot manipulator tracks the path of four-leaf-clover. The initial joint angle $\theta(0) = [1.675; 2.843; -3.216; 4.187; -1.71; -2.65]$ rad and the motion task duration $T = 5$ s, design parameters as $\upsilon = 20$, $\Lambda = 20$, $p = 10^{-4}$, $\sigma = 400$, $\gamma = 2$, and $\beta = 10^7$.

Firstly, Fig. 2 shows the end-effector of JACO^2 redundant robot manipulator tracks the path of four-leaf-clover. As shown in Fig. 2(a) and (b), the end-effector can complete the path tracking task well. As shown in Fig. 2(c), the joint angle θ is always within the physical limits when executing the task. As shown in Fig. 2(a) and (c), After completing a task cycle, the joints return to the initial position. The second column of Table 3 shows the detailed joint angle drifts, which deviation reaches 10^{-4} rad levels. From Fig. 2(d), when executing the path tracking task, the $\dot{\theta}_1$ and $\dot{\theta}_4$ reach the limits ± 1.5 rad/s, but never exceed them, which can effectively avoid high velocity and damage to the robot manipulator. In short, this four-leaf-clover path tracking simulation results verify the feasibility and accuracy of the proposed PNMCO scheme.

Secondly, VP-RNN-PF (2.30) and S-LVI-PDNN (2.35) are applied to solve the QP problem (2.16)-(2.18) of the CMCO scheme, and the robot manipulator is controlled to track the four-leaf-clover path for comparisons. In simulations, the CMCO schemes with two different weight distribution are adopted, which are the CMCO scheme ($\eta = 0.2$ and

TABLE 2. Comparisons of VP-RNN-PF and S-LVI-PDNN for resolving CMCO scheme ($\eta = 0.2$ and $\delta = 0.4$) when tracking the four-leaf-clover path.

joint drift	VP-RNN-PF (linear)	VP-RNN-PF (sinh)	VP-RNN-PF(power-sigmoid)	S-LVI-PDNN
$\theta_1(5) - \theta_1(0)$	1.96×10^{-5}	5.18×10^{-5}	3.78×10^{-4}	7.26×10^{-5}
$\theta_2(5) - \theta_2(0)$	-1.71×10^{-5}	6.86×10^{-6}	8.41×10^{-5}	1.75×10^{-4}
$\theta_3(5) - \theta_3(0)$	2.59×10^{-5}	5.28×10^{-4}	5.39×10^{-4}	1.21×10^{-3}
$\theta_4(5) - \theta_4(0)$	9.68×10^{-5}	8.95×10^{-4}	6.72×10^{-4}	1.82×10^{-3}
$\theta_5(5) - \theta_5(0)$	6.63×10^{-5}	3.86×10^{-4}	3.68×10^{-4}	9.23×10^{-4}
$\theta_6(5) - \theta_6(0)$	-9.24×10^{-6}	-3.31×10^{-4}	-2.82×10^{-4}	-7.08×10^{-4}
$\|\theta(5) - \theta(0)\|_2$	$\mathbf{1.23\times10^{-4}}$	$\mathbf{1.16\times10^{-3}}$	$\mathbf{1.05\times10^{-3}}$	2.49×10^{-3}
Computation time (s)	**2.1566**	**2.3381**	**2.3825**	5.1153

TABLE 3. Comparisons of VP-RNN-PF and S-LVI-PDNN for resolving CMCO scheme ($\eta = 0.1$ and $\delta = 0.8$) when tracking the four-leaf-clover path.

joint drift	VP-RNN-PF (linear)	VP-RNN-PF (sinh)	VP-RNN-PF(power-sigmoid)	S-LVI-PDNN
$\theta_1(5) - \theta_1(0)$	-1.62×10^{-4}	-1.12×10^{-5}	-1.52×10^{-4}	-5.06×10^{-4}
$\theta_2(5) - \theta_2(0)$	-4.77×10^{-5}	-6.74×10^{-5}	4.37×10^{-5}	4.64×10^{-5}
$\theta_3(5) - \theta_3(0)$	4.41×10^{-5}	2.69×10^{-4}	2.29×10^{-4}	-1.09×10^{-4}
$\theta_4(5) - \theta_4(0)$	9.11×10^{-6}	2.93×10^{-4}	2.07×10^{-4}	4.06×10^{-4}
$\theta_5(5) - \theta_5(0)$	-2.14×10^{-5}	1.40×10^{-4}	3.33×10^{-5}	2.68×10^{-5}
$\theta_6(5) - \theta_6(0)$	3.53×10^{-5}	-1.09×10^{-4}	-4.06×10^{-5}	-8.07×10^{-5}
$\|\theta(5) - \theta(0)\|_2$	$\mathbf{1.80\times10^{-4}}$	$\mathbf{4.41\times10^{-4}}$	$\mathbf{3.51\times10^{-4}}$	6.65×10^{-4}
Computation time (s)	**2.0490**	**2.2613**	**2.2698**	4.7664

$\delta = 0.4$) and CMCO scheme ($\eta = 0.1$ and $\delta = 0.8$). The VP-RNN-PF applies three different activation functions (the linear (2.27), sinh (2.28), and power-sigmoid (2.29) activation function). The joint drifts of the two CMCO schemes with different weight distributions are demonstrated in Table 2 and Table 3, respectively. $\theta_i(5) - \theta_i(0)$ denote the drifts of the ith joint between the final state and initial state of the joint angle. $\|\theta(5) - \theta(0)\|_2$ denotes the two norm of joint drifts. As shown in Table 2 and Table 3, all joint drifts $\|\theta(5)-\theta(0)\|_2$ of CMCO schemes synthesized by VP-RNN-PF with three different activation functions are less than the CMCO scheme synthesized by S-LVI-PDNN. Specifically, the $\|\theta(5)-\theta(0)\|_2$ of the CMCO scheme ($\eta = 0.2$ and $\delta = 0.4$) synthesized by S-LVI-PDNN is 2.49×10^{-3} rad, while synthesized by VP-RNN-PF with linear, sinh, and power-sigmoid activation functions are 1.23×10^{-4} rad, 1.16×10^{-3} rad, and 1.05×10^{-3} rad, respectively. The $\|\theta(5) - \theta(0)\|_2$ of the CMCO scheme ($\eta = 0.1$ and $\delta = 0.8$) synthesized by S-LVI-PDNN is 6.65×10^{-4} rad, while synthesized by VP-RNN-PF with linear, sinh, and power-sigmoid activation functions are 1.80×10^{-4} rad, 4.41×10^{-4} rad, and 3.51×10^{-4} rad, respectively.

Thirdly, the runtime of VP-RNN-PF (2.30) and S-LVI-PDNN (2.35) can be obtained from the last line of Table 2 and Table 3 when the end-effector tracks the four-leaf-clover path. All runtime of VP-RNN-PF with linear, sinh, and power-sigmoid activation functions are very tiny. Among them, the VP-RNN-PF with linear activation function is the fastest, The runtime of the CMCO scheme ($\eta = 0.2$ and $\delta = 0.4$) and CMCO scheme ($\eta = 0.1$ and $\delta = 0.8$) are 2.1566 s and 2.0490 s, respectively. However, the runtime of S-LVI-PDNN is more than twice that of VP-RNN-PF with linear activation function. The runtime of the CMCO scheme ($\eta = 0.2$ and $\delta = 0.4$) solved by S-LVI-PDNN is 5.1153 s. The runtime is over the task execution time T = 5 s, which means it cannot effectively solve the CMCO scheme.

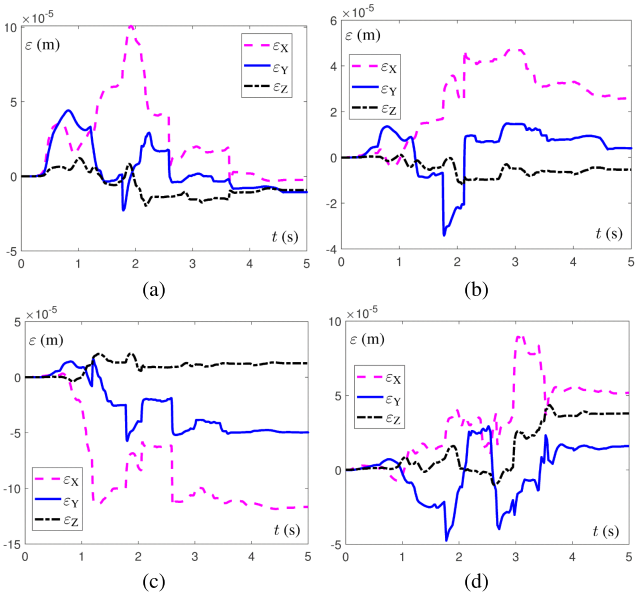

FIGURE 3. The positioning error of the end effector when using the VP-RNN-PF (three activation functions) and S-LVI-PDNN solvers to solve the CMCO scheme ($\eta = 0.2$ and $\delta = 0.4$) of tracking the four-leaf-clover trajectory (a) Position error of VP-RNN-PF (linear). (b) Position error of VP-RNN-PF (sinh). (c) Position error of VP-RNN-PF (power-sigmoid). (d) Position error of S-LVI-PDNN.

Fourthly, compare the position errors of the CMCO scheme ($\eta = 0.2$ and $\delta = 0.4$) synthesized by VP-RNN-PF(three different activation functions) with S-LVI-PDNN during the four-leaf-clover path tracking task. we can see that there is not much difference between VP-RNN-PF with linear activation function Fig.3(a) and VP-RNN-PF with power-sigmoid activation function Fig.3(c) and S-LVI-PDNN Fig.3(d), but the VP-RNN-PF with sinh activation function Fig.3(b) is less than the S-LVI-PDNN. Specifically, the position error of VP-RNN-PF with sinh activation function ε is less than 5.0×10^{-5} m. The position error with S-LVI-PDNN ε is less than 10×10^{-5} m. Fig.4 shows the position errors of

TABLE 4. Comparisons of VP-RNN-PF and S-LVI-PDNN for resolving CMCO scheme ($\eta = 0.1$ and $\delta = 0.8$) when tracking the chinese character "dai" path.

joint drift	VP-RNN-PF (linear)	VP-RNN-PF (sinh)	VP-RNN-PF(power-sigmoid)	S-LVI-PDNN
$\theta_1(5) - \theta_1(0)$	1.03×10^{-4}	1.66×10^{-4}	1.97×10^{-4}	2.41×10^{-4}
$\theta_2(5) - \theta_2(0)$	1.04×10^{-5}	-1.63×10^{-5}	-7.53×10^{-6}	-2.84×10^{-5}
$\theta_3(5) - \theta_3(0)$	1.05×10^{-5}	-8.01×10^{-5}	3.30×10^{-5}	-1.88×10^{-4}
$\theta_4(5) - \theta_4(0)$	-8.73×10^{-5}	-1.15×10^{-5}	-7.81×10^{-5}	7.29×10^{-6}
$\theta_5(5) - \theta_5(0)$	-8.69×10^{-6}	5.01×10^{-5}	2.62×10^{-5}	9.73×10^{-5}
$\theta_6(5) - \theta_6(0)$	1.09×10^{-5}	2.85×10^{-5}	-1.01×10^{-5}	-5.76×10^{-5}
$\|\theta(5) - \theta(0)\|_2$	$\mathbf{1.72\times10^{-4}}$	$\mathbf{1.94\times10^{-4}}$	$\mathbf{2.17\times10^{-4}}$	3.27×10^{-4}
Computation time (s)	**2.4921**	**2.7917**	**2.7859**	17.6546

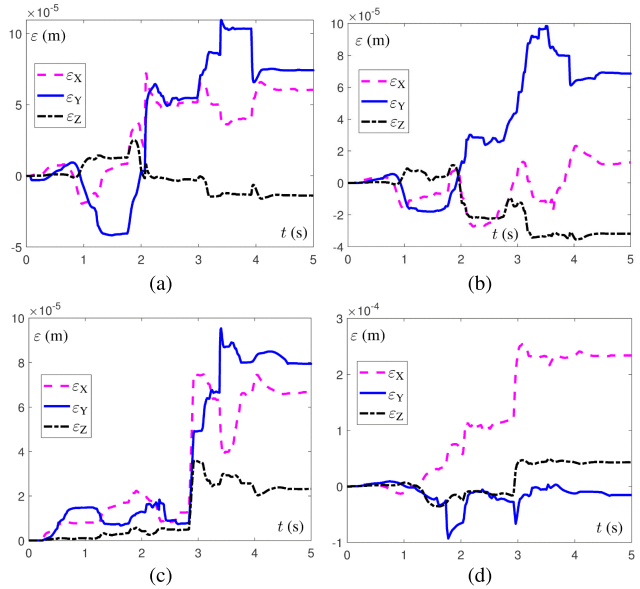

FIGURE 4. The positioning error of the end effector when using the VP-RNN-PF (three activation functions) and S-LVI-PDNN solvers to solve the CMCO scheme ($\eta = 0.1$ and $\delta = 0.8$) of tracking the four-leaf-clover trajectory (a) Position error of VP-RNN-PF (linear). (b) Position error of VP-RNN-PF (sinh). (c) Position error of VP-RNN-PF (power-sigmoid). (d) Position error of S-LVI-PDNN.

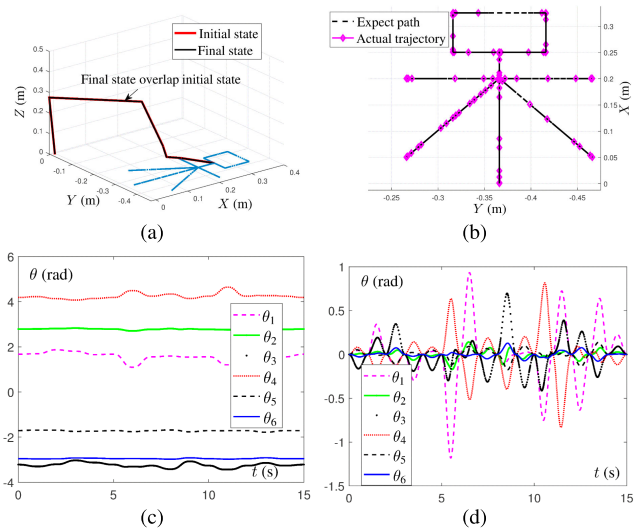

FIGURE 5. Chinese character "dai" path synthesized by PNMCO scheme. (a) Joint angle coincidence states. (b) End-effector motion trajectory. (c) Joint angle θ profiles. (d) Joint-velocity $\dot{\theta}$ profiles.

the CMCO scheme ($\eta = 0.1$ and $\delta = 0.8$) synthesized by VP-RNN-PF (three different activation functions) and S-LVI-PDNN, the position errors of three VP-RNN-PF with different activation function are all less than those of S-LVI-PDNN. The linear activation function with the largest position error ε among the three activation functions of VP-RNN-PF is less than 1.1×10^{-4} m, while the position error with S-LVI-PDNN ε is less than 2.6×10^{-4} m. All in all, the VP-RNN-PF has less position error than S-LVI-PDNN.

In summary, this four-leaf-clover path tracking simulation verifies that VP-RNN-PF can effectively and accurately solve the CMCO scheme.

B. CHINESE CHARACTER "Dai" PATH TRACKING

This subsection illustrates another path tracking task Chinese character "dai" to further verify the effectiveness and practicability of the PNMCO scheme. In the simulations, the initial joint angle $\theta(0) = [1.675; 2.79; -3.216; 4.187; -1.71; -2.95]$ rad and motion task duration $T = 15$ s, the design parameters as $\upsilon = 20$, $\Lambda = 20$, $p = 10^{-5}$, $\sigma = 200$, $\gamma = 2$, $\beta = 10^7$.

Firstly, Fig.5 shows the end-effector of the JACO^2 redundant robot manipulator tracks Chinese character "dai" path. The path tracking task is completed well, which can be seen in Fig.5(a) and (b). After completing a task cycle, the joint angle drifts are very small, which can be regarded as the joints of the manipulator fully return to the initial position. Table 4 shows the specific data of joint angle drifts of the CMCO scheme ($\eta = 0.1$ and $\delta = 0.8$) synthesized by VP-RNN-PF. Fig.5(c) and (d) show the joint angle θ and joint-velocity $\dot{\theta}$ are always within the physical limits during the execution of the task.

Secondly, comparison results of the CMCO scheme ($\eta = 0.1$ and $\delta = 0.8$) synthesized by the VP-RNN-PF and S-LVI-PDNN are shown in Table 4. All joint drifts $\|\theta(5) - \theta(0)\|_2$ of the CMCO scheme synthesized by VP-RNN-PF with their different activation functions are less than the CMCO scheme synthesized by S-LVI-PDNN. The most significant advantage of VP-RNN-PF over S-LVI-PDNN is the short calculation time. The runtime of VP-RNN-PF with linear, sinh, and power-sigmoid activation functions are 2.4921 s, 2.7917 s, and 2.7859 s, respectively, which can be seen from the last line of Table 4. However, the runtime of S-LVI-PDNN is 17.6546 s.

Thirdly, Compare the position errors of the CMCO scheme ($\eta = 0.1$ and $\delta = 0.8$) synthesized by VP-RNN-PF

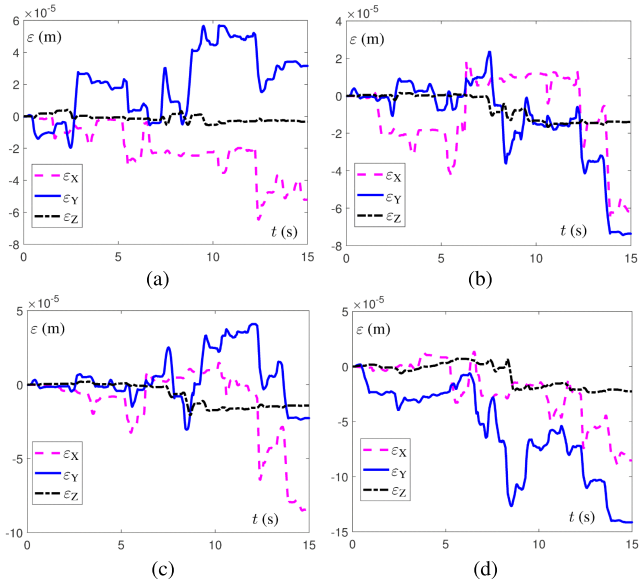

FIGURE 6. The positioning error of the end effector when using the VP-RNN-PF (three activation functions) and S-LVI-PDNN solvers to solve the CMCO scheme ($\eta = 0.1$ and $\delta = 0.8$) of tracking the Chinese character "dai" trajectory (a) Position error of VP-RNN-PF (linear). (b) Position error of VP-RNN-PF (sinh). (c) Position error of VP-RNN-PF (power-sigmoid). (d) Position error of S-LVI-PDNN.

(three different activation functions) and S-LVI-PDNN when tracking the path of Chinese character "dai". From Fig.6, the position errors of three VP-RNN-PF with different activation function are all less than those of S-LVI-PDNN, the maximum errors of VP-RNN-PF with linear, sinh, and power-sigmoid activation functions ε are 6.5×10^{-5} m, 7.5×10^{-5} m, and 8.5×10^{-5} m. But the maximum error of S-LVI-PDNN is 15×10^{-5} m.

In summary, the above two path tracking task results illustrate that the VP-RNN-PF can effectively solve the CMCO scheme. In addition, compared with the traditional S-LVI-PDNN, the proposed PNMCO scheme with different activation functions is more effective and more accurate.

IV. CONCLUSION

In this paper, a PNMCO scheme has been proposed and investigated to solve the multi-objective motion planning problem for redundant robot manipulators. The scheme consists of CMCO and VP-RNN-PF subsystems. The CMCO subsystem can achieve energy minimization, repetitive motion, and avoidance of speed peaks. In addition, the CMCO subsystem considers a novel variable joint physical constraint, and then the CMCO subsystem is reformulated as a QP problem. Finally, the QP problem has been solved by the proposed VP-RNN-PF. Comparison experiments have validated that the proposed VP-RNN-PF is more accurate and more efficient compared with the traditional S-LVI-PDNN for solving the multi-objective motion planning problem. The future research direction will be the VP-RNN-PF solver investigated to solve multi-criteria schemes at different levels.

REFERENCES

[1] H. Shen, "Research on the design and control strategy of surgical manipulator for knee surgery assistant robot," in *Proc. IEEE 4th Adv. Inf. Technol., Electron. Autom. Control Conf. (IAEAC)*, vol. 1, Dec. 2019, pp. 374–377.

[2] J. Burgner-Kahrs, D. C. Rucker, and H. Choset, "Continuum robots for medical applications: A survey," *IEEE Trans. Robot.*, vol. 31, no. 6, pp. 1261–1280, Dec. 2015.

[3] F. C. Samavati, M. Iranikhah, P. Dastangoo, and M. R. S. Alashti, "Mechanical basic and detailed design for the redundant arm SAAM applied on a domestic service robot," in *Proc. Artif. Intell. Robot. (IRANOPEN)*, Apr. 2017, pp. 77–83.

[4] A. Sameh, M. Fanni, and A. M. Mohamed, "New 3D translational interconnected manipulator for industrial applications," in *Proc. IEEE Int. Conf. Mechatronics Autom. (ICMA)*, Aug. 2018, pp. 469–474.

[5] I. V. Voinov, A. M. Kazantsev, and M. V. Nosikov, "Robot-manipulator MR-48 for nuclear industry," in *Proc. Int. Conf. Ind. Eng., Appl. Manuf. (ICIEAM)*, May 2018, pp. 1–6.

[6] D. Chen and Y. Zhang, "A hybrid multi-objective scheme applied to redundant robot manipulators," *IEEE Trans. Autom. Sci. Eng.*, vol. 14, no. 3, pp. 1337–1350, Jul. 2017.

[7] T. Fung Chan and R. V. Dubey, "A weighted least-norm solution based scheme for avoiding joint limits for redundant joint manipulators," *IEEE Trans. Robot. Autom.*, vol. 11, no. 2, pp. 286–292, Apr. 1995.

[8] Z. Zhang, L. Zheng, J. Yu, Y. Li, and Z. Yu, "Three recurrent neural networks and three numerical methods for solving a repetitive motion planning scheme of redundant robot manipulators," *IEEE/ASME Trans. Mechatronics*, vol. 22, no. 3, pp. 1423–1434, Jun. 2017.

[9] Z. Zhang and Y. Zhang, "Repetitive motion planning and control on redundant robot manipulators with Push-Rod-Type joints," *J. Dyn. Syst., Meas., Control*, vol. 135, no. 2, Mar. 2013.

[10] Z. Zhang, S. Chen, X. Zhu, and Z. Yan, "Two hybrid end-effector posture-maintaining and obstacle-limits avoidance schemes for redundant robot manipulators," *IEEE Trans. Ind. Informat.*, vol. 16, no. 2, pp. 754–763, Feb. 2020.

[11] 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.

[12] C. A. Klein and C.-H. Huang, "Review of pseudoinverse control for use with kinematically redundant manipulators," *IEEE Trans. Syst., Man, Cybern.*, vol. SMC-13, no. 2, pp. 245–250, Mar. 1983.

[13] Z. Zhang and Y. Zhang, "Variable joint-velocity limits of redundant robot manipulators handled by quadratic programming," *IEEE/ASME Trans. Mechatronics*, vol. 18, no. 2, pp. 674–686, Apr. 2013.

[14] F.-T. Cheng, T.-H. Chen, and Y.-Y. Sun, "Resolving manipulator redundancy under inequality constraints," *IEEE Trans. Robot. Autom.*, vol. 10, no. 1, pp. 65–71, Feb. 1994.

[15] H. Ding and J. Wang, "Recurrent neural networks for minimum infinity-norm kinematic control of redundant manipulators," *IEEE Trans. Syst., Man, Cybern. A, Syst. Humans*, vol. 29, no. 3, pp. 269–276, May 1999.

[16] Y. Zhang, J. Yin, and B. Cai, "Infinity-norm acceleration minimization of robotic redundant manipulators using the LVI-based primal–dual neural network," *Robot. Comput.-Integr. Manuf.*, vol. 25, no. 2, pp. 358–365, Apr. 2009.

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

[18] D. Guo, K. Zhai, Z. Xiao, H. Tan, and Y. Zhang, "Acceleration-level minimum kinetic energy (MKE) scheme derived via ma equivalence for motion planning of redundant robot manipulators," in *Proc. 7th Int. Symp. Comput. Intell. Design*, vol. 1, Dec. 2014, pp. 26–30.

[19] Y. Zhang, W. Li, D. Guo, Z. Zhang, and S. Fu, "Feedback-type MWVN scheme and its acceleration-level equivalent scheme proved by Zhang dynamics," in *Proc. Int. Conf. Control Eng. Commun. Technol.*, Dec. 2012, pp. 180–183.

[20] Y. Zhang, B. Cai, L. Zhang, and K. Li, "Bi-criteria velocity minimization of robot manipulators using a linear variational inequalities-based primal-dual neural network and PUMA560 example," *Adv. Robot.*, vol. 22, nos. 13–14, pp. 1479–1496, Jan. 2008.

[21] 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.

[22] D. Guo, K. Zhai, X. Yu, B. Cai, and Y. Zhang, "Zhang equivalence of different-level robotic schemes: An MVN case study based on PA10 robot manipulator," in *Proc. IEEE Int. Conf. Robot. Biomimetics (ROBIO)*, Dec. 2013, pp. 1592–1597.
[23] Y. Zhang, D. Guo, J. Li, and K. Li, "Unification and comparison on bi-criteria velocity, acceleration and torque minimization illustrated via three-link planar robot arm," in *Proc. Chin. Control Decis. Conf. (CCDC)*, May 2011, pp. 3410–3415.
[24] Z. Zhang, Y. Lin, S. Li, Y. Li, Z. Yu, and Y. Luo, "Tricriteria optimization-coordination motion of dual-redundant-robot manipulators for complex path planning," *IEEE Trans. Control Syst. Technol.*, vol. 26, no. 4, pp. 1345–1357, Jul. 2018.
[25] Z. Jia, S. Chen, X. Qu, P. Zhang, and N. Zhong, "Velocity-level tri-criteria optimization scheme for different complex path tracking of redundant manipulators," *IEEE Access*, vol. 7, pp. 64289–64296, 2019.
[26] Y. Zhang, S. Fu, Z. Zhang, L. Xiao, and X. Li, "On the LVI-based numerical method (E47 algorithm) for solving quadratic programming problems," in *Proc. IEEE Int. Conf. Autom. Logistics (ICAL)*, Aug. 2011, pp. 125–130.
[27] Y. Zhang, J. Li, M. Mao, W. Li, and S. Fu, "Complete theory for E47 and 94LVI algorithms solving inequality-and-bound constrained quadratic program efficiently," in *Proc. Chin. Autom. Congr. (CAC)*, Nov. 2015, pp. 183–189.
[28] Z. Zhang, T. Fu, Z. Yan, L. Jin, L. Xiao, Y. Sun, Z. Yu, and Y. Li, "A varying-parameter convergent-differential neural network for solving joint-angular-drift problems of redundant robot manipulators," *IEEE/ASME Trans. Mechatronics*, vol. 23, no. 2, pp. 679–689, Apr. 2018.
[29] Y. Leung, K.-Z. Chen, Y.-C. Jiao, X.-B. Gao, and K. S. Leung, "A new gradient-based neural network for solving linear and quadratic programming problems," *IEEE Trans. Neural Netw.*, vol. 12, no. 5, pp. 1074–1083, Sep. 2001.
[30] L. Xiao and Y. Zhang, "Zhang neural network versus gradient neural network for solving time-varying linear inequalities," *IEEE Trans. Neural Netw.*, vol. 22, no. 10, pp. 1676–1684, Oct. 2011.
[31] Y. Zhang, D. Jiang, and J. Wang, "A recurrent neural network for solving Sylvester equation with time-varying coefficients," *IEEE Trans. Neural Netw.*, vol. 13, no. 5, pp. 1053–1063, Sep. 2002.
[32] P. Miao, Y. Shen, Y. Huang, and Y.-W. Wang, "Solving time-varying quadratic programs based on finite-time zhang neural networks and their application to robot tracking," *Neural Comput. Appl.*, vol. 26, no. 3, pp. 693–703, Apr. 2015.
[33] Z. Zhang, Y. Lu, L. Zheng, S. Li, Z. Yu, and Y. Li, "A new varying-parameter convergent-differential neural-network for solving time-varying convex QP problem constrained by linear-equality," *IEEE Trans. Autom. Control*, vol. 63, no. 12, pp. 4110–4125, Dec. 2018.
[34] 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.
[35] 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.
[36] Y. Zhang, Z. Li, H.-Z. Tan, and Z. Fan, "On the simplified LVI-based primal-dual neural network for solving LP and QP problems," in *Proc. IEEE Int. Conf. Control Autom.*, May 2007, pp. 3129–3134.
[37] Z. Zhang, S. Chen, L. Zheng, and J. Zhang, "MATLAB simulink of varying-parameter convergent-differential neural-network for solving online time-varying matrix inverse," in *Proc. 9th Int. Symp. Comput. Intell. Design (ISCID)*, vol. 1, Dec. 2016, pp. 320–325.
[38] Z. Zhang, S. Yang, and L. Zheng, "A penalty strategy combined varying-parameter recurrent neural network for solving time-varying multi-type constrained quadratic programming problems," *IEEE Trans. Neural Netw. Learn. Syst.*, early access, Jul. 29, 2020, doi: 10.1109/TNNLS.2020.3009201.
[39] J. P. Evans, F. J. Gould, and J. W. Tolle, "Exact penalty functions in nonlinear programming," *Math. Program.*, vol. 4, no. 1, pp. 72–97, Dec. 1973.
[40] Z. Zhang and Z. Yan, "A varying parameter recurrent neural network for solving nonrepetitive motion problems of redundant robot manipulators," *IEEE Trans. Control Syst. Technol.*, vol. 27, no. 6, pp. 2680–2687, Nov. 2019.
[41] Y. Zhang, "On the LVI-based primal-dual neural network for solving online linear and quadratic programming problems," in *Proc. Amer. Control Conf.*, 2005, pp. 1351–1356.

NAN ZHONG received the Ph.D. degree in agricultural mechanization engineering from South China Agricultural University, Guangzhou, China, in 2006. She was a Visiting Professor with the Institute of Applied Mathematics, University of British Columbia, Canada, in 2009 to 2010, and a Senior researcher with the Institute of Agricultural Engineering, University of Hohenheim, Germany, in 2015 to 2016. Since 2006, she has been a Professor with the College of Engineering, South China Agricultural University. Her current research interests include robotics, neural networks, pattern recognition, and intelligent systems.

QINGYU HUANG received the B.E. degree from Jiangxi Agricultural University, Nanchang, China, in 2019. He is currently pursuing the M.E. degree with the College of Engineering, South China Agricultural University, Guangzhou, China. His current research interests include robotics and neural networks.

SONG YANG received the B.S. degree in automation from the North China Institute of Science and Technology, Langfang, China, in 2018. He is currently pursuing the M.S. degree in control engineering with the School of Automation Science and Engineering, South China University of Technology, Guangzhou, China. His current research interests include robotics and neural networks.

FAN OUYANG received the Ph.D. degree in mechanical engineering-robotics and automation from the South China University of Technology, China, in 2013. From 2014 to 2016, he had conducted a postdoctoral research with the College of Material Science and Engineering, South China University of Technology. Since 2016, he has been an Assistant Researcher with the College of Engineering, South China Agricultural University, Guangzhou, China. His current research interest includes the research and application of open-source agricultural UAV (Unmanned Aerial Vehicle) platform combined with robotic technology.

ZHIJUN ZHANG (Senior Member, IEEE) received the Ph.D. degree from Sun Yat-sen University, Guangzhou, China, in 2012.

From 2013 to 2015, he was a Postdoctoral Research Fellow with the Institute for Media Innovation, Nanyang Technological University, Singapore. From 2015 to 2019, he worked as an Associate Professor with the School of Automation Science and Engineering, South China University of Technology, where he has been a Full Professor, since 2020. He has published more than 80 research papers on IEEE TRANSACTIONS ON AUTOMATIC CONTROL, IEEE TRANSACTIONS ON NEURAL NETWORKS AND LEARNING SYSTEMS, IEEE TRANSACTIONS ON MECHTRONICS, IEEE TRANSACTIONS ON CYBERNETICS, and top international conferences. His current research interests include neural networks, robotics, machine learning, human machine interaction, and optimal control. He is currently a member of the Committee on Visual Cognition and Computation of Chinese Society of Image Graphics and the Hybrid Intelligence Professional Committee of Chinese Association of Automation. He received the Guangdong Science Fund for Distinguished Young Scholars and the Youth Talents in Science and Technology Innovation of Guangdong Special Support Plan. He is also an Executive Editor-in-Chief for *Global Journal of Neuroscience*, an Associate Editor for *International Journal of Robotics and Control*, a Review Editor for *Frontiers in Robotics and AI*, and reviewers of over 20 international journals.