Hindawi
Complexity
Volume 2020, Article ID 2305459, 11 pages
https://doi.org/10.1155/2020/2305459

Research Article

A Gradient-Based Recurrent Neural Network for Visual Servoing of Robot Manipulators with Acceleration Command

Zhiguan Huang,[1] **Zhengtai Xie**,[2] **Long Jin**,[1,2] **and Yuhe Li**[1]

[1]*Guangdong Provincial Engineering Technology Research Center for Sports Assistive Devices, Guangzhou Sport University, Guangzhou, China*
[2]*School of Information Science and Engineering, Lanzhou University, Lanzhou, China*

Correspondence should be addressed to Long Jin; jinlongsysu@foxmail.com and Yuhe Li; 13922738963@139.com

Received 31 July 2020; Revised 11 November 2020; Accepted 15 December 2020; Published 30 December 2020

Academic Editor: Ning Wang

Recent decades have witnessed the rapid evolution of robotic applications and their expansion into a variety of spheres with remarkable achievements. This article researches a crucial technique of robot manipulators referred to as visual servoing, which relies on the visual feedback to respond to the external information. In this regard, the visual servoing issue is tactfully transformed into a quadratic programming problem with equality and inequality constraints. Differing from the traditional methods, a gradient-based recurrent neural network (GRNN) for solving the visual servoing issue is newly proposed in this article in the light of the gradient descent method. Then, the stability proof is presented in theory with the pixel error convergent exponentially to zero. Specifically speaking, the proposed method is able to impel the manipulator to approach the desired static point while maintaining physical constraints considered. After that, the feasibility and superiority of the proposed GRNN are verified by simulative experiments. Significantly, the proposed visual servo method can be leveraged to medical robots and rehabilitation robots to further assist doctors in treating patients remotely.

1. Introduction

As one of the greatest human inventions in the 20th century, robot technology has undoubtedly made great progress in the past decades with brilliant research achievements [1–4]. After the birth, growth, and maturity of robots, they have become the indispensable core equipment in the manufacturing industry due to their high automation and efficiency. Especially as the rising star of the family of robots, redundant robots, which possess more degrees of freedom (DOFs) than the task requires, are capable of performing complicated tasks efficiently with the great property and versatility. In detail, the redundancy characteristic assists the redundant robots in fulfilling additional task demands, for example, repetitive motion planning [5], physical constraint avoidance [6], and manipulability optimization [7, 8]. In combination with medical technology, various medical robots have been developed and explored for patient rehabilitation and surgical execution as an important application prospect. Relying on high reliability and flexibility, medical robots are able to perform complex medical tasks, thus reducing the burden on doctors and improving treatment. The learning and control ability of various robots is also valued and explored by many scholars [9–11]. A novel learning framework for the robot learning and generalizing human-like variable impedance skills is developed in [9] with great research and practical value. Further, some adaptive control methods are presented for estimating the unknown model of manipulator dynamic, which achieves great parameter estimation and tracking effects [10, 11].

In current years, the kinematic control of redundant robots has become a research hotspot, thus drawing the attention of abundant scholars to expand their applications [12–16]. Zhang and Zhang present a minimum-velocity-norm (MVN) scheme for redundancy resolution of the redundant manipulators, which retains the robot joints within safe bounds [17]. A modified neural network approach in [18] is well designed for precise control of the robot manipulator, which can eliminate the error accumulation with accurate results. Moreover, the authors in [19] research an ingenious transformation method to deal with the acceleration limitation problem from the velocity level, and the experimental results illustrate the superiority of the method. It is deserved to notice that the above investigations [15–19] all transform the kinematic control issue of redundant robots into quadratic programming and then exploit the Karush–Kuhn–Tucker (KKT) conditions [20] or Lagrange multiplier method to solve the optimization schemes. In addition, the mentioned schemes in [15–19] are all velocity-level solutions such that they cannot interfere directly with the acceleration level.

With the continuous development of sensors and Internet of Things technology, robot applications have become very rich owing to information acquisition and processing. The sensor can transmit the external information directly to the control center of the robot and give appropriate feedback to the information through specific intelligent algorithms. As a greatly important robot application, the vision servoing technology drives the robot to accurately feedback the external vision in real time through the visual information collected by the vision sensor [21–23]. This technology is already being used in industrial production and robotic surgery [24, 25]. However, it is worth pointing out that the existing techniques [26–28] for solving the vision servoing problem often rely on the implementation of the pseudoinverse method to converge errors, which has achieved great results in both the acceleration-level schemes and the velocity-level schemes. By means of proportional-differential control, acceleration command for the visual servoing control is generated with excellent stability [26]. Moreover, an effective method to detect and compensate for faults in visual servoing systems is presented in literature [27], which is verified by simulation and experimental results. Based on the pseudoinverse operation of the Jacobian matrix, the robotic ball catching task is implemented [28]. This method takes advantage of the eye-in-hand construction to establish the motion capture system for locating fast-moving objects. However, a large number of investigations do not consider the existence of joint constraints and have potential for damage to the robot manipulators [21–24, 26–28]. Due to the physical limitations of the robot motor and robot structure, the control signals need to be kept within a reasonable range to maintain the normal operation of the robot manipulators. To this end, this paper formulates the visual servoing problem as a quadratic programming scheme with equality and inequality constraints in consideration of physical constraints.

The rise of intelligent algorithms in recent years has solved many difficult problems in electronic and engineering fields [29–31]. Numerous intelligent algorithms have been designed for powerful performance, such as noise suppression [32], simplified computation [33, 34], and predictive learning [35, 36]. Among the intelligent algorithms to solve the visual servoing of the manipulator, the neural network method stands out due to its fast parallel processing performance and learning ability [37–41]. In [42], a recurrent neural network is constructed for the visual servoing issue to force the feature point of the manipulator to approach the designed target point. Then, the extended research [43] eliminates the pseudoinversion operation and equips the neural network with powerful robustness. In addition, as a common optimization method, the gradient descent method has made some progress in the design of robot control algorithms in recent years [44, 45]. It can be used to accurately locate and control the robot by minimizing the position error [46]. Based on the above research, we establish the visual servoing issue based on acceleration commands and transform it into a quadratic programming scheme solved by the neural network method. Besides, the contributions of this paper are summarized below:

(1) The proposed method regards the visual servoing problem as a constrained quadratic programming scheme with acceleration command and meanwhile considers the joint constraints to ensure the safety of the manipulator

(2) This paper proposes a gradient-based recurrent neural network (GRNN) for dealing with the research on the robot visual servoing via the gradient descent method and exploiting compensation item

(3) The simulation example and illustrative experiment illustrate the feasibility and superiority of the proposed method

The remainder of this paper is summarized as follows. Section 2 covers the preliminaries and the visual servoing kinematics. In Section 3, the visual servoing problem is transformed into a constrained quadratic programming scheme at the acceleration level with the corresponding GRNN deduced. The theoretical analyses of the proposed method are presented by using the Lyapunov method in Section 4. Section 5 carries out a simulation example to demonstrate the feasibility of the proposed method. In the end, we summarily conclude the whole paper in Section 7.

2. Preliminaries

In this section, the visual servoing kinematics is introduced, which records the conversion relationship between the joint space and the image space.

Primarily, in consideration of an eye-in-hand vision system [28], i.e., an n-DOF manipulator with a camera attached to the end effector, the forward kinematics of the manipulator is given as follows:

$$\mathscr{P}(\vartheta(t)) = p(t), \tag{1}$$

where $\mathscr{P}(\cdot)$ describes the transformation relationship between the joint space and Cartesian space; $\vartheta(t) \in \mathbb{R}^n$ represents the joint angle of the manipulator; and $p(t) \in \mathbb{R}^m$ denotes the Cartesian coordinates of the end effector. The investigation of visual servoing issue always takes both the position and posture of the end effector, and thus $p(t)$ is set as a six-dimensional vector hereinafter ($m = 6$). Taking the derivative of time with respect to formula (1) leads to

$$J(\vartheta(t))\dot{\vartheta}(t) = \dot{p}(t), \tag{2}$$

where $J(\vartheta(t)) \in \mathbb{R}^{m\times n}$ stands for the robot Jacobian matrix, which is determined by the manipulator structure; $\dot{\vartheta}(t)$ signifies the joint velocity of the manipulator; and $\dot{p}(t)$ is the end effector velocity containing angular velocity and translational velocity. In addition, the physical constraints, involving joint velocity $\dot{\vartheta}$ and joint acceleration $\ddot{\vartheta}$, to maintain the safe operation of the manipulator system are provided as below:

$$\dot{\vartheta}^{-} \le \dot{\vartheta} \le \dot{\vartheta}^{+}, \tag{3a}$$

$$\ddot{\vartheta}^{-} \le \ddot{\vartheta} \le \ddot{\vartheta}^{+}, \tag{3b}$$

with $\dot{\vartheta}^{-}$ and $\dot{\vartheta}^{+}$ being the upper and lower bounds of joint velocity and $\ddot{\vartheta}^{-}$ and $\ddot{\vartheta}^{+}$ denoting the upper and lower bounds of joint acceleration. As for the camera frame and image frame, the corresponding relationship is deduced by means of similar triangle and given as follows [27, 42]:

$$\begin{bmatrix} a_{\mathrm{i}} \\ b_{\mathrm{i}} \end{bmatrix} = \frac{l}{d}\begin{bmatrix} a_{\mathrm{c}} \\ b_{\mathrm{c}} \end{bmatrix}, \tag{4}$$

of which $[a_i, b_i]^{\mathrm{T}}$ is a point coordinate in the image frame with the superscript T denoting the transpose of a matrix or a vector; $[a_{\mathrm{c}}, b_{\mathrm{c}}, d]^{\mathrm{T}}$ stands for the coordinate in the camera frame; and l denotes the focal length of the camera. Besides, in the image frame, point coordinates can be converted to pixel coordinates $v = [u, v]^{\mathrm{T}}$ by the following formula [43]:

$$u = u_{\mathrm{p}} + \kappa_a a_{\mathrm{i}}, \tag{5a}$$

$$v = v_p + \kappa_b b_i, \tag{5b}$$

where $[u_p, v_p]^{\mathrm{T}}$ stands for the designed original point and κ_a and κ_b are the pixel standard size. Furthermore, the relationship between the camera velocity, i.e., the end effector velocity $\dot{p}$, and pixel coordinate velocity $\dot{v}$ can be introduced as

$$H(v,d)\dot{p} = \dot{v}, \tag{6}$$

where $H(v,d) \in \mathbb{R}^{2\times 6}$ denotes the image Jacobian matrix [47, 48] with its expression being

$$H(v,d) = M\begin{bmatrix} -\dfrac{l}{d} & 0 & \dfrac{la_i}{d} & \dfrac{a_i b_i}{l} & -\dfrac{a_i^2 + l^2}{l} & b_i \\ 0 & -\dfrac{l}{d} & \dfrac{b_i}{d} & \dfrac{b_i^2 + l^2}{l} & -\dfrac{a_i b_i}{l} & -a_i \end{bmatrix}, \tag{7}$$

with

$$a_i = \frac{u - u_p}{\kappa_a},$$
$$b_i = \frac{v - v_p}{\kappa_b}, \tag{8}$$
$$M = \begin{bmatrix} \kappa_a & 0 \\ 0 & \kappa_b \end{bmatrix}.$$

Based on the above instructions, especially formula (2) and formula (6), it can be readily obtained that $H(v,d)J\dot{\vartheta} = \dot{v}$, which involves the relationship between the joint space and the image space. To simplify the presentation, one designs

$$\mathcal{J}\dot{\vartheta} = \dot{v} \text{ with } \mathcal{J} = H(v,d)J. \tag{9}$$

Furthermore, the kinematic relationship at the acceleration level is derived by taking time derivative as

$$\ddot{v} = \mathcal{J}\ddot{\vartheta} + \dot{\mathcal{J}}\dot{\vartheta}, \tag{10}$$

where $\ddot{v}$ represents the acceleration of feature point in the image frame and $\dot{\mathcal{J}}$ denotes the time derivative of $\mathcal{J}$.

3. Acceleration-Level IBVS Scheme and Its Solution

The robot vision servoing controls the robot manipulator to interact with circumstances according to the visual information. This issue can be simplified to find the static point in the image frame by feeding back the image information. To this end, we turn this visual servoing problem into a constrained quadratic programming problem and design a neural network-based solver.

3.1. Quadratic Programming Scheme with Constraints. Above all, the visual servoing problem is formulated at the acceleration level into the following quadratic programming scheme:

$$\text{minimize } \frac{1}{2}\ddot{\vartheta}^{\mathrm{T}}\ddot{\vartheta}, \tag{11}$$

$$\text{subject to } \ddot{v} = \mathcal{J}\ddot{\vartheta} + \dot{\mathcal{J}}\dot{\vartheta}, \tag{12}$$

$$v = v^{*}, \tag{13}$$

$$\ddot{\vartheta} \in \rho, \tag{14}$$

where v^{*} denotes the desired feature point, which is a designed constant vector and, $\rho = \left\{\ddot{\vartheta} \in \mathbb{R}^n, \rho^{-} \le \ddot{\vartheta} \le \rho^{+}\right\}$ is an inequality constraint corresponding to the physical limit (3) with ρ^{-} and ρ^{+} devised as

$$\begin{aligned}\rho_i^- &= \max\left\{\alpha\left(\dot{\vartheta}_i^- - \dot{\vartheta}_i\right), \ddot{\vartheta}_i^-\right\},\\ \rho_i^+ &= \min\left\{\alpha\left(\dot{\vartheta}_i^+ - \dot{\vartheta}_i\right), \ddot{\vartheta}_i^+\right\},\end{aligned} \tag{15}$$

where $\alpha > 0$ stands for the design parameter. Via (15), the physical constraints of joint acceleration and joint velocity could be considered and controlled within bounds simultaneously [15]. In this regard, take the upper limit of the physical constraint $\rho_i^+ = \min\left\{\alpha(\dot{\vartheta}_i^+ - \dot{\vartheta}_i), \ddot{\vartheta}_i^+\right\}$ as an example. For the joint velocity, when the joint velocity $\dot{\vartheta}_i$ approaches the upper bound of velocity-level joint constraint $\dot{\vartheta}_i^+$, $\alpha(\dot{\vartheta}_i^+ - \dot{\vartheta}_i)$ gets small and even close to zero. Afterwards, ρ_i^+ becomes tiny or even zero, so that the joint velocity stops growing and stays in joint constraints. Simultaneously, the upper bound of acceleration-level joint constraint $\ddot{\vartheta}_i^+$ is activated to realize acceleration-level joint constraint. Similarly, ρ_i^- is able to realize the velocity-level joint constraint and the acceleration-level joint constraint simultaneously.

3.2. Neural Network Solution. Differing from the traditional method to deal with equality constraints and inequality constraints, the gradient descent method [49] is exploited to derive the solution to the quadratic programming scheme (11)–(14). Design an error function $\varepsilon = v - v^*$ to start the derivation. Utilizing neural dynamic formula $\dot{\varepsilon} = -\delta\varepsilon\,(\delta > 0)$ [50] and $\dot{v}^* = \ddot{v}^* = 0$, one can get

$$\dot{v} = -\delta\left(v - v^*\right), \tag{16}$$

which can be arranged and rewritten into the form of two norms as follows:

$$e = \frac{\left\|\dot{v} + \delta\left(v - v^*\right)\right\|_2^2}{2}. \tag{17}$$

Given the gradient descent formula [51],

$$\ddot{\vartheta} = -\nabla e = -\gamma\frac{\partial(e)}{\partial\dot{\vartheta}}, \tag{18}$$

with $\gamma > 0$, it would be readily deduced that

$$\ddot{\vartheta} = \gamma\mathcal{J}^{\mathrm{T}}\left(-\dot{v} - \delta\left(v - v^*\right)\right). \tag{19}$$

Then, a compensation item ϖ is presented to make up for the lagging error in equation (19) as below:

$$\ddot{\vartheta} = \gamma\mathcal{J}^{\mathrm{T}}\left(-\dot{v} - \delta\left(v - v^*\right)\right) + \varpi. \tag{20}$$

Via deliberating the final desired stable state, i.e., $v - v^* = \dot{v} = 0$, one can simply get the expression of ϖ referring to the derivation below. Multiplying both sides of equation (20) by $\mathcal{J}$ one gains

$$\mathcal{J}\ddot{\vartheta} = \gamma\mathcal{J}\mathcal{J}^{\mathrm{T}}\left(-\dot{v} - \delta\left(v - v^*\right)\right) + \mathcal{J}\varpi. \tag{21}$$

Set $v - v^* = \dot{v} = 0$, and it can be obtained that

$$\mathcal{J}\ddot{\vartheta} = \mathcal{J}\varpi. \tag{22}$$

Then, taking the time derivative of $\dot{v} = \mathcal{J}\dot{\vartheta} = 0$ as

$$\mathcal{J}\ddot{\vartheta} = -\dot{\mathcal{J}}\dot{\vartheta}. \tag{23}$$

Comparing the two formulas above, one has

$$\mathcal{J}\varpi = -\dot{\mathcal{J}}\dot{\vartheta}. \tag{24}$$

Hence, it can be easily got that

$$\varpi = -\mathcal{J}^{\dagger}\dot{\mathcal{J}}\dot{\vartheta}, \tag{25}$$

with superscript † being the pseudoinverse operator of a matrix and $J^{\dagger} = J^{\mathrm{T}}(JJ^{\mathrm{T}})^{-1}$. Consequently, the GRNN solver is structured for solving the quadratic programming scheme (11)–(14) as follows:

$$\ddot{\vartheta} = \mathcal{F}_{\rho}\left(\gamma\mathcal{J}^{\mathrm{T}}\left(-\dot{v} - \delta\left(\mathbf{v} - \mathbf{v}^*\right)\right) - \mathcal{J}^{\dagger}\dot{\mathcal{J}}\dot{\vartheta}, \tag{26}$$

where $\mathcal{F}_{\rho}(x) = \arg\min_{y\epsilon\rho}\|y - x\|$ can be regarded as a bounded activation function and the usage of arg min can be referred to [52, 53], which is equivalent to the inequality constraint (14). As Figure 1 depicts, visual servoing scheme (11)–(14) aided with GRNN solver (26) integrates the robot frame and image frame and can be regarded as a restricted online acceleration controller. For GRNN (26) and scheme (11)–(14), the following corresponding relation is given. Owing to the derivative process that GRNN (26) originates from the error function (17), the gradient descent formula is designed to reduce the image error, thus ultimately achieving equality constraint (13). In the next place, the output control command is established at the acceleration level, which corresponds to the acceleration-level kinematics formula (12). Note that compensation item ϖ is the pseudoinverse solution of the system function in a stable state, i.e., the minimization of joint acceleration, which is equivalent to minimizing objective function (11). As for joint constraint (14), introducing $\mathcal{F}_{\rho}(\cdot)$ is able to impose restrictions on joint velocity and joint acceleration. In short, the proposed GRNN solver (26) corresponds to the quadratic programming scheme (11)–(14).

Remarks. Compared with the existing visual servo technologies, the innovations of this paper are worth emphasizing as follows: regarding the scheme (11)–(14) construction level, most of the previous strategies on visual servoing are controlled at the joint velocity level, few of which are controlled and driven by joint acceleration. In addition, none of the existing acceleration-level visual servo schemes takes joint limits into account, which is considered in the quadratic programming scheme (11)–(14). From the perspective of the intelligent algorithm, a majority of the existing techniques apply the pseudoinverse method to directly deal with the errors, which incurs additional computational overhead. However, GRNN (26) is deduced according to the gradient descent method and compensation term, which provides a novel approach to dealing with the visual servoing problem.

4. Stability Proof

In this section, the stability proof is provided to prove the feasibility and effectiveness of the proposed method (26) to

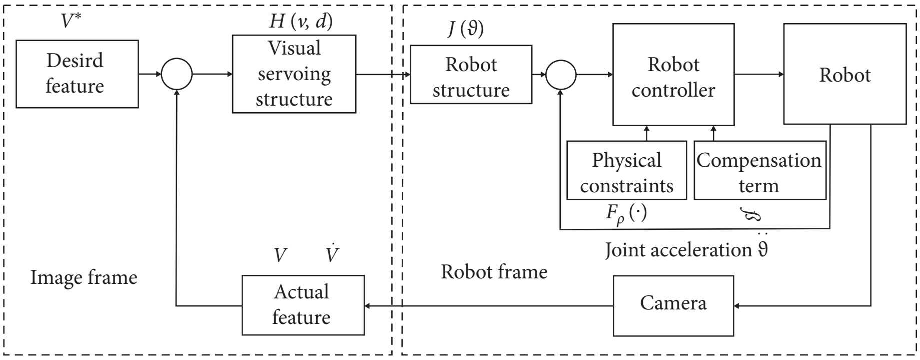


Figure 1: Control flowchart of the visual servoing scheme (11)–(14).

dispose of the visual servoing issue. The relevant theorem is given as follows.

Theorem 1. *The error $\varepsilon = v - v^*$ synthesized by GRNN (26) can approach zero globally, provided that $-\mathscr{J}^\dagger \dot{\mathscr{J}}\dot{\vartheta} \in \rho$.*

Proof. Declare that the setting of precondition $-\mathscr{J}^\dagger \dot{\mathscr{J}}\dot{\vartheta} \in \rho$ has two core functions. The first is to determine the minimum joint constraints, thus ensuring the safe operation of the manipulator. It is easy to image that forcing the joint to remain within the constraints may lead to the increase of error as reported in [54]. The second point is worth mentioning that precondition $-\mathscr{J}^\dagger \dot{\mathscr{J}}\dot{\vartheta} \in \rho$ is of necessities for the proper derivation of the theorem. According to (10), one can get

$$\mathscr{J}\ddot{\vartheta} + \dot{\mathscr{J}}\dot{\vartheta} = \mathscr{J}\mathscr{F}_\rho\left(\gamma\mathscr{J}^{\mathrm{T}}\left(-\dot{v} - \delta\left(v - v^*\right)\right) - \mathscr{J}^\dagger \dot{\mathscr{J}}\dot{\vartheta}\right) + \dot{\mathscr{J}}\dot{\vartheta}. \tag{27}$$

In the light of $\varepsilon = v - v^*$, $\dot{\varepsilon} = \dot{v}$, and $\ddot{\varepsilon} = \ddot{v}$, equation (27) can be rearranged as

$$\ddot{\varepsilon} = \mathscr{J}\left(\mathscr{F}_\rho\left(\gamma\mathscr{J}^{\mathrm{T}}(-\dot{\varepsilon} - \delta\mathscr{E}) - \mathscr{J}^\dagger \dot{\mathscr{J}}\dot{\vartheta}\right) + \mathscr{J}^\dagger \dot{\mathscr{J}}\dot{\vartheta}\right). \tag{28}$$

Let $V = \dot{\varepsilon}^T\dot{\varepsilon}/2$ stand for a Lyapunov candidate. Therefore, calculating its time derivative $\dot{V} = \dot{\varepsilon}^{\mathrm{T}}\ddot{\varepsilon}$ results in

$$\begin{aligned}\dot{V} &= \dot{\varepsilon}^{\mathrm{T}}\mathscr{J}\left(\mathscr{F}_\rho\left(\gamma\mathscr{J}^{\mathrm{T}}(-\dot{\varepsilon} - \delta\mathscr{E}) - \mathscr{J}^\dagger \dot{\mathscr{J}}\dot{\vartheta}\right) + \mathscr{J}^\dagger \dot{\mathscr{J}}\dot{\vartheta}\right)\\ &= -\frac{1}{\gamma}\left(\gamma\mathscr{J}^{\mathrm{T}}(-\dot{\varepsilon} - \delta\mathscr{E}) - \mathscr{J}^\dagger \dot{\mathscr{J}}\dot{\vartheta} + \delta\mathscr{J}^{\mathrm{T}}\mathscr{E} + \mathscr{J}^\dagger \dot{\mathscr{J}}\dot{\vartheta}\right)\\ &\quad\times\left(\mathscr{F}_\rho\left(\gamma\mathscr{J}^{\mathrm{T}}(-\dot{\varepsilon} - \delta\mathscr{E}) - \mathscr{J}^\dagger \dot{\mathscr{J}}\dot{\vartheta}\right) + \mathscr{J}^\dagger \dot{\mathscr{J}}\dot{\vartheta}\right).\end{aligned} \tag{29}$$

Consider the inequality relation $\|\mathscr{F}_\rho(x) - x\|^2 \le \|x - y\|^2$, $\forall y \in \rho$. We simply devise $x = \gamma\mathscr{J}^{\mathrm{T}}(-\dot{\varepsilon} - \delta\varepsilon) - \mathscr{J}^\dagger \dot{\mathscr{J}}\dot{\vartheta}$ and $y = -\gamma\delta\mathscr{J}^{\mathrm{T}}\varepsilon - \mathscr{J}^\dagger \dot{\mathscr{J}}\dot{\vartheta}$ and get

$$\begin{aligned}&\left\|\mathscr{F}_\rho\left(\gamma\mathscr{J}^{\mathrm{T}}(-\dot{\varepsilon} - \delta\varepsilon) - \mathscr{J}^\dagger \dot{\mathscr{J}}\dot{\vartheta}\right) - \gamma\mathscr{J}^{\mathrm{T}}(-\dot{\varepsilon} - \delta\mathscr{E}) + \mathscr{J}^\dagger \dot{\mathscr{J}}\dot{\vartheta}\right\|^2\\ &\le \left\|\gamma\mathscr{J}^{\mathrm{T}}(-\dot{\varepsilon} - \delta\varepsilon) - \mathscr{J}^\dagger \dot{\mathscr{J}}\dot{\vartheta} + \gamma\delta\mathscr{J}^{\mathrm{T}}\varepsilon + \mathscr{J}^\dagger \dot{\mathscr{J}}\dot{\vartheta}\right\|^2.\end{aligned} \tag{30}$$

Expanding the left side of the above equation generates

$$\begin{aligned}&\left\|\mathscr{F}_\rho\left(\gamma\mathscr{J}^{\mathrm{T}}(-\dot{\varepsilon} - \delta\varepsilon) - \mathscr{J}^\dagger \dot{\mathscr{J}}\dot{\vartheta}\right) + \mathscr{J}^\dagger \dot{\mathscr{J}}\dot{\vartheta} + \gamma\delta\mathscr{J}^{\mathrm{T}}\varepsilon\right\|^2\\ &\quad + \left\|\gamma\mathscr{J}^{\mathrm{T}}(-\dot{\varepsilon} - \delta\varepsilon) - \mathscr{J}^\dagger \dot{\mathscr{J}}\dot{\vartheta} + \delta\mathscr{J}^{\mathrm{T}}\varepsilon + \mathscr{J}^\dagger \dot{\mathscr{J}}\dot{\vartheta}\right\|^2\\ &\quad - 2\left(\gamma\mathscr{J}^{\mathrm{T}}(-\dot{\varepsilon} - \delta\varepsilon) - \mathscr{J}^\dagger \dot{\mathscr{J}}\dot{\vartheta} + \delta\mathscr{J}^{\mathrm{T}}\varepsilon + \mathscr{J}^\dagger \dot{\mathscr{J}}\dot{\vartheta}\right)^{\mathrm{T}}\\ &\quad \times\left(\mathscr{F}_\rho\left(\gamma\mathscr{J}^{\mathrm{T}}(-\dot{\varepsilon} - \delta\varepsilon) - \mathscr{J}^\dagger \dot{\mathscr{J}}\dot{\vartheta}\right) + \mathscr{J}^\dagger \dot{\mathscr{J}}\dot{\vartheta} + \gamma\delta\mathscr{J}^{\mathrm{T}}\varepsilon\right).\end{aligned} \tag{31}$$

Observe the two formulas above, and it can be easily gained that

$$\begin{aligned}&\left\|\mathscr{F}_\rho\left(\gamma\mathscr{J}^{\mathrm{T}}(-\dot{\varepsilon} - \delta\varepsilon) - \mathscr{J}^\dagger \dot{\mathscr{J}}\dot{\vartheta}\right) + \mathscr{J}^\dagger \dot{\mathscr{J}}\dot{\vartheta} + \gamma\delta\mathscr{J}^{\mathrm{T}}\varepsilon\right\|^2\\ &\le 2\left(\gamma\mathscr{J}^{\mathrm{T}}(-\dot{\varepsilon} - \delta\varepsilon) - \mathscr{J}^\dagger \dot{\mathscr{J}}\dot{\vartheta} + \delta\mathscr{J}^{\mathrm{T}}\mathscr{E} + \mathscr{J}^\dagger \dot{\mathscr{J}}\dot{\vartheta}\right)^{\mathrm{T}}\\ &\left(\mathscr{F}_\rho\left(\gamma\mathscr{J}^{\mathrm{T}}(-\dot{\varepsilon} - \delta\varepsilon) - \mathscr{J}^\dagger \dot{\mathscr{J}}\dot{\vartheta}\right) + \mathscr{J}^\dagger \dot{\mathscr{J}}\dot{\vartheta} + \gamma\delta\mathscr{J}^{\mathrm{T}}\varepsilon\right).\end{aligned} \tag{32}$$

Substituting equation (29) into equation (32) deduces

$$\begin{aligned}&\dot{V} - \gamma\delta\dot{\varepsilon}^{\mathrm{T}}\mathscr{J}\mathscr{J}^{\mathrm{T}}\mathscr{E} \le\\ &-\frac{1}{2\gamma}\left\|\mathscr{F}_\rho\left(\gamma\mathscr{J}^{\mathrm{T}}(-\dot{\varepsilon} - \delta\varepsilon) - \mathscr{J}^\dagger \dot{\mathscr{J}}\dot{\vartheta}\right) + \mathscr{J}^\dagger \dot{\mathscr{J}}\dot{\vartheta} + \gamma\delta\mathscr{J}^{\mathrm{T}}\varepsilon\right\|^2 \le 0.\end{aligned} \tag{33}$$

Evidently, one has

$$\dot{V} \le \gamma\delta\dot{\varepsilon}^{\mathrm{T}}\mathscr{J}\mathscr{J}^{\mathrm{T}}\varepsilon. \tag{34}$$

Recalling the neural dynamic formula $\dot{\varepsilon} = -\delta\varepsilon$, it is evident that

$$\gamma\delta\dot{\varepsilon}^{\mathrm{T}}\mathscr{J}\mathscr{J}^{\mathrm{T}}\varepsilon = -\gamma\delta^2\varepsilon^{\mathrm{T}}\mathscr{J}\mathscr{J}^{\mathrm{T}}\varepsilon \le -\gamma\delta^2\sigma\varepsilon^{\mathrm{T}}\varepsilon \le 0, \tag{35}$$

with design parameter $\gamma > 0$, $\delta > 0$, and $\sigma > 0$ denoting the minimum eigenvalue of positive definite matrix $\mathscr{J}\mathscr{J}^{\mathrm{T}}$. Therefore, it can be naturally concluded that $\dot{\varepsilon}$ is of great convergence with $\dot{V} \le 0$. Referring to the Lasalle invariance principle [55], we let $\dot{V} = 0$ to derive the stable state and get the following two conditions:

$$\dot{\varepsilon} = 0 \text{ or } \ddot{\varepsilon} = \mathscr{F}_\rho\left(\gamma\mathscr{J}^{\mathrm{T}}(-\dot{\varepsilon} - \delta\mathscr{E}) - \mathscr{J}^\dagger \dot{\mathscr{J}}\dot{\vartheta}\right) = -\mathscr{J}^\dagger \dot{\mathscr{J}}\dot{\vartheta}. \tag{36}$$

Given that $-\mathscr{J}^\dagger \dot{\mathscr{J}}\dot{\vartheta} \in \rho$, the solutions to the above two conditions can be gained:

$$\dot{\varepsilon} = \varepsilon = 0. \tag{37}$$

In this regard, a conclusion can be readily drawn that ε is convergent to zero globally. The proof is thus competed (Figure 2). □

5. Simulation Example

This section provides a simulation example to demonstrate the performance of GRNN (26) when confronted with the robot visual servoing issue. Specifically speaking, the PUMA 560 manipulator (6-DOF) is modeled with a camera attached to its end effector to track the desired static point in the image frame. In addition, the structure information of the PUMA 560 manipulator can be referred to the existing literature [43] with the photo of PUMA 560 shown in Figure 2. It is worth pointing out that when considering only one desired feature point, the kinematic control of the PUMA 560 manipulator can be regarded as utilizing the 6-dimensional joint space to control 2-dimensional image space, which can approximately treat the PUMA 560 manipulator as a redundant manipulator.

In the first place, the simulation setting and the neural network parameters are introduced. Simply put, the parameters of the neural network and camera system are set as $u_p = v_p = 256$ pixel, $\kappa_a = \kappa_b = 8 \times 10^4$pixel/m, $l = 8 \times 10 - 3$m, $d = 2$m, $v^* = [256, 256]^{\mathrm{T}}$pixel, $\delta = 10$, $\gamma = 10^3$, and $\alpha = 20$. As to the state and physical constraints of the PUMA 560 manipulator, the states are chosen as $\vartheta_0 = [0.3, -0.9, 0.4, 0.3, -1, -0.2]^{\mathrm{T}}$rad, the initial coordinate of feature point $v_0 = [169, 104]^{\mathrm{T}}$ pixel, $\dot{\vartheta}^+ = -\dot{\vartheta}^- = [0.4]_{6\times1}$rad/s, and $\ddot{\vartheta}^+ = -\ddot{\vartheta}^- = [3]_{6\times1}$ rad/s^2.

The simulation results are provided in Figure 3. As depicted in Figure 3, the PUMA 560 manipulator successfully achieves the desired feature point driven by GRNN (26). The error ε in Figure 3(b) and $\dot{\varepsilon}$ in Figure 3(c) converge to zero in 1 s. With regard to joint information, Figure 3(d) through Figure 3(f) record the joint acceleration, joint velocity, and joint angle during the simulation, respectively. It is worth emphasizing that joint acceleration and joint velocity are maintained within the designed physical constraints, which ensures the safe execution of the task. Overall, the above results indicate the feasibility and efficiency of the proposed GRNN (26) when handling the visual servoing issue.

To demonstrate the superiority of the proposed method, the traditional pseudoinverse method is employed to deal with the visual servoing problem with results provided in Figure 4. The control law adopted by the traditional pseudoinverse method is generalized as

$$\ddot{\vartheta} = \mathcal{J}^{\dagger}\left(-\beta\dot{v} - \varepsilon\left(v - v^*\right) - \dot{\mathcal{J}}\dot{\vartheta}\right), \tag{38}$$

with $\beta = 25$ and $\varepsilon = 100$. It is worth pointing out that the investigations of visual servoing based on the pseudoinversion operation of the Jacobian matrix are common and effective in the existing method [21, 26, 28]. Nevertheless, the pseudoinversion operation of a matrix brings more computational complexity, and the conventional pseudoinverse methods do not take joint limits into account, which are regarded as the deficiencies of existing methods [21, 26, 28]. As depicted in Figure 4(a), the error ε quickly converges to zero in 1.5 s, i.e., the manipulator successfully tracks the desired feature point. However, Figure 4(b) indicates that due to the large value of the initial error ε, the generated initial accelerations are even more than 10 rad/s^2, which would damage the PUMA 560 manipulator. On the contrary, the proposed method (26) limits the acceleration in the physical constraints, which emphasizes the superiority of the proposed method (26).

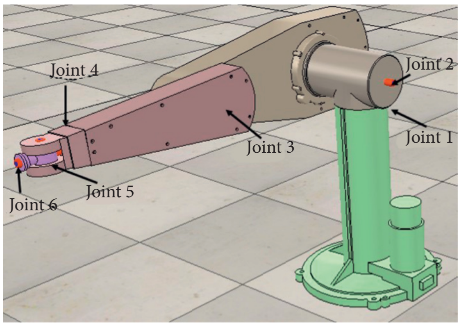
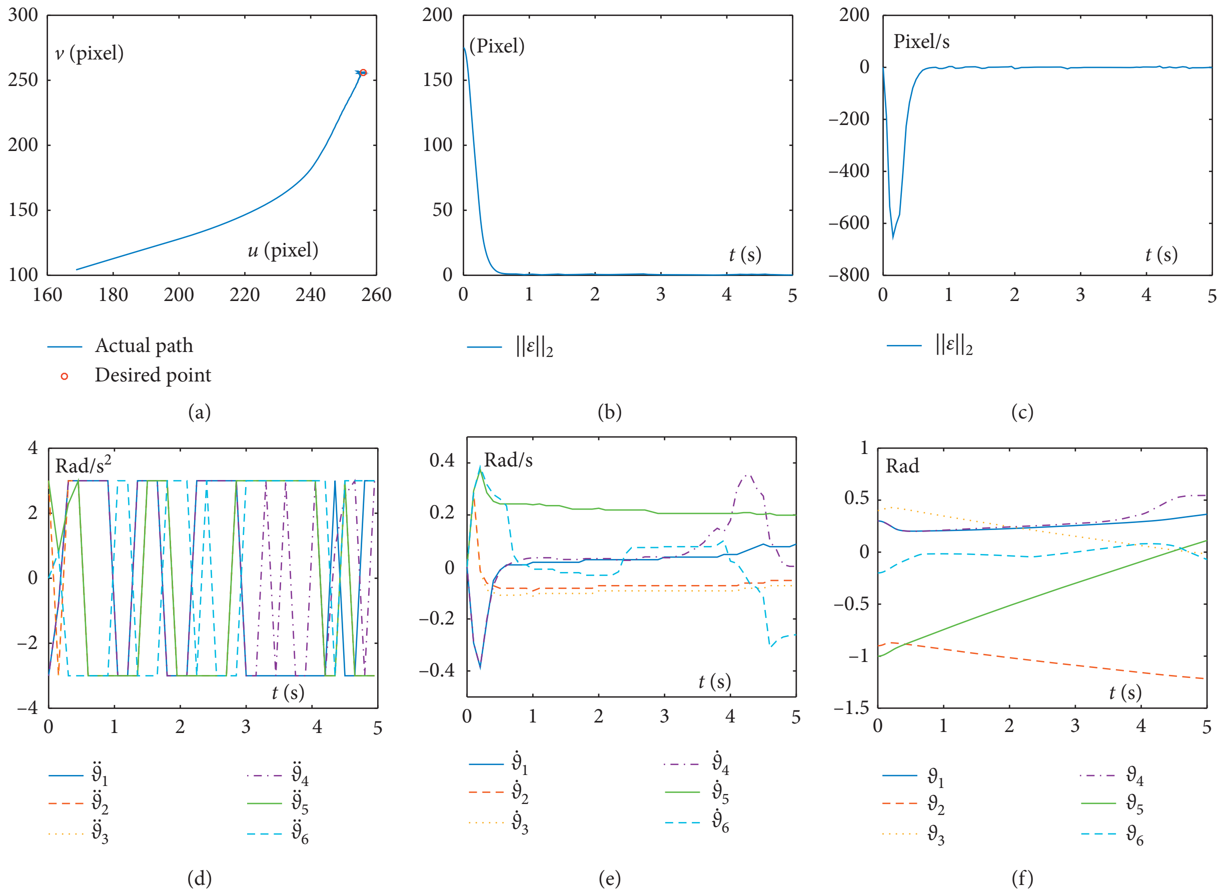


Figure 2: Photo of PUMA 560 with six joints.

Beyond that, an illustrative experiment is conducted on a UR5 manipulator (6-DOF) [25] with a visual sensor installed on its end effector, which is assisted by Virtual Robot Experimentation Platform (V-rep). The experiment results plotted in Figure 5 are synthesized by the proposed GRNN (26). Note that in Figure 5(a), the measured object is regarded as the desired point v^*, which can be captured by the visual sensor, and that the center of the sensor view is the feature point v of the robot visual system. By constantly transmitting the error information $\varepsilon = v - v^*$ to GRNN (26), the visual servoing issue can be solved with v approaching v^* as described in Figures 5(b) and 5(c), which implies the validity of the proposed GRNN (26).

6. Comparisons

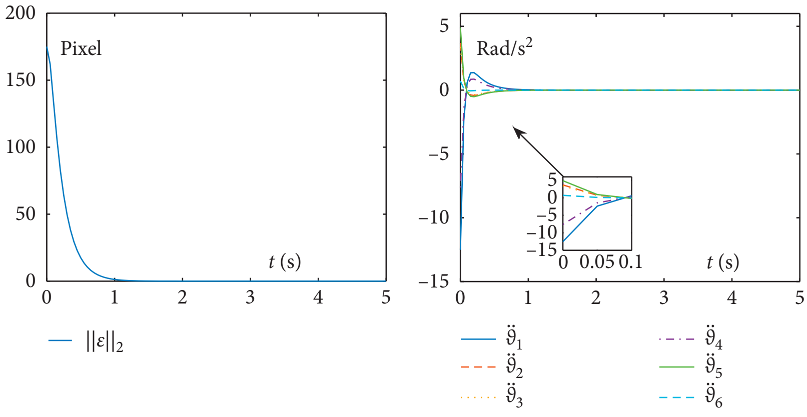
In this section, some existing visual servoing approaches [21, 25–27, 42, 43, 48] are assembled in Table 1 to highlight the superiority of the proposed quadratic programming scheme (11)–(14). The following points can be determined. A majority of the existing techniques [21, 26, 27, 48] utilize the pseudoinverse method to carry out research. These approaches often take no account of joint physical constraints, which may lead to a large generated control signal and even cause damage to the manipulator. On the other hand, it is well known that the pseudoinverse operations involved are computationally onerous. Thirdly, the present research on visual servoing at acceleration level is relatively lacking [21, 26]. Therefore, in terms of joint acceleration, the quadratic programming scheme (11)–(14) avoids the pseudoinverse operation by utilizing the matrix transpose operation and meanwhile takes the joint constraints into account. This demonstrates the superiority of the proposed quadratic programming scheme (11)–(14) (Table 1).



(a) (b) (c)

(d) (e) (f)

Figure 3: Simulation results on the PUMA 560 manipulator driven by the quadratic programming scheme (11)–(14) aided with GRNN (26) for the visual servoing task with execution time $T = 5$s. (a) Actual path and desired point in image space. (b) Time history of 2-norm of error ε. (c) Time history of 2-norm of error $\dot{\varepsilon}$. (d) Time history of joint acceleration. (e) Time history of joint velocity. (f) Time history of joint angle.



(a) (b)

Figure 4: Simulation results on the PUMA 560 manipulator driven by the pseudoinverse method (38) for the visual servoing task with execution time $T = 5$s. (a) Time history of 2-norm of error ε. (b) Time history of joint acceleration.

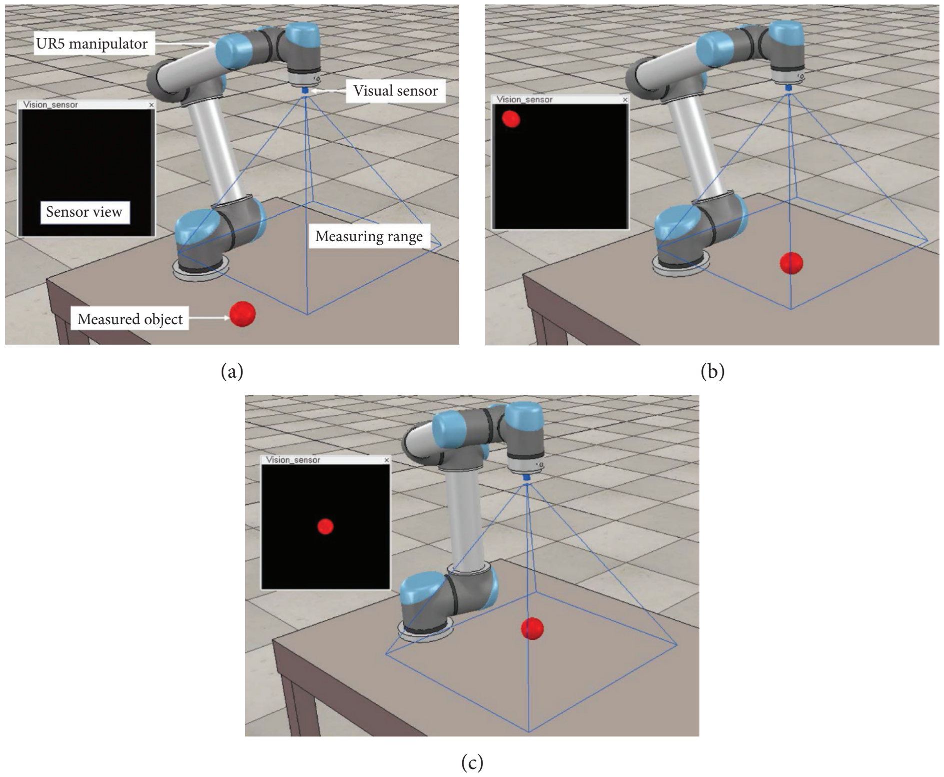


Figure 5: V-rep experiment results on the UR5 manipulator driven by the quadratic programming scheme (11)–(14) aided with GRNN (26) for the visual servoing task. (a) UR5 manipulator installed with a visual sensor. (b) Initial state of the robot visual system. (c) Final state of the robot visual system.

Table 1: Comparisons among different approaches for visual servoing of redundant manipulators.

	Command level	Gradient descent method	Joint velocity constraints	Joint acceleration constraints	Pseudoinverse operation	Number of neurons
Scheme (26)	Acceleration	Yes	Yes	Yes	No	n
Scheme in [21]	Acceleration	No	No	No	Yes	n
Scheme in [25]	Velocity	No	Yes	No	No	$n+2$
Scheme in [26]	Acceleration	No	No	No	Yes	n
Scheme in [27]	Velocity	No	No	No	Yes	n
Scheme in [42]	Velocity	No	Yes	No	No	$n+2$
Scheme in [43]	Velocity	No	Yes	No	No	$n+2$
Scheme in [48]	Velocity	No	No	No	Yes	n

7. Conclusion

In this paper, the vision servoing issue has been formulated as a constrained quadratic programming scheme at the acceleration level with physical constraints considered. Then, a GRNN has been proposed via the gradient descent method and compensation term with the stability analyses provided. After that, simulation examples have been carried out to demonstrate the correctness of theoretical analyses and the validity of the proposed method. Note that the proposed method has resolved the visual servoing issue at the acceleration level and also has considered the joint constraints of the manipulator to guarantee the safe operation of the manipulator. For the further research direction, the authors are going to investigate the uncertain conditions and optimization in the visual system, such as noise suppression [56] or Jacobian estimation [57] and manipulability optimization [58].

Data Availability

The data in the paper are not made public online.

Conflicts of Interest

The authors declare that they have no conflicts of interest.

Acknowledgments

This work was supported in part by the Guangzhou Sport University Innovation and Strengthen Project under Grant 5200080589, in part by the Ministry of Education Industry-Academic Cooperation Collaborative Education Program of China under Grant 201901007048, in part by the Research and Development Foundation of Nanchong (China) under Grant 20YFZJ0018, and in part by the Fundamental Research Funds for the Central Universities under Grant lzujbky-2019-89.

References

[1] Z. Shen, A. Elibol, and N. Y. Chong, "Understanding nonverbal communication cues of human personality traits in human-robot interaction," *IEEE/CAA Journal of Automatica Sinica*, vol. 7, no. 6, pp. 1465–1477, 2020.

[2] T. Li, H. Zhao, and Y. Chang, "Delay-dependent stability in uncalibrated image-based dynamic visual servoing robotic system," *Complexity*, vol. 2018, Article ID 1360874, 14 pages, 2018.

[3] C. Yang, C. Chen, W. He, R. Cui, and Z. Li, "Robot learning system based on adaptive neural control and dynamic movement primitives," *IEEE Transactions on Neural Networks and Learning Systems*, vol. 30, no. 3, pp. 777–787, 2019.

[4] L. Jin, S. Li, L. Xiao, R. Lu, and B. Liao, "Cooperative motion generation in a distributed network of redundant robot manipulators with noises," *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, vol. 48, no. 10, pp. 1715–1724, 2018.

[5] Z. Xie, L. Jin, X. Du, X. Xiao, H. Li, and S. Li, "On generalized RMP scheme for redundant robot manipulators aided with dynamic neural networks and nonconvex bound constraints," *IEEE Transactions on Industrial Informatics*, vol. 15, no. 9, pp. 5172–5181, 2019.

[6] Y. Zhang, S. Li, S. Kadry, and B. Liao, "Recurrent neural network for kinematic control of redundant manipulators with periodic input disturbance and physical constraints," *IEEE Transactions on Cybernetics*, vol. 49, no. 12, pp. 4194–4205, 2019.

[7] J. Zhang, L. Jin, and L. Cheng, "RNN for perturbed manipulability optimization of manipulators based on a distributed scheme: a game-theoretic perspective," *IEEE Transactions on Neural Networks and Learning Systems*, vol. 31, no. 12, p. 5116, 2020.

[8] L. Jin, S. Li, H. M. La, and X. Luo, "Manipulability optimization of redundant manipulators using dynamic neural networks," *IEEE Transactions on Industrial Electronics*, vol. 64, no. 6, pp. 4710–4720, 2017.

[9] C. Yang, C. Zeng, Y. Cong, N. Wang, and M. Wang, "A learning framework of adaptive manipulative skills from human to robot," *IEEE Transactions on Industrial Informatics*, vol. 15, no. 2, pp. 1153–1161, 2019.

[10] H. Huang, T. Zhang, C. Yang, and C. L. P. Chen, "Motor learning and generalization using broad learning adaptive neural control," *IEEE Transactions on Industrial Electronics*, vol. 67, no. 10, pp. 8608–8617, 2020.

[11] D. Huang, C. Yang, Y. Pan, and L. Cheng, "Composite learning enhanced neural control for robot manipulator with output error constraints," *IEEE Transactions on Industrial Informatics*, vol. 17, no. 1, pp. 209–218, 2021.

[12] Y. Kong, R. Zhang, Y. Jiang, and X. Xia, "A repeatable optimization for kinematic energy system with its mobile manipulator application," *Complexity*, vol. 2019, Article ID 8642027, 16 pages, 2019.

[13] H. Zhang, H. Fang, D. Zhang, X. Luo, and Q. Zou, "Adaptive fuzzy sliding mode control for a 3-DOF parallel manipulator with parameters uncertainties," *Complexity*, vol. 2020, Article ID 2565316, 16 pages, 2020.

[14] S. K. Pradhan and B. Subudhi, "Position control of a flexible manipulator using a new nonlinear self-tuning PID controller," *IEEE/CAA Journal of Automatica Sinica*, vol. 7, no. 1, pp. 82–95, 2020.

[15] L. Jin and Y. Zhang, "G2-type SRMPC scheme for synchronous manipulation of two redundant robot arms," *IEEE Transactions on Cybernetics*, vol. 45, no. 2, pp. 153–164, 2015.

[16] 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 Transactions on Cybernetics*, vol. 49, no. 7, pp. 2568–2579, 2019.

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

[18] S. Li, Y. Zhang, and L. Jin, "Kinematic control of redundant manipulators using neural networks," *IEEE Transactions on Neural Networks and Learning Systems*, vol. 28, no. 10, pp. 2243–2254, 2017.

[19] Y. Zhang, S. Li, and X. Zhou, "Recurrent-neural-network-based velocity-level redundancy resolution for manipulators subject to a joint acceleration limit," *IEEE Transactions on Industrial Electronics*, vol. 66, no. 5, pp. 3573–3582, 2019.

[20] S. Boyd and L. Vandenberghe, *Convex Optimization*, Cambridge University Press, Cambridge, UK, 2004.

[21] A. Anwar, W. Lin, X. Deng, J. Qiu, and H. Gao, "Quality inspection of remote radio units using depth-free image-based visual servo with acceleration command," *IEEE Transactions on Industrial Electronics*, vol. 66, no. 10, pp. 8214–8223, 2019.

[22] F. Ke, Z. Li, and C. Yang, "Robust tube-based predictive control for visual servoing of constrained differential-drive mobile robots," *IEEE Transactions on Industrial Electronics*, vol. 65, no. 4, pp. 3437–3446, 2018.

[23] D. Tsai, D. G. Dansereau, T. Peynot, and P. Corke, "Image-based visual servoing with light field cameras," *IEEE Robotics and Automation Letters*, vol. 2, no. 2, pp. 912–919, 2017.

[24] T. Osa, C. Staub, and A. Knoll, "Framework of automatic robot surgery system using visual servoing," in *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems*, Taipei, Taiwan, October 2010.

[25] W. Li, P. W. Y. Chiu, and Z. Li, "An accelerated finite-time convergent neural network for visual servoing of a flexible surgical endoscope with physical and RCM constraints," *IEEE Transactions on Neural Networks and Learning Systems*, vol. 31, no. 12, p. 5272, 2020.

[26] M. Keshmiri, W. Wen-Fang Xie, and A. Mohebbi, "Augmented image-based visual servoing of a manipulator using

acceleration command," *IEEE Transactions on Industrial Electronics*, vol. 61, no. 10, pp. 5444–5452, 2014.

[27] M. Van, S. S. Ge, and D. Ceglarek, "Fault estimation and accommodation for virtual sensor bias fault in image-based visual servoing using particle filter," *IEEE Transactions on Industrial Informatics*, vol. 14, no. 4, pp. 1312–1322, 2018.

[28] P. Cigliano, V. Lippiello, F. Ruggiero, and B. Siciliano, "Robotic ball catching with an eye-in-hand single-camera system," *IEEE Transactions on Control Systems Technology*, vol. 23, no. 5, pp. 1657–1671, 2015.

[29] P. M. Kebria, A. Khosravi, S. M. Salaken, and S. Nahavandi, "Deep imitation learning for autonomous vehicles based on convolutional neural networks," *IEEE/CAA Journal of Automatica Sinica*, vol. 7, no. 1, pp. 136–149, 2020.

[30] Y. Huang, J. Na, X. Wu, and G. Gao, "Approximation-free control for vehicle active suspensions with hydraulic actuator," *IEEE Transactions on Industrial Electronics*, vol. 65, no. 9, pp. 7258–7267, 2018.

[31] Y. Li, S. Li, and B. Hannaford, "A model-based recurrent neural network with randomness for efficient control with applications," *IEEE Transactions on Industrial Informatics*, vol. 15, no. 4, pp. 2054–2063, 2019.

[32] Y. Qi, L. Jin, Y. Wang, L. Xiao, and J. Zhang, "Complex-valued discrete-time neural dynamics for perturbed time-dependent complex quadratic programming with applications," *IEEE Transactions on Neural Networks and Learning Systems*, vol. 31, no. 9, pp. 3555–3569, 2020.

[33] P. S. StanimiroviĆ, ivkoviĆ and Y. Wei, "Recurrent neural network for computing the drazin inverse," *IEEE Transactions on Neural Networks and Learning Systems*, vol. 26, no. 11, pp. 2830–2843, 2015.

[34] P. S. Stanimirovi, M. Ciri, I. Stojanovic, and D. Gerontitis, "Conditions for existence, representations, and computation of matrix generalized inverses," *Complexity*, vol. 2017, Article ID 6429725, 27 pages, 2017.

[35] X. Luo, M. Zhou, Y. Xia, Q. Zhu, A. C. Ammari, and A. Alabdulwahab, "Generating highly accurate predictions for missing QoS data via aggregating nonnegative latent factor models," *IEEE Transactions on Neural Networks and Learning Systems*, vol. 27, no. 3, pp. 524–537, 2016.

[36] X. Luo, H. Wu, H. Yuan, and M. Zhou, "Temporal pattern-aware QoS prediction via biased non-negative latent factorization of tensors," *IEEE Transactions on Cybernetics*, vol. 50, no. 5, pp. 1798–1809, 2020.

[37] L. Cheng, W. Liu, T. Huang, Z.-G. Hou, and M. Tan, "A neural-network-based controller for piezoelectric-actuated stick-slip devices," *IEEE Transactions on Industrial Electronics*, vol. 65, no. 3, pp. 2598–2607, 2018.

[38] S. A. A. Shah, I. Uddin, F. Aziz, S. Ahmad, M. A. Al-Khasawneh, and M. Sharaf, "An enhanced deep neural network for predicting workplace absenteeism," *Complexity*, vol. 2020, Article ID 5843932, 12 pages, 2020.

[39] 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 Transactions on Industrial Informatics*, vol. 15, no. 1, pp. 236–246, 2019.

[40] D. Guo, F. Xu, Z. Li, Z. Nie, and H. Shao, "Design, verification, and application of new discrete-time recurrent neural network for dynamic nonlinear equations solving," *IEEE Transactions on Industrial Informatics*, vol. 14, no. 9, pp. 3936–3945, 2018.

[41] D. Guo, S. Li, and P. S. Stanimirovic, "Analysis and application of modified ZNN design with robustness against harmonic noise," *IEEE Transactions on Industrial Informatics*, vol. 16, no. 7, pp. 4627–4638, 2020.

[42] Y. Zhang, S. Li, B. Liao, L. Jin, and L. Zheng, "A recurrent neural network approach for visual servoing of manipulators," in *Proceedings of the International Conference on Intelligent Manufacturing and Automation*, pp. 614–619, Melbourne, Australia, August 2017.

[43] Y. Zhang and S. Li, "A neural controller for image-based visual servoing of manipulators with physical constraints," *IEEE Transactions on Neural Networks and Learning Systems*, vol. 29, no. 11, pp. 5419–5429, 2018.

[44] Q. Song, Y. Wu, and Y. C. Soh, "Robust adaptive gradient-descent training algorithm for recurrent neural networks in discrete time domain," *IEEE Transactions on Neural Networks*, vol. 19, no. 11, pp. 1841–1853, 2008.

[45] N. Zhang, W. Wu, and G. Zheng, "Convergence of gradient method with momentum for two-Layer feedforward neural networks," *IEEE Transactions on Neural Networks*, vol. 17, no. 2, pp. 522–525, 2006.

[46] L. Jin, Z. Xie, M. Liu, C. Ke, C. Li, and C. Yang, "Novel joint-drift-free scheme at acceleration level for robotic redundancy resolution with tracking error theoretically eliminated," *IEEE/ASME Transactions on Mechatronics*, p. 1, 2020, In press.

[47] S. Hutchinson, G. D. Hager, and P. I. Corke, "A tutorial on visual servo control," *IEEE Transactions on Robotics and Automation*, vol. 12, no. 5, pp. 651–670, 1996.

[48] K. Hashimoto, T. Kimoto, T. Ebine, and H. Kimura, "Manipulator control with image-based visual servo," in *IEEE International Conference on Robotics and Automation*, Sacramento, CA, USA, April 1991.

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

[50] Y. Zhang and S. S. Ge, "Design and analysis of a general recurrent neural network model for time-varying matrix inversion," *IEEE Transactions on Neural Networks*, vol. 16, no. 6, pp. 1477–1490, 2005.

[51] X. Luo, D. Wang, M. Zhou, and H. Yuan, "Latent factor-based recommenders relying on extended stochastic gradient descent algorithms," *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, p. 1, 2019, In press.

[52] X. Luo, M. Zhou, S. Li et al., "Incorporation of efficient second-order solvers into latent factor models for accurate prediction of missing QoS data," *IEEE Transactions on Cybernetics*, vol. 48, no. 4, pp. 1216–1228, 2018.

[53] X. Luo, M. Zhou, S. Li, and M. Shang, "An inherently nonnegative latent factor model for high-dimensional and sparse matrices from industrial applications," *IEEE Transactions on Industrial Informatics*, vol. 14, no. 5, pp. 2011–2022, 2018.

[54] Z. Xie, L. Jin, X. Luo, Z. Sun, and M. Liu, "RNN for repetitive motion generation of redundant robot manipulators: an orthogonal projection-based scheme," *IEEE Transactions on Neural Networks and Learning Systems*, p. 1, 2020, In press.

[55] H. K. Khalil, *Nonlinear Systems*, Prentice-Hall, Englewood Cliffs, NJ, USA, 3rd edition, 2001.

[56] L. Xiao, S. Li, K. Li, L. Jin, and B. Liao, "Co-design of finite-time convergence and noise suppression: a unified neural model for time varying linear equations with robotic applications," *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, vol. 50, no. 12, p. 5233, 2020.

[57] Z. Xie, L. Jin, X. Luo, S. Li, and X. Xiao, "A data-driven cyclic-motion generation scheme for kinematic control of redundant manipulators," *IEEE Transactions on Control Systems Technology*, vol. 29, no. 1, p. 53, 2021.

[58] L. Jin, Z. Zhang, M. Liu, S. Li, L. Xiao, and Z. Yang, "Perturbed manipulability optimization in a distributed network of redundant robots," *IEEE Transactions on Industrial Electronics*, p. 1, 2020, In press.