Industrial Robot: An International Journal

Adaptive neural network visual servoing of dual-arm robot for cyclic motion

Jiadi Qu, Fuhai Zhang, Yili Fu, Guozhi Li, Shuxiang Guo,

Adaptive neural network visual servoing of dual-arm robot for cyclic motion

Jiadi Qu, Fuhai Zhang, Yili Fu, Guozhi Li and Shuxiang Guo
State Key Laboratory of Robotics and System, Harbin Institute of Technology, Harbin, China

Abstract

Purpose – The purpose of this paper is to develop a vision-based dual-arm cyclic motion method, focusing on solving the problems of an uncertain grasp position of the object and the dual-arm joint-angle-drift phenomenon.

Design/methodology/approach – A novel cascade control structure is proposed which associates an adaptive neural network with kinematics redundancy optimization. A radial basis function (RBF) neural network in conjunction with a conventional proportional–integral (PI) controller is applied to compensate for the uncertainty of the image Jacobian matrix which includes the estimated grasp position. To avoid the joint-angle-drift phenomenon, a dual neural network (DNN) solver in conjunction with a PI controller and dual-arm-coordinated constraints is applied to optimize the closed-chain kinematics redundancy.

Findings – The proposed method was implemented on an industrial robotic MOTOMAN with two 7-degrees of freedom robotic arms. Two experiments of carrying a tray repeatedly and turning a steering wheel were carried out, and the results indicate that the closed-trajectories tracking is achieved successfully both in the image plane and the joint spaces with the uncertain grasp position, which validates the accuracy and realizability of the proposed PI-RBF-DNN control strategy.

Originality/value – The adaptive neural network visual servoing method is applied to the dual-arm cyclic motion with the uncertain grasp position of the object. The proposed method enhances the environmental adaptability of a dual-arm robot in a practical manipulation task.

Keywords Adaptive neural network, Dual-arm cyclic motion, Dual neural network, Visual servoing

Paper type Research paper

1. Introduction

The cyclic motion has a wide range of applications in the industrial settings, such as welding, spraying and assembling (Zhang and Zhang, 2012). It requires keeping the joint configurations in accordance with the initial state and the finished state when the end-effector achieves a closed-path task (Xiao and Zhang, 2013). Furthermore, the dual-arm [a humanoid robot with two 7-degrees of freedom (DOF) arms] cyclic motion (Zhang *et al.*, 2015) is usually applied in service settings, such as clapping, tray carrying repeatedly and wheel turning. Compared with the single-arm cyclic motion, a closed-chain kinematics redundancy resolution for two 7-DOF arms has to be considered to avoid the joint-angle-drift phenomenon (Zhang *et al.*, 2008).

Due to the service setting generally being a complex environment, it is necessary to introduce visual servoing techniques to the dual-arm cyclic motion to enhance the environmental adaptability. To the best of our knowledge, a contribution to the literature regarding vision-based dual-arm cyclic motion does not exist. In the visual servoing control, the image Jacobian matrix needs to be established (Chaumette and Hutchinson, 2006); this is related to the parameters of the camera calibration, the depth of the feature point and the grasp position. Inexact parameters will lead to uncertainty of the image Jacobian matrix, which is capable of causing a task failure (Ma and Su, 2015). Particularly, the exact grasp position is crucial to the robot manipulation task. However, it is difficult to obtain the real value of the grasp position in an actual application. If a specified grasp position is explicitly provided, platform flexibility will be greatly reduced. Thus, the uncertainty of the image Jacobian matrix caused by the uncertain grasp position needs to be taken into account. Based on the above discussion, it is clear that the two problems of the uncertain grasp position and joint-angle-drift phenomenon must be resolved for reliable task completion.

The estimation methods in existing literatures for solving the uncertainty of the image Jacobian matrix can be divided into three categories:

1. The recursive Jacobian matrix estimation method: the overall estimation of image Jacobian matrix is used regardless of the uncertainties from the camera calibration, the depth or the grasp position, for example, the weighted Broyden method; the recursive least-square method; the recursive Gauss–Newton method; and the dynamic quasi-Newton method (Piepmeier *et al.*, 2004).
2. Adaptive image Jacobian matrix method: the adaptive law, by satisfying the Lyapunov stability condition, is designed to estimate the unknown parameters online based on linear parameterization of the image Jacobian matrix.

The current issue and full text archive of this journal is available on Emerald Insight at: **www.emeraldinsight.com/0143-991X.htm**

Industrial Robot: An International Journal
44/2 (2017) 210–221

[DOI 10.1108/IR-06-2016-0154]

This work is supported by the National Natural Science Foundation of China (Grant No. 61673134) and the Self-Planned Task (No. SKLRS201501C) of State Key Laboratory of Robotics and System (HIT).

Received 7 June 2016
Revised 14 October 2016
Accepted 23 October 2016

Wang *et al.* (2012) and Liang *et al.* (2015) proposed a depth-independent image Jacobian matrix which can be linearly parameterized and designed an adaptive visual controller to compensate for the uncertain parameters of camera calibration, depth and robot dynamics.

3 Adaptive neural networks: the online neural network learning was used for the adaptive visual servoing controller to compensate for the uncertainties of the Jacobian matrix.

Notably, the above linear parameterization was not needed when adaptive neural networks are used. Zhao and Cheah (2009) proposed an adaptive radial basis function (RBF) neural network controller for a multi-fingered robot hand that compensates for the uncertainties in kinematics, Jacobian matrices and dynamics. For a planar robot manipulator, Yu and Moreno-Armendariz (2005) used a RBF neural network combined with a robust controller to compensate for uncertainties in dynamics. With the compensation for dynamic uncertainties using a RBF neural network, Xie *et al.* (2011) compared RBF networks with other networks [back propagation (BP), Kohonen networks, etc.], and the comparison of the results indicate that the non-linear function approximation capability of RBF networks is prior to that of other networks. Zhu *et al.* (2008) validated that RBF networks can effectively improve the robustness of the controller when the system parameters have a large uncertainty. Therefore, this paper will adopt a RBF network to deal with the uncertainty of the grasp position. In the above-mentioned contributions, the parameter uncertainties including the camera calibration, depth and dynamics were discussed in the context of a variety of applications; however, the uncertainty of the grasp position for robot manipulation remains unfounded.

On the other hand, the non-unique inverse kinematic solutions of two 7-DOF redundant arms led to the uncertainties of closed-chain joint configurations. Cai and Zhang (2012) formulated the inverse redundancy kinematic problem into a quadratic programming (QP) form. To solve the QP problem, the neural network approach is introduced. In Xia and Wang (2000), a linear variational inequality-based primal–dual neural network (LVI-PDNN) was designed with simple piecewise linear dynamics. Zhang *et al.* (2013) used the BP neural network and the Tank–Hopfield neural network with online learning to solve the kinematic problem; Khoogar *et al.* (2011) presented a dual neural network (DNN) to deal with the limit constraints. Zhang *et al.* (2009) presented a comparison of a DNN, LVI-PDNN and a simplified LVI-PDNN to evaluate these solutions for the QP problem for online cyclic motion of redundant robot manipulators. Simulation results showed that all three methods were effective for the joint-angle-drift problem of robot manipulators.

To solve the uncertainty of the grasp position of the object in a dual-arm cyclic-motion application, this paper proposed a novel cascade control strategy. A RBF neural network in conjunction with a conventional proportional–integral (PI) controller is applied to compensate the uncertainty of the image Jacobian matrix which includes an estimated grasp position. A DNN solver in conjunction with a PI controller and dual-arm coordinated constraints is applied to optimize the closed-chain kinematics redundancy for achieving the dual-arm cyclic motion.

Notation: $[*; *] = [*^T, *^T]^T$, R^n = real n-vectors, $R^{n\times m}$ = real $n \times m$ matrices, $0_{n\times m}$ = zero $n \times m$ matrix, $I_{n\times m}$ = $n \times m$ identity matrix. $\mathcal{J}$ = Jacobian matrix, T = transformation matrix, q = joint angle vector, R = rotation matrix, x = pose vector (position and orientation). Superscripts and subscripts: b = robot base, c = camera, d = desire, o = object, img = image, l = left, r = right, $dual$ = dual arm, nn = neural network, pi = PI controller, $+$ = upper limit and $-$ = lower limit.

2. Problem formulation

2.1 Dual-arm robot model and coordinated constraints

The industrial robotic platform MOTOMAN SDA5F with two 7-DOFs robotic arms is used in this paper. The frames of all joints are shown in Figure 1. The forward kinematics mappings of the dual-arm robot are given as follows:

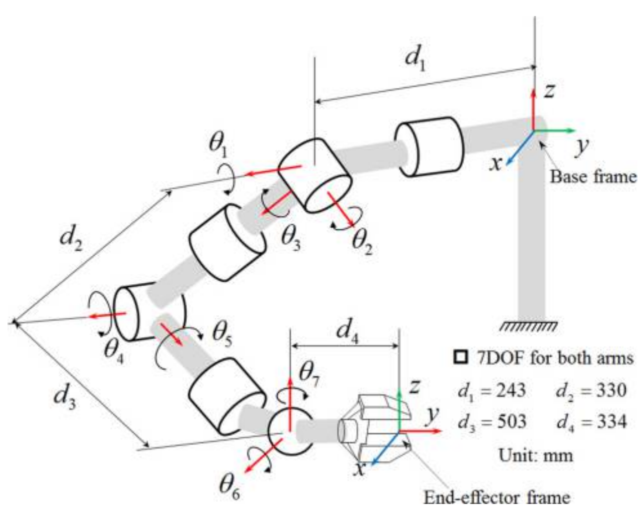
$$\begin{bmatrix} x_l \\ x_r \end{bmatrix} = \begin{bmatrix} {}^b_lT(q_l) \\ {}^b_rT(q_r) \end{bmatrix}, \quad \begin{bmatrix} \dot{x}_l \\ \dot{x}_r \end{bmatrix} = \begin{bmatrix} \mathcal{J}_l(q_l) & 0 \\ 0 & \mathcal{J}_r(q_r) \end{bmatrix} \begin{bmatrix} \dot{q}_l \\ \dot{q}_r \end{bmatrix} \tag{1}$$

where $x, \dot{x} \in R^6$, $q, \dot{q} \in R^7$, $T \in R^{4\times 4}$, $\mathcal{J} \in R^{6\times 7}$ and $q = [\theta_1; \theta_2; \theta_3; \theta_4; \theta_5; \theta_6; \theta_7]$. Dual-arm joint angle and velocity vector are defined as $q_{dual} = [q_l; q_r] \in R^{14}$ and $\dot{q}_{dual} = [\dot{q}_l; \dot{q}_r] \in R^{14}$; accordingly, the limit vectors are defined as $q^{\mp}_{dual} = [q^{\mp}_l; q^{\mp}_r] \in R^{14}$ and $\dot{q}^{\mp}_{dual} = [\dot{q}^{\mp}_l; \dot{q}^{\mp}_r] \in R^{14}$. In addition, the linear mapping function from the control output q_{dual} (rad) to the input command of robot $Y \in R^{14}$ (pulse) is given:

$$Y = \frac{Y^+ - Y^-}{q_{dual}{}^+ - q_{dual}{}^-}(q_{dual} - q_{dual}{}^-) \tag{2}$$

where (q^-_{dual}, q^+_{dual}) and (Y^-, Y^+) can be found by the robot manual.

Figure 1 Kinematic structure of the dual-arm robot MOTOMAN SDA5F



As shown in Figure 2, several frames are introduced. The pose constraints between object and left end-effector are given as follows:

$$p_l = p_o + {}^b_lR \times p_{ol} \tag{3}$$

$$ {}^b_lR = {}^b_oR^o_lR \tag{4}$$

where $p_{ol} \in R^3$ is the translation vector between Σ_o and Σ_l, which is the grasp position vector of left end-effector to the object. Due to the relative lack of motion between the object and left end-effector at the run time, p_{ol} is a constant vector which can be obtained at the initial state of the robot. However, the real value of p_{ol} is difficult to be obtained in a real environment, and the estimated value $\hat{p}_{ol}$ may be used. Similarly, the pose constraints between left and right end-effector are as follows:

$$p_r = p_l + {}^b_lR \times p_{rl} \tag{5}$$

$$ {}^b_rR = {}^b_lR^l_rR \tag{6}$$

where $p_{rl} \in R^3$ is the translation vector between Σ_r and Σ_l, which is a constant vector because of the relative lack of motion between the two end-effectors at the run time, which can be obtained accurately at the initial state. Differentiating equations (3)-(6) and combining into two equations:

$$J_l \times \dot{q}_l = \begin{bmatrix} \dot{p}_o \\ \omega_o \end{bmatrix} + \begin{bmatrix} \hat{C} \\ 0_{3\times7} \end{bmatrix} \times \dot{q}_l, \quad with \quad \hat{C} = \frac{\partial\left[{}^b_lR(q_l) \times \hat{p}_{ol}\right]}{\partial q_l} \tag{7}$$

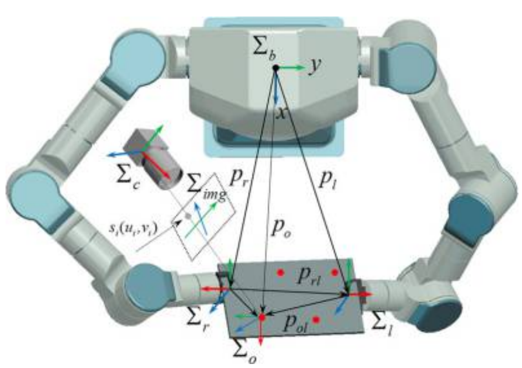
$$J_r \times \dot{q}_r = J_l \times \dot{q}_l + \begin{bmatrix} D \\ 0_{3\times7} \end{bmatrix} \times \dot{q}_l, \quad with \quad D = \frac{\partial\left[{}^b_lR(q_l) \times p_{rl}\right]}{\partial q_l} \tag{8}$$

where $\hat{C} \in R^{3\times7}$ and $D \in R^{3\times7}$ can be calculated by the following expression:

$$\frac{\partial[R(t) \times N]}{\partial q} = \left[I_{3\times3} \quad sk(R(t) \times N)\right]\begin{bmatrix} R(t) & 0 \\ 0 & R(t) \end{bmatrix} J(q(t))$$

where N is an arbitrary vector, sk is a matrix operator and $n = [n_1; n_2; n_3]$.

Figure 2 Dual-arm-coordinated constraints and coordinate frames



Let $\dot{x} = [\dot{p};\omega]$ and by equation (1), equations (7) and (8) can be rewritten as follows:

$$\hat{\dot{x}}_l = \dot{x}_o - \begin{bmatrix} \hat{C} \\ 0 \end{bmatrix} \dot{q}_l \tag{9}$$

$$\hat{\dot{x}}_r = \hat{\dot{x}}_l + \begin{bmatrix} D \\ 0 \end{bmatrix} \dot{q}_l \tag{10}$$

Equation (10) is the dual-arm-coordinated constraint. Due to the estimated value $\hat{p}_{ol}$ being used in equation (9), although $\dot{x}_o$ is known accurately, only the estimated values of the velocity screws of two end-effectors $\hat{\dot{x}}_l$ and $\hat{\dot{x}}_r$ can be obtained.

2.2 Image Jacobian matrix analysis

As shown in Figure 2, a fixed camera is used to observe the object gripped by the end-effectors. The image Jacobian matrix denotes the velocity-level mappings from the feature points of the object in the image plane to the pose of the left end-effector in the Cartesian space.

The coordinate transformation of the feature point i from the image plane Σ_{img} to the camera frame Σ_c can be obtained as follows:

$$\begin{bmatrix} s_i \\ 1 \end{bmatrix} = \frac{1}{z_i} \times M \times \begin{bmatrix} X_i^c \\ 1 \end{bmatrix} \tag{11}$$

where $s_i(u_i, v_i) \in R^2$, $X({}^cx, {}^cy, {}^cz) \in R^3$, $M \in R^{3\times4}$ is the camera intrinsic matrix and z_i is the depth of the feature point i. Similarly, the coordinate transformation of the feature point i from Σ_c to Σ_l can be obtained as follows:

$$\begin{bmatrix} X_i^c \\ 1 \end{bmatrix} = ({}^b_cT)^{-1} \times {}^b_lT \begin{bmatrix} \hat{p}_{ol} \\ 1 \end{bmatrix} \tag{12}$$

where ${}^b_cT$ is the camera extrinsic matrix; ${}^b_lT$ is the forward kinematics mapping of left end-effector in equation (1).

Differentiating equations (11) and (12), the following combination is achieved:

$$\dot{s}_i = \frac{1}{z_i}\begin{bmatrix} m_1 - u_i \times m_3 \\ m_2 - v_i \times m_3 \end{bmatrix} {}^c_bR\left[I_3 \quad -sk\left({}^b_lR \times \hat{p}_{ol}\right)\right]\dot{x}_l \tag{13}$$

where $m_j \in R^{1\times3}, j = 1, 2, 3$ is the j-th row, the first three columns of M. Because the estimated value $\hat{p}_{ol}$ is used, the estimated image Jacobian matrix of the feature point i is defined as follows:

$$\hat{J}_{imgi} = \frac{1}{z_i}\begin{bmatrix} m_1 - u_i \times m_3 \\ m_2 - v_i \times m_3 \end{bmatrix} {}^c_bR\left[I_3 \quad -sk\left({}^b_lR \times \hat{p}_{ol}\right)\right] \in R^{2\times6} \tag{14}$$

According to Chaumette and Hutchinson (2006), image-based visual servoing (IBVS) only has a local asymptotic stability when the dimension of the image coordinates is greater than the number of 6-DOF of cameras. Thus, the four feature points of the object (8-DOF) are considered. The image feature vector is defined as $s = [s_1; s_2; s_3; s_4] \in R^8$, and the desired trajectory is $s_d(t) =$

Industrial Robot: An International Journal
Volume 44 · Number 2 · 2017 · 210–221

$[s_{1d}(t); s_{2d}(t); s_{3d}(t); s_{4d}(t)] \in R^8$. Then, the overall estimated image Jacobian matrix of four points is:

$$\dot{s} = \hat{\mathcal{J}}_{img}\dot{x}_l, \quad with \quad \hat{\mathcal{J}}_{img} = \left[\hat{\mathcal{J}}_{img1}; \hat{\mathcal{J}}_{img2}; \hat{\mathcal{J}}_{img3}; \hat{\mathcal{J}}_{img4}\right] \in R^{8\times 6} \tag{15}$$

The uncertainty of the grasp position p_{ol} can degrade the performance and even destabilize the dual-arm robot system. An accurate estimation of the image Jacobian matrix is essential for the visual servoing controller design. The aim of this paper is to ensure the stability of the image error in the presence of uncertainties associated with the image Jacobian matrix.

3. Visual servoing development for dual-arm cyclic motion

3.1 The proposed control structure

The proposed cascade control structure is shown in Figure 3, which includes two parts: visual servoing and closed-chain kinematic optimization. In the visual servoing control, the visual tracking error e is projected to the desired pose error of the left end-effector $\boldsymbol{\xi}$ by the estimated image Jacobian matrix. Then, the PI controller is used to generate the pose screw $\dot{x}_{pi}$ of the left end-effector. Further, a RBF neural network compensation control is used for the uncertainty of the grasp position to obtain the pose screw $\dot{x}_{nn}$. Finally, the desired pose screw of the left end-effector $\dot{x}_{ld}$ is obtained. In the closed-chain kinematic optimization, the current pose error of the left end-effector $\boldsymbol{\zeta}$ is defined. Then, the PI controller and dual-arm-coordinated constraints are used to generate the pose screws $(\dot{x}_l, \dot{x}_r)$ of two end-effectors. According to the initial joint configurations of the dual-arm robot, a DNN solver is used to optimize the closed-chain kinematic redundancy to avoid the joint-angle-drift phenomenon.

3.2 Adaptive neural network control design

The aim of this section is to achieve the desired closed trajectories tracking of the image feature points. Firstly, given the desired closed trajectories $s_d(t)$ in the image space, the image error is:

$$e = s_d(t) - s(t) \in R^8 \tag{16}$$

Then, the time derivative of equation (16) is obtained as, follows:

$$\dot{e} = \mathcal{J}_{img}(\dot{x}_{ld} - \dot{x}_l) \tag{17}$$

In some certain Δt, the estimated Jacobian matrix $\hat{\mathcal{J}}_{img}$ is introduced, and the desired pose error $\boldsymbol{\xi}$ of the left end-effector mapping from the image space to the Cartesian space is as follows:

$$\xi = \hat{\mathcal{J}}^{+}_{img} \times e \in R^6 \tag{18}$$

where $\hat{\mathcal{J}}^{+}_{img} \in R^{6\times 8}$ is the Moore–Penrose pseudo inverse of the estimated Jacobian matrix $\hat{\mathcal{J}}_{img}$ with $\hat{\mathcal{J}}^{+}_{img} = (\hat{\mathcal{J}}^{T}_{img}\hat{\mathcal{J}}_{img})^{-1}\hat{\mathcal{J}}^{T}_{img}$. Differentiating the desired pose error ξ in equation (18), then equations (16)-(17) are substituted into $\dot{\boldsymbol{\xi}}$, and we have:

$$\dot{\xi} = \dot{\hat{\mathcal{J}}}^{+}_{img} \times e + \hat{\mathcal{J}}^{+}_{img} \times \dot{e} = -\dot{x}_l + \Delta_{\xi}(\dot{x}_l, s, e) \tag{19}$$

where $\Delta_{\xi}(\dot{x}_l, s, e) = \dot{\hat{\mathcal{J}}}^{+}_{img} \times e + \hat{\mathcal{J}}^{+}_{img} \times \mathcal{J}_{img}\dot{x}_{ld} + \hat{\mathcal{J}}^{+}_{img} \times \tilde{\mathcal{J}}_{img}\dot{x}_l \in R^6$ is the visual servoing modeling error and $\tilde{\mathcal{J}}_{img} = \mathcal{J}_{img} - \hat{\mathcal{J}}_{img}$ is the estimation error of the image Jacobian matrix. To regulate the desired pose error $\boldsymbol{\xi}$, the desired velocity screw in the Cartesian space is designed as follows:

$$\dot{x}_{ld} = \dot{x}_{pi} - \dot{x}_{nn} \tag{20}$$

where $\dot{x}_{pi}$ is the PI control signal:

$$\dot{x}_{pi} = K_{p1}\xi + K_{i1}\int \xi \tag{21}$$

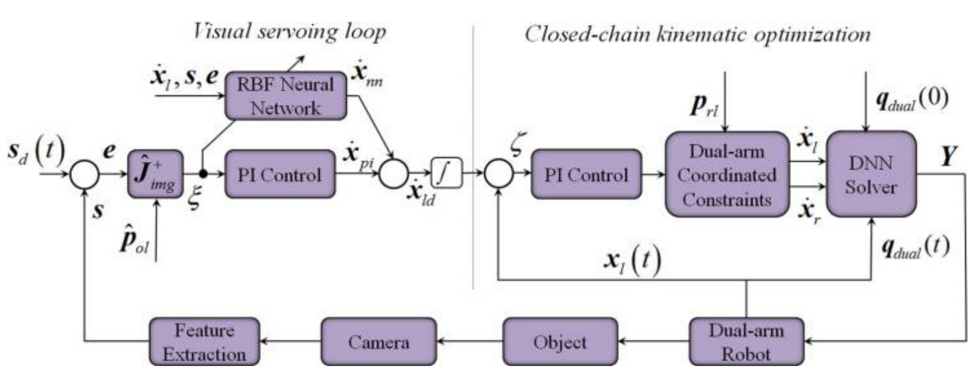
with $K_{p1} > 0, K_{i1} > 0$ as the control parameter matrices. $\dot{x}_{nn}$ is the feed-forward neural network control to compensate for the visual servoing modeling error $\Delta_{\xi}(\dot{x}_l, s, e)$. If $\dot{x}_{nn} = 0$, it indicates that the visual servoing is only the general PI control without the compensation of system uncertainty. To eliminate the system error $\Delta_{\xi}(\dot{x}_l, s, e)$, the control variable $\dot{x}_{nn}$ needs to be designed, and the neural network control needs to be adopted to approximate the nonlinear function $\Delta_{\xi}(\dot{x}_l, s, e)$.

The system error vector of visual servoing is $\Sigma = [\int \xi; \xi] \in R^{12}$. Furthermore, the state equation of the system error is obtained by equations (19) and (21):

$$\dot{\Sigma} = A \times \Sigma + B \times (\dot{x}_{nn} - \Delta_{\xi}(\dot{x}_l, s, e)) \tag{22}$$

where the stated matrix $A \in R^{12\times 12}$ and the input matrix $B \in R^{12\times 6}$ are:

Figure 3 The proposed cascade control structure of vision-based dual-arm cyclic-motion with the RBF neural network compensation and DNN solver



$$A = \begin{bmatrix} 0_{6\times6} & I_{6\times6} \\ -K_{i1} & -K_{p1} \end{bmatrix}, B = \begin{bmatrix} 0_{6\times3} & 0_{6\times3} \\ I_{6\times3} & 0_{6\times3} \end{bmatrix},$$

The RBF network is used to approximate Δ_ξ. The input vector of the RBF network is $x_{in} = [\dot{x}_l; s; e] \in R^{22}$, and $h = [h_1; h_2; \ldots; h_n] \in R^n$ is the radial basis vector with Gaussian function h_i:

$$h_i = exp\frac{\|x_{in} - c_i\|^2}{b_i^2}, \quad i = 1, 2, \ldots, n \tag{23}$$

where c_i is the center and b_i is the distance of the i-th neuron of the basis function. The output vector Δ_ξ of the RBF network is:

$$\Delta_\xi(\dot{x}_l, s, e) = W^T h(x_{in}) \tag{24}$$

where $W \in R^{n\times6}$ is the weight matrix, and n is the neuron number of the hidden layer. The approximate error $\boldsymbol{\varepsilon}$ of the nonlinear function Δ_ξ is introduced as follows:

$$\Delta_\xi(\dot{x}_l, s, e) = W^T h(x_{in}) + \varepsilon \tag{25}$$

The following assumptions are stated for the approximation error ε:

- *Assumption* 1: The optimal weight matrix W^* is defined on the compact set Θ, and the upper bound of the approximation error can be defined as:

$$\varepsilon^* = \sup_{x_{in} \in \Theta} \|\Delta_\xi(\dot{x}_l, s, e) - W^T h(x_{in})\|$$

Thus, the approximation error ε that corresponds to the optimal weights W^* is bounded by $\|\varepsilon\| \le \varepsilon^*$:

- *Assumption* 2: The optimal weight W^* is bounded by a known positive value $\|W^*\|_F \le W_{max}$.

The RBF neural network control is designed as:

$$\dot{x}_{nn} = \hat{W}^T h(x_{in}) - K_r(\|\hat{W}\|_F + W_{max})(\|\Sigma\|/\|r\|)r \tag{26}$$

where $\hat{W}$ is the estimation matrix of W, the estimation errors are defined as $\widetilde{W} = W - \hat{W}$, the last term in equation (26) is the robustifying signal with a diagonal matrix $K_r > 0$ and $r = (\Sigma^T PB)^T \in R^6$. $P \in R^{12\times12}$ is the positive definite solution for the Lyapunov equation $A^T P + PA + Q = 0$ with a positive definite matrix $Q \in R^{12\times12}$.

The network is trained online by the following adaptive law:

$$\dot{\hat{W}} = \Gamma h r^T + \kappa\Gamma\|\Sigma\|\hat{W} \tag{27}$$

where $\Gamma > 0$, $\kappa > 0$ are the adaptation design parameters.

By equations (21), (26) and (27), the desired velocity screw $\dot{x}_{ld}$ in equation (20) is obtained, thus the desired pose screw x_{ld} can be obtained by the integral operation. The following theorem is given about the stability of the visual servoing and weight matrix of the network:

- *Theorem 1: For the RBF neural network control law* (26) *with the weight adaptation laws* (27), *the system error* Σ *and the neural network weights* $\widetilde{W}$ *are uniformly ultimately bounded in the compact set* $x_{in} \in \Theta$.

Proof: First, the approximation error of the uncertainty function Δ_ξ by the RBF neural network is defined as:

$$e_{nn} = \Delta_\xi(\dot{x}_l, s, e) - \hat{W}^T h(x_{in})$$

Substituting equation (25) into the above equation:

$$e_{nn} = \varepsilon - \widetilde{W}^T h(x_{in})$$

Constructing the Lyapunov candidate function:

$$V_{nn} = \frac{1}{2}\Sigma^T P\Sigma + \frac{1}{2\gamma}tr\left(\widetilde{W}^T\widetilde{W}\right)$$

Differentiating V_{nn} along the error dynamics (22):

$$\begin{aligned}\dot{V}_{nn} &= \frac{1}{2}[\Sigma^T P\dot{\Sigma} + \dot{\Sigma}^T P\Sigma] + \frac{1}{\gamma}tr\left(\dot{\widetilde{W}}^T\widetilde{W}\right) \\ &= -\frac{1}{2}\Sigma^T Q\Sigma + \varepsilon^T B^T P\Sigma - h^T\widetilde{W}B^T P\Sigma + \frac{1}{\gamma}tr\left(\dot{\widetilde{W}}^T\widetilde{W}\right)\end{aligned}$$

Considering $h^T\widetilde{W}B^T P\Sigma = tr[B^T P\Sigma h^T\widetilde{W}]$, we have:

$$\dot{V}_{nn} = -\frac{1}{2}\Sigma^T Q\Sigma + \frac{1}{\gamma}tr\left(-\gamma B^T P\Sigma h^T\widetilde{W} + \dot{\widetilde{W}}^T\widetilde{W}\right) + \varepsilon^T B^T P\Sigma$$

Substituting equation (27) into $\dot{V}_{nn}$:

$$\dot{V}_{nn} = -\frac{1}{2}\Sigma^T Q\Sigma + k_1\|x_{in}\|tr\left(\hat{W}^T\widetilde{W}\right) + \varepsilon^T B^T P\Sigma$$

By using the inequalities, $tr[\tilde{\Sigma}^T(\Sigma - \tilde{\Sigma})] \le \|\tilde{\Sigma}\|_F\|\Sigma\|_F - \|\tilde{\Sigma}\|_F^2$, then:

$$\begin{aligned}tr\left[\hat{W}^T\widetilde{W}\right] = tr\left[\widetilde{W}^T\hat{W}\right] = tr\left[\widetilde{W}^T\left(W^* + \widetilde{W}\right)\right] \le \|\widetilde{W}\|_F\|W^*\|_F \\ - \|\widetilde{W}\|_F^2\end{aligned}$$

The following equality is used:

$$-k_1\|\widetilde{W}\|_F\omega_{max} + k_1\|\widetilde{W}\|_F^2 = k_1\left(\|\widetilde{W}\|_F - \frac{\omega_{max}}{2}\right)^2 - \frac{k_1}{4}\omega_{max}^2$$

Then:

$$\begin{aligned}\dot{V}_{nn} &\le -\frac{1}{2}\Sigma^T Q\Sigma + k_1\|\Sigma\|\left(\|\widetilde{W}\|_F\|W^*\|_F - \|\widetilde{W}\|_F^2\right) + \varepsilon^T B^T P\Sigma \\ &\le -\frac{1}{2}\lambda_{min}(Q)\|\Sigma\|^2 + k_1\|\Sigma\|\,\|\widetilde{W}\|_F\|W^*\|_F - k_1\|\Sigma\|\,\|\widetilde{W}\|_F^2 \\ &\quad + \|\varepsilon_0\|\lambda_{max}(P)\|\Sigma\| = -\|\Sigma\|\left(\frac{1}{2}\lambda_{min}(Q)\|\Sigma\| + k_1\left(\|\widetilde{W}\|_F - \frac{\omega_{max}}{2}\right)^2 \right. \\ &\quad \left. - \frac{k_1}{4}\omega_{max}^2 - \|\varepsilon_0\|\lambda_{max}(P)\right)\end{aligned}$$

To ensure $\dot{V}_{nn} \le 0$, the following conditions need to be satisfied:

$$\begin{aligned}&\frac{1}{2}\lambda_{min}(Q)\|\Sigma\| \ge \|\varepsilon_0\|\lambda_{max}(P) + \frac{k_1}{4}\omega_{max}^2 \\ &or\ k_1\left(\|\widetilde{W}\|_F - \frac{\omega_{max}}{2}\right)^2 \ge \frac{k_1}{4}\omega_{max}^2 + \|\varepsilon_0\|\lambda_{max}(P)\end{aligned}$$

Then, $\dot{V}_{nn}$ is negative if:

$$\|\Sigma\| \geq \frac{2}{\lambda_{min}(Q)}\left(\|\varepsilon_0\|\lambda_{min}(P) + \frac{k_1}{4}\omega_{max}^2\right)$$
$$or \quad \|\widetilde{W}\|_F \geq \frac{\omega_{max}}{2} + \sqrt{\frac{1}{k_1}\left(\|\varepsilon_0\|\lambda_{max}(P) + \frac{k_1}{4}\omega_{max}^2\right)}$$

Thus, the system tracking error Σ and weight matrices $\widetilde{W}$ are uniformly ultimately bounded.

3.3 Dual neural network-based closed-chain kinematic optimization

The current pose error ζ of the left end-effector between x_{ld} in previous section and x_l from the robot feedback is:

$$\boldsymbol{\zeta} = x_{ld} - x_l \tag{28}$$

To regulate the error $\boldsymbol{\zeta}$, the PI control is used to obtain the velocity screw of the left end-effector:

$$\dot{x}_l = K_{p2}\zeta + K_{i2}\int \zeta \tag{29}$$

with $K_{p2} > 0$ and $K_{i2} > 0$ as the control parameter matrices. Further, the velocity screw of the right end-effector $\dot{x}_r$ can be obtained by equation (10).

To achieve the dual-arm cyclic-motion, it needs to satisfy the following condition:

$$when \begin{bmatrix} x_l(t_f) = x_l(0) \\ x_r(t_f) = x_r(0) \end{bmatrix}, \quad then \quad q_{dual}(t_f) = q_{dual}(0).$$

where 0 and t_f are the time instant of initial and final, respectively. It is expected that the closed trajectories of two end-effectors in Cartesian space may yield the closed trajectories in the joint space. To achieve the drift-free closed-chain redundancy resolution, Zhang *et al.* (2009) presented the above problem as the following quadratic program (QP) with the physical constraints:

$$\begin{cases} \text{minimize} & \frac{1}{2}\dot{q}_{dual}^T \cdot \dot{q}_{dual} + [\lambda(q_{dual}(t) - q_{dual}(0))]^T \cdot \dot{q}_{dual}, \\ \text{subject to} & \begin{bmatrix} \mathcal{J}_l & 0_{6\times7} \\ 0_{6\times7} & \mathcal{J}_r \end{bmatrix} \cdot \dot{q}_{dual} = \begin{bmatrix} \dot{x}_l \\ \dot{x}_r \end{bmatrix}, \\ & \varsigma^- \leq \dot{q}_{dual} \leq \varsigma^+ \end{cases} \tag{30}$$

with:

$$\varsigma^- = max\{\dot{q}_{dual}^-, \mu(q_{dual}^- - q_{dual}(t))\},$$
$$\varsigma^+ = min\{\dot{q}_{dual}^+, \mu(q_{dual}^+ - q_{dual}(t))\},$$

The dynamic bound constraint $\varsigma^- \leq \dot{q}_{dual} \leq \varsigma^+$ is a unified manner, where the following transformation from q_{dual} to $\dot{q}_{\mathrm{dual}}$ is used:

$$\mu(q_{dual}^- - q_{dual}(t)) \leq \dot{q}_{dual} \leq \mu(q_{dual}^+ - q_{dual}(t))$$

where $\mu > 0$ is the intensity coefficient. The joint-angle-drift error of dual arm $(q_{dual}(t) - q_{dual}(0))$ is introduced in equation (30), and $\lambda > 0$ is the convergence rate, especially if $\lambda = 0$, the drift-free closed-chain redundancy resolution is ineffective.

Next, a DNN-based QP solver for drift-free redundancy resolution of the dual-arm robot will be presented for solving the QP problem [equation (30)], and the effectiveness of the DNN solver has also been verified for the redundancy resolution of a single-arm robot manipulation (Zhang *et al.*, 2009).

Firstly, the equality and inequality constraints in equation (30) are combined into one bilateral constraint, and the following notations are defined as:

$$\gamma^-: = \begin{bmatrix} \dot{x}_l \\ \dot{x}_r \\ \varsigma^- \end{bmatrix}, \quad \gamma^+: = \begin{bmatrix} \dot{x}_l \\ \dot{x}_r \\ \varsigma^+ \end{bmatrix}, \quad \mathcal{J}_{dual} = \begin{bmatrix} \mathcal{J}_l & 0_{6\times7} \\ 0_{6\times7} & \mathcal{J}_r \\ I_{14\times7} & I_{14\times7} \end{bmatrix} \tag{31}$$

where $\mathcal{J}_{dual} \in R^{26\times14}$ is defined as the dual-arm robot Jacobian matrix. Thus, equation (30) can be rewritten as:

$$\begin{cases} \text{minimize} & \frac{1}{2}\dot{q}_{dual}^T \times \dot{q}_{dual} + [\lambda(q_{dual}(t) - q_{dual}(0))]^T \times \dot{q}_{dual}, \\ \text{subject to} & \gamma^- \leq \mathcal{J}_{dual} \times \dot{q}_{dual} \leq \gamma^+. \end{cases} \tag{32}$$

Secondly, equation (32) can be considered as a parametric optimization problem. $\dot{q}_{dual}$ is a solution to equation (32) if and only if there is a dual decision variable vector $u \in R^{26}$ such that:

$$\dot{q}_{dual} - \mathcal{J}_{dual}^T u + \lambda(q_{dual}(t) - q_{dual}(0)) = 0$$

and:

$$\begin{cases} [\mathcal{J}_{dual} \times \dot{q}_{dual}]_i = \gamma_i^-, & u_i > 0, \\ \gamma_i^- \leq [\mathcal{J}_{dual} \times \dot{q}_{dual}]_i \leq \gamma_i^+, & u_i = 0, \\ [\mathcal{J}_{dual} \times \dot{q}_{dual}]_i = \gamma_i^+, & u_i < 0 \end{cases} \tag{33}$$

In addition, equation (33) is equivalent to piecewise linear equation $\mathcal{J}_{dual}\dot{q}_{dual} = g(\mathcal{J}_{dual}\dot{q}_{dual} - u)$, where $g_\Omega(\times)$ is a projection operator from R^{26} onto $\Omega: = \{u|\gamma^- \leq u \leq \gamma^+\} \in R^{26}$, and the i-th output of $g_\Omega(u)$ is defined as:

$$g_\Omega(u_i) = \begin{cases} \gamma_i^-, & u_i < \gamma_i^-, \\ u_i, & \gamma_i^- \leq u_i \leq \gamma_i^+ \quad i = 1, \ldots, 26 \\ \gamma_i^+, & u_i > \gamma_i^+, \end{cases} \tag{34}$$

Thirdly, the necessary and sufficient condition for solving equation (32) is that $\dot{q}_{dual}$ and u need to satisfy: $\dot{q}_{dual} - \mathcal{J}_{dual}{}^T u + \lambda(q_{dual}(t) - q_{dual}(0)) = 0$ and $\mathcal{J}_{dual}\dot{q}_{dual} = g_\Omega(\mathcal{J}_{dual}\dot{q}_{dual} - u)$. Further, the following condition is given:

$$\begin{cases} \dot{q}_{dual} = \mathcal{J}_{dual}{}^{\mathrm{T}}u - \lambda(q_{dual}(t) - q_{dual}(0)), \\ g_\Omega(\mathcal{J}_{dual}\mathcal{J}_{dual}{}^{\mathrm{T}}u - \lambda\mathcal{J}_{dual}(q_{dual}(t) - q_{dual}(0)) - u) \\ \quad = \mathcal{J}_{dual}\mathcal{J}_{dual}{}^{\mathrm{T}}u - \lambda\mathcal{J}_{dual}(q_{dual}(t) - q_{dual}(0)). \end{cases} \tag{35}$$

By equation (35), the dynamic equation and the output equation of DNN is obtained:

$$\begin{cases} \Lambda\dot{u} = g_\Omega(\mathcal{J}_{dual}\mathcal{J}_{dual}{}^{\mathrm{T}}u - \lambda\mathcal{J}_{dual}(q_{dual}(t) - q_{dual}(0)) - u) \\ \qquad - \mathcal{J}_{dual}\mathcal{J}_{dual}{}^{\mathrm{T}}u + \lambda\mathcal{J}_{dual}(q_{dual}(t) - q_{dual}(0)), \\ \dot{q}_{dual} = \mathcal{J}_{dual}{}^{\mathrm{T}}u - \lambda(q_{dual}(t) - q_{dual}(0)). \end{cases} \tag{36}$$

where $\Lambda \in R^{26\times26}$ is a design parameter used to scale the convergence rate of DNN. Finally, q_{dual} can be obtained by the integral operation of $\dot{q}_{dual}$ solved by equation (36). Further, the

Industrial Robot: An International Journal
Volume 44 · Number 2 · 2017 · 210–221

input command of dual-arm robot Y can be obtained by equation (2). The following theorem is given to ensure the convergence and the optimal solution of (36) for the QP problem (30):

- Theorem 2: *For any initial joint configuration* $\dot{q}_{dual}$ *of the dual-arm robot, the DNN* [equation (36)] *is exponentially convergent to an equilibrium point* u^*. *The output* $\dot{q}^*_{dual} = \mathcal{J}^T_{dual}u^* - \lambda(q_{dual}(t) - q_{dual}(0))$ *is the optimal solution to the original QP problem* [equation (30)].

Proof: The proof is similar to the work in Zhang *et al.* (2009), and it is omitted.

4. Experiments and results

4.1 Experiments setup

The experimental setup is shown in Figure 4. A Point Grey Blackfly color camera is fixed above the robot workspace with the accuracy of 0.1 mm, resolution of 640 × 480 pixels and 24 frames/s. The camera intrinsic and extrinsic matrices are calibrated by Camera Calibration Toolbox for MATLAB:

$$M = \begin{bmatrix} 1007.6 & 0 & 639.5 & 0 \\ 0 & 1077.9 & 511.5 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix},$$

$$ {}^b_cT = \begin{bmatrix} -0.4355 & 0.6729 & -0.5952 & 337 \\ 0.8238 & 0.5632 & -0.0354 & -364 \\ 0.3609 & -0.4760 & -0.8000 & 868 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

Two experiments are carried out: carrying a tray and turning a steering wheel. Four marks are set on both the tray and wheel; the centroids of which are extracted by the image processing algorithm of the color space.

As shown in Figure 5, the initial joint configurations of two experiments are set. The grasp positions are set with an arbitrary pose of the two end-effectors, and the relative pose errors of the two end-effectors are recorded. It should be noted that the rigid contact between the end-effector and the object has to be considered. To enhance the passive compliance, the syntactic foam is pasted on each fingertip to ensure the experiment is carried out with an allowable pose error. By the test experiments, the desired accuracies of two experiments are defined as: the image tracking error is less than 2 pixels, and the joint-angle-drift error is less than 0.1 rad.

Figure 4 Experimental setup

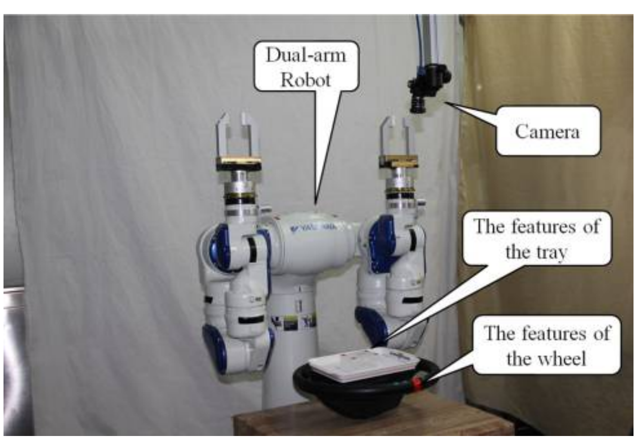
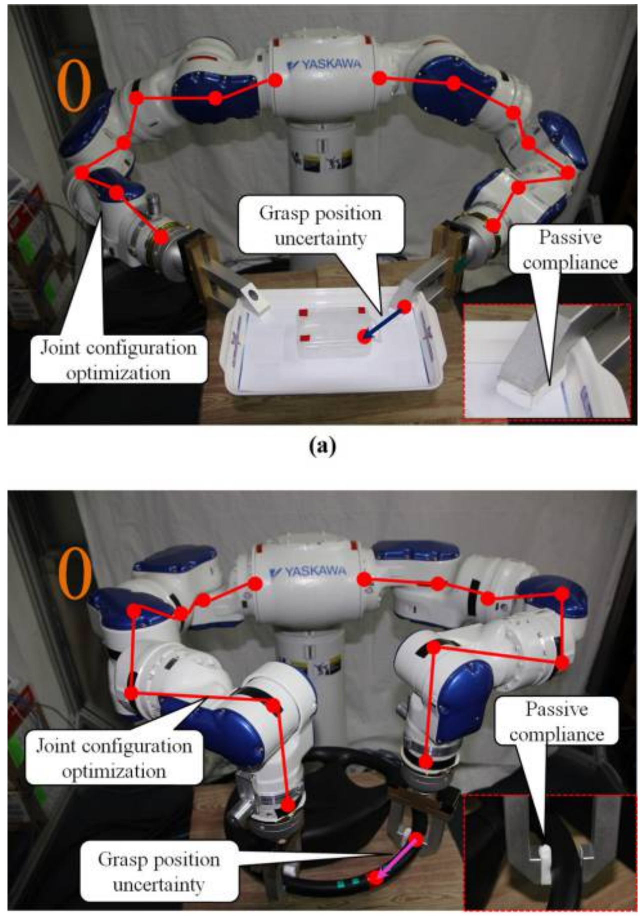


Figure 5 The initial states of two experiments



Notes: "0" indicates the initial state; (a) Carrying the tray; (b) Turning the steering wheel

The parameters of the RBF network are fixed: the centers c_i are chosen so that they were evenly distributed to span the input space of the network, $b_i = 10$ and the neuron number of the hidden layer is $n = 45$.

4.2 Experiment 1: carrying the tray repeatedly

The initial joint configurations are set as $q_l = [-1.50; -0.46; 1.79; 0.78; 1.36; 0.06; -0.88]$ and $q_r = [-1.28; -0.43; 1.55; 0.81; 2.53; 0.13; -2.49]$ rad. The relative position $p_{\rm rl} = [7.61; 467.61; 0.03]$ mm; the relative orientation is $[-2.955; -0.475; 0.064]$ rad. The Cartesian coordinates with respect to Σ_o are set as ${}^ox = [0; 0; 0]$, ${}^ox = [-43; 3; 0]$, ${}^ox = [2; -72; 0]$ and ${}^ox = [-43; -72; 0]$ mm. The real value of the grasp position is $p_{\rm ol} = (88; 73; 32)$ mm, and the estimated value is set as $\hat{p}_{\rm ol} = [70; 60; 32]$ mm. The initial image coordinate vectors of the four feature points are set as $s_0 = [144; 699; 338; 634; 281; 540; 43; 615]$ pixels. The tray is expected to move along a trapezoid trajectory. The desired image closed-trajectories consist of five piecewise line equations, where four trapezoid vertices are defined as the path points: s_A, s_B, s_C and s_D.

The image displacements at the four path points with respect to s_d are set as: $\Delta s_{A\to 0} = [37; 5]$, $\Delta s_{B\to 0} = [69; 45]$,

$\Delta s_{C\to 0} = [-74; 28]$, $\Delta s_{D\to 0} = [-36; -3]$. The finished image coordinate s_1 is equal to s_0.

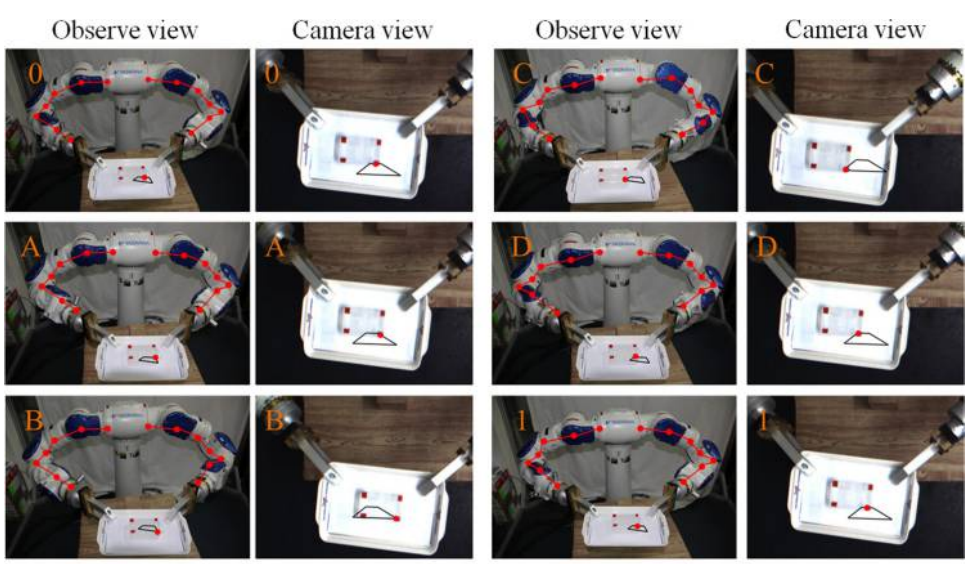
The control parameters are set as: $K_{p1} = 2.5I_{6\times 6}$, $K_{i1} = 0.2I_{6\times 6}$, $\Gamma = 4$, $\kappa = 0.1$, $K_r = I_{6\times 6}$, $K_{p2} = diag$, {10, 10, 10, 20, 20, 20} $K_{i2} = diag$, {0.05, 0.05, 0.05, 0.1, 0.1, 0.1}, $\mu = 15$, $\lambda = 5$ and $\Lambda = 10^{-4}$.

The experiment snapshots from two viewpoints are shown in Figure 6. An extra camera is placed near the workspace to observe the experiment process, and the scene in the fixed camera is shown. For clarity, only one feature point is described. The full task is divided into five phases along the path points from A to D. Although the estimated grasp position is used, the image feature points of the tray achieved successfully the closed-trajectory tracking task along the desired trapezoidal trajectory, and the joint configurations maintain invariability.

The experiment results are shown in Figure 7. It should be noted that when an exact grasp position is known, the conventional PI control can be competent to the dual-arm motion control. When the grasp position is uncertain, the PI-RBF method is proposed on the basis of PI control. To indicate the superiority of the proposed method, the results when using the conventional PI control [$\dot{x}_{nn} = 0$ in equation (20), $\lambda = 0$ in equation (30)] are introduced for comparison. The three image trajectories of one feature point are shown in Figure 7(a). At the beginning, the real tracking trajectory has a large error compared with the desired trajectory. Subsequently, the trajectory is asymptotically stabilized to the desired location near the path point A and benefits from the uncertainty $\hat{p}_{ol}$ compensation by the RBF network. In contrast, the tracking trajectory generated by the PI control has a large error value when compared to the desired location during the whole process because of the lack of uncertainty compensation. The image tracking errors of four feature points of the proposed and conventional method are shown in Figure 7(b). The average tracking errors of the proposed PI - RBF is 1.2 pixels, which is much less than 5.8 pixels of the conventional PI control. All joint-angle trajectories of dual-arm robot are shown in Figure 7(c)-(d). Obviously, when compared to the initial state, the trajectories without compensation (no symbol) have large drifts on the left: q_{l1}, q_{l2}, q_{l3} and q_{l6}, and the right: q_{r2}, q_{r3}, q_{r5} and q_{r6}. The average drift joint errors of the left and right arms reach 0.963 and 0.492 rad, respectively. However, the trajectories of all joints with DNN compensation (with symbol) show few fluctuations. Accordingly, the trajectories using DNN compensation (with symbol) have relatively few fluctuations in all joints. Compared to the initial state, the average error of the dual-arm robot is (0.079, 0.034) rad, which is less than one-tenth of the errors observed using the PI method.

To indicate the feasibility of the proposed method, additional four cases are added to Table I (the results of Figure 7 are recorded as Case 1). It should be noted that because of the fixed coordination constraint of dual-arm manipulation, the pose error of the right end-effector will be the same as the one of the left end-effector. Therefore, only the Cartesian errors of the left end-effector are given in both experiments. The real values of the grasp position are varied, whereas the estimated values are fixed. The error between the real and estimated value is defined as $\tilde{p}_{ol} = p_{ol} - \hat{p}_{ol}$, and the average error of $\tilde{p}_{ol}$ between these five trials is (25,27,0) mm. The errors in three spaces had an increasing trend along with $\tilde{p}_{ol}$; however, the error is limited within a small range, where the maximum errors are 2 pixels (image), 7.9 mm (position), 0.034 rad (orientation) and (0.095, 0.073) rad (joint), which meets our experimental objective. The final average error for these five cases was 1.5 pixels for image tracking, (5.3 mm, 0.027 rad) for Cartesian tracking and (0.083, 0.051) rad for joint-angle-drift – this is significantly less than the average error of 6.4 pixels, (14.1 mm,0.046 rad) and (1.034, 0.708) rad obtained by the conventional method.

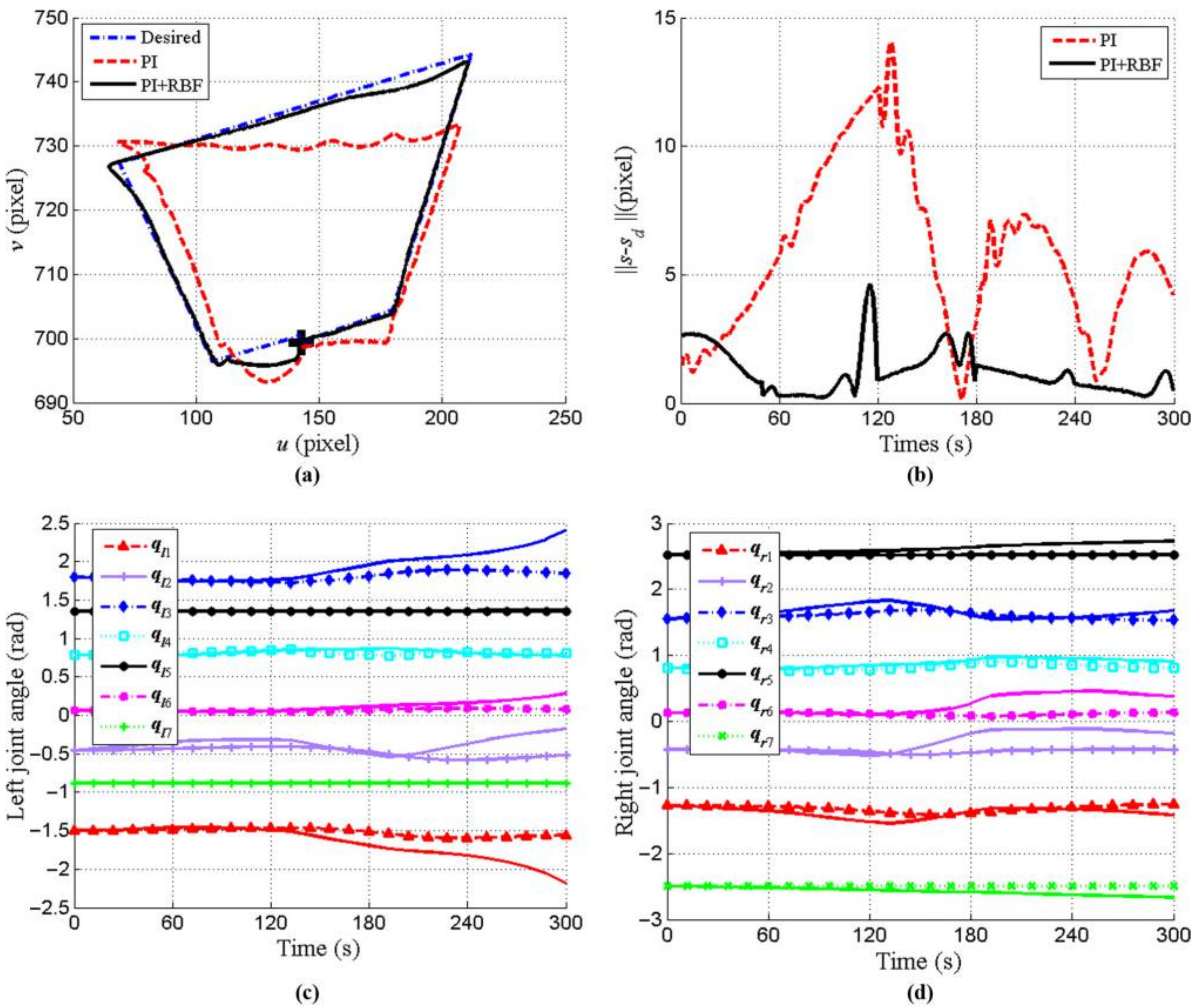
Figure 6 Experiment snapshots for carrying the tray



Notes: "0" and "1" are the initial state and the finished state, respectively, and "A-B-C-D" are the path points during the whole task

Industrial Robot: An International Journal
Volume 44 · Number 2 · 2017 · 210–221

Figure 7 The experiment results of carrying the tray in the image space (a)-(b), and in the joint spaces (c)-(d)



Notes: (a) The image trajectories; (b) the average image errors of the four feature points; (c) the joint trajectories of the left arm (Symbol: proposed, no symbol: PI); (d) the joint trajectories of the right arm (Symbol: proposed, no symbol: PI)

Table I Grasp positions and average errors in comparison experiments of carrying the tray

Case	Grasp position (p_{ol}) Real (mm)	Estimated (mm)	Conventional method (PI) Image (pixel)	Left Cartesian (mm, rad)	Joint angle (left, right) (rad)	Proposed method (PI + RBF + DNN) Image (pixel)	Left Cartesian (mm, rad)	Joint angle (left, right) (rad)
1	(88, 73, 32)	(70, 60, 32)	5.8	(10.6, 0.037)	(0.963, 0.492)	1.2	(3.3, 0.021)	(0.079, 0.034)
2	(74, 79, 32)	(70, 60, 32)	5.6	(11.8, 0.031)	(0.841, 0.693)	1.3	(3.9, 0.019)	(0.076, 0.046)
3	(95, 86, 32)	(70, 60, 32)	6.2	(13.6, 0.044)	(0.981, 0.587)	1.4	(5.2, 0.028)	(0.075, 0.041)
4	(106, 96, 32)	(70, 60, 32)	6.8	(16.6, 0.056)	(1.243, 0.812)	1.8	(6.2, 0.031)	(0.095, 0.062)
5	(112, 102, 32)	(70, 60, 32)	7.6	(18.1, 0.062)	(1.142, 0.955)	2.0	(7.9, 0.034)	(0.088, 0.073)
Average: $\widetilde{p}_{ol}$ (25, 27, 0)			6.4	(14.1, 0.046)	(1.034, 0.708)	1.5	(5.3, 0.027)	(0.083, 0.051)

4.3 Experiment 2: turning the steering wheel

First, it is necessary to explain the characteristics of Experiment 2. Due to the rigid construction of both the steering wheel and end-effector, uncertainty of the grasp position may lead to task failure, the tracking errors both in the image plane and in the Cartesian space must be very small regardless of the control method used and the tracking trajectory must be circular.

The initial joint configurations are set as q_l $[-0.04; 0.71; 0; 1.14; -1.51; -1.59; -1.52]$ and $q_r = [-0.03; 1.21; 0; 1.79; -1.52; -1.60; 0.32]$ rad. The relative position $p_{rl} = [-72.34; 233.22; 1.43]$ mm; the relative orientation is $[0.007; -0.021; 0.197]$ rad. The Cartesian coordinates with respect to $\sum_o$ are set as $^o x = [0; 0; 0]$, $^o x = [-2; 27; 0]$, $^o x = [-3; -29; 0]$ and $^o x[3; -53; 0]$ mm. The real value of the grasp position is $p_{ol} = [85; 120; 5]$ mm, and the estimated value is set as $\hat{p}_{ol} = [70; 100; 5]$ mm. The initial image coordinates vector of four feature points are set as $s_0 = [216; 337; 208; 328; 219; 330; 221; 327]$ pixels. The steering wheel is expected to achieve the movement of a closed-circular trajectory. The desired image closed-trajectories equation is given as:

$$s_d(t) = s_{circle} + \frac{d}{2}\begin{bmatrix}\cos(g\theta_m t/t_m)\\ \sin(g\theta_m t/t_m)\end{bmatrix}, \quad t \in [0, 2t_m]$$

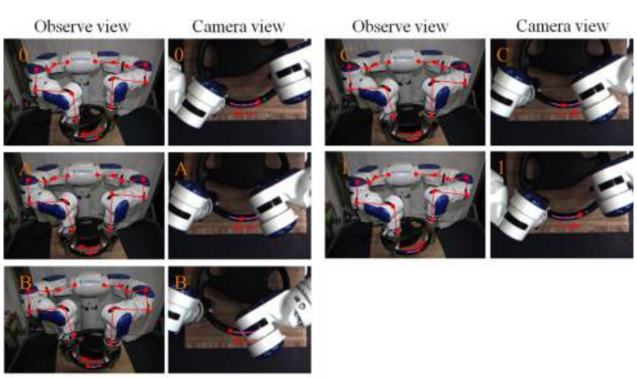
where $\theta_m = \arccos((s_B - s_{circle}) \times (s_0 - s_{circle})/(\|s_B - s_{circle}\|\|s_0 - s_{circle}\|))$; $s_{circle} = [105; 422]$ is the image coordinate of the circle center which is obtained in advance. $s_B = [340; 336]$ is the finished point of the front half of the entire closed-circular trajectory. d is the diameter of the steering wheel. g indicates the direction of the angular velocity, $g = 1$ when $t \in [0, 2t_m]$ in the front half of the closed-circular trajectory and $g = -1$ when $t \in [t_m, 2t_m]$ in the last half of the one. $t_m = 70$ (s) is the half time of the entire experiment. The control parameters are set as: $K_{p1} = 5I_{6\times6}$, $K_{i1} = 0.05I_{6\times6}$, $\Gamma = 10$, $\kappa = 0.1$, $K_r = I_{6\times6}$, $K_{p2} = diag$, $\{2, 2, 2, 5, 5, 5,\}$ $K_{i2} = diag$, $\{0.02, 0.02, 0.02, 0.5, 0.5, 0.5\}$ $\mu = 10$, $\lambda = 10$ and $\Lambda = 10^{-6}$.

The experiment snapshots are shown in Figure 8. Again, for the sake of clarity, one feature point is described. The path points A and C are introduced only to explain the process of the full task. From the initial state 0 to the finished state 1 along with the path points A-B-C, the image feature points of the wheel successfully achieved the closed-trajectory tracking task along the desired closed-circular trajectory, and the joint configurations remain invariable.

The experiment results are shown in Figure 9. The three image trajectories of one feature point are shown in Figure 9(a). In accordance with expectations, the tracking trajectories by the conventional and proposed method are both coincident with the desired trajectory within a very small visual measured error. However, the conventional trajectory (dash line) terminated in some iteration because of the pose error of the left end-effector caused by uncertainty of the grasp position, ultimately resulting in task failure. The image tracking errors of four feature points of the proposed and conventional method are shown in Figure 9(b). The average tracking error of the proposed method is 1.5 pixels. Furthermore, the tracking error of the conventional trajectory is terminated at 89 s, thus the average error of the whole task is imponderable. The joint-angle trajectories are shown in Figure 9(c)-(d). The trajectories obtained while using the proposed method (with symbol) have only a few fluctuations in all joints. Compared to the initial state, the average error is (0.049, 0.062) rad. Furthermore, at the terminal instant of the conventional trajectories, there is large joint-angle-drift error in some joints, such as the left: q_{l2} and q_{l6} and the right: q_{r2}, q_{r4}, q_{r5} and q_{r6}.

As with the Experiment 1, additional four cases are added for comparison, and the data are presented in Table II (the results of Figure 9 are recorded as Case 1). It should be noted that because the end-effectors and the steering wheel are regarded as the fixed connection rigidly, that is the desired and real position trajectories of the end-effectors are the same, only the orientation error of the left end-effector are given in this experiment. Due to the large estimated error $\tilde{p}_{ol}$ leading to the task failure, $\tilde{p}_{ol}$ were decreased accordingly in other cases [the average error is (15.2, 8.6, 0) mm]. The average errors are 1.3 pixels (image) and (0.049, 0.054) rad (joint). Importantly, the average orientation error of the left end-effector is 0.068 rad, well within the 0.1 rad range, which ensures the wheel turning task is achieved successfully. In contrast, the first three cases of the conventional method failed because of the large estimated error $\tilde{p}_{ol}$ which leads to the large average orientation errors of the left end-effector (>0.1 rad). In Case 4-5, a small value of $\tilde{p}_{ol}$ is selected, and the large average orientation errors of the left end-effector are limited within 0.1 rad. Thus, the task is achieved in these iterations with the average error of 1.1 pixels and (1.232, 1.081) rad; however, the joint angle-drift-error is still large due to the lack of the closed-chain kinematics redundancy optimization.

Figure 8. Experiment snapshots for turning the steering wheel



Notes: "0" and "1" are the initial state and the finished state, respectively, and "A-B-C" are the path points during the whole task

5. Conclusion

In this paper, a vision-based dual-arm cyclic-motion method is developed; the PI-RBF-DNN control method is proposed to compensate for the uncertainty of the grasp position and optimize the closed-chain kinematics redundancy for avoiding the joint-angle-drift phenomenon. The two experiments carried out by the MOTOMAN dual-arm robot indicate that the trapezoidal trajectory tracking of the tray is successfully achieved with an average image error of 1.5 pixels and the joint-angle-drift error of (0.083, 0.051) rad. The closed-circular trajectory tracking of the wheel is also achieved successfully with an average image error of 1.3 pixels and a joint-angle-drift error of (0.049, 0.054) rad. The data satisfy the experiment requirements and validate the effectiveness, accuracy and realizability of the PI-RBF-DNN method.

Industrial Robot: An International Journal
Volume 44 · Number 2 · 2017 · 210–221

Figure 9 The experiment results of turning the steering wheel in the image space (a)-(b) and in the joint spaces (c)-(d)(a) The image trajectories, (b) The average image errors of the four feature points, (c) The joint trajectories of the left arm (symbol: proposed, no symbol: PI), (d) The joint trajectories of the right arm (symbol: proposed, no symbol: PI)

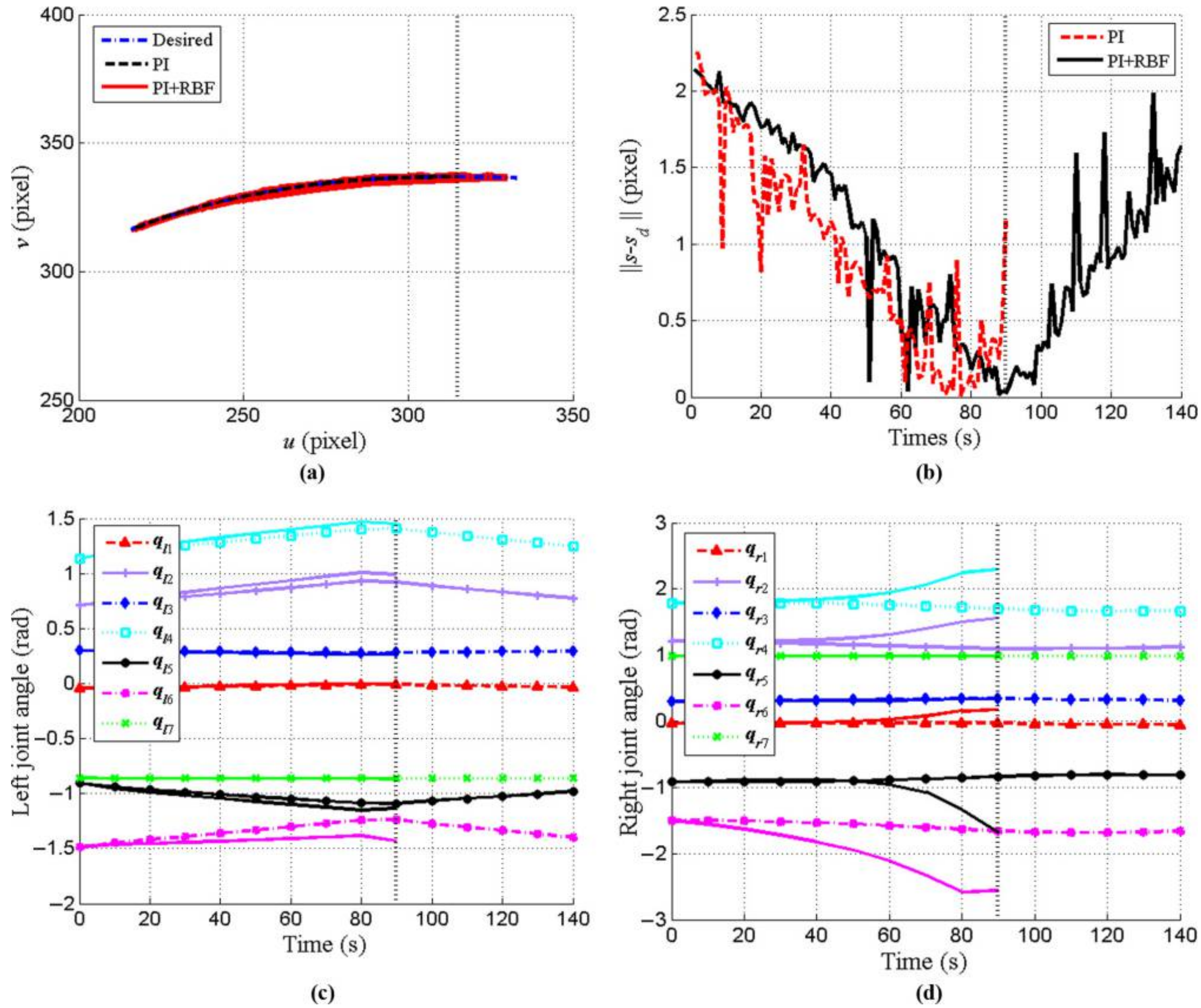


Table II Grasp positions and average errors in comparison experiments of turning the steering wheel

	Grasp position (p_{ol})		Conventional method (PI)			Proposed method (PI + RBF + DNN)		
Case	Real (mm)	Estimated (mm)	Image (pixel)	Left orientation (rad)	Joint angle (left, right) (rad)	Image (pixel)	Left orientation (rad)	Joint angle (left, right) (rad)
1	(85, 120, 5)	(70, 100, 5)	–	0.107	–	1.5	0.070	(0.049, 0.062)
2	(92, 115, 5)	(70, 100, 5)	–	0.112	–	1.4	0.082	(0.057, 0.074)
3	(100, 95, 5)	(70, 100, 5)	–	0.120	–	1.3	0.088	(0.076, 0.056)
4	(64, 93, 5)	(70, 100, 5)	1.2	0.064	(1.272, 1.055)	1.1	0.048	(0.039, 0.043)
5	(73, 106, 5)	(70, 100, 5)	1.1	0.074	(1.191, 1.107)	1.1	0.052	(0.024, 0.033)
Average: $\widetilde{p}_{ol}$ (15.2, 8.6, 0)			1.1	0.095	(1.232, 1.081)	1.3	0.068	(0.049, 0.054)

The proposed method can enhance the flexibility of the manipulation object in a dual-arm cyclic motion task. In other words, an arbitrary size of the tray in service and nursing settings or the valve in industrial and rescue settings can be used. On the other hand, the desired image trajectories are given in advance in this paper. Therefore, object recognition and autonomous planning technique will need to be developed in the future.

References

Cai, B. and Zhang, Y. (2012), "Different-level redundancy-resolution and its equivalent relationship analysis for robot manipulators using gradient-descent and Zhang's neural-dynamic methods", *IEEE Transactions on Industrial Electronics*, Vol. 59 No. 8, pp. 3146-3155.

Chaumette, F. and Hutchinson, S. (2006), "Visual servo control. Part I: basic approaches", *IEEE Robotics & Automation Magazine*, Vol. 13 No. 4, pp. 82-90.

Khoogar, A.R., Tehrani, A.K. and Tajdari, M. (2011), "A dual neural network for kinematic control of redundant manipulators using input pattern switching", *Journal of Intelligent & Robotic Systems*, Vol. 63 No. 1, pp. 101-113.

Liang, X., Wang, H. and Chen, W. (2015), "Adaptive image-based trajectory tracking control of wheeled mobile robots with an uncalibrated fixed camera", *IEEE Transactions on Control Systems Technology*, Vol. 23 No. 6, pp. 2266-2282.

Ma, Z. and Su, J. (2015), "Robust uncalibrated visual servoing control based on disturbance observer", *ISA Transactions*, Vol. 59 No. 6, pp. 193-204.

Piepmeier, J.A., McMurray, G.V. and Lipkin, H. (2004), "Uncalibrated dynamic visual servoing", *IEEE Transactions on Robotics and Automation*, Vol. 20 No. 1, pp. 143-147.

Wang, H., Jiang, M. and Chen, W. (2012), "Visual servoing of robots with uncalibrated robot and camera parameters", *Mechatronics*, Vol. 22 No. 6, pp. 661-668.

Xia, Y. and Wang, J. (2000), "A recurrent neural network for solving linear projection equations", *Neural Networks*, Vol. 13 No. 3, pp. 337-350.

Xiao, L. and Zhang, Y. (2013), "Acceleration-level repetitive motion planning and its experimental verification on a six-link planar robot manipulator", *IEEE Transactions on Control Systems Technology*, Vol. 21 No. 3, pp. 906-914.

Xie, T., Yu, H. and Wilamowski, B. (2011), "Comparison between traditional neural networks and radial basis function networks", *2011 IEEE International Symposium on Industrial Electronics (ISIE 2011)*, *Gdansk*, pp. 1194-1199.

Yu, W. and Moreno-Armendariz, M.A. (2005), "Robust visual servoing of robot manipulators with neuro compensation", *Journal of the Franklin Institute*, Vol. 342 No. 7, pp. 824-838.

Zhang, Y., Guo, D. and Li, Z. (2013), "Common nature of learning between back-propagation and Hopfield-type neural networks for generalized matrix inversion with simplified models", *IEEE Transactions on Neural Networks and Learning Systems*, Vol. 24 No. 4, pp. 579-592.

Zhang, Y., Lv, X. and Li, Z. (2008), "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.

Zhang, Y., Tan, Z. and Chen, K. (2009), "Repetitive motion of redundant robots planned by three kinds of recurrent neural networks and illustrated with a four-link planar manipulator's straight-line example", *Robotics and Autonomous Systems*, Vol. 57 No. 6, pp. 645-651.

Zhang, Z., Li, Z. and Zhang, Y. (2015), "Neural-dynamic-method-based dual-arm CMG scheme with Time-varying constraints applied to humanoid robots", *IEEE Transactions on Neural Networks and Learning Systems*, Vol. 26 No. 12, pp. 3251-3262.

Zhang, Z. and Zhang, Y. (2012), "Acceleration-level cyclic-motion generation of constrained redundant robots tracking different paths", *IEEE Transactions on Systems, Man, and Cybernetics, Part B: Cybernetics*, Vol. 42 No. 4, pp. 1257-1269.

Zhao, Y. and Cheah, C.C. (2009), "Neural network control of multi-fingered robot hands using visual feedback", *IEEE Transactions on Neural Networks*, Vol. 20 No. 5, pp. 758-767.

Zhu, Q., Fei, S. and Zhang, T. (2008), "Adaptive RBF neural-networks control for a class of time-delay nonlinear systems", *Neurocomputing*, Vol. 71 No. 16, pp. 3617-3624.

Corresponding author

Fuhai Zhang can be contacted at: zfhhit@hit.edu.cn