

# An RBF-PD Control Method for Robot Grasping of Moving Object



Yong Tao, Xianwu Xie and Hegen Xiong

**Abstract** In order to solve the uncertainty of robot's grabbing position of moving objects, a control method based on RBF (radial basis function) neural network and PD (proportional-derivative) for crawling dynamic targets is proposed. The Kalman filter algorithm is used to estimate the pose of the moving target. The information of the pose estimator is used as the input of the adaptive neural network controller. An adaptive robust control scheme based on RBF neural network and PD is proposed. It ensures that the trajectories are accurately tracked even in the presence of external disturbances and uncertainties. The machine learning method is implemented into a vision-based control scheme to compensate for the uncertainty of the estimated grasping position and improve the success rate of the robot's accurate grasping. Finally, the experiment was carried out to verify the effectiveness of the proposed method.

**Keywords** Radial basis function neural network · RBF-PD control  
Visual servoing PBVS · Track and capture

## 1 Introduction

In recent years, visual servo has been increasing attention paid to the field of robotics and control. Robot motion control uses direct visual sensory information to achieve a desired relative position between the robot and a moving object in the robot environment, accurate access to the target pose and motion to automatically control the robot to track and grasp the moving target. Because of non-invasive, non-destructive and non-contact, computer vision is specifically used as a sensing

---

Y. Tao  
Beihang University, 100191 Beijing, China  
e-mail: taoy@buaa.edu.cn

X. Xie (✉) · H. Xiong  
Wuhan University of Science and Technology, 430081 Wuhan, China  
e-mail: 905288566@qq.com

system for obtaining the required information [1–7]. Thus, a visual servo control system has been developed to control the posture of the end effector of the robot arm relative to the target based on the feedback of the visual system. A single moving camera can be used to track the position of a known moving object in the image plane based on past images and past control inputs to the mobile platform [8]. Automated capture of non-cooperative targets by robotic manipulators requires not only tracking the motion of the target [9, 10], but also predicting the rendezvous point and following the specific approximation trajectory of the end effector based on the estimated posture and motion of the target [2, 4]. The control loop in Visual Servoing has different architectures, such as look-and-move structure and perweiss structure [11]. The look-and-move structure has an internal feedback controller, as being used in many industrial robots. Such structure may accept Cartesian velocity or incremental position commands and permits to simplify the design of control signal [11].

There are three main approaches in VS [12], Position-Based Visual Servoing (PBVS) [13], Image-Based Visual Servoing (IBVS) [14], and “2&1/2 D” visual servoing [15], where PVBS is the most frequently used method [13]. In PBVS, the control signal is produced based on the estimation of position and orientation (pose) of the target with respect to the camera. The accuracy of the estimated pose is directly related to the measurement noise and the camera calibration [16]. Extended Kalman Filter (EKF) and Unscented Kalman Filter (UKF) have been developed to deal with the pose estimation in the noisy and uncertain situations. The aforementioned estimators have shown to be quite effective in practice [17–19]. In order to calculate the velocity and acceleration of the target, the appropriate dynamic model for the relative motion between the camera and the target is necessary. Conventional models are applied based on the constant velocity or the acceleration model which assumes invariable relative velocity or acceleration at each sample time [20].

After estimating the pose of the target object, the main goal in VS problem is to enhance the performance of tracking via a controller. Since the system eye to hand (a robot) has nonlinear dynamics, a nonlinear controller has to be designed for this purpose. For such a design, we use adaptive neural network control in order to achieve the robust performance in the noisy environment and in the industrial environments. Generally, in many forms of VS, the path planning and controlling the end effector of robot are performed separately. By using adaptive neural network control approach, the aforementioned tasks can be combined together. In the adaptive neural network controller, the error between the actual target and the desired target position is fed to the PD controller using the visual information provided by the camera, and The PD controller calculates the joint speed when the error is zero, and then the RBF neural network controller is designed so that the joint speed of the robot converges to a given speed input. In this paper, the estimated values are obtained from a UKF cascade structure which has been recently proposed in [21]. The information of the estimated model and the observation inherits uncertainties which can be directly considered in the proposed controller.

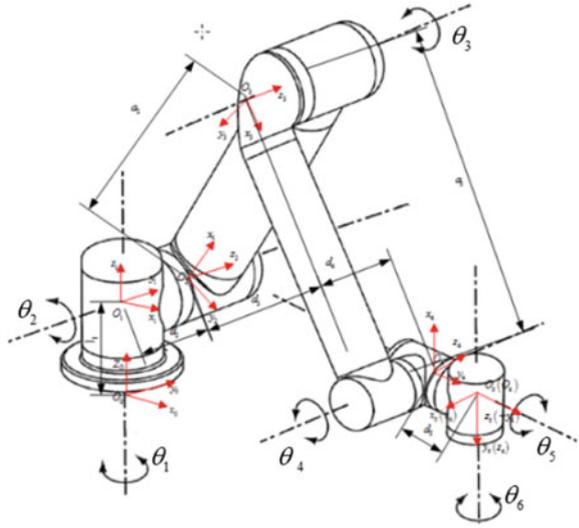
Neural network has a well-known property that it can approximate arbitrary nonlinear functions and learn through examples, and hence it allows robot control without structure assumed in the aforementioned adaptive control laws. For a planar robot manipulator, Kim et al. [22] used an RBF neural network combined with a robust controller to compensate for uncertainties in dynamics. Chen [23] 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. With the compensation for dynamic uncertainties using a RBF neural network, Xie T, Yu H et al. compared RBF networks with other networks [back propagation (BP), Kohonen networks, etc.], and the results of the comparison indicate that the non-linear function approximation capability of RBF networks is prior to that of other networks [24]. Yang H. J. et al. validated that RBF networks can effectively improve the robustness of the controller when the system parameters have a large uncertainty [25]. In the above-mentioned contributions, the parameter uncertainties including the camera calibration, depth and dynamics were discussed; however, there is a lot of uncertainty in the real world. Because of uncertainty, the models we use in grasping are basically inaccurate, or even wrong. At the same time, there is not enough good sensor which can real-time feedback real state. Hence, it is very difficult to accurately control the robots to grasp the pose we desire. To sum up, the uncertainty of the robotic grasping position is still not better resolved so far. Therefore, aiming at the uncertainties such as time delay and occlusion in the process of robot grasping, this paper presents a PD based adaptive neural network control method, adopting a RBF network to deal with the uncertainty of the grasp position. The information of the pose estimator is used as the input of the adaptive neural network controller, and control command is generated. In combination with the software gripper, the robot's grasping was achieved. Finally, experiments are carried out to verify the tracking performance of the industrial robot in grasping control of the moving object.

## 2 Kinematics of Robot Arm

The robot manipulator is shown in Fig. 1. It has six joints from base to the end of robot arm, namely, base joint ( $\theta_1$ ), shoulder joint ( $\theta_2$ ), elbow joint ( $\theta_3$ ), wrist joint ( $\theta_4$ ), wrist joint ( $\theta_5$ ) and wrist joint ( $\theta_6$ ). The soft hand is installed at the robot's end-effector, and the gripper is usually not considered as a robotic joint since it will be activated only after the end-effector is aligned with the target. Therefore, only the degrees of freedom (DOF) of the six rotational joints will be considered in the robot controller.

For the PBVS, it is natural to describe the target in relative to the soft hand. The spatial position of end effector is described by the stationary Cartesian coordinate frame ( $x_0$ ). The local coordinate frame ( $X_g$ ) defined attached to the gripper with y-axis and z-axis aligned with the rotational axes of wrist( $\theta_6$ ). Next the kinematics

**Fig. 1** UR robot and joint axis



relationship between the rotational joint positions and the corresponding Cartesian position of the end-effector in the workspace is defined as:

$$\begin{Bmatrix} X_0 \\ 1 \end{Bmatrix} = T_{0g}(\theta) \begin{Bmatrix} X_g \\ 1 \end{Bmatrix} \tag{1}$$

where  $T_{0g}(\theta)$  is the  $4 \times 4$  DH transformation matrix from the gripper coordinate frame to the global coordinate system and  $\theta = \{\theta_1, \theta_2, \theta_3, \theta_4, \theta_5, \theta_6\}$  is the vector of joint angles which is defined in the joint space respectively, as shown in Table 1.

Similarly, transformation from the camera coordinate system ( $X_c$ ) to the soft-hand coordinate system ( $X_g$ ) can be expressed as:

$$\begin{Bmatrix} X_g \\ 1 \end{Bmatrix} = T_{gc}(\theta) \begin{Bmatrix} X_c \\ 1 \end{Bmatrix} \tag{2}$$

**Table 1** D–H parameters of manipulator

$i$	$a_{i-1}$ (mm)	$\alpha_{i-1}$	$d_i$ (mm)	$\theta_i$ (Zero position)	Joint variable	Range of joint angle
1	0	$0^\circ$	$d_1 = 94.5$	$\theta_1 = 0^\circ$	$\theta_1$	$-180^\circ \sim 180^\circ$
2	0	$-90^\circ$	$d_2 = 130.5$	$\theta_2 = -90^\circ$	$\theta_2$	$-180^\circ \sim 180^\circ$
3	$a_2 = 425$	$0^\circ$	$d_3 = 120$	$\theta_3 = -90^\circ$	$\theta_3$	$-180^\circ \sim 180^\circ$
4	$a_3 = 392.5$	$-90^\circ$	$d_4 = 93$	$\theta_4 = 0^\circ$	$\theta_4$	$-180^\circ \sim 180^\circ$
5	0	$-90^\circ$	$d_5 = 93$	$\theta_5 = 180^\circ$	$\theta_5$	$-180^\circ \sim 180^\circ$
6	0	$0^\circ$	0	$\theta_6 = 0^\circ$	$\theta_6$	$-180^\circ \sim 180^\circ$

where  $T_{gc}(\theta)$  is the  $4 \times 4$  DH transformation matrix from the camera coordinate system to the gripper coordinate system. The camera frame  $X_c$  is defined as  $x$ -axis and  $y$ -axis lie in the image plane, while the  $z$ -axis is parallel to the axis of forearm and pointing towards the target.

Therefore, the velocity and acceleration velocity relationships between the end-effector and joints are:

$$\dot{X}_e = J(\theta)\dot{\theta} \text{ and } \ddot{X}_e = J(\theta)\ddot{\theta} + \dot{J}(\theta)\dot{\theta} \tag{3}$$

where  $J(\theta)$  is the Jacobi matrix of robot.

If Jacobi is invertible, it can gain from inverse kinematics that:

$$\dot{\theta} = J^+(\theta)\dot{X}_e \tag{4}$$

where  $J^+(\theta) = (J^T(\theta)J(\theta))^{-1}J^T(\theta)$  is Moore-Penrose pseudo-inverse of the Jacobi matrix.

### 3 Attitude Estimation of Camera Model/Moving Object

#### 3.1 Attitude Estimation of Camera Model

Suppose the coordinates  $\{x_T, y_T, z_T\}^T$  of the target feature point are known in the target coordinate system. In other words, the visual system has to be calibrated in advance. Then, the uniform relationship between target and camera frame can be described by the Eq. (5):

$$\begin{Bmatrix} x_C \\ y_C \\ z_C \\ 1 \end{Bmatrix} = \begin{bmatrix} & & & x_{T0} \\ R_{TC} & & & y_{T0} \\ & & & z_{T0} \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{5}$$

where  $\{x_C, y_C, z_C\}^T$  is the coordinates of the same point in the camera coordinate system. The pose of a target can be described by the Cartesian coordinates  $\{x_{T0}, y_{T0}, z_{T0}\}$  of the frame origin which is fixed on the target in relative to the camera frame and the Eulerian angle  $\{\theta_x, \theta_y, \theta_z\}^T$  of this frame regarding to the camera frame.

The pinhole camera model which is applied in this paper is shown in Fig. 2. RTC represents the transformation between the target frame and the camera frame, RTC elements are expressed as  $r_{ij}$ . The feature points on the target are projected onto the physical image plane through the Eq. (6):

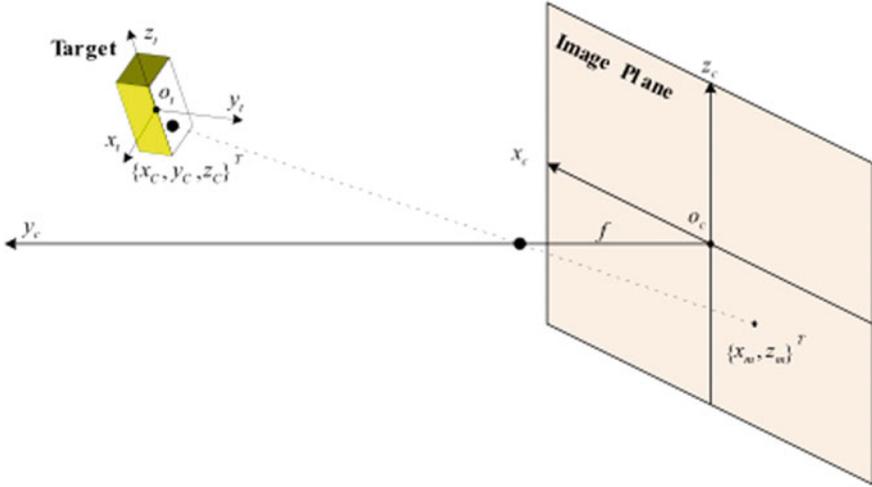


Fig. 2 Pin-hole camera model

$$\begin{cases} x_m = -f \frac{x_C}{y_C - f} = -f \frac{r_{11}x_T + r_{12}y_T + r_{13}z_T + x_{T0}}{r_{21}x_T + r_{22}y_T + r_{23}z_T + y_{T0} - f} \\ z_m = -f \frac{z_C}{y_C - f} = -f \frac{r_{31}x_T + r_{32}y_T + r_{33}z_T + z_{T0}}{r_{21}x_T + r_{22}y_T + r_{23}z_T + y_{T0} - f} \end{cases} \quad (6)$$

where  $f$  is the focal length of camera, and  $\{x_m, z_m\}^T$  stands for the projection image coordinates of the feature point. For a calibrated camera, the focal length is known in advance.

As described above, the photogrammetry is memoryless and prone to the image noises, which may result in large fluctuation of estimated target pose. Besides, photogrammetry can't estimate the motion of target directly, which is an important parameter for trajectory planning of the robotic manipulator to perform autonomous grasping in a dynamic environment. It has poor real-time performance of pose estimation. To address these challenges, an unscented Kalman filter (UKF) with photogrammetry is presented in the following.

### 3.2 Pose Estimation of Moving Object

The Kalman filter is an optimal estimation algorithm for a linear system with independent white noise of normal distribution [26]. The camera model in Eq. (6) is highly nonlinear, while the traditional Kalman filter is the linearization of nonlinear functions. There are disadvantages such as low accuracy, poor stability, slow response to the target maneuver. The UKF approximates the probability density distribution of the nonlinear function, and uses a series of determined samples to

approximate the posterior probability density of the state, overcoming the above shortcomings. Therefore, in order to ensure stability of robot control and smoothness of target attitude estimation, attitude and motion of the moving object were estimated by combining photogrammetry and UKF in this paper. This robust method leads to a better performance in an uncertain and noisy environment. Output of the photogrammetry was used as the input of Kalman filter. Then,

Let  $\{X\}$  be the state vector of the target, including the pose, velocity and acceleration velocity with respect to the camera coordinate system:

$$\{X\} = \left\{ x_{T0}, \dot{x}_{T0}, \ddot{x}_{T0}, y_{T0}, \dot{y}_{T0}, \ddot{y}_{T0}, z_{T0}, \dot{z}_{T0}, \ddot{z}_{T0}, \theta_x, \dot{\theta}_x, \ddot{\theta}_x, \theta_y, \dot{\theta}_y, \ddot{\theta}_y, \theta_z, \dot{\theta}_z, \ddot{\theta}_z \right\} \quad (7)$$

Then, the Kalman filter model of the target in discrete time form can be defined as:

$$\{x\}_{k+1} = [A]\{x\}_k + [B]\{w\}_k \quad (8)$$

where the subscript  $(k + 1)$  refers to the state of time step  $k + 1$ ,  $[A]$  is the state transfer matrix, and  $[B]$  is the disturbance transition matrix related with the process noise vector  $\{w\}_k$ .

The  $18 \times 18$  transition matrix  $[A]$  is composed of six  $3 \times 3$  block diagonal sub-matrixes  $[a]$ , such as:

$$[a] = \begin{bmatrix} 1 & d_t & d_t^2/2 \\ 0 & 1 & d_t \\ 0 & 0 & 1 \end{bmatrix}$$

where the disturbance transition matrix  $[B]$  is the  $18 \times 6$  sparse matrix which has the following non-zero elements:

$$B_{3(i-1)+1}, i = \frac{d_t^3}{6}, B_{3(i-1)+2}, i = \frac{d_t^2}{2}, B_{3(i-1)+3}, i = d_t$$

Here,  $d_t$  refers to the sampling period and  $i = 1, 2, \dots, 6$ . The process noise vector  $\{w\}_k$  includes vibration of the target and is hypothesized obeying to the zero-mean white Gaussian distribution with its covariance,  $[Q]_k$ .

$$\{w\}_k = \{\ddot{x}_{T0}, \ddot{y}_{T0}, \ddot{z}_{T0}, \ddot{\theta}_z, \ddot{\theta}_y, \ddot{\theta}_x\}^T \sim N(0, [Q]_k) \quad (9)$$

Generally speaking, the covariance matrix of process noise  $[Q]$  is difficult to be determined in advance due to non-synchronization of target and unknown motion of the camera. In the current work, it is found that the following constant process noise covariance matrix works well after tuning the Kalman filter in experiments.

$$[Q] = \begin{bmatrix} 5 & 0 & 0 & 0 & 0 & 0 \\ 0 & 5 & 0 & 0 & 0 & 0 \\ 0 & 0 & 5 & 0 & 0 & 0 \\ 0 & 0 & 0 & 5 & 0 & 0 \\ 0 & 0 & 0 & 0 & 5 & 0 \\ 0 & 0 & 0 & 0 & 0 & 5 \end{bmatrix} \times 10^{-6}$$

## 4 RBF-PD Control Method for Robot Grasping of Moving Object

In visual servo cycle, firstly, an image is captured by a camera and then the desired features are extracted from the mentioned image. Finally, the position, velocity and acceleration velocity of the object are estimated by UKF estimator. Offset in relative to pose is defined as the input and is used in the adaptive neural network controller. Then, the control signal (velocity) is generated by the controller and is proportional to each DOF of robot. According to singularity avoidance, this signal is used to solve the inverse kinematic problem in order to calculate angle of each joint. Finally, the outcome signal is commanded the internal controller of our industrial robot. Attentions should be paid to that the controller can be viewed as a perfect tracker due to the internal controller. The whole process is repeated until the target object is tracked perfectly.

In this paper, an adaptive neural network controller is used to perform visual servo on the mentioned tasks, and stability of the whole control system is proved by the Lyapunov theory. Moreover, robustness of the adaptive neural network controller to uncertainty of grasping position and estimated noise provides appropriate tracking performances to the whole visual servo system.

Structure of the proposed control method is shown in Fig. 3. In the visual servoing control, the desired pose error of end effector in visual tracking is  $\xi$ . Then, the PD controller is used to generate the pose screw ( $\dot{x}_{pd}$ ) of end effector. In addition, RBF neural network control is used to compensate uncertainty of the grasp position and visual servo error to gain the pose screw ( $\dot{x}_{nn}$ ). Finally, the desired pose screw of the end effector ( $\dot{x}_{id}$ ) is acquired. During the closed-chain control, the current pose error of the end effector is defined as  $\xi$ .

The desired pose error of end effector ( $\xi$ ) which is mapped from image space to the Cartesian coordinates is:

$$\xi = x_d(t) - x(t) \quad (10)$$

Here,  $x_d(t)$  and  $x(t)$  are desired pose and actual pose of end effector of robot in the world coordinate system.  $x_d(t)$  is calculated from the Eq. (8) and  $x(t)$  is measured by encoder.

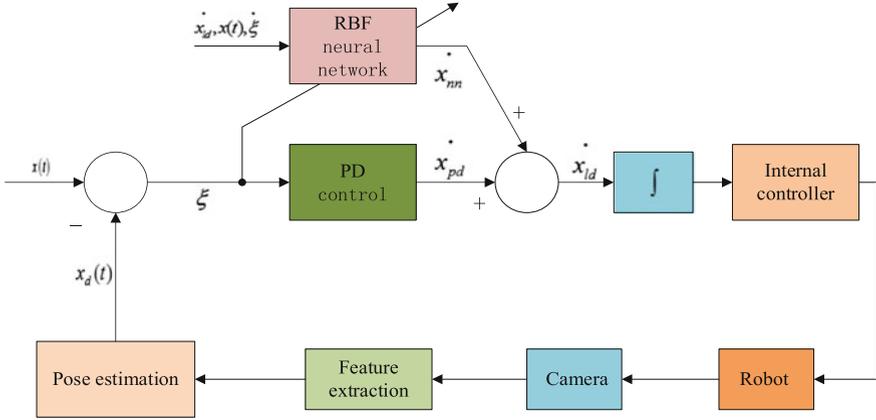


Fig. 3 Adaptive neural network controller

Next, it can be gained from derivation of the Eq. (10):

$$\dot{\xi} = -\dot{x}_{ld} + \Delta_{\xi}(\dot{x}_{ld}, s, \dot{\xi}) \tag{11}$$

To regulate the desired pose error  $\xi$ , the desired velocity screw in the Cartesian space is designed as follows:

$$\dot{x}_{ld} = \dot{x}_{pd} - \dot{x}_{nn} \tag{12}$$

where  $\dot{x}_{pd}$  is the PD control signal:

$$\dot{x}_{pd} = K_p \xi + K_d \dot{\xi} \tag{13}$$

where  $K_p > 0$  and  $K_d > 0$  are the control parameter matrixes.  $\dot{x}_{nn}$  is the feedback neural network control to compensate modeling error of the visual servoing system  $\Delta_{\xi}(\dot{x}_{ld}, s, \dot{\xi})$ . If  $\dot{x}_{nn} = 0$ , it indicates that the visual servoing is only the general PD control without the compensation of system uncertainty. To eliminate system error  $\Delta_{\xi}(\dot{x}_{ld}, s, \dot{\xi})$ , a control variable  $\dot{x}_{nn}$  needs to be designed and the neural network control needs to be adopted to approximate nonlinear function  $\Delta_{\xi}(\dot{x}_{ld}, s, \dot{\xi})$ .

The system error vector of the visual servo is  $\Sigma = [r\xi; \xi] \in R^{12}$ . Additionally, state equation of system error is obtained by Eqs. (11) and (13):

$$\dot{\Sigma} = A \times \Sigma + B \times (\dot{x}_{nn} - \Delta_{\xi}(\dot{x}_{ld}, s, \dot{\xi})) \tag{14}$$

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

$$A = \begin{bmatrix} 0_{6 \times 6} & I_{6 \times 6} \\ -K_d & -K_p \end{bmatrix}, B = \begin{bmatrix} 0_{6 \times 3} & 0_{6 \times 3} \\ I_{6 \times 3} & 0_{6 \times 3} \end{bmatrix}$$

RBF network is used to approximate  $\Delta_\xi$ . The input vector of RBF network is  $x_{in} = [\dot{x}_{ld}; s; \dot{\xi}] \in R^{22}$ , and  $h = [h_1; h_2; \dots; h_n] \in R^n$ . The radial basis vector with Gaussian function ( $h_i$ ) is as follows:

$$h_i = \exp\left(-\frac{\|x_{in} - c_i\|^2}{b_i^2}\right), \quad i = 1, 2, \dots, n \quad (15)$$

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

$$\Delta_\xi(\dot{x}_{ld}, s, \dot{\xi}) = W^T h(x_{in}) \quad (16)$$

where  $W \in R^{n \times 6}$  is the weight matrix and  $n$  is number of neurons in the hidden layer. Based on Eq. (11), the approximation error ( $\varepsilon$ ) of nonlinear function  $\Delta_\xi$  is introduced as follows:

$$\Delta_\xi(\dot{x}_{ld}, s, \dot{\xi}) = W^T h(x_{in}) + \varepsilon \quad (17)$$

With respect to the approximation error ( $\varepsilon$ ), if the optimal weight matrix  $W^*$  is defined on the compact set  $\Theta$ , the upper limit of the approximation error can be defined as:

$$\varepsilon^* = \sup \left\| \Delta_\xi(\dot{x}_{ld}, s, \dot{\xi}) - W^T h(x_{in}) \right\|, x_{in} \in \Theta \quad (18)$$

Therefore, the approximation error ( $\varepsilon$ ) that corresponds to the optimal weights  $W^*$  is bounded by  $\|\varepsilon\| \leq \varepsilon^*$ .

For the approximation error ( $\varepsilon$ ), if the optimal weight  $W^*$  is bounded by a known positive value  $\|W^*\|_F \leq W_{max}$ .

The RBF neural network control algorithm is designed as:

$$\dot{x}_{nn} = \widehat{W}^T h(x_{in}) - K_r \left( \left\| \widehat{W} \right\|_F + W_{max} \right) (\|\Sigma\| / \|r\|) r \quad (19)$$

where  $\widehat{W}$  is the estimation matrix of  $W$  and the estimation errors are defined as  $\widehat{W} = W - \widehat{W}$ . The last term in Eq. (19) is the robust signal with a diagonal matrix

$Kr > 0$  and  $r = (\Sigma^T PB)^T \epsilon R^6 . P \in R^{12 \times 12}$  is the positive definite solution for the Lyapunov equation  $ATP + PA + Q = 0$ , where  $Q \in R^{12 \times 12}$  is a positive definite matrix.

The network is trained online by the following adaptive laws:

$$\dot{\hat{W}} = \Gamma hr^T + \kappa \Gamma \|\Sigma\| \hat{W} \tag{20}$$

where  $\Gamma > 0$  and  $\kappa > 0$  are the adaptive design parameters.

The desired velocity  $\dot{x}_{ld}$  screw in Eq. (12) can be gained from Eqs. (11), (19) and (20), thus enabling to obtain the desired pose screw ( $x_{ld}$ ) by the integral operation. With respect to RBF neural network control law (19) and weight adaptive law (20), the system error ( $\Sigma$ ) and neural network weight ( $\hat{W}$ ) are uniformly ultimately bounded in the compact set  $x_{in} \in \Theta$ .

The approximate error of uncertainty function  $\Delta_\xi$  by RBF neural network is defined as:

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

Substitute the Eq. (17) into the above equation:  $e_{nn} = \varepsilon - \hat{W}^T h(x_{in})$ .

The Lyapunov candidate function is established:  $V_{nn} = \frac{1}{2} \Sigma^T P \Sigma + \frac{1}{2\Upsilon} \text{tr}(\hat{W}^T \dot{W})$

Calculate differential of  $V_{nn}$  along the error dynamics (14):

$$\dot{V}_{nn} = -\frac{1}{2} \Sigma^T Q \Sigma + \varepsilon^T B^T P \Sigma - h^T \tilde{W} B^T P \Sigma + \frac{1}{\Upsilon} \text{tr}(\dot{\hat{W}}^T \tilde{W})$$

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

$$\dot{V}_{nn} = -\frac{1}{2} \Sigma^T Q \Sigma + \varepsilon^T B^T P \Sigma + \frac{1}{\Upsilon} \text{tr}(-\Upsilon B^T P \Sigma h^T \tilde{W} + \dot{\hat{W}}^T \tilde{W})$$

Substitute the Eq. (24) into  $\dot{V}_{nn}$ :

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

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

$$\text{tr}[\hat{W}^T \tilde{W}] \leq \|\tilde{W}\|_F \|W^*\|_F - \|\tilde{W}\|_F^2$$

The following equation is used:

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

Then:

$$\dot{V}_{nn} \leq -\|\Sigma\| \left( \frac{1}{2} \lambda_{min} Q \|\Sigma\| + k_1 \left( \|\tilde{W}\|_F - \frac{w_{max}}{2} \right)^2 - \frac{k_1}{4} w_{max}^2 - \|\varepsilon_0\| \lambda_{max} P \right)$$

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

$$\frac{1}{2} \lambda_{min} Q \|\Sigma\| \geq \frac{k_1}{4} w_{max}^2 + \|\varepsilon_0\| \lambda_{max} P$$

Then, if  $\|\Sigma\| \geq \frac{2}{\lambda_{min} Q} (\|\varepsilon_0\| \lambda_{min} P) + \frac{k_1}{4} w_{max}^2$ ,  $\dot{V}_{nn}$  is negative.

Therefore, the system tracking error  $\Sigma$  and weight matrices  $\tilde{W}$  are uniformly ultimately bounded.

For the proposed control law (19), the hybrid control consists of a closed loop system controlled by a deviation and an open loop system directly controlled by a disturbance signal. In this paper, the RBF neural network controller implements feedforward control to achieve the inverse dynamic model of the controlled object; PD implements feedback control to ensure system stability while suppressing disturbances. This hybrid control greatly improves the system's tracking accuracy and dynamics.

The proposed scheme is non-regressor based and requires no information about dynamic uncertainties and external disturbances of robot grasping. Using the RBF neural network to approximate the uncertain term  $\Delta_\xi(\dot{x}_{ld}, s, \dot{\xi})$ , can avoid using fixed large boundedness of robust controller to guarantee good performance, because large boundedness implies high noise amplification and high control cost.

The designed controller consists of two components. The first component is common PD control term which is used to guarantee the stability of the system and achieve uniformly ultimately performance. The second component is the adaptive RBF neural network to approximate the uncertain term  $\Delta_\xi(\dot{x}_{ld}, s, \dot{\xi})$ , and it is the robust term to attenuate disturbances.

As far as we know, all adaptive control methods should satisfy the persistent excitation condition due to the assumption that all the uncertainties in the system can be parameterized by the constant coefficient, but for the real systems, the assumption cannot be satisfied completely. Therefore, in our approach, we use the RBF neural network to deal with the uncertainties in the systems of the robot grasping, and avert the complex computation. There will relax the requirement of persistent excitation condition, and only require that the weight of neural network is convergent. Through the proof of the theorem (19), the designed tuning law of the

weight of RBF neural network can converge, consequently, the proposed control approach doesn't require the persistent excitation condition.

## 5 Experimental Analysis

### 5.1 Experimental Platform

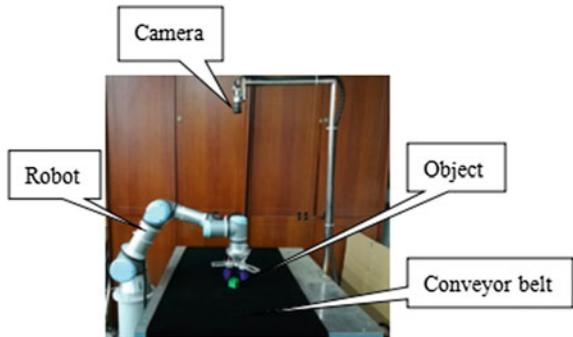
The industrial camera of experimental platform is fixed above the robot workspace with the accuracy of 0.1 mm, resolution of  $1600 \times 1200$  pixels and 30 frame/s (Fig. 4). The camera intrinsic and extrinsic matrices are calibrated by camera calibration tool box:

$$M = \begin{bmatrix} 1817.84 & 0 & 787.222 & 0 \\ 0 & 1818.79 & 595.201 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix},$$

$$T = \begin{bmatrix} -0.010705 & 0.999331 & -5.19069 & 354 \\ 0.998846 & 0.012369 & -3.2054 & 317 \\ 0.046804 & -0.034422 & 22.99 & 665 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

The initial joint angles  $q = [-0.563, -0.183, -0.656, -0.996, 1.283, 0.996]$  are given, and unit of joint angle values is rad. In the Cartesian coordinate system, the coordinates of end effector of robot is  $p = [7.61, 467.61, 0.03]$  mm. Parameters of RBF neural network are fixed: the node centers ( $c_i$ ) in the hidden layer 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  $n = 45$ . Control parameters are set as follows.

Fig. 4 Experimental platform

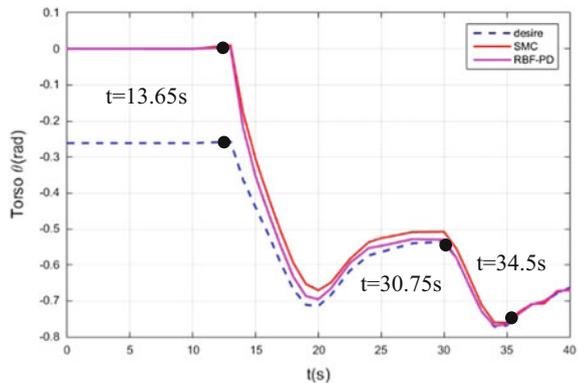


## 5.2 Experiment and Analysis

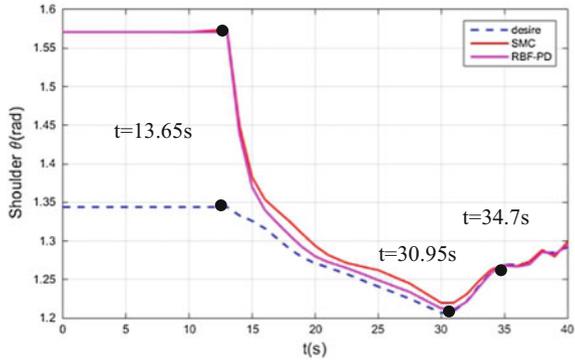
The experimental results of autonomous grasping of moving target will be presented in here to illustrate the desired and current joint positions of the torso  $\theta_1$ , the shoulder  $\theta_2$  and the elbow  $\theta_3$ . The grasping operation will be examined in detail where the target is acquired in stage 1 and the Kalman filtering algorithm begins to output the desired values for joint angles. Stage 2 denotes the beginning of tracking and the input of actual joint positions. In this stage, robot arm moves to the target quickly and its joint positions change greatly, causing violent vibration of the motion. To avoid collision with targets and improve the reliability and accuracy of grasping operation, the robot arm comes close to the target slowly, which is accompanied with slow changes of joint positions and light vibration of movement. When the end effector of the robot arm is close enough to the moving object, stage 3 takes effect and the robot grasps the object.

The experimental process of grasping the moving target can be divided into three stages. Experimental results of three stages are shown in Figs. 5, 6 and 7. The 0–13.65 s is at the first Stage, which searches and locks the object by the visual system and calculates inverse kinematics of robot. 13.65–34.7 s for the SMC (sliding mode controller) [27] algorithm and 13.65–30.99 s for the RBF-PD control algorithm are in the second stage. Visual system tracks movement of the object according to color invariant moment feature in order to increase imaging efficiency. Based on estimations of object pose and motion, the robot begins to track and approach to the object. When the end effector approaches to the object, both desired and actual joint angles begin to change simultaneously. This is because continuous change of positions of the object and end-effector takes place simultaneously when the attitude error between end-effector and object decreases gradually. However, the estimated desired joint angle is smoother than actual joint angle, because there's some residual vibration of joints in actual movement. After 34.7 s for the SMC algorithm and after 30.99 s for the RBF-PD control algorithm are in the third stage. The robot arm grasps the moving object successfully.

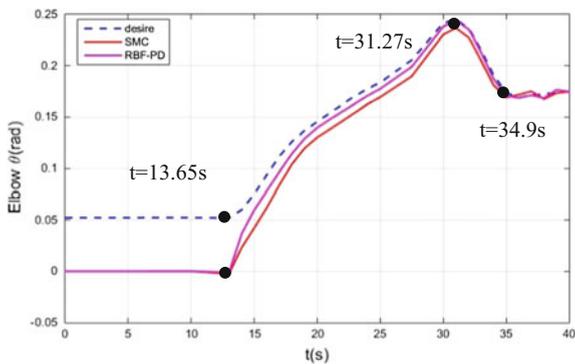
**Fig. 5** The angle curve of torso ( $\theta_1$ ) joint



**Fig. 6** The angle curve of shoulder ( $\theta_2$ ) joint



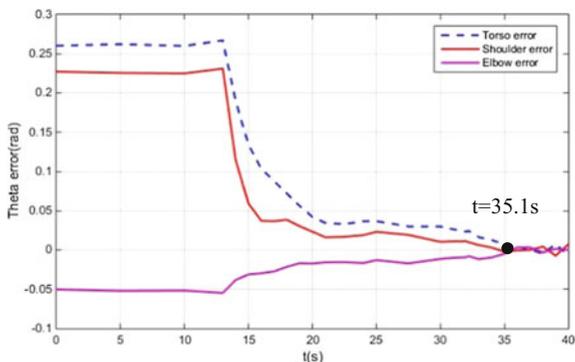
**Fig. 7** The angle curve of elbow ( $\theta_3$ ) joint



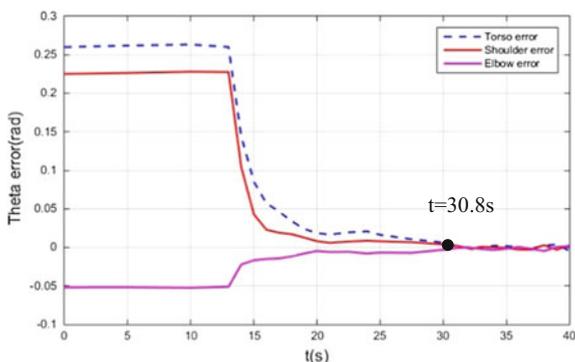
Angle curves of  $\theta_1$ ,  $\theta_2$  and  $\theta_3$  of the robot arm are shown in Figs. 5, 6 and 7. The blue dotted line is the desired joint angle, the red solid line is the actual joint angle using the traditional PD control algorithm, and purple solid line is the actual joint angle using RBF-PD control algorithm. As can be seen from the figure, the traditional PD control algorithm takes long adjustment time, and it costs about 22.58 s from 13.65 s to about 36 s before stabilization. The RBF-PD control algorithm takes short adjustment time, about 17.52 s from 13.65 s to about 31 s. Therefore, the RBF-PD control algorithm is superior to traditional PD control algorithm in term of adjustment time, which shows that it can track and move close to the object in the short time and can also increase reliability and accuracy of grasping.

The measurement error curves of the first three joint angles by the SMC algorithm are shown in Fig. 8. The measurement error curves of the RBF-PD control algorithm are shown in Fig. 9. Blue dotted line is the actual joint angle error of  $\theta_1$ , red solid line is the actual joint angle error of  $\theta_2$ , and the purple solid line is the actual joint angle error of  $\theta_3$ . It can be seen from Fig. 9 that the average errors of three joint angles from 13.65 to 35.1 s are about 0.042, 0.025 and 0.018 rad, respectively. In Fig. 10, the average errors of three joint angles from 13.65 to 30.8 s are about 0.021, 0.013 and 0.007 rad. 20 s later, the SMC algorithm still has great

**Fig. 8** Joint angle error of robot based on SMC control



**Fig. 9** Joint angle error of robot based on RBF-PD control



average error and great disturbance, but the RBF-PD control algorithm has stable error, accompanied with slow changes, and the error value approaches to 0 gradually. Therefore, joint angle error of the RBF-PD control algorithm is smaller than that of the SMC algorithm. Moreover, it can be seen from the graphs in Figs. 9 and Fig. 10 that joint angle changes of the RBF-PD control algorithm are gentler than those of the SMC algorithm, indicating that the RBF-PD control algorithm has strong anti-interference from noises and high robustness. Finally, experimental results demonstrated that compared to the SMC algorithm, the RBF-PD control algorithm achieves higher grasping accuracy and efficiency, 20% shorter adjustment time. Specifically, errors of  $\theta_1$ ,  $\theta_2$  and  $\theta_3$  are decreased by 27, 16 and 35%, respectively.

## 6 Conclusion

To solve the uncertainty for robot grasping of moving object, a RBF-PD control algorithm is proposed by combining PBVS-based RBF neural network and PD. The machine learning method is applied in the visual grasping control scheme of robot to offset uncertainty of grasping positions. Based on Lyapunov stability theory, the proposed control scheme can guarantee the stability, the uniformly ultimately bounded of the closed-system and the tracking performance of robot grasping system. Through experiment, the proposed controller is verified that it is robust not only to external disturbances but also to the parameter and non-parameter uncertainties.

**Acknowledgements** The authors would like to acknowledge the fund of Beijing Advanced Innovation Center for Intelligent Robots and Systems.

## References

1. Mehta SS, Burks TF (2014) Vision-based control of robotic manipulator for citrus harvesting. *Comput Electron Agric* 102:146–158
2. Larouche BP, Zhu ZH (2014) Autonomous robotic capture of non-cooperative target using visual servoing and motion predictive control. *Auton Robots* 37(2):157–167
3. Wang HS, Yang BH, Liu YT (2017) Visual servoing of soft robot manipulator in constrained environments with an adaptive controller. *IEEE/ASME Trans Mechatron* 22(1):41–50
4. Kazemi M, Gupta K, Mehrandezh M (2012) Path planning for image-based control of wheeled mobile manipulators. *IEEE/RSJ international conference on intelligent robots and systems (IROS)*, vol 2012. IEEE, Vilamoura, pp p5306–p5312
5. Hajiloo A, Keshmiri M, Xie WF (2016) Robust online model predictive control for a constrained image-based visual servoing. *IEEE Trans Industr Electron* 63(4):2242–2250
6. Lazar C, Burlacu A (2016) Image-based visual servoing for manipulation via predictive control—a survey of some results. *Mem Sci Sections Rom Acad* 39(11):71–81
7. Corke PI, Hutchinson SA (2015) A new partitioned approach to image-based visual servo control. *IEEE Trans Robot Autom* 17(4):507–515
8. Cazy N, Wieber PB, Giordano PR et al (2015) Visual servoing when visual information is missing: experimental comparison of visual feature prediction schemes. *IEEE international conference on robotics & automation*, vol 2015. IEEE, Seattle, pp p6031–p6036
9. Ghasemi MH, Kashiri N, Dardel M (2012) Time-optimal trajectory planning of robot manipulators in point-to-point motion using an indirect method. *ARCHIVE Proc Inst Mech Eng* 226(2):473–484
10. Zhang Q, Li SR, Gao XS (2013) Practical smooth minimum time trajectory planning for path following robotic manipulators. *American control conference*, vol 2013. IEEE, Washington, pp p2778–p2783
11. Banas W, Sekala A, Gwiazda A et al (2015) Determination of the robot location in a workcell of a flexible production line. *IOP Conf Ser: Mater Sci Eng* 95(1):1–6
12. Chesi G, Vicino A (2004) Visual servoing for large camera displacements. *IEEE Trans Rob* 20(4):724–735
13. Dong GQ, Zhu ZH (2015) Position-based visual servo control of autonomous robotic manipulators. *Acta Astronaut* 115:291–302

14. Henrik IC, Oussama K (2016) Modular design of image based visual servo control for dynamic mechanical systems. Springer Tracts Adv Robot 100:129–146
15. Malis E, Chaumette F, Boudet S (1999) 2&1/2-D visual servoing. IEEE Trans Robot Autom 15(2):238–250
16. Sadeghzadeh M, Calvert D, Abdullah HA (2015) Self-learning visual servoing of robot manipulator using explanation-based fuzzy neural networks and Q-learning. J Intell Rob Syst 78(1):83–104
17. Taghirad HD, Atashzar SF, Shahbazi M (2012) A robust solution to three-dimensional pose estimation using composite extended Kalman observer and Kalman filter. IET Comput Vision 6(2):140–152
18. Liu WH et al (2016) Real time pose estimation based on extended Kalman filter for binocular camera. Asia-pacific conference on intelligent robot systems (ACIRS), vol 2016. IEEE, Tokyo, pp p142–p145
19. Salehian M, Rayatdoost S, Taghirad HD (2011) Robust unscented Kalman filter for visual Servoing system. 2nd international conference on control, instrumentation and automation (ICCIA), vol 2011. IEEE, Shiraz, pp p1006–p1011
20. Zhang K, Chen J, Jia B et al (2017) Unified tracking and regulation visual servoing of wheeled mobile robots with euclidean reconstruction. American control conference (ACC), vol 2017. IEEE, Seattle, pp p1942–p1947
21. Atia KR (2013) A new variable structure controller for robot manipulators with a nonlinear PID sliding surface. Robotica 31:503–510
22. Kim J, Kumar N, Panwar V et al (2012) Adaptive neural controller for visual servoing of robot manipulators with camera-in-hand configuration. J Mech Sci Technol 26(8):2313–2323
23. Chen DS, Li GF, Jiang GZ et al (2015) Intelligent computational control of multi-fingered dexterous robotic hand. J Comput Theor Nanosci 12(7):6126–6132
24. Xie T, Yu H, Wilamowski H (2011) Comparison between traditional neural networks and radial basis function networks. IEEE Int Symp Ind Electron 19(5):1194–1199
25. Yang HJ, Liu JK (2018) An adaptive RBF neural network control method for a class of nonlinear systems. IEEE/CAA J Automatica Sinica 5(2):457–462
26. Saeed EA, Eleni C, Costas P (2015) A dual Kalman filter approach for state estimation via output-only acceleration measurements. Mech Syst Signal Process 60:866–886
27. Parsapour M, RayatDoost S et al (2015) A 3D sliding mode control approach for position based visual servoing system. Sci Iranica Trans B Mech Eng 22(3):844–853