

We are IntechOpen, the world's leading publisher of Open Access books Built by scientists, for scientists

6,900

Open access books available

186,000

International authors and editors

200M

Downloads

Our authors are among the

154

Countries delivered to

TOP 1%

most cited scientists

12.2%

Contributors from top 500 universities



WEB OF SCIENCE™

Selection of our books indexed in the Book Citation Index
in Web of Science™ Core Collection (BKCI)

Interested in publishing with us?
Contact book.department@intechopen.com

Numbers displayed above are based on latest data collected.
For more information visit www.intechopen.com



Visual Feedback Control of a Robot in an Unknown Environment (Learning Control Using Neural Networks)

Xiao Nan-Feng and Saeid Nahavandi

1. Introduction

When a robot has no transcendental knowledge about an object to be traced and an operation environment, a vision sensor is needed to attach to the robot in order to recognize the object and the environment. On the other hand, it is also desirable that the robot has learning ability in order to improve effectively the trace operation in the unknown environment.

Many methods⁽¹⁾⁻⁽¹¹⁾ have been so far proposed to control a robot with a camera to trace an object so as to complete a non-contact operation in an unknown environment. e.g., in order to automate a sealing operation by a robot, Hosoda, K.⁽¹⁾ proposed a method to perform the sealing operation by the robot through off-line teaching beforehand. This method used a CCD camera and slit lasers to detect the sealing line taught beforehand and to correct on line the joint angles of the robot during the sealing operation.

However, in those methods⁽¹⁾⁻⁽³⁾, only one or two image feature points of the sealing were searched per image processing period and the goal trajectory of the robot was generated using an interpolation. Moreover, those methods must perform the tedious CCD camera calibration and the complicated coordinate transformations. Furthermore, the synchronization problem between the image processing system and the robot control system, and the influences of the disturbances caused by the joint friction and the gravity of the robot need to be solved.

In this chapter, a visual feedback control method is presented for a robot to trace a curved line in an unknown environment. Firstly, the necessary conditions are derived for one-to-one mapping from the image feature domain of the curved line to the joint angle domain of the robot, and a multilayer neural network which will be abbreviated to NN hereafter is introduced to learn the mapping. Secondly, a method is proposed to generate on line the goal trajectory through computing the image feature parameters of the curved line. Thirdly, a multilayer neural network-based on-line learning algorithm is developed for the present visual feedback control. Lastly, the present approach is applied to trace a curved line using a 6 DOF industrial robot with a CCD cam-

era installed in its end-effector. The main advantage of the present approach is that it does not necessitate the tedious CCD camera calibration and the complicated coordinate transformations.

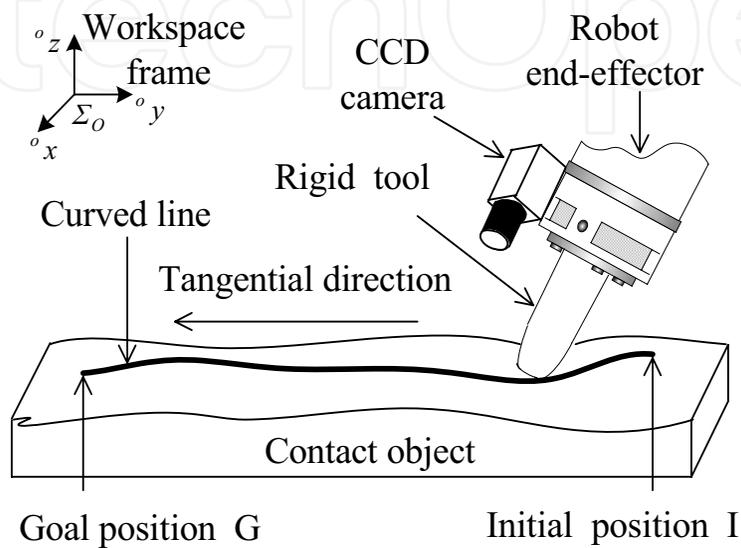


Figure 1. Vision-based trace operation

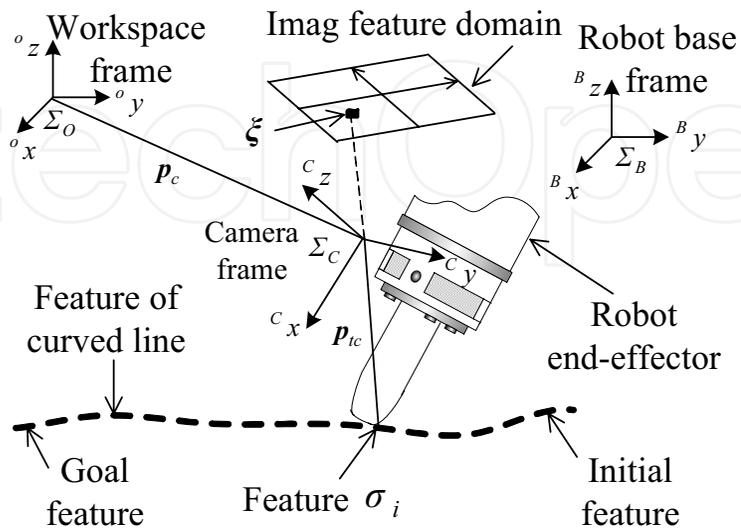


Figure 2. Image features and mapping relation

2. A Trace Operation by a Robot

When a robot traces an object in an unknown environment, visual feedback is necessary for controlling the position and orientation of the robot in the tangential and normal directions of the operation environment. Figure 1 shows a robot with a CCD camera installed in its end-effector. The CCD camera guides the end-effector to trace a curved line from the initial position I to the target position G. Since the CCD camera is being fixed at the end-effector, the CCD camera and the end-effector always move together.

3. Mapping from Image Feature Domain to Joint Angle Domain

3.1 Visual feedback control

For the visual feedback control shown in Fig. 1, the trace error of the robot in the image feature domain needs to be mapped to the joint angle domain of the robot. That is, the end-effector should trace the curved line according to a_j, a_{j+1} in the image domain of the features A_j, A_{j+1} shown in Fig. 2.

Let $\xi, \xi_d \in R^{6 \times 1}$ be the image feature parameter vectors of a_j, a_{j+1} in the image feature domain shown in Fig. 2, respectively. The visual feedback control shown in Fig. 1 can be expressed as

$$e = \left\| \xi_d - \xi \right\|, \quad (1)$$

where $\| \cdot \|$ is a norm, and e should be made into a minimum.

From the projection relation shown in Fig. 2, we know

$$\xi = \varphi(p_{tc}), \quad (2)$$

where $\varphi \in R^{6 \times 1}$ is a nonlinear mapping function which realizes the projection transformation from the workspace coordinate frame \sum_O to the image feature domain shown in Fig. 2.

It is assumed that $p_{tc} \in R^{6 \times 1}$ is a position/orientation vector from the origin of the CCD camera coordinate frame \sum_C to the gravity center of A_j . Linearizing Eq.(2) at a minute domain of p_{tc} yields

$$\delta \xi = J_f \cdot \delta p_{tc}, \quad (3)$$

where $\delta \xi$ and δp_{tc} are minute increments of ξ and p_{tc} , respectively, and $J_f = \partial \varphi / \partial p_{tc} \in R^{6 \times 6}$ is a feature sensitivity matrix.

Furthermore, let θ and $\delta \theta \in R^{6 \times 1}$ are a joint angle vector of the robot and its minute increment in the robot base coordinate frame Σ_B . If we map $\delta \theta$ from Σ_B to δp_{tc} in Σ_O using Jacobian matrix of the CCD camera $J_c \in R^{6 \times 6}$, we can get

$$\delta p_{tc} = J_c \cdot \delta \theta. \quad (4)$$

From Eqs.(3) and (4), we have

$$\delta \theta = (J_f J_c)^{-1} \cdot \delta \xi. \quad (5)$$

Therefore, the necessary condition for realizing the mapping expressed by Eq.(5) is that $(J_f J_c)^{-1}$ must exist. Moreover, the number of the independent image feature parameters in the image feature domain (or the element numbers of ξ) must be equal to the degrees of freedom of the visual feedback control system.

3.2 Mapping relations between image features and joint angles

Because J_f and J_c are respectively linearized in the minute domains of p_{tc} and θ , the motion of the robot is restricted to a minute joint angle domain, and Eq.(5) is not correct for large $\delta \theta$. Simultaneously, the mapping is weak to the change of $(J_f J_c)^{-1}$. In addition, it is very difficult to calculate $(J_f J_c)^{-1}$ on line during the trace operation. Therefore, NN is introduced to learn such mapping.

Firstly, we consider the mapping from Δp_{tc} in Σ_O to $\Delta \xi$ in the image feature domain. $\Delta \xi$ and Δp_{tc} are increments of ξ and p_{tc} for the end-effector to move from A_j to A_{j+1} , respectively. As shown in Fig. 2, the mapping is depend on p_{tc} , and the mapping can be expressed as

$$\Delta \xi = \varphi_1 (\Delta p_{tc}, p_{tc}), \quad (6)$$

where $\varphi_1 (\cdot) \in R^{6 \times 1}$ is a continuous nonlinear mapping function. We have from Eq.(6),

$$\Delta p_{tc} = \varphi_2 (\Delta \xi, p_{tc}), \quad (7)$$

where $\varphi_2 (\cdot) \in R^{6 \times 1}$ is a continuous nonlinear mapping function. When ξ is uniquely specified in the image feature domain, there is a one-to-one mapping relationship between p_{tc} and ξ . Therefore, Eq.(7) is expressed as

$$\Delta p_{ic} = \varphi_3(\Delta \xi, \xi), \quad (8)$$

where $\varphi_3(\cdot) \in R^{6 \times 1}$ is a continuous nonlinear mapping function.

Secondly, we consider the mapping from $\Delta \theta$ in Σ_B to Δp_c in Σ_O . Let $p_c \in R^{6 \times 1}$ be a position/ orientation vector of the origin of Σ_C with respect to the origin of Σ_O , $\Delta \theta$ and Δp_c be increments of θ and p_c for the end-effector to move from A_j to A_{j+1} , respectively. Δp_c is dependent on θ as shown in Fig. 2, and we obtain from the forward kinematics of the CCD camera

$$\Delta p_c = \varphi_3'(\Delta \theta, \theta), \quad (9)$$

where $\varphi_3'(\cdot) \in R^{6 \times 1}$ is a continuous nonlinear mapping function.

Since the relative position and orientation between the end-effector and the CCD camera are fixed, the mapping from Δp_c to Δp_{ic} is also one-to-one. We get

$$\Delta p_{ic} = \varphi_4(\Delta p_c), \quad (10)$$

where $\varphi_4(\cdot) \in R^{6 \times 1}$ is a continuous nonlinear mapping function. Combining Eq.(9) and Eq.(10) gives

$$\Delta p_{ic} = \varphi_4'(\Delta \theta, \theta), \quad (11.a)$$

and we have from Eq.(11.a)

$$\Delta \theta = \varphi_5(\Delta p_{ic}, \theta), \quad (11.b)$$

where $\varphi_5(\cdot) \in R^{6 \times 1}$ is a continuous nonlinear mapping function. It is known from Eq.(11.b) that if the CCD camera moves from A_j to A_{j+1} , the robot has a unique $\Delta \theta$.

Combining Eq.(8) and Eq.(11.b) yields

$$\Delta \theta = \varphi_6(\Delta \xi, \xi, \theta), \quad (12)$$

where $\varphi_6(\cdot) \in R^{6 \times 1}$ is a continuous nonlinear mapping function. In this paper, NN is used to learn $\varphi_6(\cdot)$.

3.2 Computation of image feature parameters

For 6 DOF visual feedback control, 6 independent image feature parameters are chosen to correspond to the 6 joint angles of the robot. An image feature parameter vector $\xi^{(j)} = [\xi_1^{(j)}, \xi_2^{(j)}, \dots, \xi_6^{(j)}]^T$ is defined at the window j shown in Fig. 3. L and W are length and height of the window j , respectively.

Defining $g_{qr}^{(j)}$ at the window j by

$$g_{qr}^{(j)} = \begin{cases} 0 & \text{(white pixel)} \\ 1 & \text{(black pixel)} \end{cases} \quad (13.a)$$

the elements of $\xi^{(j)}$ are defined and calculated by the following equations:

(1) The gravity center coordinates

$$\xi_1^{(j)} = \frac{\sum_{q=1}^L \sum_{r=1}^W g_{qr}^{(j)} \cdot q}{\sum_{q=1}^L \sum_{r=1}^W g_{qr}^{(j)}} \quad (13.b)$$

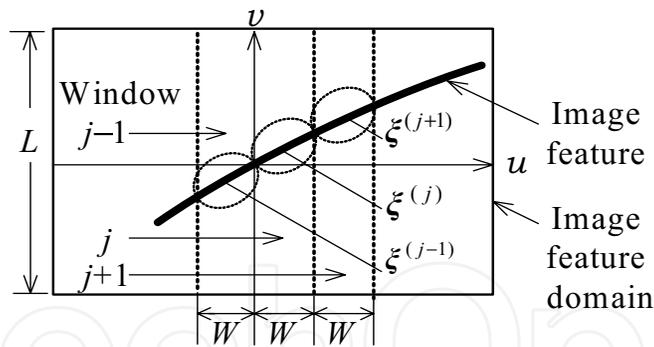


Figure 3. Definition of image feature parameter vector

$$\xi_2^{(j)} = \frac{\sum_{q=1}^L \sum_{r=1}^W g_{qr}^{(j)} \cdot r}{\sum_{q=1}^L \sum_{r=1}^W g_{qr}^{(j)}} \quad (13.c)$$

(2) The area

$$\xi_3^{(j)} = \sum_{q=1}^L \sum_{r=1}^W g_{qr}^{(j)} \quad (13.d)$$

(3) The lengths of the main and minor axes of the equivalent ellipse

$$\xi_4^{(j)} = \frac{\lambda_{20}^{(j)} + \lambda_{02}^{(j)} + \sqrt{(\lambda_{20}^{(j)} - \lambda_{02}^{(j)})^2 + 4(\lambda_{11}^{(j)})^2}}{2\xi_3^{(j)}}, \quad (13.e)$$

$$\xi_5^{(j)} = \frac{\lambda_{20}^{(j)} + \lambda_{02}^{(j)} - \sqrt{(\lambda_{20}^{(j)} - \lambda_{02}^{(j)})^2 + 4(\lambda_{11}^{(j)})^2}}{2\xi_3^{(j)}}, \quad (13.f)$$

Where

$$\lambda_{11}^{(j)} = \sum_{q=1}^L \sum_{r=1}^W g_{qr}^{(j)} \cdot [q - \xi_1^{(j)}][r - \xi_2^{(j)}], \quad (13.g)$$

$$\lambda_{20}^{(j)} = \sum_{q=1}^L \sum_{r=1}^W g_{qr}^{(j)} \cdot [q - \xi_1^{(j)}]^2, \quad (13.h)$$

$$\lambda_{02}^{(j)} = \sum_{q=1}^L \sum_{r=1}^W g_{qr}^{(j)} \cdot [r - \xi_2^{(j)}]^2, \quad (13.i)$$

(4) The orientation

$$\xi_6^{(j)} = \frac{1}{2} \tan^{-1} \frac{\lambda_{11}^{(j)}}{\lambda_{20}^{(j)} - \lambda_{02}^{(j)}}. \quad (13.j)$$

At time $t=imT$, ξ and $\Delta\xi$ in Eq.(12) are given by

$$\xi = \xi^{(j)}(im), \quad (14.a)$$

$$\Delta\xi = \Delta\xi(im), \quad (14.b)$$

where $\xi^{(j)}(im)$ and $\xi^{(j+1)}(im)$ are image feature parameter vectors in the window j and $j+1$ shown in Fig. 3. $\xi^{(0)}(im)$, $\xi^{(1)}(im)$ and $\xi^{(2)}(im)$ can be calculated for $j=0,1,2$ in Eq.(13.a)~(13.j).

4. Goal Trajectory Generation Using Image Feature Parameters

In this paper, a CCD camera is used to detect the image feature parameters of the curved line, which are used to generate on line the goal trajectory. The se-

quences of trajectory generation are shown in Fig. 4. Firstly, the end-effector is set to the central point of the window 0 in Fig. 4(a). At time $t=0$, the first image of the curved line is grasped and processed, and the image feature parameter vectors $\xi^{(0)}(0)$, $\xi^{(1)}(0)$ and $\xi^{(2)}(0)$ in the windows 0,1,2 are computed respectively. From time $t=mT$ to $t=2mT$, the end-effector is only moved by $\Delta\xi(0) = \xi^{(1)}(0) - \xi^{(0)}(0)$. At time $t=mT$, the second image of the curved line is grasped and processed, the image feature parameter vector $\xi^{(2)}(m)$ shown in Fig. 4(b) is computed. From $t=2mT$ to $t=3mT$, the end-effector is only moved by $\Delta\xi(m) = \xi^{(2)}(0) - \xi^{(1)}(0)$. At time $t=imT$, $(i+1)$ th image is grasped and processed, the image feature parameter vector $\xi^{(2)}(im)$ shown in Fig. 4(c) is computed. From $t=imT$ to $t=(i+1)mT$, the end-effector is only moved by $\Delta\xi(im) = \xi^{(1)}[(i-1)m] - \xi^{(0)}[(i-1)m]$.

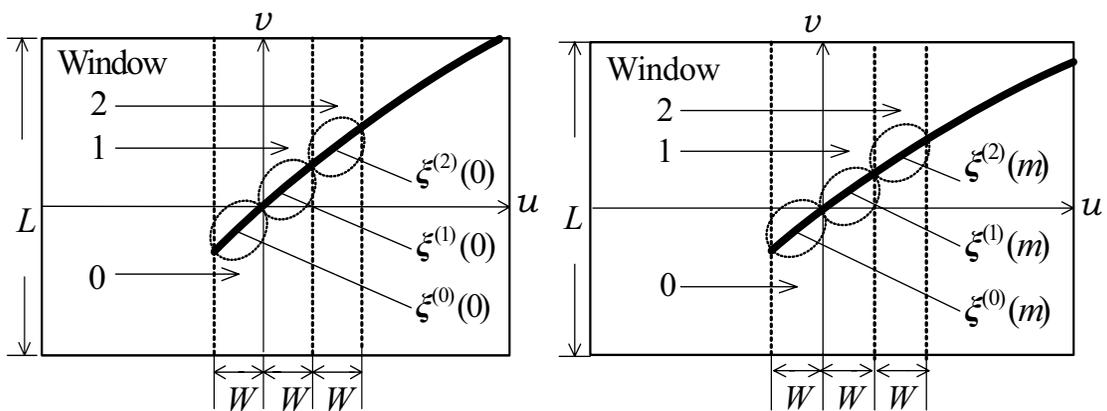
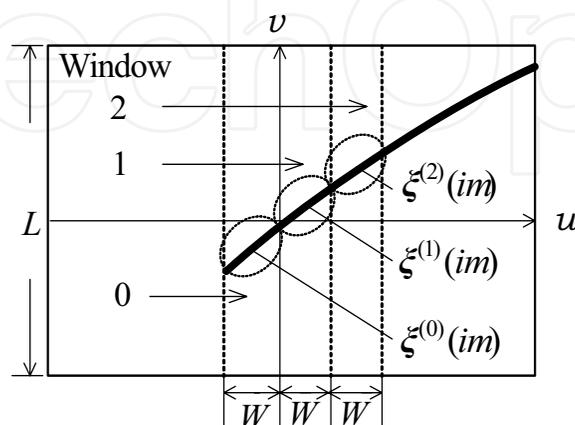
(a) Image feature domain at $t=0$ (b) Image feature domain at $t=mT$ (c) Image feature domain $t=imT$

Figure 4. Image trajectory generation sequences

5. Neural Network-Based Learning Control System

5.1 Off-line learning algorithm

In this section, a multilayer NN is introduced to learn $\phi_s(\cdot)$. Figure 5 shows the structure of NN which includes the input level A, the hidden level B, C and the output level D. Let M, P, U, N be the neuron number of the levels A, B, C, D, respectively, $g=1,2,\dots,M, l=1,2,\dots,P, j=1,2,\dots,U$ and $i=1,2,\dots,N$. Therefore, NN can be expressed as

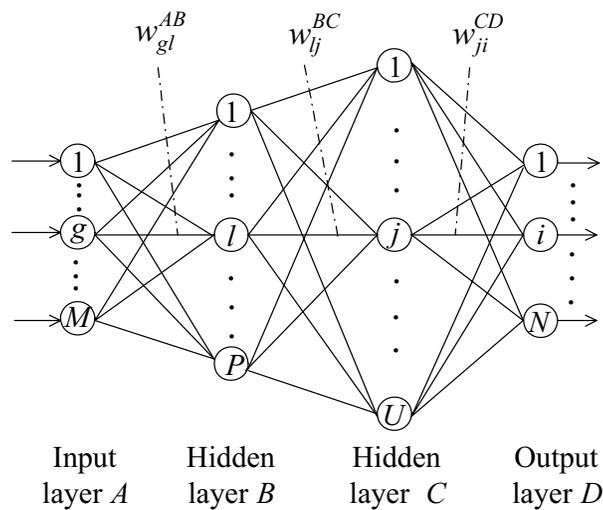


Figure 5. Neural network (NN) structure

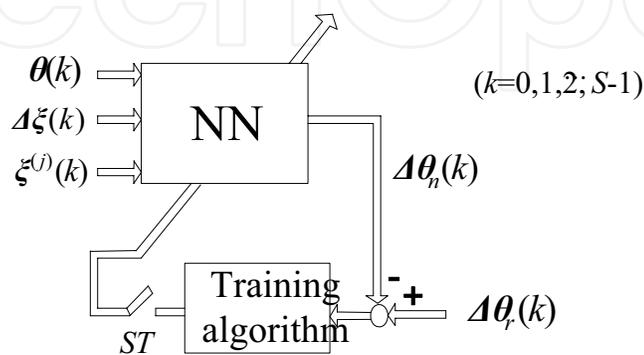


Figure 6. Off-line learning of NN

$$\left. \begin{aligned} x_l^B &= \sum_{g=1}^M w_{gl}^{AB} \cdot y_g^A + z_l^B, & x_j^C &= \sum_{l=1}^P w_{lj}^{BC} \cdot y_l^B + z_j^C, & x_i^D &= \sum_{j=1}^U w_{ji}^{CD} \cdot y_j^C + z_i^D \\ y_g^A &= x_g^A, & y_l^B &= f(x_l^B), & y_j^C &= f(x_j^C), & y_i^D &= f(x_i^D) \end{aligned} \right\}, \quad (15)$$

where x_m^R , y_m^R and z_m^R ($m=g, l, j, i$; $R=A, B, C, D$) are the input, the output and the bias of a neuron, respectively. w_{gl}^{AB} , w_{lj}^{BC} and w_{ji}^{CD} are the weights between y_g^A and x_l^B , y_l^B and x_j^C , y_j^C and x_i^D , respectively. The sigmoid function of NN is defined by

$$f(x) = \frac{1 - e^{-\beta x}}{1 + e^{-\beta x}}, \quad (16)$$

where β is a real number which specifies the characteristics of $f(x)$. The learning method of NN is shown in Fig. 6, and its evaluation function is defined as

$$E_f = \frac{1}{2S} \sum_{k=0}^{S-1} [\Delta \theta_r(k) - \Delta \theta_n(k)]^T [\Delta \theta_r(k) - \Delta \theta_n(k)], \quad (17)$$

where $\Delta \theta_r(k)$ and $\Delta \theta_n(k) \in R^{6 \times 1}$ are the increments of the robot joint angle vector and the output vector of the neural network, respectively. The positive integer S is the number of the learning samples $\theta(k)$, $\Delta \theta_r(k)$, $\xi^{(j)}(k)$, $\Delta \xi(k)$ for the end-effector to move along the curved line from the initial position I to the goal position G. $\theta(k)$, $\Delta \theta_r(k)$, $\xi^{(j)}(k)$ and $\Delta \xi(k)$ are off-line measured in advance, respectively. $\theta(k)$, $\xi^{(j)}(k)$ and $\Delta \xi(k)$ are divided by their maximums before inputting to NN, respectively. w_{ji}^{CD} of NN is changed by

$$\Delta w_{ji}^{CD} = -\eta_f \frac{\partial E_f}{\partial w_{ji}^{CD}}, \quad (j=1,2,\dots,U; i=1,2,\dots,N), \quad (18.a)$$

where Δw_{ji}^{CD} is an increment of w_{ji}^{CD} , η_f is a learning rate of NN. From Eqs.(15)~(17), we obtain

$$\frac{\partial E_f}{\partial w_{ji}^{CD}} = -[\Delta \theta_{ri}(k) - \Delta \theta_{ni}(k)] f'(x_i^D) y_j^C, \quad (j=1,2,\dots,U; i=1,2,\dots,N), \quad (18.b)$$

where $\Delta \theta_{ri}(k)$ and $\Delta \theta_{ni}(k)$ are the i th element of $\Delta \theta_r(k)$ and $\Delta \theta_n(k)$, respectively. w_{gl}^{AB} and w_{lj}^{BC} of NN are changed by the error back-propagation algorithm. Here, the detailed process of error back propagation is omitted.

The present learning control system based on NN is shown Fig. 7. The initial w_{gl}^{AB} , w_{lj}^{BC} and w_{ji}^{CD} of NN are given by random real numbers between -0.5 and 0.5 . When NN finishes learning, the reference joint angle $\theta_n(k+1)$ of the robot is obtained by

$$\left. \begin{aligned} \theta_n(k+1) &= \theta_n(k) + \Delta\theta_n(k), \\ \theta_n(0) &= \theta(0), \quad \Delta\theta_n(0) = \mathbf{d}, \quad (k = 0, 1, \dots, S-1) \end{aligned} \right\} \quad (19)$$

where $\mathbf{d} \in R^{6 \times 1}$ is an output of NN when $\theta(0)$, $\xi^{(0)}$ and $\Delta\xi(0)$ are used as initial inputs of NN. The PID controller $G_c(z)$ is used to control the joint angles of the robot.

5.2 On-line learning algorithm

For the visual feedback control system, a multilayer neural network NN_c is introduced to compensate the nonlinear dynamics of the robot. The structure and definition of NN_c is the same as NN, and its evaluation function is defined as

$$E_c(k) = e^T(k)W e(k) \quad (20)$$

where $e(k) \in R^{6 \times 1}$ is an input vector of $G_c(z)$, and $W \in R^{6 \times 6}$ is a diagonal weighting matrix. w_{ji}^{CD} of NN_c is changed by

$$\Delta w_{ji}^{CD} = -\eta_c \frac{\partial E_c}{\partial w_{ji}^{CD}}, \quad (j=1, 2, \dots, U; i=1, 2, \dots, N), \quad (21.a)$$

where Δw_{ji}^{CD} is an increment of w_{ji}^{CD} , and η_c is a learning rate. From Eqs.(15)~(17), we have

$$\frac{\partial E_c}{\partial w_{ji}^{CD}} = -e_i w_i f'(x_i^D) y_j^C \quad (j=1, 2, \dots, U; i=1, 2, \dots, N), \quad (21.b)$$

where e_i is the i th element of $e(k)$, and w_i is the i th diagonal element of W . w_{gl}^{AB} and w_{lj}^{BC} of NN_c are changed by the error back-propagation algorithm, the process is omitted.

The initial w_{gl}^{AB} , w_{lj}^{BC} and w_{ji}^{CD} of NN_c are given by random number between -0.5 and 0.5 . $\theta_n(k+1)$, $\Delta\theta_n(k)$ and $\Delta\theta_n(k-1)$ are divided by their maximums before inputting to NN_c , respectively. K is a scalar constant which is specified by the experiments. While NN_c is learning, the elements of $e(k)$ will become smaller and smaller.

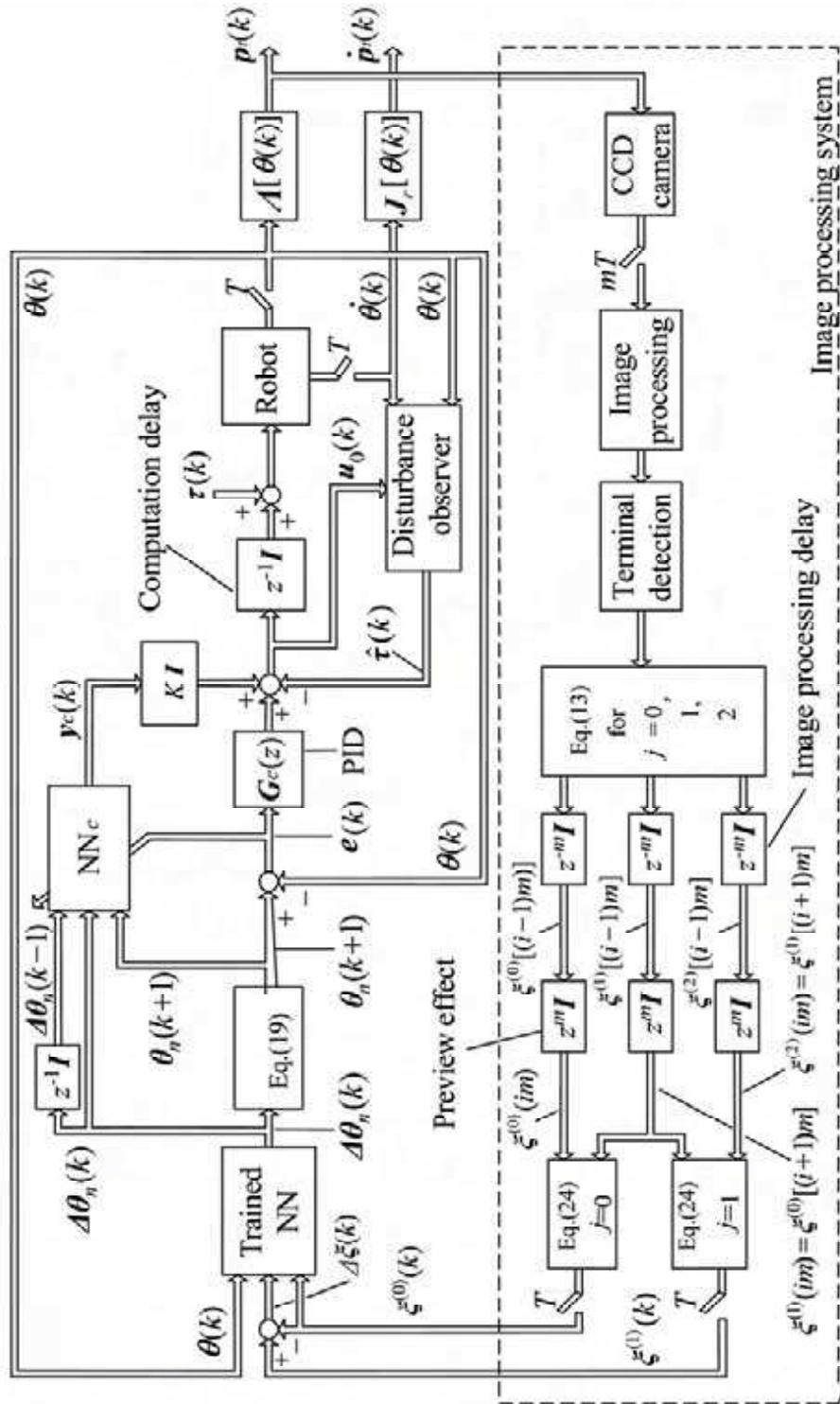


Figure 7. Block diagram of learning control system for a robot with visual feedback

In Fig. 7, I is a 6×6 unity matrix. $A[\theta(k)] \in R^{6 \times 1}$ and $J_r[\theta(k)] = \partial A[\theta(k)] / \partial \theta(k) \in R^{6 \times 6}$ are the forward kinematics and Jacobian matrix of the end-effector (or the rigid tool), respectively. Let $p_t(k) = [p_{t1}, p_{t2}, \dots, p_{t6}]^T$ be a position/orientation vector which is defined in Σ_O and corresponds to the tip of the rigid tool, we have

$$p_t(k) = A[\theta(k)], \quad (22.a)$$

$$\dot{p}_t(k) = J_r[\theta(k)] \cdot \dot{\theta}(k). \quad (22.b)$$

The disturbance observer is used to compensate the disturbance torque vector $\tau(k) \in R^{6 \times 1}$ produced by the joint friction and gravity of the robot. The PID controller $G_c(z)$ is given by

$$G_c(z) = K_p + K_i \frac{z}{1-z} + K_d (1-z^{-1}), \quad (23)$$

where K_p , K_i and $K_d \in R^{6 \times 6}$ are diagonal matrices which are empirically determined.

5.3 Work sequence of the image processing system

The part circled by the dot line shown in Fig. 7 is an image processing system. The work sequence of the image processing system is shown in Fig. 8. At time $t=imT$, the CCD camera grasps a 256-grade gray image of the curved line, the image is binarized, and the left and right terminals of the curved line are detected. Afterward, the image parameters $\xi^{(0)}(0), \xi^{(1)}(0)$ and $\xi^{(2)}(0)$ (or $\xi^{(2)}(im)$) are computed using Eqs. (13.a)~(13.j) in the windows 0,1,2 shown in Figs. 4(a)~(c). Furthermore, in order to synchronize the image processing system and the robot joint control system, the 2nd-order holder $G_{h2}(z)$ in Section 5.4 is introduced. $\xi^{(0)}(im), \xi^{(1)}(im)$ and $\Delta \xi(im)$ are processed by $G_{h2}(z)$, and their discrete time values $\xi^{(0)}(k), \xi^{(1)}(k)$ and $\Delta \xi(k)$ are solved at time $t=kT$.

5.4 Synchronization of image processing system and robot control system

Generally, the sampling period of the image processing system is much longer than that of the robot control system. Because the sampling period of $\xi^{(0)}(im), \xi^{(1)}(im)$ and $\Delta \xi(im)$ is m times of the sampling period of $\theta(k)$, it is difficult to synchronize $\xi^{(0)}(im), \xi^{(1)}(im), \Delta \xi(im)$ and $\theta(k)$ by zero-order holder or 1st-order holder. Otherwise, the robot will drastically accelerate or decelerate during the visual feedback control.

In this section, $\xi^{(0)}(im)$ and $\xi^{(1)}(im)$ are processed by the 2nd-order holder $G_{h2}(z)$. For instance, $\xi_l^{(j)}$ is the l th element of $\xi^{(j)}$, and $\xi_l^{(j)}$ is compensated by the 2nd-order curved line from $t=imT$ to $t=(i+1)mT$. At time $t=kT$, $\xi^{(j)}(k)$ ($j=0,1$) is calculated by

$$\xi^{(j)}(k) = \xi^{(j)}(im) + \frac{2(k-im)^2}{m^2} \{ \xi^{(j)}[(i+1)m] - \xi^{(j)}(im) \}, \quad (0 \leq k-im < m/2), \quad (24.a)$$

$$\xi^{(j)}(k) = \xi^{(j)}(im) + \left\{ 1 - \frac{2[k-(i+1)m]^2}{m^2} \right\} \{ \xi^{(j)}[(i+1)m] - \xi^{(j)}(im) \}, \quad (m/2 \leq k-im < m). \quad (24.b)$$

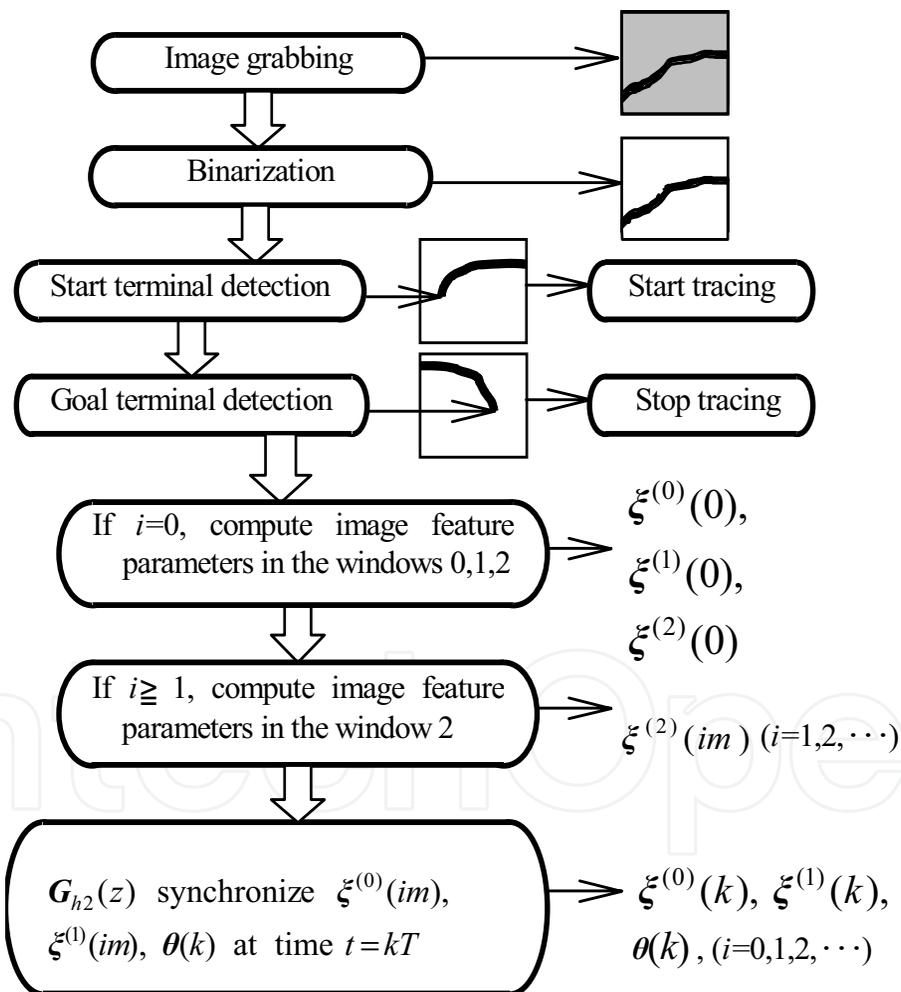


Figure 8. Work sequence of image processing

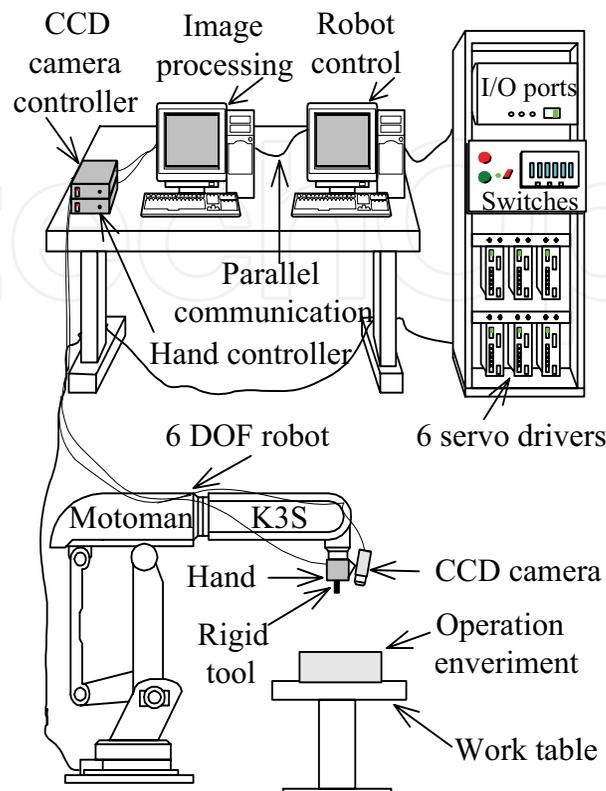


Figure 9. Trace operation by a robot

6. Experiments

In order to verify the effectiveness of the present approach, the 6 DOF industrial robot is used to trace a curved line. In the experiment, the end-effector (or the rigid tool) does not contact the curved line. Figure 9 shows the experimental setup. The PC-9821 Xv21 computer realizes the joint control of the robot. The Dell XPS R400 computer processes the images from the CCD camera. The robot hand grasps the rigid tool, which is regarded as the end-effector of the robot. The diagonal elements of K_p , K_i and K_d are shown in Table 1, the control parameters used in the experiments are listed in Table 2, and the weighting matrix W is set to be an unity matrix I .

Figures 10 and 11 show the position responses (or the learning results of NN and NN_c) $p_{x_1}(k)$, $p_{x_2}(k)$ and $p_{z_3}(k)$ in the directions of x , y and z axes of $\sum O$. $p_{x_1}(k)$, $p_{x_2}(k)$ and $p_{z_3}(k)$ shown in Fig. 10 are teaching data. Figure 11 show the trace responses after the learning of NN_c. Figure 12 shows the learning processes of NN and NN_c as well as the trace errors. E_f converges on 10^{-9} rad², the learning error E^* shown in Fig. 12(b) is given by $E^* = [\sum_{k=0}^{N-1} E_c(k)]/N$, where $N=1000$. After the 10 trials (10000 iterations) using NN_c, E^* converges on 7.6×10^{-6} rad², and the end-

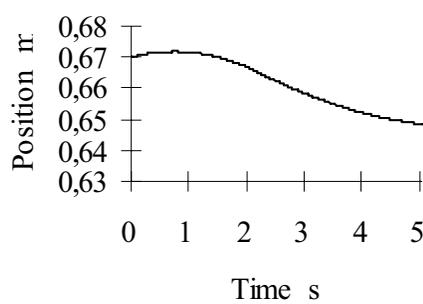
effector can correctly trace the curved line. Figure 12(c) shows the trace errors of the end-effector in x , y , z axis directions of Σ_O , and the maximum error is lower than 2 mm.

i	K_p	K_I	K_D
	N·m/rad	N·m/rad	N·m/rad
1	25473	0.000039	0.0235
2	8748	0.000114	0.0296
3	11759	0.000085	0.0235
4	228	0.004380	0.0157
5	2664	0.000250	0.0112
6	795	0.001260	0.0107

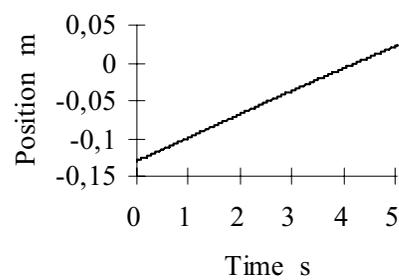
Table 1. Diagonal i th element of K_p , K_I and K_D

Neuron numbers of NN and NN _c	$M=18, P=36$ $U=72, N=6$
Sampling period	$T=5$ ms, $mT=160$ ms
Control Parameters	$\eta_f=0.9, \eta_c=0.9, \beta=1,$ $K=100, S=732$
Window size	$L=256, W=10$

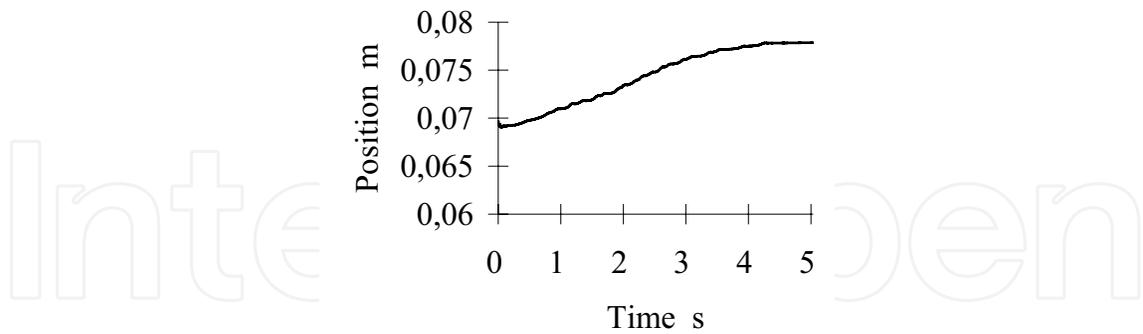
Table 2. Parameters used in the experiment



(a) Position p_{t1} in x direction

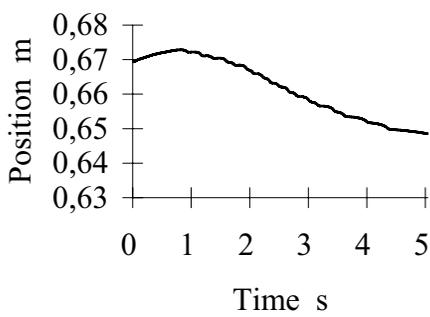


(b) Position p_{t2} in y direction

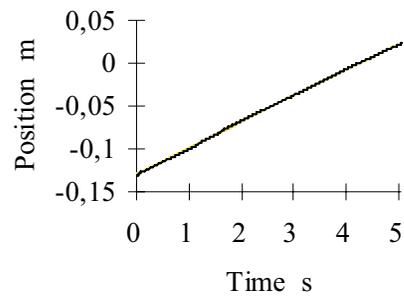


(c) Position p_{13} in z direction

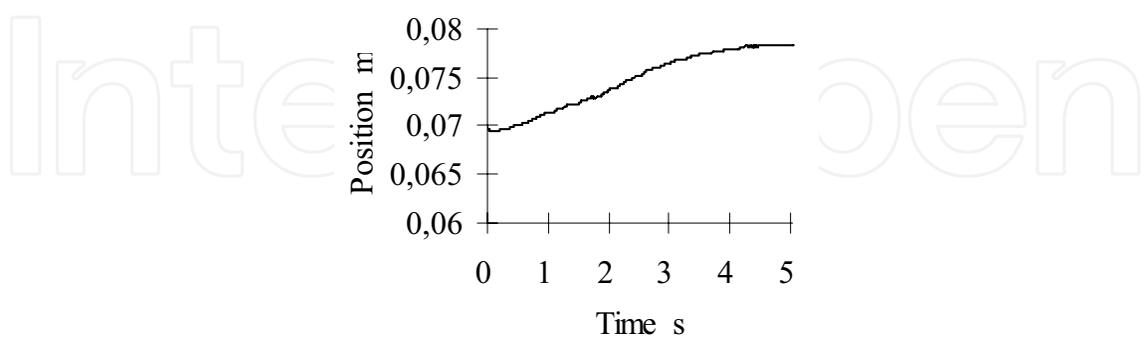
Figure 10. The teaching data for the learning of NN



(a) Response p_{11} in x direction



(b) Response p_{12} in y direction



(c) Response p_{13} in z direction

Figure 11. The trace responses after the learning of NN_c (after 10th trials)

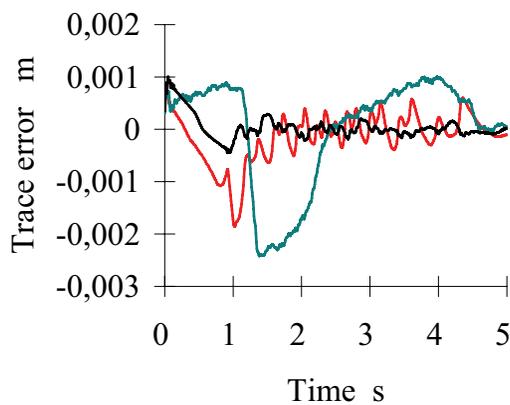
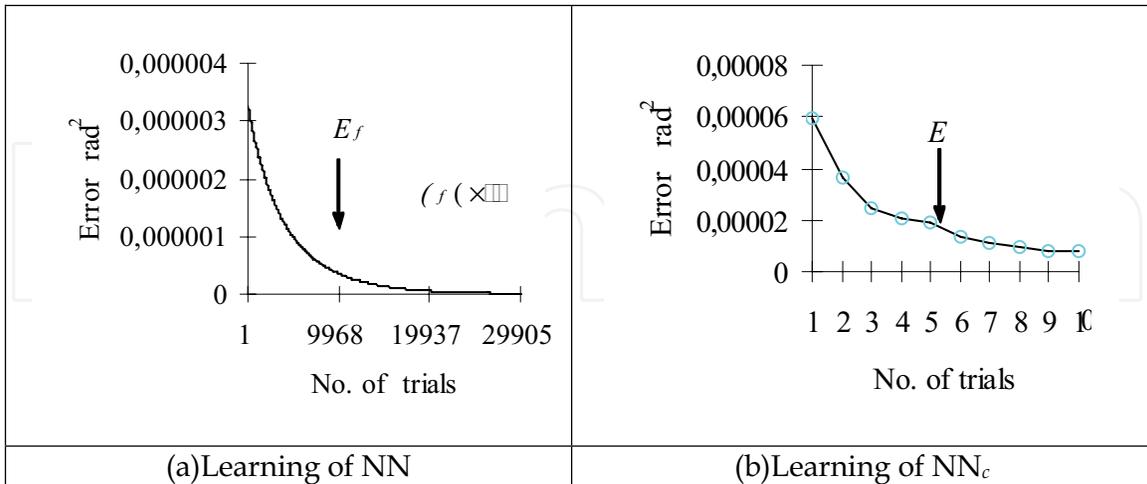
(c) Trace errors in x, y, z directions

Figure 12. Learning processes and trace errors

7. Conclusions

In this chapter, a visual feedback control approach based on neural network is presented for a robot with a camera installed on its end-effector to trace an object in an unknown environment. Firstly, the necessary conditions for mapping the image features of the object to be traced to the joint angles of the robot are derived. Secondly, a method is proposed to generate a goal trajectory of the robot by measuring the image feature parameters of the object to be traced. Thirdly, a multilayer neural network is used to learn off-line the mapping in order to produce on line the reference inputs for controlling the robot. Fourthly, a multilayer neural network-based learning controller is designed for the compensation the nonlinear robotic dynamics. Lastly, the effectiveness of

the present approach is verified by tracing a curved line using a 6 DOF industrial robot with a CCD camera installed on its end-effector.

Through the above research, the following conclusions are obtained:

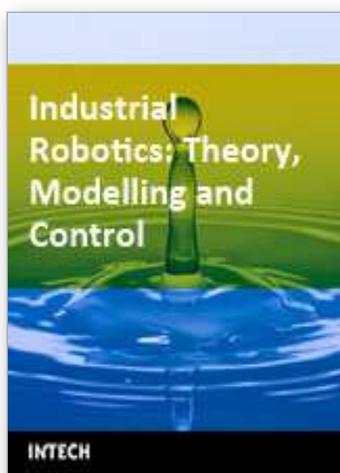
1. If the mapping relations between the image feature domain of the object and the joint angle domain of the robot are satisfied, NN can learn the mapping relations.
2. By computing the image feature parameters of the object, the goal trajectory for the end-effector to trace the object can be generated.
3. The approach does not necessitate the tedious CCD camera calibration and the complicated coordinate transformation.
4. Using the 2nd-order holder and the disturbance observer, the synchronization problem and the influences of the disturbances can be solved.

The above research is supported by the Natural Science Foundation of China (No. 60375031) and the Natural Science Foundation of Guangdong (No. 36552), the authors express their heartfelt thanks to the Foundations.

8. References

- Hosoda, K. & Asada, M., (1996). Adaptive Visual Servoing Controller with Feed-forward Compensator without Knowledge of True Jacobian, *Journal of Robot Society of Japan*, (in Japanese), Vol. 14, No. 2, pp. 313-319.
- Hosoda, K., Igarashi, K. & Asada, M., (1997). Adaptive Hybrid Visual/Force Servoing Control in Unknown Environment, *Journal of Robot Society of Japan*, (in Japanese), Vol. 15, No. 4, pp. 642-647.
- Weiss, L.E. & Sanderson, A.C., (1987). Dynamic Sensor-Based Control of Robots with Visual Feedback, *IEEE Journal of Robotics and Automation*, Vol.3, No. 5, pp. 404-417.
- Nikolaos, P. & Pradeep, K., (1999). Visual Tracking of a Moving Target by a Camera Mounted on a Robot: A Combination of Control and Vision, *IEEE Journal of Robotics and Automation*, Vol. 9, No. 1, pp. 14-35.
- Bernardino, A. & Santos-Victor J, (1999). Binocular Tracking: Integrating Perception and Control, *IEEE Journal Robotics and Automation*, Vol. 15, No. 6, pp. 1080-1093.
- Malis, E., Chaumette, F. and Boudet, S., (1999). 2-1/2-D Visual Servoing, *IEEE Journal of Robotics and Automation*, Vol. 15, No. 2, pp. 238-250.
- Hashimoto, K., Ebine, T. & Kimura, H., (1996). Visual Servoing with Hand-Eye Manipulator-Optimal Control Approach, *IEEE Journal of Robotics and Automation*, Vol. 12, No. 5, pp. 766-774.

- Wilson, J.W., Williams, H. & Bell, G.S., (1996). Relative End-Effector Control Using Cartesian Position Based Visual Servoing, *IEEE Trans. Robotics and Automation*, Vol. 12, No. 5, pp. 684-696.
- Ishikawa, J., Kosuge, K. & Furuta, K., Intelligent Control of Assembling Robot Using Vision Sensor, 13-18 May 1990, Cincinnati, OH, USA, Proceedings 1990 IEEE International Conference on Robotics and Automation (Cat. No.90CH2876-1), Vol. 3, pp. 1904-1909.
- Yamada, T. & Yabuta, T., (1991). Some Remarks on Characteristics of Direct Neuro-Controller with Regard to Adaptive Control, *Trans. Soc. Inst. Contr. Eng.*, (in Japanese), Vol. 27, No. 7, pp. 784-791.
- Verma, B., (1997). Fast Training of Multilayer Perceptrons, *IEEE Trans. Neural Networks*, Vol. 8, No. 6, pp. 1314-1319.



Industrial Robotics: Theory, Modelling and Control

Edited by Sam Cubero

ISBN 3-86611-285-8

Hard cover, 964 pages

Publisher Pro Literatur Verlag, Germany / ARS, Austria

Published online 01, December, 2006

Published in print edition December, 2006

This book covers a wide range of topics relating to advanced industrial robotics, sensors and automation technologies. Although being highly technical and complex in nature, the papers presented in this book represent some of the latest cutting edge technologies and advancements in industrial robotics technology. This book covers topics such as networking, properties of manipulators, forward and inverse robot arm kinematics, motion path-planning, machine vision and many other practical topics too numerous to list here. The authors and editor of this book wish to inspire people, especially young ones, to get involved with robotic and mechatronic engineering technology and to develop new and exciting practical applications, perhaps using the ideas and concepts presented herein.

How to reference

In order to correctly reference this scholarly work, feel free to copy and paste the following:

Xiao Nan-Feng and Saeid Nahavandi (2006). Visual Feedback Control of a Robot in an Unknown Environment (Learning Control Using Neural Networks), Industrial Robotics: Theory, Modelling and Control, Sam Cubero (Ed.), ISBN: 3-86611-285-8, InTech, Available from:

http://www.intechopen.com/books/industrial_robotics_theory_modelling_and_control/visual_feedback_control_of_a_robot_in_an_unknown_environment__learning_control_using_neural_networks

INTECH
open science | open minds

InTech Europe

University Campus STeP Ri
Slavka Krautzeka 83/A
51000 Rijeka, Croatia
Phone: +385 (51) 770 447
Fax: +385 (51) 686 166
www.intechopen.com

InTech China

Unit 405, Office Block, Hotel Equatorial Shanghai
No.65, Yan An Road (West), Shanghai, 200040, China
中国上海市延安西路65号上海国际贵都大饭店办公楼405单元
Phone: +86-21-62489820
Fax: +86-21-62489821

© 2006 The Author(s). Licensee IntechOpen. This chapter is distributed under the terms of the [Creative Commons Attribution-NonCommercial-ShareAlike-3.0 License](https://creativecommons.org/licenses/by-nc-sa/3.0/), which permits use, distribution and reproduction for non-commercial purposes, provided the original is properly cited and derivative works building on this content are distributed under the same license.

IntechOpen

IntechOpen