Robot Visual Servoing With Iterative Learning Control

P. Jiang and R. Unbehauen

Abstract—**This paper presents an iterative learning scheme for vision-guided robot trajectory tracking. At first, a stability criterion for designing iterative learning controller is proposed. It can be used for a system with initial resetting error. By using the criterion, one can convert the design problem into finding a positive definite discrete matrix kernel and a more general form of learning control can be obtained. Then, a three-dimensional (3-D) trajectory tracking system with a single static camera to realize robot movement imitation is presented based on this criterion.**

Index Terms—**Adaptive control, iterative learning control, visual servoing.**

I. INTRODUCTION

Most of the industry manipulators can be programmed by so-called "teaching by showing," which involves moving the manipulator manually with a teaching pendant to a desired position and then recording the position in a memory. The manipulator then plays back the recorded coordinates. However, for a trajectory tracking, it is difficult to move the manipulator exactly manually and then record the trajectory. Recently, for the purpose of programming a humanoid robot, several researchers investigated more general and flexible "teaching by showing" from the point of learning strategy [1], [2], where camera, laser range finder, and the marker-based optical recording equipment act as percept device to replace the time-consuming manual programming. In this paper, we use a camera instead of a teaching pendant for robot continuous trajectory imitation, especially for manipulator curve tracking in industry application. First, a teacher grasps a tool or simply an object and does a demonstration. At the same time, a static camera records the trajectory of some selected features of the object on the image plane. It describes the desired trajectory of the object. Then, let the manipulator grasp the same object and do the tracking repetitively. After several times learning, it can be expected that a perfect replay of the demonstrated trajectory can be achieved ultimately. It is a repetitive learning scheme to track the demonstrated trajectory in image plane and both learning and visual servoing techniques involve in it. From this process, the iterative learning control (ILC) [3] fits this application very well. For trajectory tracking, it improves the performance of each point along the trajectory by the experience of the previous tracking of the same trajectory. Usually, the learning operator is designed by using the discrete Lyapunov method and the system input is updated in an affine fashion such as a P type or a D type learning. One of known problem existing in the ILC is the influence of the initial error. Some researchers have studied this problem [4]–[6]. The results show that the final tracking performances are related to the initial error or depend on initial state learning.

To the visual serving, more attention was paid in the last decade. Wijesoma *et al.* [7] employed an overhead static camera to detect the position of the end-effector on the image plane, which was directly fed to the joint servo controller. Allen *et al.* [8] used two static cameras

to follow the motion of a toy train. Papanikolopoulos *et al.* [9]–[11] worked in the eye-in-hand manner, where they introduced SSD optical flow and studied LQG, self-tuning regulators for object tracking. Espiau *et al.* [12] studied visual servoing from a more general point of task function by image Jacobian matrix. The key to designing a successful visual servo control is how to estimate the image Jabobian to meet the so-called modeling condition [12]. The image Jacobian includes the information of depth, which is usually unmeasurable with only a single camera. Some approaches simply allocate the nominal one for the image Jacobian matrix. It works well if a robot is commanded to grasp a static object [12], [13]. However, it becomes worse when the robot tracks a trajectory, as shown in the experiment result of [13].

In this paper, an iterative learning controller based on a modified error with deadzone is presented. The initial width of the deadzone is designed such that the initial state is located within it, then the modified error exhibits zero-error initially. In the period of trajectory tracking, the width of the deadzone is decreased properly so that the control error is eliminated. We proposed two possible methods, quadratic and exponential, to adjust the width. Both of them lead to the final tracking performances not related to the initial error. In addition, this paper proposes a more general iterative learning form in addition to P-type or D-type learning. The learning operator is generally represented as a positive discrete matrix kernel. Based on it, we design a learning controller for the task of "teaching by showing" with visual servoing, where we use a single static camera to do 3-D trajectory tracking control. In order to calculate the image Jacobian matrix, we present an on-line pose estimation that identifies the coordinates of the feature points on the tool frame of robot's end-effector. Because these coordinates are constant for a particular task, the convergence can be guaranteed. At the end of this paper, the simulations are carried out to verify the efficacy of the proposed learning scheme.

II. STABILITY CRITERION FOR ITERATIVE LEARNING CONTROL

For a desired trajectory $x_d(t)$ with a finite time interval $0 \leq t \leq t_f$, the objective of the ILC is to design a learning law such that the system states can converge to the $x_d(t)$ by tracking it repetitively. Suppose that the ith tracking of a time-varying system can be described by the following equation:

$$\dot{x}(t,i) = f(t,i) + u(t,i) \quad i = 1 \ldots k, \quad 0 \leq t \leq t_f \qquad (1)$$

where $f(t,i)$ is an unknown vector but satisfies the repetitive condition, i.e., $f(t,i) = f(t), x(t,i) \in R^n$ and $u(t,i) \in R^n$ are the state and the control input of the ith learning at instant t, respectively.

Most of the iterative learning schemes require an initial resetting condition, i.e., $x(0,i) = x_d(0)$. However, a practicable assumption is to set the initial errors within a given region. We introduce a modified equivalent error $s_\Delta(t,i)$ to ILC with a deadzone

$$s_\Delta(t,i) = s(t,i) - \phi(t,i)$$
$$\phi(t,i) = [\varepsilon_{f1}(t) \operatorname{sat}(s_1(t,i)/\varepsilon_{f1}(t)) \ldots$$
$$\varepsilon_{fn}(t) \operatorname{sat}(s_n(t,i)/\varepsilon_{fn}(t))]^T \qquad (2)$$

where $s(t,i)$ is a vector of an equivalent error, $s_j(t,i)$ is its jth component, and the $\varepsilon_f(t) = [\varepsilon_{f1}(t), \ldots, \varepsilon_{fn}(t)]^T$ is an n-dimensional width of the deadzone that is time-varying but independent of the iterative index i.

If we specify the deadzone width $\varepsilon_f(0)$ such that the initial resetting errors of the system are always within it, then $s_\Delta(0,i) = 0$ and the usual assumption of a vanishing initial error is satisfied.

Theorem 1: If a system satisfies a weak initial resetting condition of $|s_j(0, i)| < \varepsilon_{fj}(0)$, then an upper-bounded condition $\lim_{k \to \infty} \sum_{i=0}^{k} s_\Delta^T(t, i)(\dot{s}(t, i) - \mathrm{sgn}(s_\Delta)\dot{\varepsilon}_f) \leq \gamma_0^2, 0 \leq t \leq t_f$, where γ_0^2 is any positive finite constant, ensures that the equivalent error of every point along the desired trajectory will converge into the deadzone, i.e., $\lim_{i \to \infty} |s_j(t, i)| \leq \varepsilon_{fj}(t), 0 \leq t \leq t_f$, as the system tracks the desired trajectory repetitively.

Proof: See Appendix 1.   □

On the basis of this theorem, we shall design an iterative learning controller for the system (1) and mainly focus our attention on two problems: 1) Except usual P-type and D-type iterative learning, what is the general learning scheme? 2) How to design the deadzone width such that a perfect tracking can be achieved.

Suppose the tracking error of system (1) is $s(t, i) = x_d(t) - x(t, i)$. From the repetitive condition $f(t, i) = f(t)$, we obtain

$$\sum_{i=0}^{k} s_\Delta^T(t, i)(\dot{s}(t, i) - \mathrm{sgn}(s_\Delta)\dot{\varepsilon}_f)$$
$$= \sum_{i=0}^{k} s_\Delta^T(t, i)(\dot{x}_d(t) - f(t) - \mathrm{sgn}(s_\Delta)\dot{\varepsilon}_f - u(t, i)). \quad (3)$$

Let the control law be

$$u(t, i) = u_l(t, i) + u_f(t, i) \quad (4)$$

where $u_l(t, i)$ is a learning term for compensating the unknown but repetitive influence of $\dot{x}_d(t) - f(t)$, and $u_f(t, i)$ is a feedback term for countermining the time varying width of the deadzone.

Substituting control law (4) into (3), we have

$$\sum_{i=0}^{k} s_\Delta^T(t, i)(\dot{s}(t, i) - \mathrm{sgn}(s_\Delta)\dot{\varepsilon}_f)$$
$$= \sum_{i=0}^{k} s_\Delta^T(t, i)(\dot{x}_d(t) - f(t) - u_l(t, i))$$
$$+ \sum_{i=0}^{k} s_\Delta^T(t, i)(-\mathrm{sgn}(s_\Delta)\dot{\varepsilon}_f - u_f(t, i)). \quad (5)$$

In order to satisfy the upper bounded condition in Theorem 1, the following property of a positive definite discrete matrix kernel [14] can provide a more general learning form.

Lemma: For any vector $y(i)$ and any unknown constant vector a, a positive definite discrete matrix kernel $F(i)$, whose z-transform is a positive real discrete transfer matrix and with a pole at $z = 1$, ensures that the following accumulation is always upper bounded

$$\sum_{i=0}^{N} y^T(i)\left(a - \sum_{j=0}^{i} F(i - j)y(j)\right) \leq \gamma_0^2, \quad \forall N.$$

In the first term of (5), although $\dot{x}_d(t) - f(t)$ is time-varying, it remains constant over iterations for any given instant t, i.e., it can be repeated. From the Lemma, if we use a learning law of the uncertainties $\dot{x}_d(t) - f(t)$ through iterations i as

$$u_l(t, i) = \sum_{j=0}^{i} F(i - j)s_\Delta(t, j) \quad (6)$$

where $F(i - j)$ is a positive definite discrete matrix kernel, then $\sum_{i=0}^{k} s_\Delta^T(t, i)(\dot{x}_d(t) - f(t) - u_l(t, i)) \leq \gamma_0^2$ for any given t and

$$\sum_{i=0}^{k} s_\Delta^T(t, i)(\dot{s}(t, i) - \mathrm{sgn}(s_\Delta)\dot{\varepsilon}_f)$$
$$\leq \gamma_0^2 + \sum_{i=0}^{k} s_\Delta^T(t, i)(-\mathrm{sgn}(s_\Delta)\dot{\varepsilon}_f - u_f(t, i)). \quad (7)$$

Further, let the feedback term be

$$u_f(t, i) = -\mathrm{sgn}(s_\Delta)\dot{\varepsilon}_f \quad (8)$$

then $\lim_{k \to \infty} \sum_{i=0}^{k} s_\Delta^T(t, i)(\dot{s}(t, i) - \mathrm{sgn}(s_\Delta)\dot{\varepsilon}_f) \leq \gamma_0^2, 0 \leq t \leq t_f$, the condition in Theorem 1 is satisfied.

Take the z transformation along the iteration axis i for the (6), and we get

$$u_l(t, z) = F(z)s_\Delta(t, z). \quad (9)$$

From the definition of a positive definite discrete matrix kernel that has a pole at $z = 1$, we can rewrite $F(z) = W(z)z/(z - 1)$, and (9) becomes

$$u_l(t, z) = z^{-1}u_l(t, z) + W(z)s_\Delta(t, z).$$

By its inverse z-transform, the learning law (6) can be written in an equivalent form

$$u_l(t, i) = u_l(t, i - 1) + Z^{-1}(W(z) \cdot s_\Delta(t, z)). \quad (10)$$

Obviously, the usual P-type law with a gain of a positive constant diagonal matrix W is a particular one of (10) because $Z(F(i)) = W \cdot \mathrm{diag}(z/(z - 1), \ldots, z/(z - 1))$ is a positive real discrete transfer matrix with a pole at $z = 1$. Therefore, (6) or (10) represents a more general learning law. It updates the control input based on the output of the error passed through a filter $W(z)$ if $F(z) = W(z)z/(z - 1)$ is a positive real discrete transfer function. This kind of filter can guarantee convergence of the learning and improve its robustness to the noise. For example, a first order low-pass filter $1/(Ts + 1)$ is often used in application, where T is the time constant of the filter and s is the Laplacian operator. For the purpose of discretization, its conversion to digital form can be written by its z-transform with a zero hold

$$G(z) = (1 - z^{-1})Z\left(\frac{1}{s(Ts + 1)}\right) = \frac{1 - e^{-\tau/T}}{z - e^{-\tau/T}}$$
$$= \frac{(1 - a)}{z - a} \quad (11)$$

where $a = e^{-\tau/T}$ is a constant satisfying $0 < a < 1$.

If we take this filter as the learning gain in (10), i.e., $W(z) = G(z)$, the learning law can be written as

$$\begin{cases} W(t, i) = aW(t, i - 1) + (1 - a)s_\Delta(t, i - 1) \\ u(t, i) = u(t, i - 1) + W(t, i) \end{cases} \quad (12)$$

In this equation, it only uses the tracking errors of last tracking, i.e., $s_\Delta(t, i - 1)$ due to the zero hold to avoid noncausal problem in normal digital control system. However, in ILC, the learning is along discrete iteration axis but control is conducted along continuous time axis. Then the ith control error $s_\Delta(t, i)$ is available for the ith learning control. Using $s_\Delta(t, i)$ instead of $s_\Delta(t, i - 1)$, the (12) becomes

$$\begin{cases} W(t, i) = aW(t, i - 1) + (1 - a)s_\Delta(t, i) \\ u(t, i) = u(t, i - 1) + W(t, i) \end{cases} . \quad (13)$$

With this low-pass filter, we have $W(z) = ((1 - a)z)/(z - a), 0 < a < 1$. Now, we shall verify if $F(z) = W(z)z/(z - 1)$ is a positive real discrete transfer matrix, which is the convergent condition of the learning control (6)

$$F(z) = W(z) \cdot \frac{z}{z - 1} = \frac{(1 - a)z^2}{(z - 1)(z - a)}.$$

1) $F(z)$ is analytic outside the unit circle. The pole on the unit circle is simple and the associated residue is positive.

2) Applying the transformation $z = (1 + j\omega)/(1 - j\omega)$ to $F(z)$, its real part is given by

$$\mathrm{Re}[F(j\omega)] = (1 - a)\frac{(2 + 2a)\omega^4 + (2 - 6a)\omega^2}{(2 + 2a)^2\omega^4 + (2 - 2a)^2\omega^2}.$$

In order to let $\mathrm{Re}[F(j\omega)] > 0$ for all real ω, we have the condition of $a < 1/3$.

So the coefficient of the filter $W(z)$ in (13) should satisfy $0 < a < 1/3$, and then the convergence of the learning control can be ensured.

Remark 1: The learning law (6) or (13) consist of the accumulation of control history $(0 \ldots i - 1)$ and the ith control error feedback. The stability can be guaranteed if $F(z) = W(z)z/(z - 1)$ is a positive real discrete transfer matrix.

Remark 2: In addition to the system (1), for a class of nonlinear system

$$\dot{x}(t, i) = f(t, i) + Y(x) * a(t, i) + u(t, i),$$
$$i = 1 \ldots k, \quad 0 \leq t \leq t_f \quad (14)$$

where $Y(x) \in R^{n \times m}$ is a known nonlinear matrix, $f(t, i) \in R^n$ and $a(t, i) \in R^m$ are unknown vectors but have the repetitive property, i.e., $f(t, i) = f(t)$ and $a(t, i) = a(t)$, we can also obtain the following learning law to meet the upper bounded condition of Theorem 1:

$$\begin{cases} u_l(t, i) = \hat{f}(t, i) + Y(x)\hat{a}(t, i) \\ \hat{f}(t, i) = \hat{f}(t, i - 1) + Z^{-1}(W_1(z)s_\Delta(t, z)) \\ \hat{a}(t, i) = \hat{a}(t, i - 1) + Z^{-1}(W_2(z)Y^T(x)s_\Delta(t, z)) \end{cases} \quad (15)$$

if $F_k(z) = W_k(z)z/(z - 1), k = 1, 2$, are positive real discrete transfer matrices.

From Theorem 1, we have $\lim_{i \to \infty} |s_j(t, i)| \leq \varepsilon_{fj}(t), 0 \leq t \leq t_f$. It only implies that the tracking error converges into the deadzone. In order to realize high accuracy tracking, we have to limit the width of the deadzone as narrow as possible. As mentioned previously, the initial width of the deadzone is a constant that must be greater than any possible uncertain initial state error, namely, $|s_j(0, i)| < \varepsilon_{fj}(0), \forall i$. Then the width $\varepsilon_f(t)$ should be decreased along the desired trajectory so that $\varepsilon_f(t) = 0$ after a specific instant T and perfect tracking can then be achieved after this instant. From the feedback law (8), this time varying deadzone width $\varepsilon_f(t)$ has to be differentiable. A lot of mathematical functions can be used to realize this attenuating width of deadzone. One of possible and simpler candidate is the quadratic function.

1) Quadratic Deadzone: The width of the deadzone for the $s_j(t, i)$ is decreased quadratically as

$$\varepsilon_{fj}(t) = \begin{cases} \varepsilon_{fj}(0)(t - T)^2/T^2, & t \leq T \\ 0, & t > T \end{cases}. \quad (16)$$

It is differentiable and the width of the deadzone reaches zero after time T. This means $\lim_{i \to \infty} |x_d(t) - x(t, i)| = 0, T \leq t \leq t_f$, and a perfect tracking can be realized. In application, for any desired trajectory, we can design a 'run up segment' from time 0 to T before normal tracking. It is only an assistant segment for the desired trajectory.

With this strategy, the feedback control term in (8) can be written as

$$u_{fj}(t, i) = \begin{cases} 2\varepsilon_{fj}(0)(T - t)\,\mathrm{sgn}(s_{\Delta j}(t, i))/T^2, & t \leq T \\ 0, & t > T \end{cases}. \quad (17)$$

This feedback term is discontinuous in the "run up segment." In normal tracking, it is removed. In order to realize a smooth feedback control and remove the additional "run up segment," an exponentially attenuating width can be considered as an alternative.

2) Exponential Deadzone: Considering the second term in (5), we have

$$\sum_{i=0}^{k}\left\{ s_\Delta^T(t, i)[-\mathrm{sgn}(s_\Delta)\dot{\varepsilon}_f - u_f(t, i)] \right\}$$
$$= \sum_{i=0}^{k}\left[-|s_\Delta(t, i)|^T\dot{\varepsilon}_f - s_\Delta^T(t, i)u_f(t, i) \right] \quad (18)$$

where $|s_\Delta(t, i)| = [|s_{\Delta 1}(t, i)|, \ldots, |s_{\Delta n}(t, i)|]^T$.

Let

$$u_f(t, i) = K_d s(t, i),$$
$$\text{where } K_d = \mathrm{diag}(K_{d1}, \ldots, K_{dn}) > 0 \quad (19).$$

Substituting (19) into (18) and based on (2), we have

$$\sum_{i=0}^{k}\left\{ s_\Delta^T(t, i)[-\mathrm{sgn}(s_\Delta)\dot{\varepsilon}_f - u_f(t, i)] \right\}$$
$$= \sum_{i=0}^{k}\left[-s_\Delta^T(t, i)K_d s_\Delta(t, i) - |s_\Delta(t, i)|^T\dot{\varepsilon}_f \right.$$
$$\left. - s_\Delta^T(t, i)K_d\phi(t, i) \right]$$
$$= \sum_{i=0}^{k}\left[-s_\Delta^T(t, i)K_d s_\Delta(t, i) - |s_\Delta(t, i)|^T\dot{\varepsilon}_f \right.$$
$$\left. - |s_\Delta(t, i)|^T K_d\varepsilon_f \right] \quad (20)$$

Therefore, when the width of the deadzone satisfies

$$\dot{\varepsilon}_f = -K_d\varepsilon_f, \quad \text{with the initial condition of } \varepsilon_f(0) \quad (21)$$

the (20) is less than or equal to zero and the inequality in Theorem 1 can be satisfied too. Then $\lim_{i \to \infty} |s_j(t, i)| \leq \varepsilon_{fj}(t), 0 \leq t \leq t_f$. It implies that the system tracking performances can be better than any given first-order linear system (21) if we use a linear feedback control (19).

III. TEACHING BY SHOWING WITH ITERATIVE LEARNING

In this section, a teaching by showing algorithm for continuous trajectory programming is proposed, where a static camera is used as trajectory recording device instead of teaching pendant. First, a human teacher grasps an object or a tool and does a movement demonstration as shown in Fig. 1, where a cube is grasped to move. At the same time, a static camera records the trajectories $p_d(t) = [x_{d1}(t), y_{d1}(t), x_{d2}(t), y_{d2}(t), x_{d3}(t), y_{d3}(t)]^T$ of three selected feature points of the object on the image plane, which are three corner points of the cube $p_1(t), p_2(t)$, and $p_3(t)$, in Fig. 1. Then, let a manipulator grasp the same object to imitate the movement of the teacher.

Let us define a moving tool frame F_t attached to the end-effector of the manipulator and a base frame F_c attached to the static camera as shown in Fig. 2. The vector that locates F_t's origin is called $^{F_c}P_{F_to}$. F_t is rotated with respect to F_c by $^{F_c}_{F_t}R$. Then, any selected feature point $P(X_s, Y_s, Z_s)$ of an object can be described in the frame of F_c as

$$P = {}^{F_c}_{F_t}R \cdot {}^{F_t}P + {}^{F_c}P_{F_to} \quad (22)$$

where ^{F_t}P is coordinate vector of the point P in the tool frame F_t. Note that it is an unknown constant vector for operating a rigid object, which represents the fixed relationship between the object and the end-effector for a given operation.
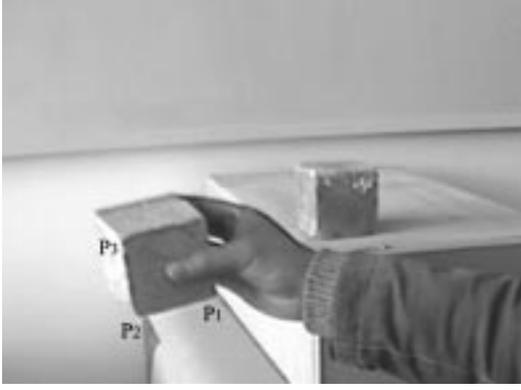
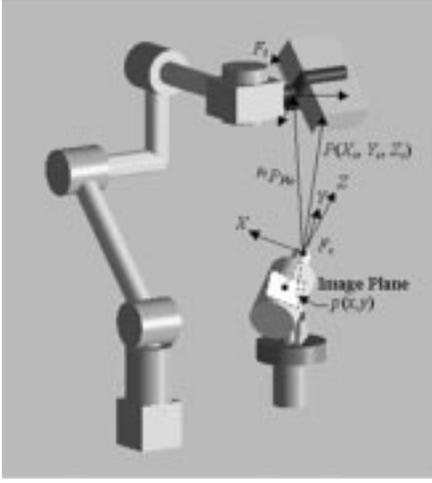Fig. 1.　Teacher's demonstration and feature points.

Fig. 2.　Frames definition.

Let us assume that the end-effector moves with a translational velocity $V = (V_x, V_y, V_z)^T$ and with an angular velocity $\omega = (\omega_x, \omega_y, \omega_z)^T$. Differentiating both side of (22) gives

$$\dot{P} = \omega \times \left({}^{F_c}_{F_t}R \cdot {}^{F_t}P \right) + V = \omega \times \left(P - {}^{F_c}P_{F_t o} \right) + V$$
$$= \omega \times P + V'. \tag{23}$$

Define $T' = \Theta T$, where $T' = [V', \omega]^T, T = [V, \omega]^T$

$$\Theta = \begin{bmatrix} I & [{}^{F_c}P_{F_t o} \times] \\ 0 & I \end{bmatrix}.$$

Now suppose the projection of the feature point $P(X_s, Y_s, Z_s)$ onto the image is $p(x, y)$ and satisfies the pinhole camera model with a unit focal length as shown in Fig. 2. Then the optical flow of the point $p(x, y)$ is written as [12]

$$\dot{p} = JT' = J\Theta T,$$
$$J = \left[\frac{1}{Z_s} \begin{bmatrix} 1 & 0 & -x \\ 0 & 1 & -y \end{bmatrix}, \begin{bmatrix} -xy & 1+x^2 & -y \\ -1-y^2 & xy & x \end{bmatrix} \right]. \tag{24}$$

It is known that, in order to compute the control input, at least three noncollinear feature points are required [10]. Suppose three feature points of the object are represented as $P_i, i = 1, 2, 3$.

How to obtain the image Jacobian matrix J of 3 feature points as defined in (24), where $J = [J_1^T, J_2^T, J_3^T]^T \in R^{6 \times 6}$, is the key of
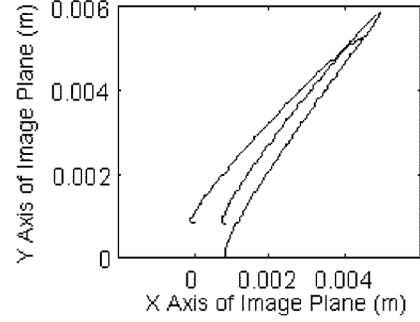

Fig. 3.　Desired trajectories on the image plane.

designing a visual servoing controller, which involves unknown depths of the feature points with only a single camera. From (22), we know that the depth of the jth feature point can be expressed as

$$Z_{sj} = P_{jz} = {}^{F_c}_{F_t}R_z \cdot {}^{F_t}P_j + Z_o \tag{25}$$

where $[\cdot]_z$ corresponds to the third row of a matrix, $Z_o = \lfloor {}^{F_c}P_{F_t o} \rfloor_z$ and ${}^{F_c}_{F_t}R_z$ are measurable from the robot position sensors, ${}^{F_t}P_j$ is an unknown constant vector that is a repetitive uncertainty and can be learned by ILC. Then, we propose the following control for the system (24) to follow the desired trajectory $p_d(t)$:

$$T(t, i) = \Theta^{-1}T'(t, i) = \Theta^{-1}\hat{J}^{-1}(u_l(t, i) + u_f(t, i)) \tag{26}$$

where $\hat{J}$ is an estimate of Jacobian matrix with estimated depths $\hat{Z}_{sj}, j = 1, 2, 3$, instead of the real one in (24) and is supposed to be not singular. The $u_l(t, i)$ is the learning term and the $u_f(t, i)$ is the feedback term.

Theorem 2: When select $u_l(t, i)$ as in (10) and use the following depth estimation for computation of $\hat{J}$:

$$\begin{cases} 1/\hat{Z}_{sj} = 1/Z_o - 1 \big/ Z_o^2 {}^{F_c}_{F_t}R_z \cdot {}^{F_t}\hat{P}_j(t, i) - |\delta|_{\max}\,\mathrm{sgn}(g_j) \\ {}^{F_t}\hat{P}_j(t, i) = \sum_{l=0}^i G_j(i-l)g_j(t, l) {}^{F_c}_{F_t}R_z^T \big/ Z_o^2 \end{cases}$$

where $G_j(i-l)$, $j = 1, 2, 3$, should be positive definite discrete matrix kernels. By iterative learning, the system can realize perfect tracking with $u_f(t, i)$ in a form of (17) or exhibit exponential decrease with $u_f(t, i)$ in a form of (19).

Proof: See Appendix 2.　　□

In order to calculate (26), the invertibility of $\hat{J}$ must be guaranteed. This means that the three estimate points must be not collinear and the camera center is not located on the singular cylinder [15]. In order to avoid the singularity, we define three cubic regions for every estimate of feature points such that they are impossible to become collinear if the estimates of X, Y, Z coordinates of ${}^{F_t}\hat{P}_j, j = 1, 2, 3$, are confined inside the regions, namely, $\lfloor {}^{F_t}P_{jm} \rfloor_{\max} > {}^{F_t}\hat{P}_{jm}(t, i) > \lfloor {}^{F_t}P_{jm} \rfloor_{\min}, m = 1, 2, 3$. Then, we modify the identification scheme in Theorem 2 with a bounded learning scheme, as shown in (27), shown at the bottom of the next page, where $m = 1, 2, 3$ and $d_j = [d_{j1}\ d_{j2}\ d_{j3}] = g_j {}^{F_c}_{F_t}R_z/Z_o^2$.

Then, (A.3) in Appendix 2 is upper bounded as well.

IV. SIMULATION RESULTS

In the showing phase, a teacher grasps a cube to do demonstration as shown in Fig. 1 and the trajectories of three corner points in 5 s are recorded by a camera (16 mm focal length) as shown in Fig. 3.

(a) The 1st learning   (b) The second learning

_ex, . . . ey, - - - ez
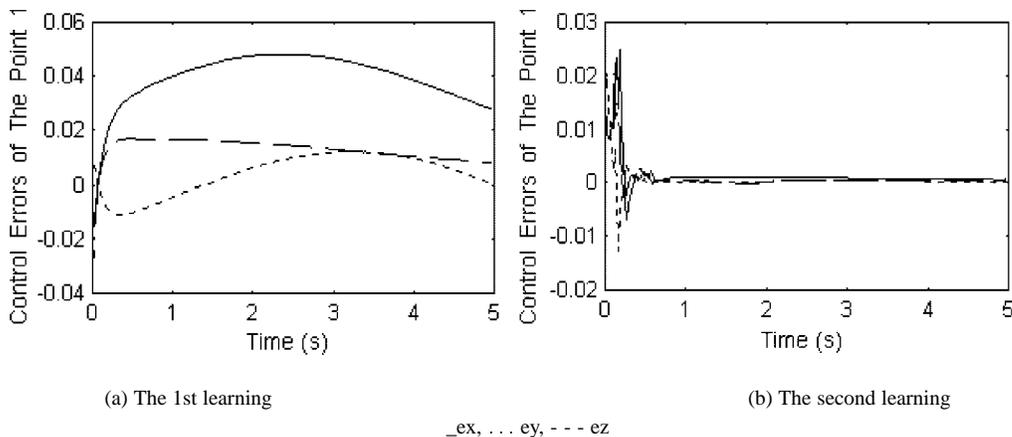
Fig. 4.   Control errors of the feature point 1 $(\Delta = \pm 0.05)$.

Then, let the manipulator grasp the same cube to learn the demonstrated trajectory repetitively.

During the training, suppose the origin of the tool frame can not coincide with the desired one with a ± 0.05 m random initial error for each tracking. It causes the initial resetting errors of the iterative learning. In order to use bounded scheme (27) to estimate the coordinates of the feature points in the tool-frame, a rough position estimation of the feature points in tool frame is needed

$$
^{Ft}P_1 = [0.1 \quad 0.1 \quad 0]^T, \quad ^{Ft}P_2 = [0.1 \quad 0 \quad 0.1]^T,
$$
$$
^{Ft}P_3 = [0 \quad 0.1 \quad 0.1]^T \tag{28}
$$

with an possible error bound of $\Delta = \pm 0.05$ for all of components, i.e., $\lfloor ^{Ft}P_{jm} \rfloor_{\max} = {}^{Ft}P_{jm} + \Delta$, $\lfloor ^{Ft}P_{jm} \rfloor_{\min} = {}^{Ft}P_{jm} - \Delta$ required in (27), shown at the bottom of the page, where $j, m = 1, 2, 3$.

The servo period is 0.02 s and the control parameters are selected to be

$$
K_d = \mathrm{diag}[10 \quad 10 \quad 10 \quad 10 \quad 10 \quad 10]^T
$$
$$
F(i-j) = \mathrm{diag}[10 \quad 10 \quad 10 \quad 10 \quad 10 \quad 10]^T
$$
$$
|\delta|_{\max} = [0.001 \quad 0.001 \quad 0.001]^T
$$
$$
G_{jm}(i-l) = 50
$$

The first tracking errors of the feature point 1 in the Cartesian space are depicted in Fig. 4(a). After FIVE iterations, the simulation results are shown in Fig. 4(b), and the control error has shown an exponential decrease for the whole trajectory. This simulation shows that the proposed control scheme is effective in eliminating the influence of initial errors. In (27), for the purpose of estimating $1/Z_{sj}$, the initial value $^{Ft}\hat{P}_{jm}(t, 0)$ cannot be selected arbitrarily. It must be maintained within the permitted region $(\Delta = \pm 0.05$ in this simulation), otherwise it may be confronted with a singular problem. One of example can be shown in Fig. 5. In this example, $\Delta = \pm 0.2$. It means we have less prior knowledge about the coordinates of the feature points in the tool frame. In this case, three estimating feature points might become
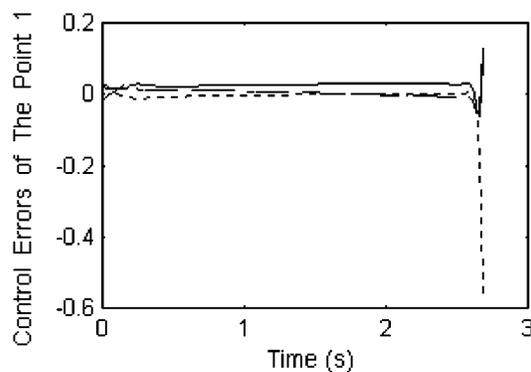
Fig. 5.   Control errors of the feature point 1 $(\Delta = \pm 0.2)$.

collinear so that the estimate of Jacobian matrix might become singular. At 2.66 s, when three feature points close to collinear, the determinant of the image Jacobian matrix is less than 5×10^{-8} and the control law faces the singular problem and further causes system divergence. How to avoid singularity for ILC without *a priori* knowledge is important for its visual servoing application. We proposed a possible solution to the problem called Indirect Iterative Lerning Control [16]. Another problem is how to avoid big deviation from the reference trajectory, especially in the first several times of learning, where the feature points might get out of camera sight and the feedback is lost. We also proposed a segmented training ILC to avoid too much deviation and keep feature point within a permitted region [17].

V. CONCLUSION

This paper presented a general iterative learning scheme for nonlinear system trajectory tracking with initial resetting error. It can be used for robot programming with "teaching by showing." Our method is different from the traditional one through a "teaching pendant." A camera is used to record the trajectory shown by a teacher. With the aid

$$
^{Ft}\hat{P}_{jm}(t, i) = \begin{cases} [^{Ft}P_{jm}]_{\max} & ^{Ft}\hat{P}_{jm} > [^{Ft}P_{jm}]_{\max} \quad \text{and} \quad d_{jm} > 0 \\ [^{Ft}P_{jm}]_{\min} & ^{Ft}\hat{P}_{jm} < [^{Ft}P_{jm}]_{\min} \quad \text{and} \quad d_{jm} < 0 \\ \sum_{l=0}^{i} G_{jm}(i-l)d_{jm} & \text{else} \end{cases} \tag{27}
$$

of this camera, the robot can track the desired trajectory after several times learning. How to avoid singularity of image Jacobian without any *a priori* knowledge is a further research topic.

APPENDIX I

Proof of Theorem 1: Define an energy function of the ith learning as $V_i(t) = s_\Delta^T(t,i)s_\Delta(t,i)$. From definition (2)

$$V_i(t) = \sum_{j=1}^{n} s_{\Delta j}^2(t,i), \quad \text{where } s_{\Delta j}^2(t,i)$$

$$= \begin{cases} (s_j - \varepsilon_{fj})^2 & s_j > \varepsilon_{fj} \\ 0 & |s_j| \le \varepsilon_{fj} \\ (s_j + \varepsilon_{fj})^2 & s_j < -\varepsilon_{fj} \end{cases}$$

and the $s_{\Delta j}^2(t,i)$ is differentiable if $\varepsilon_{fj}(t)$ is differentiable.

Then we have

$$\frac{d}{dt}\left(s_{\Delta j}^2(t,i)\right) = 2s_{\Delta j}(t,i)(\dot{s}_j - \mathrm{sgn}(s_j)\dot{\varepsilon}_{fj}).$$

Define $\mathrm{sgn}(s_\Delta) = \mathrm{diag}[\mathrm{sgn}(s_{\Delta 1}),\ldots,\mathrm{sgn}(s_{\Delta n})]$, then

$$\dot{V}_i(t) = 2s_\Delta^T(t,i)(\dot{s}(t,i) - \mathrm{sgn}(s_\Delta)\dot{\varepsilon}_f)$$

and $V_i(t) = \int_0^t \dot{V}_i(t)\,dt + s_\Delta^T(0,i)s_\Delta(0,i)\ 0 \le t \le t_f$.

From the condition $|s_j(0,i)| < \varepsilon_{fj}(0)$ and (2), we have $V_i(t) = \int_0^t \dot{V}_i(t)\,dt$.

Then the total energy of k iterations is

$$\sum_{i=0}^{k} V_i(t) = \sum_{i=0}^{k} \int_0^t \dot{V}_i(t)\,dt$$

$$= 2\int_0^t \sum_{i=0}^{k} s_\Delta^T(t,i)(\dot{s}(t,i) - \mathrm{sgn}(s_\Delta)\dot{\varepsilon}_f)\,dt.$$

When the repetitive times go into infinite, the energy becomes

$$\lim_{k\to\infty} \sum_{i=0}^{k} V_i(t)$$

$$= 2\int_0^t \left[\lim_{k\to\infty} \sum_{i=0}^{k} s_\Delta^T(t,i)(\dot{s}(t,i) - \mathrm{sgn}(s_\Delta)\dot{\varepsilon}_f)\right]dt.$$

If $\lim_{k\to\infty} \sum_{i=0}^{k} s_\Delta^T(t,i)(\dot{s}(t,i) - \mathrm{sgn}(s_\Delta)\dot{\varepsilon}_f) \le \gamma_0^2, 0 \le t \le t_f$, then $\lim_{k\to\infty} \sum_{i=0}^{k} V_i(t) \le 2\int_0^t \gamma_0^2\,dt \le 2\gamma_0^2 t_f$.

Because the time interval t_f is bounded, then $\lim_{k\to\infty} \sum_{i=0}^{k} V_i(t)$ is bounded. We have $\lim_{i\to\infty} V_i(t) = 0$, which means $\lim_{i\to\infty} s_\Delta(t,i) = 0$, for $0 \le t \le t_f$. □

APPENDIX II

Proof of Theorem 2: Let the ith equivalent error be $s(t,i) = [s_1^T(t,i), s_2^T(t,i), s_3^T(t,i)]^T = p_d(t) - p(t,i) \in R^6$, then from (24)

$$\sum_{i=0}^{k} s_\Delta^T(t,i)(\dot{s}(t,i) - \mathrm{sgn}(s_\Delta)\dot{\varepsilon}_f)$$

$$= \sum_{i=0}^{k} s_\Delta^T(t,i)(\ddot{p}_d(t) - \mathrm{sgn}(s_\Delta)\dot{\varepsilon}_f - \hat{J}T'(t,i))$$

$$+ \sum_{i=0}^{k} s_\Delta^T(t,i)(\hat{J} - J)T'(t,i).$$

Substitute the control law (26) into this equation and let $u_l(t,i)$ as in (10) and $u_f(t,i)$ as in (17) or (19), the first term of the above equation is

upper bounded. Then, from the definition of the image Jacobian matrix in (24), we have

$$\sum_{i=0}^{k} s_\Delta^T(t,i)(\dot{s}(t,i) - \mathrm{sgn}(s_\Delta)\dot{\varepsilon}_f)$$

$$\le \gamma_0^2 + \sum_{i=0}^{k} s_\Delta^T(t,i)(\hat{J} - J)T'(t,i)$$

$$= \gamma_0^2 + \sum_{i=0}^{k}\sum_{j=1}^{3} g_j(t,i)(1/\hat{Z}_{sj} - 1/Z_{sj}) \qquad (A.1)$$

where

$$g_j(t,i) = s_{\Delta j}^T(t,i)\begin{bmatrix} V_x' - x_j V_z' \\ V_y' - y_j V_z' \end{bmatrix}.$$

From (25), we can approximate $1/Z_{sj}$ by the Taylor series around F_t's origin as

$$1/Z_{sj} = 1/Z_o - 1/Z_o^2\,{}^{F_c}_{F_t}R_z \cdot {}^{F_t}P_j + \delta \qquad (A.2)$$

where δ is a small value for a task with a small size/distance ratio.

Substituting (A.2) and the depth estimating law of the Theorem 2 into (A.1) gives

$$\sum_{i=0}^{k} s_\Delta^T(t,i)(\dot{s}(t,i) - \mathrm{sgn}(s_\Delta)\dot{\varepsilon}_f)$$

$$\le \gamma_0^2 + \sum_{i=0}^{k}\sum_{j=1}^{3} g_j(t,i)\left(\frac{1}{Z_o^2}{}^{F_c}_{F_t}R_z\right.$$

$$\times \left({}^{F_t}P_j - {}^{F_t}\hat{P}_j(t,i)\right) - \delta - |\delta|_{\max}\,\mathrm{sgn}(g_j)\Big)$$

$$\le \gamma_0^2 + \sum_{j=1}^{3}\sum_{i=0}^{k} g_j(t,i){}^{F_c}_{F_t}R_z$$

$$\times \frac{1}{Z_o^2}\left({}^{F_t}P_j - \sum_{l=0}^{i} G_j(i-l)g_j(t,l){}^{F_c}_{F_t}R_z^T\frac{1}{Z_o^2}\right). \quad (A.3)$$

From the Lemma, it satisfies the upper bounded condition in Theorem 1. □

REFERENCES

[1] H. Miyamoto and M. Kawato, "A tennis serve and upswing learning robot based on bi-directional theory," *Neural Networks*, vol. 11, pp. 1331–1344, 1998.

[2] S. Schaal, "Is imitation learning the route to humanoid robots?," *Trends Cogn. Sci.*, vol. 3, no. 6, pp. 233–242, 1999.

[3] S. Arimoto, S. Kawamura, and F. Miyazaki, "Bettering operation of robots by learning," *J. Robot. Syst.*, vol. 1, no. 2, pp. 123–140, 1984.

[4] G. Heinzinger, D. Fenwick, B. Paden, and F. Miyazaki, "Stability of learning control with disturbances and uncertain initial condition," *IEEE Trans. Automat. Contr.*, vol. 37, pp. 110–114, Jan. 1992.

[5] S. S. Saab, "On the P-type learning control," *IEEE Trans. Automat. Contr.*, vol. 39, pp. 2298–2302, Nov. 1994.

[6] Y. Chen, C. Wen, Z. Gong, and M. Sun, "An iterative learning controller with initial state learning," *IEEE Trans. Automat. Contr.*, vol. 44, pp. 371–376, Feb. 1999.

[7] S. W. Wijiesoma, D. F. H. Wolfe, and R. J. Richards, "Eye-to-hand co-ordination for vision-guided robot control applications," *Int. J. Robot. Res.*, vol. 12, no. 1, pp. 64–78, 1993.

[8] P. K. Allen, A. Timcenko, B. Yoshimi, and P. Michelman, "Automated tracking and grasping of a moving object with a robotic hand-eye system," *IEEE Trans. Robot. Automat.*, vol. 9, pp. 152–165, Feb. 1993.

[9] N. P. Papanikolopoulos, P. K. Khosla, and T. Kanade, "Visual tracking of a moving target by camera mounted on a robot: A combination of control and vision," *IEEE Trans. Robot. Automat.*, vol. 9, pp. 14–35, Jan. 1993.

[10] N. P. Papanikolopoulos and P. K. Khosla, "Adaptive robot visual tracking theory and experiments," *IEEE Trans. Automat. Contr.*, vol. 38, pp. 429–445, Mar. 1993.

[11] N. P. Papanikolopoulos, B. J. Nelson, and P. K. Khosla, "Six degree of freedom hand/eye visual tracking with uncertain parameters," *IEEE Trans. Robot. Automat.*, vol. 11, pp. 725–732, May 1995.

[12] B. Espiau, F. Francois, and P. Rives, "A new approach to visual servoing in robotics," *IEEE Trans. Robot. Automat.*, vol. 8, pp. 313–326, Mar. 1992.

[13] R. Horaud, F. Dornaika, and B. Espiau, "Visually guided object grasping," *IEEE Trans. Robot. Automat.*, vol. 14, pp. 525–532, Apr. 1998.

[14] Y. D. Landau, *Adaptive Control—The Model Reference Approach*. New York: Marcel–Dekker, 1979.

[15] H. Michel and P. Rives, "Singularities in the Determination of the Situation of a Robot Effector From the Perspective View of 3 Points," INRIA, Sophia-Antipolis, France, Res. Rep. RR-1805, Feb. 1993.

[16] P. Jiang, R. Unbehauen, and P. Y. Woo, "Singularity-free indirect iterative learning control," *Proc. 40th IEEE Conf. Decision and Control*, pp. 4903–4908, Dec. 2001.

[17] P. Jiang and Y. Q. Chen, "Repetitive robot visual servoing via segmented training neural network controller," *Proc. IEEE Int. Symp. Computational Intelligence in Robotics and Automation*, pp. 260–265, July 2001.