A Repetitive Learning Method Based on Sliding Mode for Robot Control

T. S. Liu
W. S. Lee
Department of Mechanical Engineering,
National Chiao Tung University,
Hsinchu 30010, Taiwan
e-mail: tsliu@cc.nctu.edu.tw

In order to make a robot precisely track desired periodic trajectories, this work proposes a sliding mode based repetitive learning control method, which incorporates characteristics of sliding mode control into repetitive learning control. The learning algorithm not only utilizes shape functions to approximate influence functions in integral transforms, but also estimates inverse dynamics functions based on integral transforms. It learns at each sampling instant the desired input joint torques without prior knowledge of the robot dynamics. To carry out sliding mode control, a reaching law method is employed, which is robust against model uncertainties and external disturbances. Experiments are performed to validate the proposed method. [S0022-0434(00)02001-3]

1 Introduction

A learning system is capable of improving its performance over time by interaction with its environment. A learning control system is designed so that its learning controller can improve the performance of closed-loop systems by generating command inputs to plants and utilizing feedback information from plants. Rather than proportional-derivative (PD) type learning control methods in the past [1], fuzzy learning control [2] and neural network based learning controller [3] have been presented. Yang and Asada [4] proposed an excitation scheduling method to enable an impedance control law to learn quasi-static, slow modes in the beginning, followed by learning faster modes. Similar to controllers presented by Horowitz [5] and Messner et al. [6], a repetitive robot controller was implemented with Cartesian trajectory description [7]. A structured singular value method was also applied to determine stability and performance robustness of repetitive control systems [8].

This study presents a sliding mode based repetitive learning control approach to robot control. The advantages of using sliding mode control instead of PD control for feedback portion of a repetitive learning control include: (1) the robust property of sliding mode control dealing with model uncertainties; (2) the flexibility in using sliding mode control. It is known that, in general, the transient dynamics of a variable structure control system [9] is accounted for by a reaching mode followed by a sliding mode [10]. Therefore the design involves, first, the design of an appropriate sliding surface and a reaching mode method for the desired sliding mode dynamics, and second, the design of a learning algorithm to ensure asymptotically stability.

Sliding mode based repetitive learning control focuses on learning rules that estimate feedforward terms (inverse dynamic functions). A class of function identification for learning algorithm compensation based on integral transforms was presented by Messner et al. [6]. This study employs a set of shape functions to approximate influence functions and estimates inverse dynamics functions based on integral transforms. The inverse dynamics function is estimated by the integral of a predefined kernel multiplied by an estimated influence function. The influence function used in integral transforms is approximated by a set of linear shape functions and this influence function is in turn represented by corresponding coefficients. An adaptation law employing kernel functions, sliding surfaces, and shape functions is thus developed in this study to update the coefficients associated with influence functions.

2 Tracking Control of Manipulator

In general, the equation of motion for a n-axis manipulator can be expressed as

$$M(q)\ddot{q}+C(q,\dot{q})\dot{q}+G(q)+d(q,\dot{q})=u \tag{1}$$

where q, $\dot{q}$, and $\ddot{q}$ are, respectively, the $n\times 1$ joint position, velocity, and acceleration vectors, u represents the $n\times 1$ torque vector generated by actuators, $M(q)$ is the symmetric positive definite generalized inertia matrix, $C(q,\dot{q})$ is the force (torque) vector resulting from Coriolis and centripetal accelerations, $G(q)$ is the generalized gravitational force vector, and $d(q,\dot{q})$ denotes the disturbance. Define a $2n$-dimensional state vector x as

$$x=\begin{bmatrix} x_1 \\ x_2 \end{bmatrix}=\begin{bmatrix} q \\ \dot{q} \end{bmatrix}$$

Thus Eq. (1) can be written as

$$\dot{x}=A(x)+B(x)u+\nu(x)$$

where

$$A(x)=\begin{bmatrix} x_2 \\ -M^{-1}(x)C(x)x_2-M^{-1}(x)G(x) \end{bmatrix}$$

$$B(x)=\begin{bmatrix} 0 \\ M^{-1}(x) \end{bmatrix}$$

and the disturbance is expressed by

$$\nu(x)=\begin{bmatrix} 0 \\ -M^{-1}(x)d(x) \end{bmatrix}$$

2.1 Sliding Mode Control. The manipulator is demanded to track a desired motion $q_d(t)$. Define an error vector

$$\mathbf{e}=\begin{bmatrix} \dot{e} \\ e \\ \int_0^t e\,dt \end{bmatrix}$$

where $e=q-q_d$ and $\dot{e}=q-\dot{q}_d$. A sliding surface s of n dimensions is of the form:

Contributed by the Dynamic Systems and Control Division for publication in the JOURNAL OF DYNAMIC SYSTEMS, MEASUREMENT, AND CONTROL. Manuscript received by the Dynamic Systems and Control Division April 15, 1999. Associate Technical Editor: E. A. Misawa.

$$s(\mathbf{e})=\mathbf{C}(\mathbf{e})=[I \quad \Lambda \quad \Gamma]\begin{bmatrix}\dot{e}\\ e\\ \int_0^t edt\end{bmatrix}=\dot{e}+\Lambda e+\Gamma\int_0^t edt \quad (2)$$

where both Λ and Γ are positive definite matrices. In addition, a reaching law [11] is defined as

$$\dot{s}=-Q\ \mathrm{sgn}(s)-Ks \quad (3)$$

where gains Q and K are diagonal matrices with positive elements q_i and k_i, respectively. Chattering can be reduced by tuning q_i and k_i in this reaching law. Near the sliding surface, $s_i \approx 0$. It follows from Eq. (3) that $|\dot{s}_i| \approx q_i$. By using a small gain, the chattering amplitude can be reduced. However, q_i cannot be chosen equal to zero since the reaching time would become infinite. Moreover, when the state is not near the sliding surface a large k_i is employed to increase the reaching rate.

Taking the time derivative of Eq. (2) gives

$$\begin{aligned}\dot{s}&=\ddot{e}+\Lambda\dot{e}+\Gamma e\\ &=\Lambda\dot{e}+\Gamma e-\ddot{q}_d-M(q)^{-1}(C(q,\dot{q})\dot{q}+G(q)+d(q,\dot{q})-u)\end{aligned} \quad (4)$$

Equating Eqs. (3) and (4) yields control input

$$\begin{aligned}u=&M(q)\{-Q\ \mathrm{sgn}(s)-Ks-\Lambda\dot{e}-\Gamma e+\ddot{q}_d\}\\ &+C(q,\dot{q})\dot{q}+G(q)+d(q,\dot{q})\end{aligned} \quad (5)$$

2.2 Sliding Mode Based Repetitive Learning Control. Tracking control is aimed at following a prescribed trajectory as closely as possible. Using inverse kinematics one can obtain joint position, velocity, and acceleration vectors denoted by q_d, $\dot{q}_d$, and $\ddot{q}_d$, respectively. The desired torque input of a manipulator, denoted by $w_d(.):R_+ \rightarrow R^n$, is defined as

$$w_d(t)=M(q_d)\ddot{q}_d+C(q_d,\dot{q}_d)\dot{q}_d+G(q_d)+d(q_d,\dot{q}_d)$$

Definition: Let $C_k(T)$ denote a subset of $C(T)$ (which is the space of continuous T-period functions $w_d:R_+ \rightarrow R^n$) such that every w_d is piecewise continuously differentiable, and

$$\sup_{t\in[0,T]}\left|\frac{d}{dt}w_d(\bullet)\right|\leq k$$

Given a collection for shape functions $\{\Phi_i\}$ and $\varepsilon>0$, there exists a finite number of shape functions $\{\Phi_0,\Phi_1,\Phi_2,\ldots,\Phi_N\}$ that uniformly approximate members of $C_k(T)$ within $\varepsilon>0$, i.e., for every $w_d\in C_k(T)$, there exists constant vectors $C_0,C_1,C_2,\ldots,C_n \in R^n$ such that

$$\sup_{t\in[0,T]}\left|w_d(t)-\sum_{i=0}^{N}C_i\Phi_i\right|<\varepsilon$$

To estimate the desired torque $w_d(t)$, it can be approximated by a linear combination of appropriately selected period shape functions Φ_i. Hence,

$$w_d(t)\cong\sum_{i=0}^{N}C_i\Phi_i(t) \quad (6)$$

where $C_i\in R^n$ represent unknown coefficient vectors for each shape function Φ_i at an instant, and N denotes the total number of shape functions. The estimated feedforward term is generated by determining the coefficient vectors $\hat{C}_i$ [7], i.e.,

$$\hat{w}_d(t)=\sum_{i=0}^{N}\hat{C}_i\Phi_i(t) \quad (7)$$

The coefficient vectors are updated on-line by conducting the following estimation law [7]:

$$\frac{d}{dt}\hat{C}_i(t)=-K_L\Phi_i(t)s \quad i=1,2,\ldots,N \quad (8)$$

where K_L is a constant positive definite matrix.

Another approximation of the ideal feedforward compensation term can be represented by

$$w_d(t)=\int_0^T K(t,\tau)I(\tau)d\tau \quad (9)$$

where the function $K(\bullet,\bullet):R\times[0,T]$ is a known Hilbert-Schmit kernel that satisfies

$$\int_0^T K(t,\tau)^2d\tau=k<\infty \quad K(t,\tau)=K(t+T,\tau) \quad (10)$$

whereas the influence function $I(\bullet):[0,T]\rightarrow R^n$ is unknown. If a kernel function is chosen to satisfy Eq. (10), then the feedforward term $w_d(t)$ can be estimated by influence function $I(\bullet)$. The following function adaptation law for estimating the unknown functions $w_d(t)$ and $I(\bullet)$ was presented by Messner et al. [6]:

$$\hat{w}_d(t)=\int_0^T K(t,\tau)\hat{I}(t,\tau)d\tau \quad (11)$$

$$\frac{\partial}{\partial t}\hat{I}(t,\tau)=-K_LK(t,\tau)s \quad (12)$$

The estimated $\hat{w}_d(t)$ is hence indirectly updated by the adaptation of $\hat{I}(t,\tau)$, which is the estimate of the influence function $I(\tau)$.

In the integral transform estimation, the feedforward term $\hat{w}_d(t)$ is estimated through updating the influence function $\hat{I}(t,\tau)$ according to the learning law Eq. (12). However, if the influence function, which belongs to the space of continuous T-period functions, satisfies

$$\sup_{t\in[0,T]}\left|\hat{I}(t,\tau)-\sum_{i=0}^{N}\hat{C}_i(t,\tau)\Phi_i(\tau)\right|<\varepsilon,$$

it can be expressed by a set of shape functions. The unknown influence function is proposed as

$$\hat{I}(t,\tau)=\sum_{i=0}^{N}\hat{C}_i(t,\tau)\Phi_i(\tau) \quad (13)$$

and the coefficient adaptation law becomes

$$\frac{\partial}{\partial t}\hat{C}_i(t,\tau)=-K_LK(t,\tau)\Phi_i(\tau)s \quad (14)$$

where $\Phi_i(\bullet)$ denotes a shape function and $\hat{C}_i(\bullet)$ its associated coefficient. The advantage of the above learning rule is that only the associated coefficients for shape functions are updated in estimating the influence function, which can be in turn obtained by a linear combination of shape functions. It is unnecessary to save all influence function values at every sampling instant, thus computer memory space can be saved. Since the value of influence function $\hat{I}(t,\tau)$ is updated on the basis of previous value $\hat{I}(t,\tau-\Delta\tau)$, the property of ‘‘interpolating’’ is achieved.

2.3 Stability Analysis. The stability of the present control method for the robotic model represented by Eq. (1) depends on the following conditions.

Condition 1: There exists an influence function $\alpha(t,\tau)$, such that

$$M(q)\dot{v}+C(q,\dot{q})v+G(q)+d(q,\dot{q})=\int_0^T K(t,\tau)\alpha(t,\tau)d\tau \quad (15)$$

where $v(t)\in R^n$ is a vector of smooth functions.

Condition 2: Using a proper definition of matrix $C(q,\dot{q})$, both $M(q)$ and $C(q,\dot{q})$ in Eq. (1) satisfy

$$x^T(\dot{M}-2C)x=0 \quad \forall x \in R^n$$

Hence, $(\dot{M}-2C)$ is a skew-symmetric matrix. In particular, the element of $C(q,\dot{q})$ can be defined as

$$C_{ij}=\frac{1}{2}\left[\dot{q}^T\frac{\partial M_{ij}}{\partial q}+\sum_{k=1}^{n}\left(\frac{\partial M_{ik}}{\partial q_j}-\frac{\partial M_{jk}}{\partial q_i}\right)\dot{q}_k\right] \tag{16}$$

Condition 3: In robot control systems, the disturbance $d(q,\dot{q})$ due to friction, sensor noise, etc. is assumed to be bounded. Generally speaking, unmodeled dynamics is bounded as follows:

$$\|d\| \leqslant L_0+L_1\|\dot{e}\|+L_2\|e\|$$

where L_0, L_1, and L_2 are positive constants.

Remark: If the structure conditions presented above are satisfied, a sliding mode based repetitive learning controller for achieving the trajectory tracking can be realized.

In the current study, the norm of vector x is defined as

$$\|x\|=\left(\sum_{i=1}^{n} x_i^2\right)^{1/2}$$

and the norm of matrix A is defined as

$$\|A\|=(\max_{\text{eigenvalue}} A^TA)^{1/2}$$

The singular value of matrix A is defined as $\alpha(A) = (\text{eigenvalue}(A^TA))^{1/2}$ and $\alpha_{\min}(A)$ denotes the smallest singular value. For positive definite matrix $A=A^T$, the matrix property [12]

$$x^TAx \geqslant \alpha_{\min}(A)\|x\|^2 \tag{17}$$

will be employed in this work to formulate the learning control method.

In the following, a brief overview of the proposed sliding mode based repetitive learning controller is given. The design problem for the proposed sliding mode based repetitive learning controller is described as follows: For any given desired trajectory $q_d \in R^n$, $\dot{q}_d \in R^n$, and $\ddot{q}_d \in R^n$, with some or all of the manipulator coefficient vectors unknown, derive a controller for the actuator torque (force), and an adaptation law for the unknown coefficient vectors, such that the manipulator joint position $q(t)$ precisely tracks $q_d(t)$.

To ensure the convergence of the trajectory tracking, define a reference velocity vector $\dot{q}_r$ as

$$\dot{q}_r=\dot{q}_d-\Lambda e-\Gamma\int_0^t e\,dt \tag{18}$$

where both Λ and Γ denote positive definite matrices whose eigenvalues are strictly in the right-half plane. Therefore, the sliding surface s defined in Eq. (2) can be expressed by

$$s=\dot{e}+\Lambda e+\Gamma\int_0^t e\,dt=\dot{q}-\dot{q}_r \tag{19}$$

Consider the plant defined in Eq. (1), a repetitive learning controller using sliding mode feedback control is proposed, i.e.,

$$u=\hat{w}_d(t)-Q\,\text{sgn}(s)-Ks \tag{20}$$

where $\hat{w}_d(t)$ can be estimated by the linear combination of shape functions Eq. (7) or by the integral of kernel function and influence function Eq. (11). The adaptation laws can be found in Eqs. (8) and (12). Treating $s=0$, where s is defined in Eq. (19), as a sliding surface, by combining Eqs. (1) and (18) and using the property that $s=\dot{q}-\dot{q}_r$, which follows from Eqs. (2) and (18), the sliding mode equation reads

$$M\dot{s}=\int_0^T K(t,\tau)\hat{I}(t,\tau)d\tau-\int_0^T K(t,\tau)\alpha(t,\tau)d\tau$$
$$-Cs-Ks-Q\,\text{sgn}(s)-d \tag{21}$$

where the following definition is used, which satisfies Condition 1,

$$M(q)\ddot{q}_r+C(q,\dot{q})\dot{q}_r+G(q)=\int_0^T K(t,\tau)\alpha(t,\tau)d\tau$$

A generalized Lyapunov function is chosen as

$$V(t)=\tfrac{1}{2}s^TMs+\tfrac{1}{2}e^TK_Le \tag{22}$$

where $K_L=\sigma_S I$ with $\sigma_S>0$. Taking the time derivative of Eq. (22) gives

$$\dot{V}=s^TM\dot{s}+s^TCs+e^TK_Le \tag{23}$$

Substituting Eq. (20) into Eq. (23) and employing Condition 2 give

$$\dot{V}=s^T\left\{\int_0^T K(t,\tau)[\hat{I}(t,\tau)-\alpha(t,\tau)]d\tau-Ks-Q\,\text{sgn}(s)-d\right\}$$
$$+e^TK_L\dot{e} \tag{24}$$

Now choosing Λ and Γ such that $s^T\alpha(t,\tau)>s^T\hat{I}(t,\tau)$ yields

$$\dot{V}\leqslant -s^TKs-s^TQ\,\text{sgn}(s)-s^Td+e^TK_L\left(s-\Lambda e-\Gamma\int_0^t e\,dt\right) \tag{25}$$

From Condition 3, one has

$$-s^Td \leqslant \|s\|\left[L_0+L_1\left(\|s\|+\Lambda\|e\|+\Gamma\left\|\int_0^t e\,dt\right\|\right)+L_2\|e\|\right]$$
$$\leqslant L_0\|s\|+L_1\|s\|^2+[\lambda_M(\Lambda)L_1+L_2]\|s\|\|e\|$$
$$+L_1\lambda_M(\Gamma)\|s\|\left\|\int_0^t e\,dt\right\| \tag{26}$$

where $\lambda_M(\Lambda)$ denotes the maximum eigenvalues of Λ. It follows that

$$\dot{V}\leqslant -[\lambda_m(K)-L_1]\|s\|^2+[\sigma_S+\lambda_M(\Lambda)L_1+L_2]\|s\|\|e\|$$
$$+L_1\lambda_M(\Gamma)\|s\|\left\|\int_0^t e\,dt\right\|-\sigma_S\lambda_m(\Gamma)\|e\|\left\|\int_0^t e\,dt\right\|$$
$$-\sigma_S\lambda_m(\Lambda)\|e\|^2+[L_0-\lambda_m(Q)]\|s\| \tag{27}$$

where $\lambda_m(\bullet)$ denotes the minimum eigenvalue of a matrix. A further manipulation of Eq. (27) leads to

$$\dot{V}\leqslant -\left[\|s\| \quad \|e\| \quad \left\|\int_0^t e\,dt\right\|\right]R\begin{bmatrix}\|s\|\\ \|e\|\\ \left\|\int_0^t e\,dt\right\|\end{bmatrix}$$
$$-\frac{\lambda_m(K)}{2}\left[\|s\|-\frac{L_0-\lambda_m(Q)}{\lambda_m(K)}\right]^2+\frac{[L_0-\lambda_m(Q)]^2}{2\lambda_m(K)}$$
$$-\lambda_M(\Gamma)\left\|\int_0^t e\,dt\right\|^2 \tag{28}$$

where

$$R=\begin{bmatrix} \frac{\lambda_m(K)}{2}-L_1 & -\frac{\lambda_M(K)L_1+L_2+\sigma_S}{2} & -\frac{L_1\lambda_M(\Gamma)}{2} \\ -\frac{\lambda_M(K)L_1+L_2+\sigma_S}{2} & \sigma_S\lambda_m(\Lambda) & \frac{\sigma_S\lambda_m(\Gamma)}{2} \\ -\frac{L_1\lambda_M(\Gamma)}{2} & \frac{\sigma_S\lambda_m(\Gamma)}{2} & \lambda_M(\Gamma) \end{bmatrix}$$

It is always feasible to adequately choose K, Λ, Γ, and σ_S such that R is positive definite. Therefore, one can prescribe values of positive constants σ_d, σ_τ, and σ_β to satisfy

$$R=\begin{bmatrix} \sigma_d & 0 & 0 \\ 0 & \sigma_\tau & 0 \\ 0 & 0 & \sigma_\beta \end{bmatrix}+\tilde{R} \tag{29}$$

where $\tilde{R}$ is a positive semidefinite matrix. With Eq. (29), it follows from Eq. (28) that

$$\dot{V}\leqslant-\sigma_d\|s\|^2-\sigma_\tau\|e\|^2+\frac{[L_0-\lambda_m(Q)]^2}{2\lambda_m(K)} \tag{30}$$

i.e.

$$\dot{V}\leqslant-2\gamma V+\varepsilon \tag{31}$$

where $\varepsilon=[L_0-\lambda_m(Q)]^2/2\lambda_m(K)$ and $\gamma=\min(\sigma_d/\lambda_M(M),\sigma_\tau/\sigma_S)$ in view of Eqs. (17) and (22). Solving Eq. (31) yields

$$V(t)\leqslant e^{-2\gamma t}\left[V(0)-\frac{\varepsilon}{2\gamma}\right]+\frac{\varepsilon}{2\gamma} \tag{32}$$

Therefore, substituting Eq. (22) and $K_L=\sigma_S I$ into Eq. (32) results in

$$\begin{aligned}\|e\|&\leqslant\sqrt{\frac{e^{-2\gamma\cdot t}}{\sigma_S}\left[V(0)-\frac{\varepsilon}{2\gamma}\right]+\frac{\varepsilon}{2\gamma\cdot\sigma_S}}\\ &\leqslant\frac{e^{-\gamma\cdot t}}{\sqrt{\sigma_S}}\left[V(0)-\frac{\varepsilon}{2\gamma}\right]^{1/2}+\sqrt{\frac{\varepsilon}{2\gamma\cdot\sigma_S}}\\ &\leqslant\frac{1}{\sqrt{\sigma_S}}e^{-\gamma\cdot t}\left[V(0)-\frac{\varepsilon}{2\gamma}\right]^{1/2}+\sqrt{\frac{\varepsilon}{2\gamma}}\end{aligned} \tag{33}$$

As a consequence,

$$\lim_{t\to\infty}\|e\|\leqslant\sqrt{\frac{\varepsilon}{2\gamma}}$$

This completes the proof of the following theorem:

Theorem: For a robot model Eq. (1) subject to sliding mode based repetitive learning control, which is accounted for by Eq. (19), the sliding surface S and the tracking error e are uniformly bounded if both gain matrices K and Q in the reaching law are adequately chosen. Furthermore, having learned a number of cycles, the ultimate tracking error e is bounded by

$$\lim_{t\to\infty}\|e\|<\sqrt{\frac{\varepsilon}{2\gamma}}$$

where

$$\varepsilon=[L_0-\lambda_m(Q)]^2/2\lambda_m(K) \tag{34}$$

and

$$\gamma=\min(\sigma_d/\lambda_M(M),\sigma_\tau/\sigma_S) \tag{35}$$

According to Eq. (34), ε can be made arbitrarily small by enlarging gains in gain matrix K, which makes the control effort to grow. In practice, however, the minimum size of the error bound is limited since too large control effort may not be available.

2.4 Chattering Elimination. Since the control law given above is discontinuous across the sliding surface $s=0$, it gives rise to chattering in a trajectory tracking process. Chattering is undesirable in practice because it introduces high control effort, and furthermore, may excite unmodeled high frequency plant dynamics, which would result in instabilities. This problem can be overcome by smoothing out the discontinuous control input in the neighborhood of the sliding surface [13]. Therefore, this study uses $s/(|s|+\delta)$ in place of sgn(s) for control law Eq. (20), i.e.,

$$u=\hat{w}_d(t)-Qs_\delta-Ks \tag{36}$$

where

$$s_\delta=\begin{bmatrix} \frac{s_1}{|s_1|+\delta_1} \\ \vdots \\ \frac{s_n}{|s_n|+\delta_n} \end{bmatrix}$$

and δ_i is a positive constant.

3 Implementation

As shown in Fig. 1, this study constructs a three-axis $R-\theta-Z$ direct-drive robot manipulator, where the first link is driven by a NSK Megatorque motor [14], the second by a NSK Megathrust motor, and the third by an electrothrust motor together with ball screw.

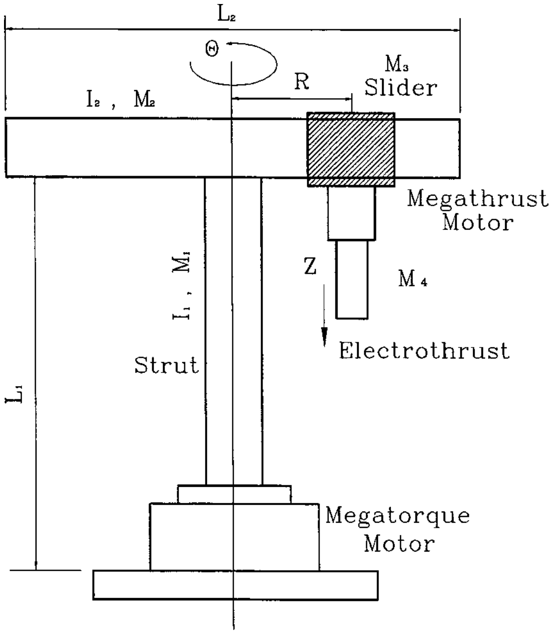

Fig. 1 Schematic diagram of direct-drive robot

3.1 Discrete Control Law.

In order to implement sliding mode control and the proposed method respectively using a DSP controller, discrete equivalents of both control laws are formulated in the following:

(a) Sliding Mode Control. The sliding mode control method adopts a new reaching law Eq. (3) to achieve the sliding surface. The control input of each axis actuator can be discretized using a zero-order hold. The discrete time form resulting from Eq. (5) is written as

$$\begin{aligned} u_\theta(k+1) = {} & [I_1+I_2+(M_3+M_4)R(k)^2]\Big\{-Q_\theta \operatorname{sgn}[s_\theta(k)] \\ & -K_\theta s_\theta(k) - \Lambda_\theta\left[\frac{e_\theta(k)-e_\theta(k-1)}{\Delta t}\right] \\ & -\Gamma_\theta e_\theta(k) + \alpha(k)^d\Big\} + 2M_3R(k)V_R(k)\omega(k) \\ & +2M_4R(k)V_R(k)\omega(k) + b_\theta\omega(k) + \mu_\theta N_\theta \operatorname{sgn}[\omega(k)] \end{aligned} \tag{37}$$

$$\begin{aligned} u_R(k+1) = {} & (M_3+M_4)\Big\{-Q_R \operatorname{sgn}[s_R(k)] - K_R s_R(k) \\ & -\Lambda_R\left[\frac{e_R(k)-e_R(k-1)}{\Delta t}\right] - \Gamma_R e_R(k) + a_R^d(k)\Big\} \\ & -(M_3+M_4)R(k)\omega^2(k) + b_R V_R(k) \\ & +\mu_R N_R \operatorname{sgn}[V_R(k)] \end{aligned} \tag{38}$$

$$\begin{aligned} u_Z(k+1) = {} & M_4\Big\{-Q_Z \operatorname{sgn}[s_Z(k)] - K_Z s_Z(k) \\ & -\Lambda_Z\left[\frac{e_Z(k)-e_Z(k-1)}{\Delta t}\right] - \Gamma_z e_z(k) + a_Z^d(k)\Big\} + M_4 g \\ & +b_Z V_Z(k) + \mu_Z N_Z \operatorname{sgn}[V_Z(k)] \end{aligned} \tag{39}$$

where $\Lambda = \mathrm{diag}(30, 30, 30)$ $\Gamma = \mathrm{diag}(30, 30, 30)$, $Q = \mathrm{diag}(1,1,1)$, and $K = \mathrm{diag}(100, 350, 380)$. Further, since friction is treated as disturbance $d(q,\dot{q})$ depicted in Eq. (5), friction compensation has been incorporated in Eqs. (37)–(39).

(b) The Present Method. Except for its employing shape functions to estimate influence functions, the structure of this learning control method is the same as learning control using integral transforms. The learning control law consists of Eqs. (11), (13), and (14). There are some typical shape functions [15] such as Fourier series shape functions, polynomial shape functions, and piecewise linear shape functions, which can be used to approximate the periodic continuous function $I(t,\tau)$. This experiment employs a set of piecewise linear functions, as depicted in Fig. 2. Accordingly, in each interval of $[iT/N,(i+1)T/N]$, only two linear shape functions, Φ_i and Φ_{i+1}, are required; i.e., there are only two corresponding coefficients, c_i and c_{i+1}, to be updated at any instant. For computational efficiency of a kernel function, a piecewise linear function shown in Fig. 3 is used as a kernel function for integral transforms. The piecewise linear functions is defined as follows:

Denote the span s of this piecewise linear function as the length of a subinterval where the function value is not zero. The piecewise linear function can be written as:
If $t \in [0,s/2]$, as shown in Fig. 3(a),

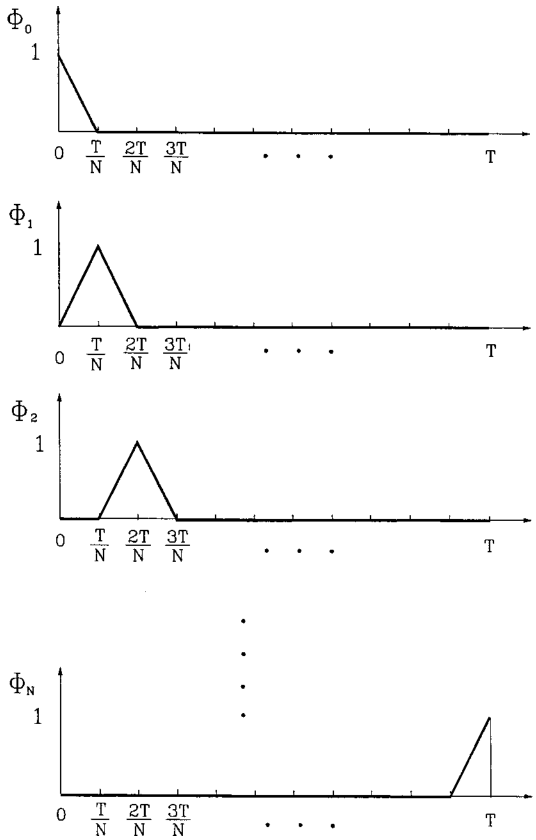

Fig. 2 Piecewise linear shape functions

$$K(t,\tau) = \begin{cases} 1+m(\tau-t) & \text{for } 0 \leqslant \tau < t \\ 1-m(\tau-t) & \text{for } t \leqslant \tau < t+\dfrac{s}{2} \\ 1+m(\tau-t+T) & \text{for } t+T-\dfrac{s}{2} \leqslant \tau < T \end{cases} \tag{40}$$

If $t \in [s/2, T-s/2]$, as shown in Fig. 3(b),

$$K(t,\tau) = \begin{cases} 1+m(\tau-t) & \text{for } t-\dfrac{s}{2} \leqslant \tau < t \\ 1-m(\tau-t) & \text{for } t \leqslant \tau < t+\dfrac{s}{2} \end{cases} \tag{41}$$

If $t \in [T-s/2, T]$, as shown in Fig. 3(c),

$$K(t,\tau) = \begin{cases} 1-m(\tau-t+T) & \text{for } 0 \leqslant \tau < t-T+\dfrac{s}{2} \\ 1+m(\tau-t) & \text{for } t-\dfrac{s}{2} \leqslant \tau < t \\ 1-m(\tau-t) & \text{for } t \leqslant \tau < T \end{cases} \tag{42}$$

The speed and acceleration profiles of the end-effector are prescribed as shown in Figs. 4(a) and 4(b), respectively. Figure 5 depicts a spatial circular trajectory to be tracked. In addition, a planar square trajectory on the $X-Z$ plane, will also be carried out in experiments. Figure 6 depicts the control block diagram. To implement the present method, Eqs. (11), (13), and (14) are rewritten to become the discretized form:

$$\hat{w}_d(\overline{k}) = \frac{1}{2}\sum_{l=0}^{n-1}[K(\overline{k},l)\hat{I}(\overline{k},l) + K(\overline{k},l+1)\hat{I}(\overline{k},l+1)]a\Delta t \tag{43}$$

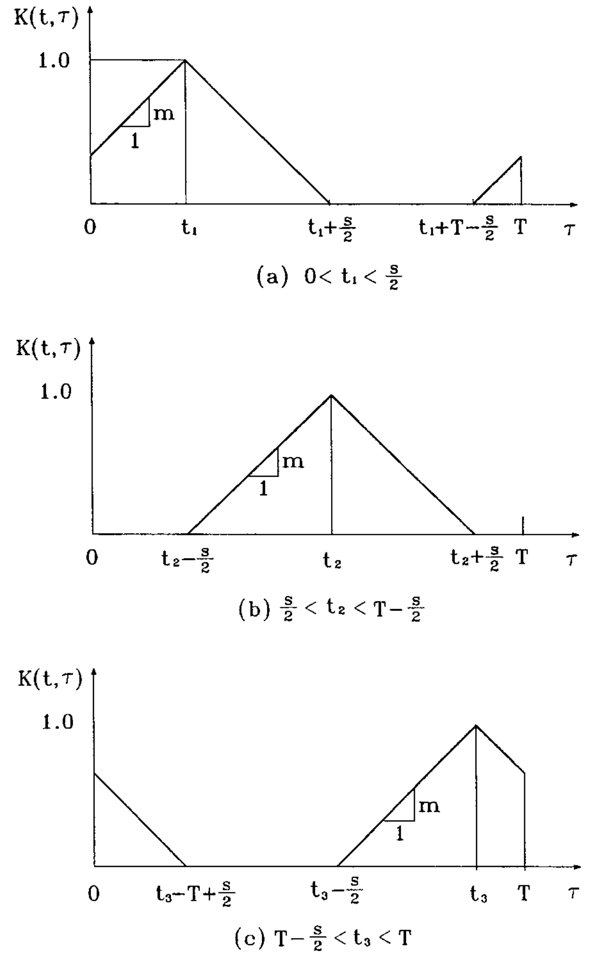

Fig. 3 The piecewise linear kernel function

$$\hat{I}(\bar{k},l)=\sum_{i=0}^{N}\hat{C}_i(\bar{k},l)\Phi_i(l) \tag{44}$$

$$\hat{C}_i(\bar{k},l)=\hat{C}(\bar{k}-1,l)-K_L K(\bar{k},l)\Phi_i(l)\sum_{i=k-a+1}^{k} s(i)\Delta t \tag{45}$$

where $K_L=\mathrm{diag}(150,850,950)$, $a=5$, and integral transforms are computed by a trapezoid method. Moreover, the sliding surface is formulated as

$$s(k)=\frac{e(k)-e(k-1)}{\Delta t}+\Lambda e(k)+\Gamma\sum_{i=0}^{k} e(i)\Delta t \tag{46}$$

where $\Lambda=\mathrm{diag}(30,30,30)$, and $\Gamma=\mathrm{diag}(30,30,30)$. It follows from Eq. (46) that the control law Eq. (20) becomes, in discretized form,

$$u(k+1)=-Q\,\mathrm{sgn}[s(k)]-K[s(k)]+\hat{w}_d(\bar{k})$$

where $Q=\mathrm{diag}(1,1,1)$, and $K=\mathrm{diag}(100,350,380)$. In this discrete control algorithm, k represents an index for the feedback portion of the controller, $\bar{k}$ and l indexes for the repetitive learning portion, and a an integer that relates these indexes. For any given k in a period of the path, $k=a\bar{k}=al$. In other words, the adaptation parameters $\hat{c}_i$ are updated at a rate a times slower than the inner feedback loop. Each increment in k represents a time step of Δt second, and each increment in $\bar{k}$ represents a time step of $a\Delta t$ second. In both sliding mode control and the present method, the gains of the reaching law and sliding surface are the same.

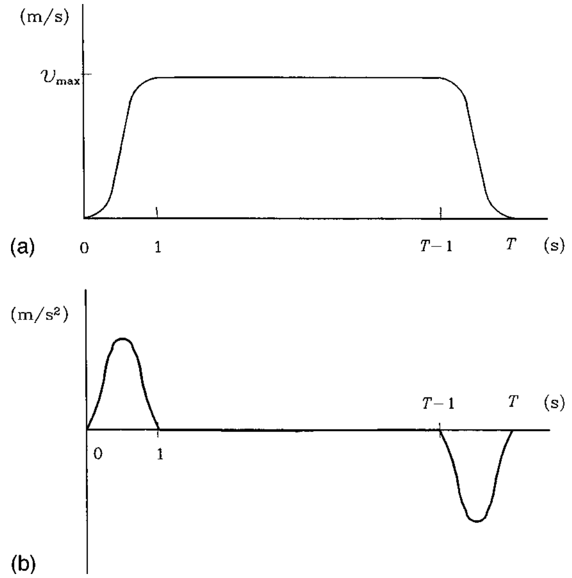

Fig. 4 (*a*) Speed profile of end-effector and (*b*) corresponding acceleration profile

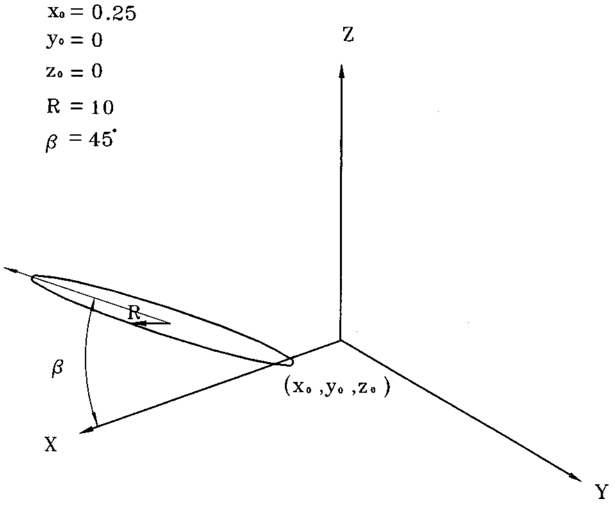

Fig. 5 Desired spatial circular trajectory

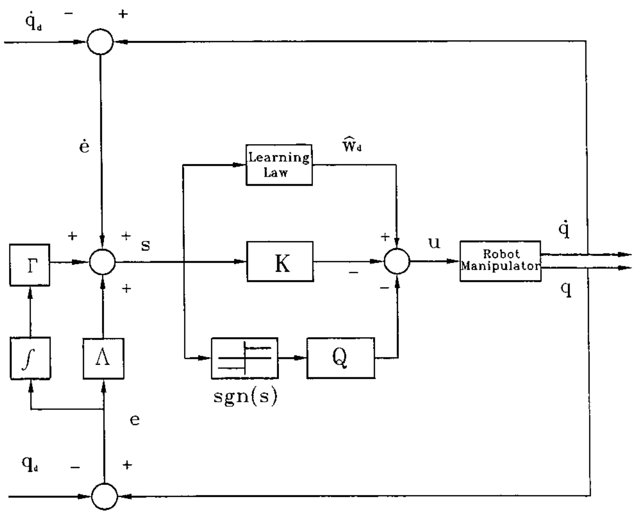

Fig. 6 Control block diagram of the present method

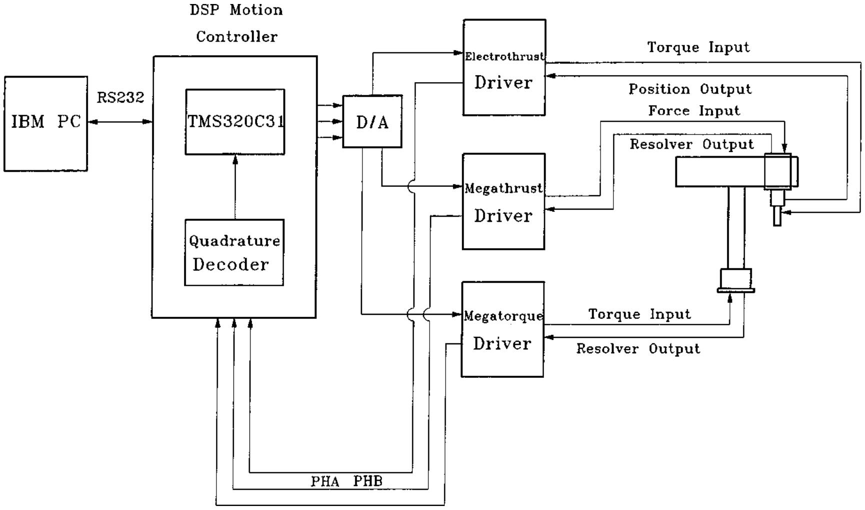

Fig. 7 Experimental setup

The radius R of the desired spatial circle is denoted as 0.1 m and period $T=10$ s. The maximum speed $v_{\max}$ is 0.068 m/s. The side length of the planar square is 0.1 m while the maximum speed $v_{\max}$ is 0.05 m/s. A total number of 100 linear shape functions, i.e. $N=100$, are prescribed to approximate influence functions. The kernel function is a piecewise linear function of slope m=2; the span s of this piecewise linear function is hence 1.0 sec.

3.2 Experimental Results. To control a three-axis R-θ-Z direct-drive robot as shown in Fig. 1, which contains three servomotors, this study employs an MX31 DSP integrated motion controller endowed with a TI TMS320C31 digital signal processor, as depicted in Fig. 7. Along the R axis is a direct-drive NSK Megathrust motor that enables the slider to undergo the force mode translational motion. For the θ axis, a direct-drive NSK Megatorque motor performs the torque mode rotational motion. In addition, motion along the Z axis is achieved by a 3 phase DC servo-motor operating in torque mode, where the motor rotation is altered into translation by a screw mechanism installed on the motor shaft.

(1) Spatial Circular Trajectory Tracking. In contrast to Fig. 8(*a*), tracking errors in Fig. 8(*b*) progressively reduce to a smaller range through learning. Hence, this sliding mode based repetitive learning controller outperforms the sliding mode controller. The large tracking error generated at the beginning of the first period is arised from friction at robot joints and the prescribed acceleration command. It is difficult to quickly acquire knowledge of the friction torque and the adequate motor torque corresponding to acceleration commands at the outset of every learning period, since the learning gains can not be infinitely large during the learning process. Variation of a sliding surface is shown in Fig. 9 and confirms that the proposed control algorithm achieves its objective through learning. Using the present method, Figs. 10 and 11 depict the first time and the fifth time learning results of a spatial circular trajectory, respectively. There is no significant improvement after five iterations.

(2) Planar Square Trajectory Tracking. Comparing with Fig. 12(*a*) for the sliding mode control result, the tracking error shown in Fig. 12(*b*) reduces to lie between ±1.5 mm. Using the present method, Figs. 13 and 14, respectively, depict the first time and the fifth time learning results for the desired trajectory. The tracking accuracy is significantly improved through learning; however, the improvement after five iterations is not significant.

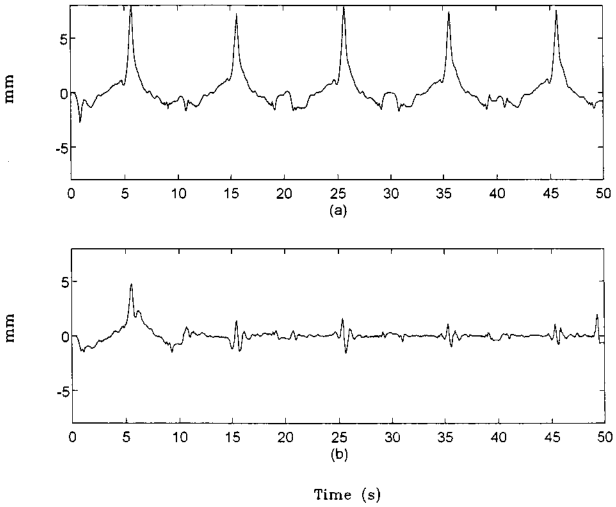

Fig. 8 Position errors in *Z*-axis using (*a*) sliding mode control and (*b*) the present method

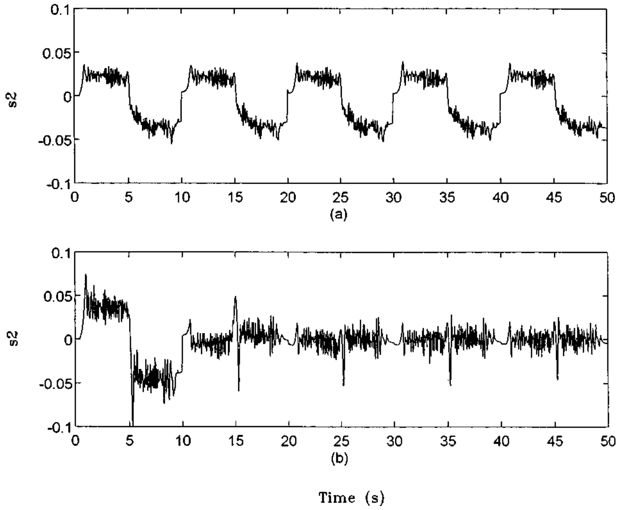

Fig. 9 Sliding variables in *R*-axis using (*a*) sliding mode control and (*b*) the present method

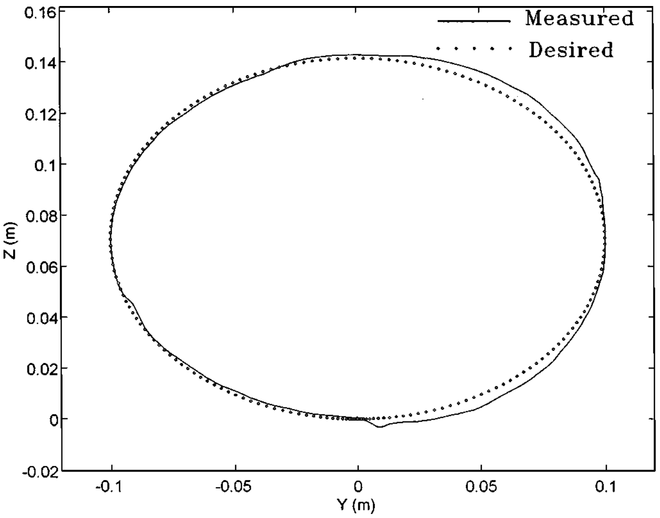

Fig. 10 Tracking result of the first time learning in *Y*-*Z* plane

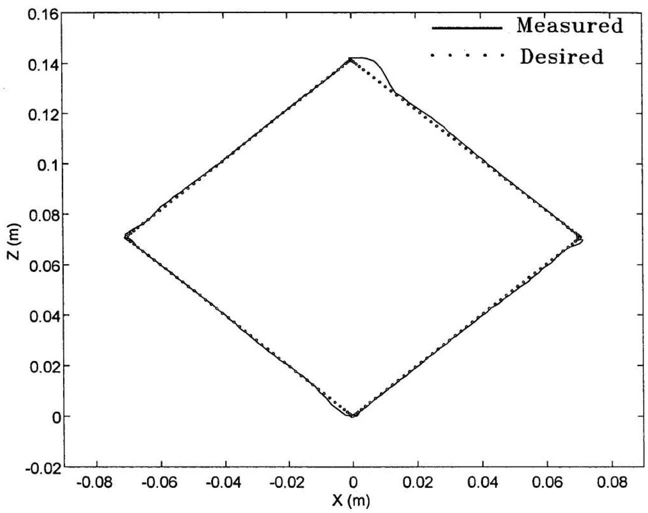

Fig. 13 Tracking result of the first time learning in *X*-*Z* plane

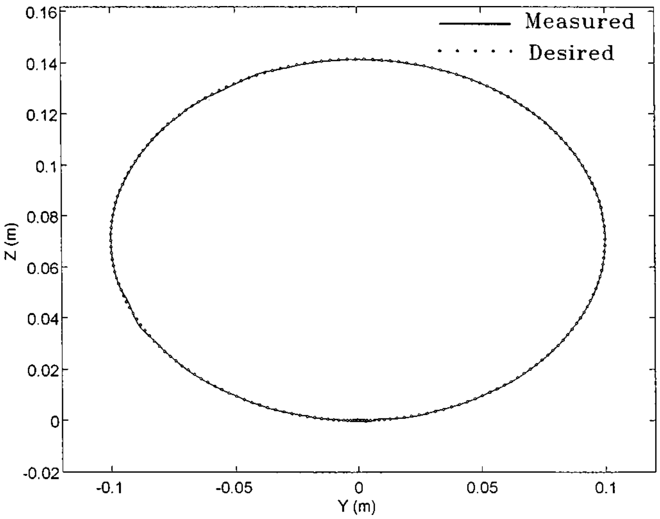

Fig. 11 Tracking result of the fifth time learning in *Y*-*Z* plane

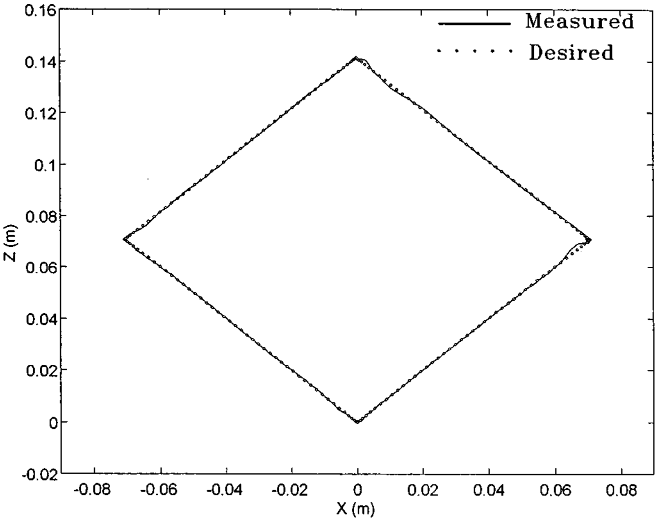

Fig. 14 Tracking result of the fifth time learning in *X*-*Z* plane

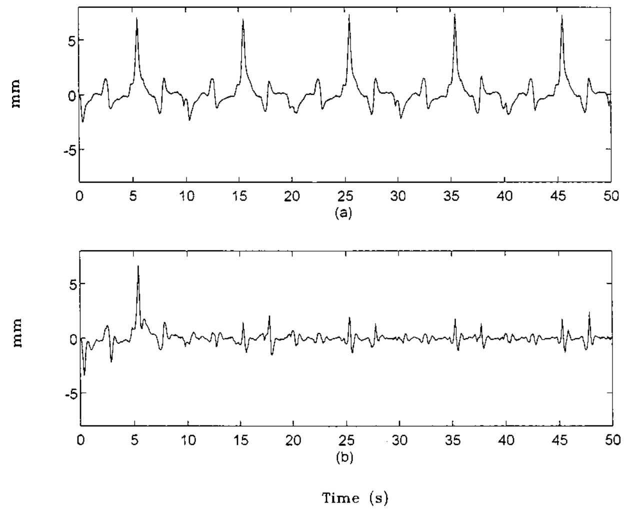

Fig. 12 Position errors in *Z*-axis using (*a*) sliding mode control and (*b*) the present method

The overshoot error at each of four corners are mainly caused by decelerating approaching followed by accelerating departure, in addition to direction alteration of the slider.

The authors have also carried out experiments using a conventional controller—the PD control method. It is found that the departure and arrival points always do not coincide due to cumulative error during trajectory tracking. Moreover, concerning tracking accuracy, both PD control and the first time learning in the present method perform in a comparable manner.

4 Conclusion

This study has presented a new learning control algorithm with robust properties to improve the performance in robot tracking task. According to experimental results, the proposed control method exhibits advantages described as follows:

1 Its error convergence is faster than the sliding mode control method.

2 The time needed for the sliding surface reaching the sliding surface is shorter in each of those trajectory tracking experiments.

3 The accumulated errors generated at the initial time can be effectively reduced by introducing an integral term to the sliding surface.

4 The choice of shape functions depends on the trajectory to be tracked. If the polynomial order of influence functions corresponding to a desired trajectory is high, higher order shape functions, instead of current linear shape functions, should be adopted to improve the estimate for feedforward term.

5 Considering the coefficients adaptation law, i.e., Eq. (14), the proposed learning control requires fewer computer memory space since only associated coefficients for shape functions are updated at every instant.

6 The more the total number N of shape functions is used, the more accurate the feedforward term estimate can accomplish. However, for the sake of computational efficiency, N cannot be too large.

7 The proposed method using shape functions to approximate the influence function for feedforward term can be extended to higher degree-of-freedom robots in a straightforward manner, since the control algorithm for each axis undergoes its own learning process using the same shape function set.

8 By using the modified control law Eq. (36) the chattering caused by the discontinuous control law Eq. (20) can be improved to an acceptable extent.

9 The proposed method is robust since it can do without the dynamic model while successfully tracking the desired trajectory.

References

[1] Arimoto, S., Kawamura, S., and Miyazaki, F., 1984, ''Bettering Operation of Robots by Learning,'' J. Robotic Systems, **1**, No. 2, pp. 123–140.

[2] Kwong, W. A., and Passino, K. M., 1995, ''Dynamically Focused Fuzzy Learning Control,'' Proc. American Control Conf., pp. 3755–3759.

[3] Teshnehlab, M., and Watanabe, K., 1995, ''Flexible Structural Learning Control of a Robotic Manipulator Using Artificial Neural Networks,'' JSME Int. J., Series C, **38**, No. 3, pp. 510–521.

[4] Yang, B. H., and Asada, H., 1995, ''Progressive Learning for Robotic Assembly: Learning Impedance with an Excitation Scheduling Method,'' Proc. IEEE Int. Conf. on Robotics and Automation, pp. 2538–2544.

[5] Horowitz, R., 1993, ''Learning Control of Robot Manipulators,'' ASME J. Dyn. Syst., Meas., Control, **115**, No. 2(B), pp. 402–411.

[6] Messner, W., Horowitz, R., Kao, W. W., and Boals, M., 1991, ''A New Adaptive Learning Rule,'' IEEE Trans. Autom. Control., **36**, No. 2, pp. 188–197.

[7] Guglielmo, K., and Sadegh, N., 1996, ''Theory and Implementation of a Repetitive Robot Controller with Cartesian Trajectory Description,'' ASME J. Dyn. Syst., Meas., Control, **118**, No. 1, pp. 15–21.

[8] Guvenc, L., 1996, ''Stability and Performance Robustness Analysis of Repetitive Control Systems Using Structured Singular Values,'' ASME J. Dyn. Syst., Meas., Control, **118**, No. 3, pp. 593–597.

[9] Emelyanov, S. V., 1967, *Variable Structure Control Systems*, Nauka, Moscow (in Russian).

[10] Slotine, J.-J. E., and Li, W., 1991, *Applied Nonlinear Control*, Prentice-Hall, Englewood Cliffs, N.J.

[11] Gao, W., and Hung, J. C., 1993, ''Variable Structure Control of Nonlinear System: A New Approach,'' IEEE Trans. Ind. Electron., **40**, No. 1, pp. 45–56.

[12] Strang, G., 1980, *Linear Algebra and Its Applications*, 2nd ed., Academic, Press.

[13] Burton, J. A., and Zinober, A. S. I., 1986, ''Continous Approximation of Variable Structure Control,'' Int. J. Syst. Sci., **17**, No. 6, pp. 876–885.

[14] NSK Corporation, 1989, *Megatorque Motor System: User's Manual*, Tokyo, Japan.

[15] Sadegh, N., Horowitz, R., and Tomizuka, M., 1990, ''A Unified Approach to Design of Adaptive and Repetitive Controllers for Robotic Manipulator,'' ASME J. Dyn. Syst., Meas., Control, **112**, No. 4, pp. 618–629.