

An Adaptive Iterative Learning Control for Robot Manipulator in Task Space

T. Ngo, Y. Wang, T.L. Mai, J. Ge, M.H. Nguyen, S.N. Wei

ThanhQuyen Ngo, M. Hung Nguyen

HCM City University of Industry
College of Electrical Engineering
HCM City, Vietnam
E-mail: thanhquyenngo2000@yahoo.com,
nmhung289@yahoo.com

YaoNan Wang, Ji Ge, ShuNing Wei

Hunan University
College of Electrical and Information Engineering
Changsha, Hunan Province 410082, P.R.China
E-mail: yaonan@hnu.cn, gechunxi@126.com,
weishuning@sina.com

T. Long Mai

HCM City University of Industry
College of Electronic Engineering
HCM City, Vietnam
E-mail: mailongtk@gmail.com

Abstract: In this paper, adaptive iterative learning control (AILC) of uncertain robot manipulators in task space is considered for trajectory tracking in an iterative operation mode. The control scheme includes a PD controller with a gain switching technique plus a learning feedforward term, is exploited to predict the desired actuator torque. By using Lyapunov method, an adaptive iterative learning control scheme is presented for robotic system with both structured and unstructured uncertainty, and the overall stability of the closed-loop system in the iterative domain is established. The validity of the scheme is confirmed through a numerical simulation.

Keywords: PD Control; Learning control; robot manipulator.

1 Introduction

In general, robotic manipulators have to face various uncertainties in their dynamics, such as friction, and external disturbance. It is difficult to establish exactly mathematical model for the design of a model-based control system. In order to deal with this problem, the branches of current control theories are broad include classical control, robust control, adaptive control, optimal control, nonlinear control, neural network control, fuzzy logic and intelligent control. However, many adaptive control approaches are rejected as being too computationally intensive because of the required of real-time parameter identification and control design

Owing to its simplicity and robustness to modelling uncertainties, are usually used for repetitive tasks. In this case, the reference trajectory is repeated over a definite operation time. So, iterative learning control (ILC) for robot manipulator has attracted considerable attention in the recent years [1]-[3], the ideal of learning control is to improve the tracking performance from iteration to iteration for applications of robotics in industry which a single repetitive task [4], [5].

ILC is a relatively recent but well-established area of study in control theory. ILC, which can be categorized as an intelligent control methodology, is an approach for improving the transient performance of system that operates repetitively over a fixed time interval [4]. Starting from the classical

Arimoto-type ILC algorithm, we can develop a PID-like update law can be given in [6]. So far some of robot manipulators control in the published literature [7]-[9] and etc, proposed an adaptive ILC to deal with parameter uncertainties, such as the link length, mass inertia, and friction nonlinearity, with a self-organizing capability.

In this paper, a new method is given based on a combination of the advantages of a several control methods into a hybrid one. In particular, it is further extended to the task space or the so-called Cartesian space. To apply robot manipulators to a wide class of tasks, it will be necessary to control not only the position of the end-effector, but also the force exerted by the end-effector on the object. By designing the control law in task space, force control can be easily formulated.

This paper is organized as follows. Section 2 described a dynamic model of an n -link robot manipulator in task space. Section 3 presents AILC and its features are discussed. By using Lyapunov method to prove the asymptotic convergence of proposed controller. Numerical simulation results of a two-link robot manipulator in task space under the possible occurrence of uncertainties are provided to demonstrate the tracking control performance of the proposed AILC system in section 4. Conclusions are drawn in section 5.

2 Robotic Dynamic Model In Task Space

In general, the dynamic of an n -link robot manipulator may be expressed in the Lagrange form [10] as:

$$D(q^i(t))\ddot{q}^i(t) + C(q^i(t), \dot{q}^i(t))\dot{q}^i(t) + G(q^i(t), \dot{q}^i(t)) + \tau_a(t) = \tau^i(t) \quad (1)$$

With $t \in [0, t_f]$ denotes the time and $i \in N$ denotes the iteration, $q^i(t) \in R^n$, $\dot{q}^i(t) \in R^n$, $\ddot{q}^i(t) \in R^n$ are the joint position, joint velocity, and joint acceleration variables vector, respectively. $D(q^i(t)) \in R^{n \times n}$ is the inertia matrix, $C(q^i(t), \dot{q}^i(t)) \in R^n$ is the coriolis-centripetal matrix, is the gravity vector plus friction force vector. Bounded unknown disturbances are denoted by $\tau_a(t) \in R^n$ and the control input torque is denoted by $\tau^i(t) \in R^n$. For convenience, a two-link robot manipulator, as shown in Fig.1, is utilized to verify dynamic properties are given in section 4.

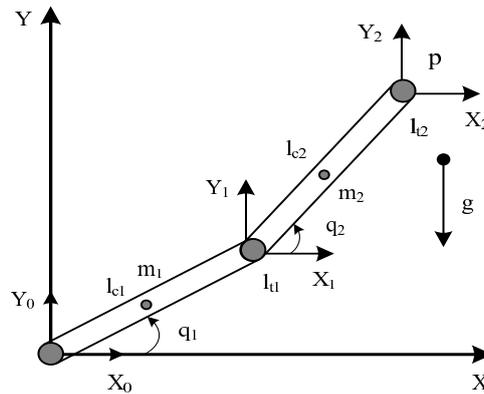


Figure 1: Architecture of two-link robot manipulator.

Usually, the manipulator task specification is given relative to the end-effector. Thus, it is natural to attempt to derive the control algorithm directly in task space, better than in the joint space. Denote the end-effector position and orientation in task space by $x \in R^n$. The task space dynamic can be rewritten as follow:

$$D_x(q^i(t))\ddot{x}^i(t) + C_x(q^i(t), \dot{q}^i(t))\dot{x}^i(t) + G_x(q^i(t), \dot{q}^i(t)) = \tau_a(t) + F_x^i(t) \quad (2)$$

Where

$$D_x(q^i(t)) = J^{-T}(q^i(t))D(q^i(t))J^{-1}(q^i(t))$$

$$C_x(q^i(t), \dot{q}^i(t)) = J^{-T}(q^i(t))(C(q^i(t), \dot{q}^i(t)) - D(q^i(t))J^{-1}(q^i(t))\dot{J}(q^i(t)))J^{-1}(q^i(t))$$

$$G_x(q^i(t), \dot{q}^i(t)) = J^{-T}(q^i(t))G(q^i(t), \dot{q}^i(t)), \quad F_x^i = J^{-T}(q^i(t))\tau^i(t)$$

and $J(q^i(t)) \in R^{n \times n}$ is the configuration-dependent Jacobian matrix, which is assumed as non-singular in the finite work space Ω . The above dynamic equation has the following useful structural properties [11], which can be exploited to facilitate the controller design in the next section.

Property 1: The inertia matrix $D_x(q^i(t))$ is symmetric and positive definite. It is also bounded as a function of q : $m_1 I \leq D_x(q^i(t)) \leq m_2 I$, where $m_1, m_2 > 0$.

Property 2: $\dot{D}_x(q^i(t)) - 2C_x(q^i(t), \dot{q}^i(t))$ is a skew symmetric matrix. Therefore, $y^T[\dot{D}_x(q^i(t)) - 2C_x(q^i(t), \dot{q}^i(t))]y = 0$, where y is a $n \times 1$ nonzero vector.

Assumption 1: The given desired joint trajectory $x_d(t)$ belongs to $C^2[0, t_f]$, where $C^2[0, t_f]$ denotes 2^{nd} -order continuously differentiable functions on $t \in [0, t_f]$.

Assumption 2: Initial condition $x^i(0) = x_d(0)$, $\dot{x}^i(0) = \dot{x}_d(0)$ and for all $i \geq 1$.

3 AILC Design

Linearizing the system (2) along the desired trajectory $x_d(t)$, $\dot{x}_d(t)$, $\ddot{x}_d(t)$ at the i th iterative, we obtain the following linear time-varying system according [8]:

$$D(t)\ddot{e}^i(t) + [C(t) + C_1(t)]\dot{e}^i(t) + R(t)e^i(t) + n(\ddot{e}^i, \dot{e}^i, e^i, t) - \tau_a(t) = S(t) - F^i(t) \quad (3)$$

Where:

$$D(t) = D_x(x_d(t)), \quad C(t) = C_x(x_d(t), \dot{x}_d(t))$$

$$R(t) = \left. \frac{\partial D_x}{\partial x} \right|_{x_d(t)} \ddot{x}_d(t) + \left. \frac{\partial C_x}{\partial x} \right|_{x_d(t), \dot{x}_d(t)} \dot{x}_d(t) + \left. \frac{\partial G_x}{\partial x} \right|_{x_d(t)}$$

$$S(t) = D_x(x_d(t))\ddot{x}_d(t) + C_x(x_d(t), \dot{x}_d(t))\dot{x}_d(t) + G_x(x_d(t)) + \tau_a(t)$$

$$e(t) = x_d(t) - x(t)$$

The term $n(\ddot{e}^i, \dot{e}^i, e^i, t)$ contains the higher order terms $\ddot{e}^i(t)$, $\dot{e}^i(t)$ and $e^i(t)$ and it can be negligible. The control problem is to find a control law so that the end-effector position $x(t)$ can track specific commands $x_d(t)$. We construct controller as follows:

$$F^i = F_e^i + H^i \quad (4)$$

Where the first term $F_e^i = K_p^i(e^i(t)) + K_d^i(\dot{e}^i(t))$ is feedback PD control law with the following gain switching rule in [8]:

$$K_p^{i+1} = \beta(i)K_p^i, \quad K_d^{i+1} = \beta(i)K_d^i, \quad \beta(i+1) > \beta(i), \quad i = 0, 1, 2, \dots, N \quad (5)$$

With K_p^i, K_d^i are the initial proportional and derivative control gain matrices that diagonal positive definite, K_p^{i+1}, K_d^{i+1} , are the control gains of the i th iterative. $\beta(i) > 1$ is the gain switching factor. The gains adaptive law in (5) are used to adjust the PD gains from iterative to iterative. And H^i is the initial predicted feedforward control input to be computed at each iterative by a learning rule. According demonstrated in [8], the feedback PD control law with the gain switching factor in (5) plus the feedforward learning control law with the input force profile, the convergence of system (3) is guaranteed. However, in order to, the trajectory tracking convergence fast in some initial iterative, we cannot increase the switching factor arbitrarily large because actuator forces are limited, especially when the system has modelling errors or nonlinearity. Hence, according [12] to deal this problem, we propose feedforward control input $H^i(t)$ with a learning rule so that $H^i(t)$ converges to $R(t)$ for all $t \in [0, t_f]$ as follow:

$$H^{i+1} = H^i + \alpha F_e^i \quad (6)$$

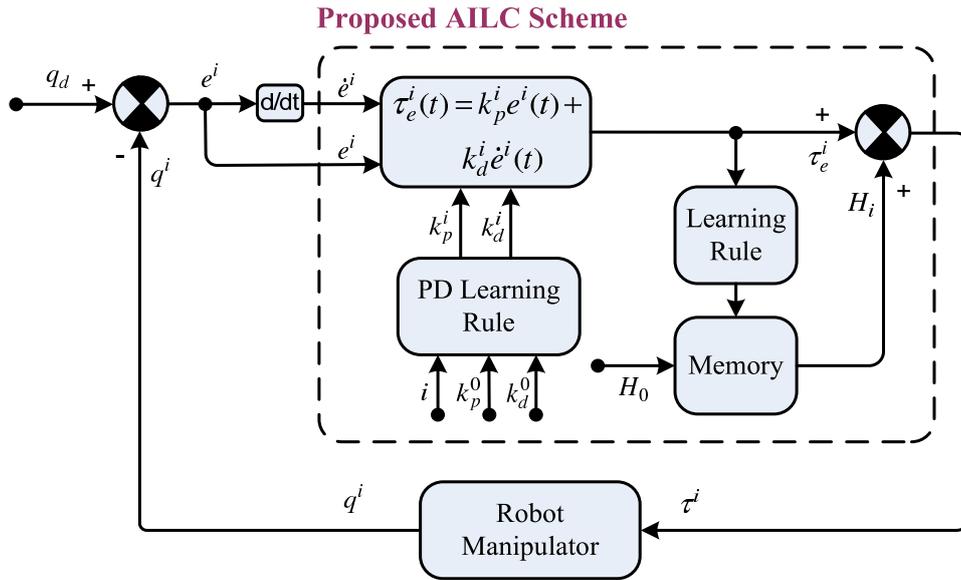


Figure 2: Schematic diagram of the learning control scheme.

At the initial stage of learning, the $H^i(t)$ are set to zero. α is a positive constant often called a training factor. Therefore, for the i th and $(i + 1)$ th iterations, applying the input (4), (6) to system (3), we obtain an error equation as follows:

$$D(t)\ddot{e}^i(t) + [C(t) + C_1(t)]\dot{e}^i(t) + R(t)e^i(t) = S(t) - F_e^i - H^i \quad (7)$$

$$D(t)\ddot{e}^{i+1}(t) + [C(t) + C_1(t)]\dot{e}^{i+1}(t) + R(t)e^{i+1}(t) = S(t) - F_e^{i+1} - H^i - \alpha F_e^i \quad (8)$$

To simplify the proof of stability, let $K_p^i = \alpha K_d^i$ for the initial iteration, and define the filter errors as follow:

$$\tilde{x}^i(t) = \dot{e}^i(t) + a e^i(t) \quad (9)$$

Also, define $\delta\tilde{x}^i = \tilde{x}^{i+1} - \tilde{x}^i$ and $\delta e^i = e^{i+1} - e^i$. Then, from (9):

$$\delta\tilde{x}^i = \delta\dot{e}^i + a\delta e^i \quad (10)$$

From (5)-(9) and (10), one can obtain the following equation:

$$D\delta\ddot{\tilde{x}}^i + (C + C_1 - aD + K_d^{i+1})\delta\dot{\tilde{x}}^i + (R - a(C + C_1 - aD))\delta e^i = -(K_d^{i+1} + (\alpha - 1)K_d^i)\tilde{x}^i \quad (11)$$

The following theorem can be proved.

Theorem: Suppose robot system (2) satisfies property (1, 2) and assumption (1, 2). With the control input (4), the gain switching rule (5) and learning rule (6). The following should hold for all $t \in [0, t_f]$.

$$x^i(t) \rightarrow x_d(t), \quad \dot{x}^i(t) \rightarrow \dot{x}_d(t), \quad i \rightarrow \infty$$

If the controller gains are selected so that the following relationships hold:

$$l_p = \lambda_{\min}((2 - \alpha)K_d^i + 2C_1 - 2aD) > 0 \quad (12)$$

$$l_r = \lambda_{\min}((2 - \alpha)K_d^i + 2C + 2R/a - 2\dot{C}_1/a) > 0 \quad (13)$$

$$l_p l_r \geq \|R/a - (C + C_1 - aD)\|_{\max}^2 \quad (14)$$

Where $\lambda_{\min}(A)$ is the minimum eigenvalue of matrix A , and $\|M\|_{\max} = \max \|M(t)\|$ for $t \in [0, t_f]$. Here, $\|M\|$ represents the Euclidean norm of M .

Proof: We select a performance index $V^i(t)$ as follow:

$$V^i = \int_0^t e^{-\rho\tau} (\tilde{x}^i)^T Q \tilde{x}^i d\tau \geq 0 \quad (15)$$

Thus, $\beta(i) > 1$ according (5) so we have $K_e^{i+1} > K_d^i$ and α is a positive constant, so $Q = K_d^{i+1} + (\alpha - 1)K_d^i > 0$. From the definition of V^j , for the $(i + 1)$ th iteration, one can get:

$$V^{i+1} = \int_0^t e^{-\rho\tau} (\tilde{x}^{i+1})^T Q \tilde{x}^{i+1} d\tau \quad (16)$$

Let $\Delta V^i = V^{i+1} - V^i$ then from (15), (16) and (11), we obtain:

$$\begin{aligned} \Delta V^i &= \int_0^t e^{-\rho\tau} ((\delta \tilde{x}^i)^T Q \delta \tilde{x}^i + 2(\delta \tilde{x}^i)^T Q \dot{\tilde{x}}^i) d\tau \\ &= \int_0^t e^{-\rho\tau} ((\delta \tilde{x}^i)^T Q \delta \tilde{x}^i + 2(\delta \tilde{x}^i)^T Q \dot{\tilde{x}}^i) d\tau \\ &= \int_0^t e^{-\rho\tau} (\delta \tilde{x}^i)^T Q \delta \tilde{x}^i d\tau - 2 \int_0^t e^{-\rho\tau} (\delta \tilde{x}^i)^T D \delta \dot{\tilde{x}}^i d\tau \\ &\quad - 2 \int_0^t e^{-\rho\tau} (\delta \tilde{x}^i)^T ((C + C_1 - aD + K_d^{i+1}) \delta \tilde{x}^i + (R - a(C + C_1 - aD))) \delta e^i d\tau \end{aligned} \quad (17)$$

Applying the partial integration, from assumption 2, and property 2, we have:

$$\begin{aligned} \int_0^t e^{-\rho\tau} (\delta \tilde{x}^i)^T D \delta \dot{\tilde{x}}^i d\tau &= e^{-\rho\tau} (\delta \tilde{x}^i)^T D (\delta \tilde{x}^i) \Big|_0^t - \int_0^t (e^{-\rho\tau} (\delta \tilde{x}^i)^T D) \rho \delta \tilde{x}^i d\tau \\ &= e^{-\rho\tau} (\delta \tilde{x}^i)^T (t) D (t) \delta \tilde{x}^i (t) + \rho \int_0^t e^{-\rho\tau} (\delta \tilde{x}^i)^T D \delta \tilde{x}^i d\tau \\ &\quad - \int_0^t e^{-\rho\tau} (\delta \tilde{x}^i)^T D \delta \dot{\tilde{x}}^i d\tau - 2 \int_0^t e^{-\rho\tau} (\delta \tilde{x}^i)^T C \delta \tilde{x}^i d\tau \end{aligned} \quad (18)$$

Substituting (18) into (17), and $Q = K_d^{i+1} + (\alpha - 1)K_d^i$ yields:

$$\begin{aligned} \Delta V^i &= -e^{-\rho\tau} (\delta \tilde{x}^i)^T (t) D (t) \delta \tilde{x}^i (t) - \rho \int_0^t e^{-\rho\tau} (\delta \tilde{x}^i)^T D \delta \tilde{x}^i d\tau \\ &\quad - 2 \int_0^t e^{-\rho\tau} (\delta \tilde{x}^i)^T (R - a(C + C_1 - aD)) \delta e^i d\tau \\ &\quad - \int_0^t e^{-\rho\tau} (\delta \tilde{x}^i)^T (K_d^{i+1} - (\alpha - 1)K_d^i + 2C_1 - 2aD) \delta \tilde{x}^i d\tau \end{aligned} \quad (19)$$

From (5), we have:

$$\int_0^t e^{-\rho\tau} (\delta\tilde{x}^i)^T K_d^{i+1} \delta\tilde{x}^i d\tau = \beta(i+1) \int_0^t e^{-\rho\tau} (\delta\tilde{x}^i)^T K_d^i \delta\tilde{x}^i d\tau \geq \int_0^t e^{-\rho\tau} (\delta\tilde{x}^i)^T K_d^i \delta\tilde{x}^i d\tau \quad (20)$$

Substituting (10) into (19) and noticing (20), we obtain:

$$\begin{aligned} \Delta V^i \leq & -e^{-\rho\tau} (\delta\tilde{x}^i)^T (t) D(t) \delta\tilde{x}^i(t) - \rho \int_0^t e^{-\rho\tau} (\delta\tilde{x}^i)^T D \delta\tilde{x}^i d\tau \\ & - \int_0^t e^{-\rho\tau} (\delta\dot{e}^i)^T ((2-\alpha)K_d^i + 2C_1 - 2aD) \delta\dot{e}^i d\tau \\ & - 2a \int_0^t e^{-\rho\tau} (\delta e^i)^T ((2-\alpha)K_d^i + 2C_1 - 2aD) \delta e^i d\tau \\ & - 2 \int_0^t e^{-\rho\tau} (\delta\dot{e}^i)^T (R - a(C + C_1 - aD)) \delta\dot{e}^i d\tau \\ & - a^2 \int_0^t e^{-\rho\tau} (\delta e^i)^T ((2-\alpha)K_d^i + 2C_1 - 2aD) \delta e^i d\tau \\ & - 2a \int_0^t e^{-\rho\tau} (\delta e^i)^T (R - a(C + C_1 - aD)) \delta e^i d\tau \end{aligned} \quad (21)$$

Applying the partial integration again gives:

$$\begin{aligned} \int_0^t e^{-\rho\tau} (\delta\dot{e}^i)^T ((2-\alpha)K_d^i + 2C_1 - 2aD) \delta\dot{e}^i d\tau &= e^{-\rho\tau} (\delta e^i)^T ((2-\alpha)K_d^i + 2C_1 - 2aD) (\delta e^i) \Big|_0^t \\ &+ \rho \int_0^t e^{-\rho\tau} (\delta e^i)^T ((2-\alpha)K_d^i + 2C_1 - 2aD) \delta e^i d\tau \\ &- \int_0^t e^{-\rho\tau} (\delta\dot{e}^i)^T ((2-\alpha)K_d^i + 2C_1 - 2aD) \delta e^i d\tau \\ &+ 2 \int_0^t e^{-\rho\tau} (\delta e^i)^T (a\dot{D} - \dot{C}_1) \delta e^i d\tau \end{aligned} \quad (22)$$

Therefore

$$\begin{aligned}
\Delta V^i &\leq -e^{-\rho\tau}(\delta\tilde{x}^i)^T D\delta\tilde{x}^i - \rho \int_0^t e^{-\rho\tau}(\delta\tilde{x}^i)^T D\delta\tilde{x}^i d\tau \\
&\quad - ae^{-\rho\tau}(\delta e^i)^T((2-\alpha)K_d^i + 2C_1 - 2aD)\delta e^i \\
&\quad - \rho a \int_0^t e^{-\rho\tau}(\delta e^i)^T((2-\alpha)K_d^i + 2C_1 - 2aD)\delta e^i d\tau - \int_0^t e^{-\rho\tau} w d\tau \\
&\leq -e^{-\rho\tau}(\delta\tilde{x}^i)^T D\delta\tilde{x}^i - ae^{-\rho\tau}(\delta e^i)^T l_p \delta e^i - \rho \int_0^t e^{-\rho\tau}(\delta\tilde{x}^i)^T D\delta\tilde{x}^i d\tau \\
&\quad - \rho a \int_0^t e^{-\rho\tau}(\delta e^i)^T l_p \delta e^i d\tau - \int_0^t e^{-\rho\tau} w d\tau
\end{aligned} \tag{23}$$

Where

$$\begin{aligned}
w &= (\delta e^i)^T((2-\alpha)K_d^i + 2C_1 - 2aD)\delta e^i + 2a(\delta e^i)^T(R/a - (C + C_1 - aD))\delta e^i \\
&\quad + a^2(\delta e^i)^T((2-\alpha)K_d^i + 2R/a + 2C - 2C_1/a)\delta e^i
\end{aligned} \tag{24}$$

Let $P = R/a - (C + C_1 - aD)$. Then from (12) and (13), we obtain

$$w \geq l_p \|\delta e\|^2 + 2a\delta e^T P \delta e + a^2 l_r \|\delta e\|^2 \tag{25}$$

Applying the Cauchy-Schwartz inequality, we obtain:

$$\delta e^T P \delta e \geq -\|\delta e\| \|P\|_{\max} \|\delta e\| \tag{26}$$

From (12)-(14)

$$\begin{aligned}
w &= l_p \|\delta e\|^2 - 2a \|\delta e\| \|P\|_{\max} \|\delta e\| + a^2 l_r \|\delta e\|^2 \\
&= l_p \left(\|\delta e\| - \frac{a}{l_p} \|P\|_{\max} \|\delta e\| \right)^2 + a^2 \left(l_p - \frac{1}{l_r} \|P\|_{\max}^2 \right) \|\delta e\|^2 \\
&\geq 0
\end{aligned} \tag{27}$$

According property 1 and (27), based on (23), it can be ensured that $\Delta V^i \leq 0$, therefore $V^{i+1} \leq V^i$. From the definition, K_d^i is a positive definite matrix. From the definition of V^i , $V^i \geq 0$ and V^i is bounded. As a result, $y^i(t) \rightarrow \infty$ when $i \rightarrow \infty$. Because $e^i(t)$ and $\dot{e}^i(t)$ are two independent variables, and a is a positive constant. Thus, if $i \rightarrow \infty$, $e^i(t) \rightarrow 0$ and $\dot{e}^i(t) \rightarrow 0$ for $t \in [0, t_f]$. Finally, the following conclusions hold for $t \in [0, t_f]$.

$$x^i(t) \rightarrow x_d(t), \quad \dot{x}^i(t) \rightarrow \dot{x}_d(t), \quad i \rightarrow \infty$$

From the above analysis it can be seen that the adaptive PD control method can guarantee that the tracking errors converge arbitrarily close to zero as the number of iterations increases. The following case studies based on simulation will demonstrate this conclusion.

4 Numerical Simulation

A two-link robot manipulator as shown in Fig.1 is utilized in this paper to verify the effectiveness of the proposed control scheme. The dynamic equation of the robot manipulator is adopted in [13].

$$D(q) = \begin{bmatrix} m_1 + m_2 + 2m_3 \cos(q_2) & m_2 + m_3 \cos(q_2) \\ m_2 + m_3 \cos(q_2) & m_2 \end{bmatrix}$$

$$C(q, \dot{q}) = \begin{bmatrix} -m_3 \dot{q}_2 \sin(q_2) & -m_3 (\dot{q}_1 + \dot{q}_2) \sin(q_2) \\ m_3 \dot{q}_1 \sin(q_2) & 0 \end{bmatrix}$$

$$G(q) = \begin{bmatrix} m_4 g \cos(q_1) + m_5 g \cos(q_1 + q_2) \\ m_5 g \cos(q_1 + q_2) \end{bmatrix}$$

and m_i are the parameters of interest given by $M = P + p_l L$ with

$$M = [m_1 \quad m_2 \quad m_3 \quad m_4 \quad m_5]^T$$

$$P = [p_1 \quad p_2 \quad p_3 \quad p_4 \quad p_5]^T$$

$$L = [l_1^2 \quad l_2^2 \quad l_1 l_2 \quad l_1 \quad l_2]^T$$

and p_l is the payload, $l_1 = 1(m)$ and $l_2 = 1(m)$ are the lengths of link 1 and link 2, respectively, P is the parameter vector of the robot itself. $g = 9.8 \text{ m/s}^2$. The jacobian matrix is known as:

$$J(q) = \begin{bmatrix} -l_1 \sin(q_1) - l_2 \sin(q_1 + q_2) & -l_2 \sin(q_1 + q_2) \\ l_1 \cos(q_1) + l_2 \cos(q_1 + q_2) & l_2 \cos(q_1 + q_2) \end{bmatrix}$$

For the convenience of the simulation, the nominal parameters of the robotic system are given as:

$$P = [1.66 \quad 0.42 \quad 0.63 \quad 3.75 \quad 1.25]^T \text{ kg.m}^2$$

The desired reference trajectories in the Cartesian space are:

$$x_{d1} = 1.0 + 0.2 \cos(\pi t), \quad x_{d2} = 1.0 + 0.2 \sin(\pi t).$$

Which represent a circle of radius $0.2m$ and its centres is located at $(x_1, x_2) = (1.0, 1.0)m$. The robot is initially rested with its end-effector positioned at the center of the circle. With initial condition are $x_1(0) = x_{d1}(0) = 1.2$, $x_2(0) = x_{d2}(0) = 1.0$, $\dot{x}_1(0) = \dot{x}_{d1}(0) = 1.0$, $\dot{x}_2(0) = \dot{x}_{d2}(0) = 1.2$ and $t = [0, 2]$.

The most important parameters that effect the control performance of the robotic system are the external disturbance $\tau_a(t)$, the friction term $f(\dot{q})$, in simulation, payload variation situation and external disturbance situation occurring at fifth the iteration are considered. The payload variation situation is that $p_l = 0(kg)$ from first the iteration to fourth the iteration, and then it was put on $p_l = 3$ after the fifth iteration. The disturbance situation is that external forces are injected into the robotic system, and their shapes are expressed as follows:

$$\tau_a(t) = [5 \sin(5t) \quad 5 \sin(5t)]^T \quad (28)$$

In addition, friction forces are also considered in this simulation and given as:

$$f(\dot{q}) = [2\dot{q}_1 + 0.8\text{sgn}(\dot{q}_1) \quad 4\dot{q}_2 + 0.1\text{sgn}(\dot{q}_2)]^T \quad (29)$$

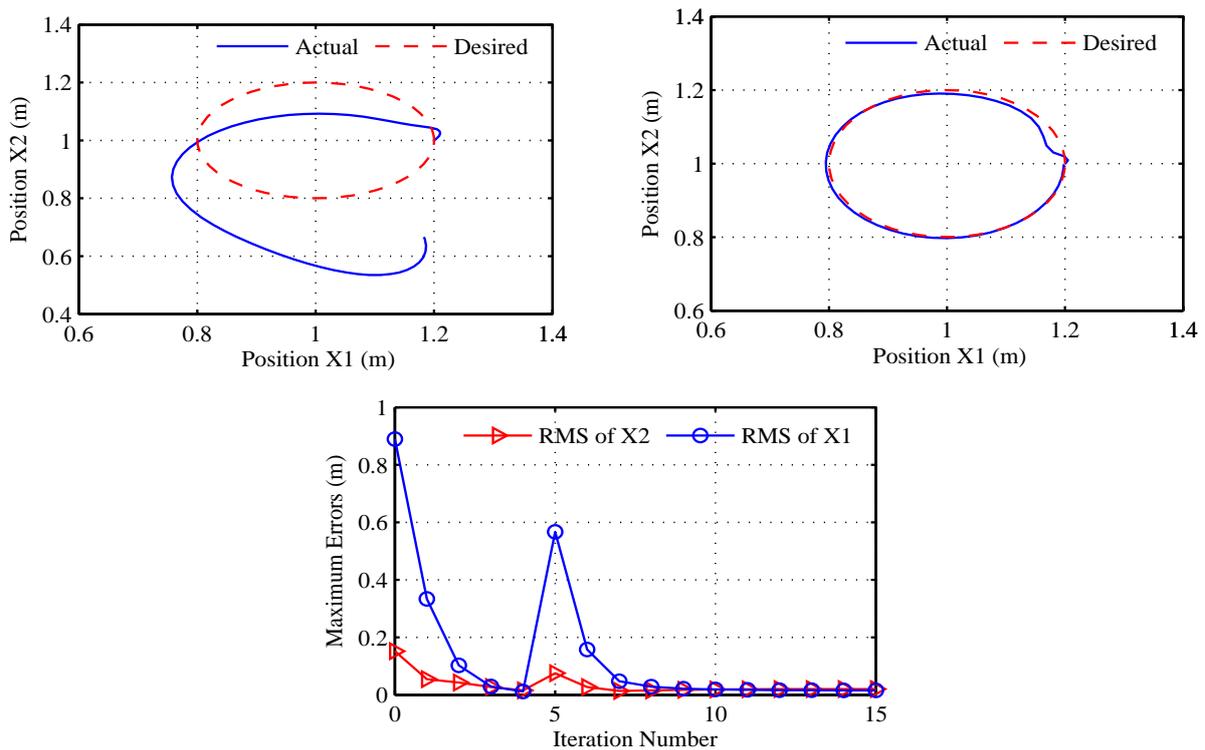


Figure 3: Reference tracking and actual trajectory for end-effector at different iterative under AILC is adopted in [11]. (a) Position tracking of end-effector at first iteration, (b) Position tracking of end-effector after fifteenth iteration and (c) profile of rms position errors payload and disturbance variation after the fifth iteration.

In order to exhibit the superior control performance of AILC system, two extra control systems including an adaptive switching learning PD control system (ASL-PD) is represented in [8], an iterative learning control (ILC) is represented in [12].

In three simulation situations, The PD control gain was set to be the same as [8]:

$$K_p^i = K_d^i = \text{diag}\{30, 30\}, \quad K_p^{i+1} = 2iK_p^i, \quad K_d^{i+1} = 2iK_d^i, \quad \beta = 0.75$$

The simulated results of ILC system, the responses of end-effector position at first and fifteenth iteration and tracking error from iteration to iteration to be depicted Fig. 3(a), (b) and (c) respectively. From the simulated results, we can be seen that the tracking performance was not acceptable because the errors were too large compared to ASL-PD and proposed controllers, especially at the fifth iteration poor tracking errors responses are resulted due to the occurrence of payload variation situation and external disturbance. In the ASL-PD system are depicted in [8]. The end-effector position responses, tracking error from iteration to iteration to be depicted in Fig. 4(a), 4(b), and 4(c), respectively. The end-effector tracking performance is obvious under the occurrence of payload variation and external disturbance. The convergence rate increased greatly compared with the ILC control method.

Now, the AILC system depicted in Fig. 2 is applied to control the robot manipulator for comparison. The simulated results of end-effector position responses and tracking error from iteration to iteration are depicted in Fig. 5(a), 5(b) and 5(c), respectively. Table 1 shows tracking performance of proposed system from the initial iteration to fifteenth are obvious. Therefore, the comparison of their method and our method demonstrated a bit fast convergence rate with the proposed control method. Specially, control

	Iterative	1	5	10	15
ILC	$\max e_1^i (m)$	0.0822	0.0773	0.0168	0.0140
	$\max e_2^i (m)$	0.1998	0.5634	0.0130	0.0105
ASL-PD	$\max e_1^i (m)$	0.0398	0.0076	0.0012	0.0010
	$\max e_2^i (m)$	0.0646	0.0593	0.0008	0.0006
AILC	$\max e_1^i (m)$	0.0377	0.0076	0.0011	0.0009
	$\max e_2^i (m)$	0.1741	0.0590	0.0008	0.0006

Table 1: Maximum tracking errors from iteration to iteration.

	Iterative	1	5	10	15
ASL-PD	$\max \tau_1^i (N.m)$	90	930	3330	7230
	$\max \tau_2^i (N.m)$	51.45	531.6	1903	3618
AILC	$\max \tau_1^i (N.m)$	82.50	772.5	2647	5647
	$\max \tau_2^i (N.m)$	47.14	441.6	1513	3228

Table 2: Maximum control forces from iteration to iteration.

torques of proposed system is much smaller than the ASL-PD controller. Detail results are depicted in table 2.

5 Conclusions

This paper has successfully implemented an AILC scheme to control the position of end-effector in task space for achieving desired position control. All the system dynamics may be unknown. A new control method is a combination of advantages some other method into a hybrid one as explained above. By using Lyapunov theorem, the asymptotic convergence of the closed-loop control system can be ensured whether or not the uncertainties occur. Simulation results of a two link robot manipulator in task space via various existing control methods including ASL-PD and ILC control were also applied in this paper to compare and display the manipulative performance of the proposed control system. According to the result as depict in Figs. 3-5 and table 1-2, the desired position tracking and tracking errors response of the AILC scheme decrease with the increase of the iteration number under wide range of payload and external disturbance.

The main of the paper is to construct a simple scheme, easy implementation, fast convergence. Especially, in this paper is applied not only to control the position of the end-effector, but also the force exerted by the end-effector on the object. By designing the control law in task space, force control can be easily formulated.

Acknowledgment

This work was supported by the National Natural Science Foundation of China (60775047; 60835004), the National High Technology Research and Development Program of China (863 Program) (2007AA04Z244; 2008AA04Z214).

The authors would like to thank the associate editor and the reviewers for their valuable comments.

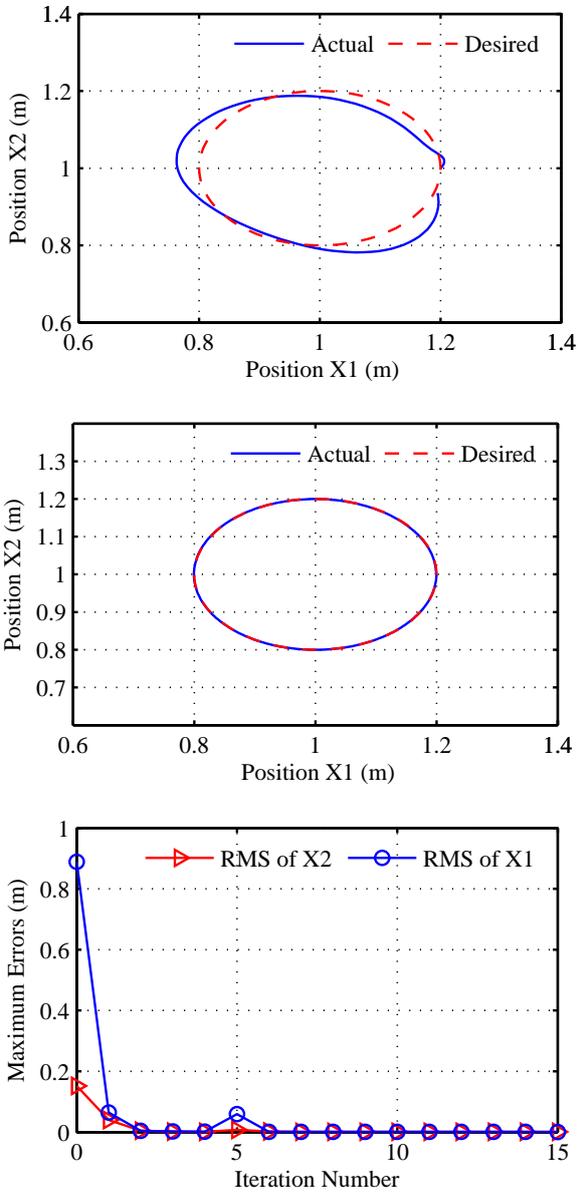


Figure 4: Reference tracking and actual trajectory for end-effector at different iterative under ASL-PD is adopted in [6]. (a) Position tracking of end-effector at first iteration, (b) Position tracking of end-effector after fifteenth iteration and (c) profile of rms position errors payload and disturbance variation after the fifth iteration.

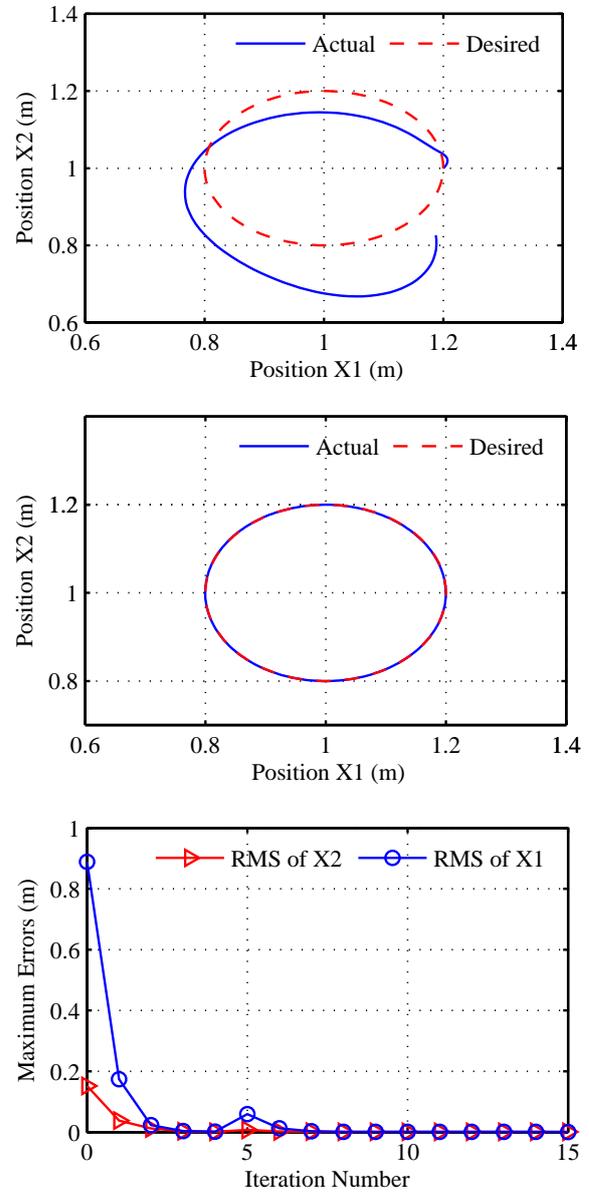


Figure 5: Reference tracking and actual trajectory for end-effector at different iterative under AILC controller. (a) Position tracking of end-effector at first iteration, (b) Position tracking of end-effector after fifteenth iteration and (c) profile of rms position errors payload and disturbance variation after the fifth iteration.

Bibliography

- [1] P. Bondi, G. Cacalini and L. Gambardella, On the iterative learning Control theory for robot manipulators, *IEEE J. On Robotics and Automation*, 4:14-22, 1984.
- [2] S. Arimoto, S. Kawamura and F. Miyazaki, Bettering operation of dynamic system by learning: A new control theory for servomechanism or mechatronics systems, *Proceedings of 23rd. CDC*, 1064-1069, 1984.
- [3] J.J. Craig, Adaptive control of mechanical manipulators, *Addison-Wesley, New York*, 1988.
- [4] Hyo-Sung Ahn, YangQuan Chen, Kevin L. Moore, Iterative learning control: Brief Survey and Categorization, *IEEE Trans. Ind. Sys*, 6:1099-1121, 2007.
- [5] Dong-II Kim, Sungkwun Kim An iterative learning control method with application for CNC machine tools, *IEEE Trans. Ind. App*, 32(1):66-72, 1996.
- [6] K.L. Moore, Iterative learning control for deterministic systems, *Advances Industrial Control, New York, Springer-Verlag*, 1993.
- [7] Tayebi A, Adaptive iterative learning control for robot manipulators, *Automatica*, 40:1195-1203, 2004.
- [8] Ouyang P R, Zhang W J, Gupta M M, An adaptive switching learning control method for trajectory tracking of robot manipulators, *it Mechatronics*, 16:51-61, 2006.
- [9] Choi JY, Lee JS, Adaptive iterative learning control of uncertain robotic systems, *IEE Proc Contr Theory*, 147(4):217-223, 2000.
- [10] B.S. Chen, H.J. Uang, and C.S. Tseng, Robust tracking enhancement of robot systems including motor dynamics: A fuzzy-based dynamic game approach, *IEEE Trans. Fuzzy syst*, 11(4):538-552, 1998.
- [11] Craig JJ. Introduction to robotics: mechanics and control. Reading, MA: *Addison-Wesley*, 1986.
- [12] T.Y Kuc, K. Nam, J.S. Lee, An iterative learning control of robot Manipulators, *IEEE Trans Robot Automat.*, 7(6): 835-842, 1991.
- [13] S.S. Ge, C.C. Hang, Adaptive neural network control of robot manipulators in task space, *IEEE Trans. Ind. Elec.*, 44(6):746-752, 1997.