1 DESIGN OF AN ITERATIVE LEARNING CONTROLLER FOR A CLASS OF LINEAR DYNAMIC SYSTEMS WITH TIME-DELAY AND INITIAL STATE ERROR

Kwang-Hyun Park*, Zeungnam Bien*, and Dong-Hwan Hwang**

*Department of Electrical Engineering
Korea Advanced Institute of Science and Technology
373-1 Kusong-dong, Yusong-gu
Taejon, 305-701, Korea
**Department of Electronics Engineering, College of Engineering
Chungnam National University
220 Kung-dong, Yusong-gu
Taejon, 305-764, Korea

akaii@ctrgate.kaist.ac.kr, zbien@ee.kaist.ac.kr, dhhwang@hanbat.chungnam.ac.kr

Abstract: In this chapter, it is first observed that, when a typical iterative learning control(ILC) algorithm is applied to a class of linear dynamic systems with time-delay, erratic estimation of delay time may cause the control input to diverge. In order to resolve such a difficulty due to uncertainty of the delay time, a new ILC algorithm is proposed, in which the holding mechanism is adopted to hold the control input at a constant value for the duration of the delay time uncertainty. As a consequence, the output of the system tracks a given desired trajectory at discrete time-points which are spaced by the size of the uncertainty of delay time. In addition to the consideration of the delay time, the effect of the initial state error is also studied with a modification of the proposed ILC algorithm. Numerical examples are given to show the effectiveness of the proposed algorithms.

1.1 INTRODUCTION

In recent years, the demand for high precision control methodology is increasing to improve the performance of the automation systems such as petro-chemical processes, industrial robot manipulators, NC machine-based manufacturing systems, Magneto-Optical Disk Drives(MODD), and so on. Among tasks utilizing these systems/machines, many tasks, such as batch job of certain chemical processes, spray painting, and arc-welding, are repetitive and require a controller which can track the given whole trajectory completely in a specified time interval.

Due to inaccuracy in modeling and/or uncertainty of some system parameters, a conventional feedback control system adopting PID control, state feedback control, or optimal path tracking control(Makowski and Neustadt, 1974) is found to be unsatisfactory in its performance, and also, well-known advanced techniques such as adaptive control may not work(Lee *et al.*, 1984; Koivo and Guo, 1983). Especially, unsatisfactory tracking performance is much more prominent for the nonminimum phase processes or time-delay systems. As a method to overcome the limitation of the conventional controllers, the iterative learning control(ILC) method was proposed by Arimoto et al. (Arimoto *et al.*, 1984), and has been further developed by many researchers(Bondi *et al.*, 1988; Hwang *et al.*, 1993; Togai and Yamano, 1985; Bien and Huh, 1989) since then. The ILC algorithm is generally expressed in the following form :

$$u_{k+1}(t) = u_k(t) + f\left(e_k(\cdot)\right)(t), 0 \leq t \leq T \tag{1.1}$$

where $u_k(t)$ is the control input and f is a functional of error function, $e_k(t)$, $0 \leq t \leq T$, between the actual output and the desired output at the k-th iteration(Lee and Bien, 1996). The functional f can be designed in various ways according to the objective of the control(Arimoto *et al.*, 1984; Lee and Bien, 1996; Kawamura *et al.*, 1988; Arimoto, 1985).

Most of the results up to now on the ILC are for the dynamic systems with no time-delay. On the other hand, in many batch chemical processes as shown in Fig. 1.1, the time-delay effect cannot be ignored. For example, consider the system in Fig. 1.1 in which the flow rates of cooling water and steam are the control inputs, and the temparature of the reactor is the output. In this system, there exists time-delay between the control input and the plant output. Suppose the temperature should track a given desired trajectory. As a means for complete tracking of the desired temperature output trajectory, the ILC method can be adopted. If the general form given in Eqn. (1.1) is to be applied to such a system with time-delay, the form of the controller may be modified as in the following Eqn. (1.2), taken into consideration of the delay time.

$$u_{k+1}(t) = u_k(t) + f\left(e_k(\cdot)\right)(t + \tau_e) \tag{1.2}$$

Here, τ_e is an estimated delay time. In Eqn. (1.2), the control input $u_{k+1}(t)$ is updated by input $u_k(t)$ and value of the functional $f\left(e_k(\cdot)\right)$ at time $t + \tau_e$. If the estimated delay time τ_e is different from the actual dealy time, however, the intput $u_{k+1}(t)$ is to be updated from incorrect response error, and in this case, there is no guarantee that the algorithm is convergent. That is, a typical ILC algorithm with some naive modification can be applied to a system with time-delay only in case the delay time is known exactly. Otherwise, the control

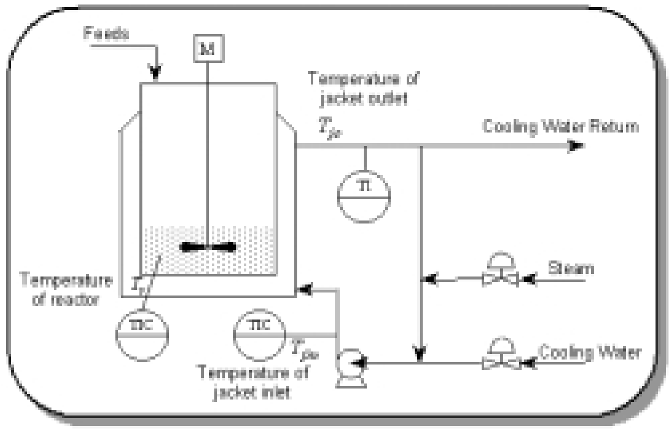


Figure 1.1 Process schematics of graft ABS (Acrylonitrile-Butadiene-Styrene) polymerization reactor(Yi, 1993)

input may be divergent due to uncertainty of the delay time. Hideg(Hideg, 1996) investigated the possibility of divergence of an ILC for a plant with time-delay in the frequency domain via some computer simulations. As an example of quick remedy, he presented an ILC algorithm with a windowing technique applied.

On the other hand, most ILC algorithms have assumed that the initial state value of the plant is equal to that of the desired trajectory for perfect tracking. Since it is impossible to set the initial state value of the plant to that of desired trajectory exactly in real application, a new algorithm to overcome this assumption and/or performance analysis on the initial state error have been recognized as open problems. Lee and Bien(Lee and Bien, 1996) showed that the trajcetory errors could be estimated in terms of the initial error and the parameters of the ILC algorithm when the PD-type ILC algorithm was applied.

In this chapter, a new ILC algorithm for a class of linear dynamic systems with time-delay is proposed using the holding mechanism, and the effect of the initial state error is investigated.

This chapter is organized as follows. In section 1.2, a new ILC algorithm for linear systems with time-delay is proposed, and the condition for convergence is presented. In section 1.3, the effect of the initial state error is discussed. In section 1.4, numerical examples are presented to show the effectiveness of the proposed algorithms, and concluding remarks follow in section 1.5.

In the sequel, for the n-dimensional Euclidean space R^n, $\|x\|$ denotes the Euclidean norm of a vector $x = (x_1, \cdots x_n)^T$. For a matrix A, $\|A\|$ denotes its induced matrix norm. $\|x\|_\infty$ denotes the ∞-norm defined by

$$\|x\|_\infty = \max_{1 \le i \le n} |x_i|$$

and for $n \times r$ matrix A with elements a_{ij},

$$\|A\|_\infty = \max_{1 \le i \le n} \sum_{j=1}^{r} |a_{ij}|.$$

As in the notation $u_k(t)$, the subscript k is employed to denote the iteration number.

1.2 ITERATIVE LEARNING CONTROL LAW FOR LINEAR SYSTEMS WITH TIME-DELAY

In this section, it is shown that the output traking performance is obtained by using holding mechanism.

Consider the linear time invariant system with time-delay described by

$$\begin{aligned} \dot{x}(t) &= Ax(t) + Bu(t - \tau') \\ y(t) &= Cx(t) \end{aligned} \tag{1.3}$$

where $x \in R^n, u \in R^r$ and $y \in R^q$ denote the state, the input and the output, respectively, and τ' is the actual delay time. A, B and C are matrices with appropriate dimensions. Let $y_d(t)$ be the desired output trajectory and $x_d(t)$ be the corresponding state trajectory. Assume that they are continuously differentiable on $[0, T]$.

Suppose the delay time τ' is estimated in terms of some lower and upper bounds, τ_1 and τ_2 so that $\tau_1 \le \tau' \le \tau_2$ as shown in Fig. 1.2. The size of uncertainty in delay time can be defined as $h = \tau_2 - \tau_1$. Now, divide the time interval $[\tau_2, T]$ forward from time τ_2 by h and the time interval $[0, \tau_1]$ backward from time τ_1 also by h. The time interval $[0, \tau_1]$ cannot be an integer multiple of h, so some initial remainder ξ exists, which is smaller than h as shown in Fig. 1.2. A similar statement applies for $[\tau_2, T]$. Let m be the index that represent the sequence of divided intervals, and let $t = mh + \xi$ to denote each discrete point as shown in Fig. 1.2. Let M be the maximum among the numbers m that satisfy $mh + \xi \le T$. Then we may write $m \in \{0, 1, \cdots, M\}$. Here, if T' is defined by $T' = Mh + \xi$, then the divided time interval in Fig. 1.2 is the same as the time interval which $[0, T]$ is divided by h from T' backward. Let d be the number that satisfies $dh + \xi = \tau_2$; then the actual delay time τ' can be represented by Eqn. (1.4).

$$\tau' = (d-1)h + \tau + \xi \tag{1.4}$$

where $\tau, 0 \le \tau < h$, is unknown value in τ'.

Now, we reformulate the problem of the ILC for a class of linear dynamic systems with uncertain time-delay as a discrete-time tracking problem.

Problem 1.2.1 *Suppose a desired trajectory $y_d(t)$, $t \in [0, T]$, is given and the initial state at each iteration is the same as the desired initial state, i.e., $x_k(0) = x_d(0)$ for $k = 0, 1, 2, \cdots$. The problem is to find a control input $u(t), 0 \le$*

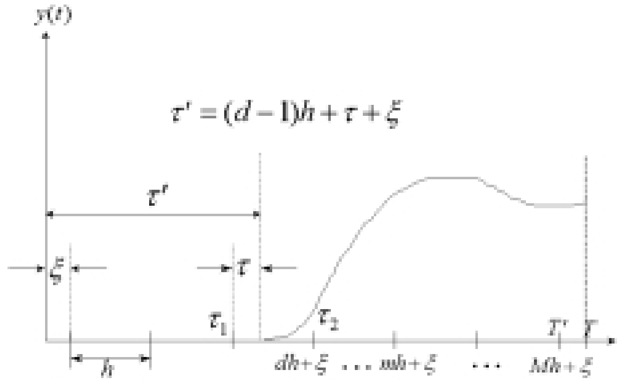


Figure 1.2 Representation of delay time

$t \leq T$, such that the output of the given linear dynamic system (1.3) matches the desired trajectory $y_d(t)$ exactly at the discrete points $y_d(mh + dh + \xi), m \in \{0, 1, \cdots, M - d\}$.

The following ILC algorithm is proposed as a solution for the Problem 1.2.1.

$$\begin{aligned} u_{k+1}(t) &= u_k(mh) + \Gamma e_k(mh + dh + \xi), \\ &\quad \forall t \in [mh, mh + h), m \in \{0, 1, \cdots, M - d\} \end{aligned} \tag{1.5}$$

where

$$e_k(mh + dh + \xi) = y_d(mh + dh + \xi) - y_k(mh + dh + \xi).$$

Before showing the convergence of the ILC algorithm (1.5), the following norm is introduced for a sequence, which is utilized in the proof of main result on convergence.

Definition 1.2.1 *For each $m = 0, 1, \cdots, M - d$, let $h(mh)$ be a vector of finite dimension. For a given $\lambda > 1$ and $\alpha > 1$, define $\|h(\cdot)\|_\lambda^\alpha$ to be*

$$\|h(\cdot)\|_\lambda^\alpha = \sup_{0 \leq m \leq M-d} \alpha^{-\lambda m} \|h(mh)\|_\infty.$$

$\|h(\cdot)\|_\lambda^\alpha$ is called a modified λ-norm.

It is clear that the norm defined in Definition 1.2.1 is equivalent to the sup-norm (Lee and Bien, 1997) since $\|h(\cdot)\|_\lambda^\alpha \leq \sup_{0 \leq m \leq M-d} \|h(mh)\|_\infty \leq \alpha^{M-d} \|h(\cdot)\|_\lambda^\alpha$.

Now, the convergence of the ILC algorithm (1.5) will be shown.

Theorem 1.2.1 *Suppose that the update law Eqn. (1.5) is applied to the system Eqn. (1.3) and the initial state at each iteration is the same as the desired initial state, i.e., $x_k(0) = x_d(0)$ for $k = 0, 1, 2, \cdots$. If*

$$\|I - \Gamma C \int_0^{h-\tau} e^{A\sigma} d\sigma B\|_\infty \leq \rho < 1 \tag{1.6}$$

then,

$$\lim_{k\to\infty} y_k(mh+dh+\xi) = y_d(mh+dh+\xi), \forall m \in \{0,1,\cdots,M-d\}.$$

Proof : From Eqn. (1.3), the state value at $t = mh + dh + \xi$ is represented by the state value at $t = mh + dh - h + \xi$ and input $u(t)$ as follows:

$$\begin{aligned} x_k(mh+dh+\xi) &= e^{Ah} x_k(mh+dh-h+\xi) \\ &\quad + \int_{mh+(d-1)h+\xi}^{mh+dh+\xi} e^{A(mh+dh+\xi-\sigma)} B u_k(\sigma-\tau') d\sigma \\ &= e^{Ah} x_k(mh+dh-h+\xi) \\ &\quad + \int_{mh-\tau}^{mh+h-\tau} e^{A(mh+h-\tau-\sigma)} B u_k(\sigma) d\sigma. \end{aligned}$$

The input $u(t)$ is constant over the time interval h. The integral above can now be separated into two parts : one where $u_k(t) = u_k(mh-h), mh-\tau \le t < mh$; and the other one where $u_k(t) = u_k(mh), mh \le t < mh + h - \tau$. This gives

$$\begin{aligned} x_k(mh+dh+\xi) &= e^{Ah} x_k(mh+dh-h+\xi) \\ &\quad + \int_0^{\tau} e^{A(h-\tau)} e^{A\sigma'} d\sigma' B u_k(mh-h) \\ &\quad + \int_0^{h-\tau} e^{A\sigma'} d\sigma' B u_k(mh). \end{aligned} \tag{1.7}$$

For simplicity of presentation, introduce the following notations:

$$\begin{aligned} \Phi(t) &= e^{At} \\ \Theta(t) &= \int_0^t e^{A\sigma} B d\sigma \\ z_k(mh+dh+\xi) &= \begin{bmatrix} x_k(mh+dh+\xi) \\ u_k(mh) \end{bmatrix} \\ F &= \begin{bmatrix} \Phi(h) & \Phi(h-\tau)\Theta(\tau) \\ 0 & 0 \end{bmatrix} \\ G &= \begin{bmatrix} \Theta(h-\tau) \\ I \end{bmatrix} \\ H &= \begin{bmatrix} C & 0 \end{bmatrix}. \end{aligned}$$

And then, rearrange Eqn. (1.7) in matrix form to obtain:

$$\begin{aligned} z_k(mh+dh+\xi) &= F z_k(mh+dh-h+\xi) + G u_k(mh) \\ y_k(mh+dh+\xi) &= H z_k(mh+dh+\xi). \end{aligned} \tag{1.8}$$

From Eqn. (1.8), $y_k(mh+dh+\xi)$ is represented as follows.

$$y_k(mh+dh+\xi) = HF^{m+1} \begin{bmatrix} e^{A(dh-h+\xi)} x_k(0) \\ 0 \end{bmatrix} + H \sum_{j=0}^{m} F^{m-j} G u_k(jh). \tag{1.9}$$

Now, let $u_d(mh)$ be a control input such that

$$y_d(mh+dh+\xi) = HF^{m+1}\begin{bmatrix} e^{A(dh-h+\xi)}x_d(0) \\ 0 \end{bmatrix} + H\sum_{j=0}^{m} F^{m-j}Gu_d(jh) \tag{1.10}$$

and define

$$\Delta u_k(mh) = u_d(mh) - u_k(mh).$$

Then it follows from Eqns. (1.5), (1.9), and Eqn. (1.10) that

$$\begin{aligned} \Delta u_{k+1}(mh) &= \Delta u_k(mh) - \Gamma e_k(mh+dh+\xi) \\ &= \left(I - \Gamma C\int_0^{h-\tau} e^{A\sigma}d\sigma B\right)\Delta u_k(mh) \\ &\quad -\Gamma HF\sum_{j=0}^{m-1} F^{m-j}G\Delta u_k(jh). \end{aligned} \tag{1.11}$$

Taking the norm $\|\cdot\|_\infty$ on both sides of Eqn. (1.11), we have

$$\|\Delta u_{k+1}(mh)\|_\infty \le \rho\|\Delta u_k(mh)\|_\infty + k\sum_{j=0}^{m-1}\left(\|F\|_\infty\right)^{m-j}\|\Delta u_k(jh)\|_\infty \tag{1.12}$$

where

$$k = \|\Gamma HAF\|_\infty\|G\|_\infty.$$

Choosing a real number α such that $\alpha > \max\{1, \|F\|_\infty\}$ and multiplying $\alpha^{-\lambda m}$ on both sides of Eqn. (1.12), we have

$$\begin{aligned} \|\Delta u_{k+1}(\cdot)\|_\lambda^\alpha &\le \rho\|\Delta u_k(\cdot)\|_\lambda^\alpha + k\sup_{0\le m\le M-d}\alpha^{-\lambda m}\sum_{j=0}^{m-1}(\|F\|_\infty)^{m-j}\alpha^{\lambda j}\|\Delta u_k(\cdot)\|_\lambda^\alpha \\ &< \rho\|\Delta u_k(\cdot)\|_\lambda^\alpha \\ &\quad + k\sup_{0\le m\le M-d}\alpha^{-(\lambda-1)m}\frac{1-\alpha^{(\lambda-1)m}}{1-\alpha^{\lambda-1}}\|\Delta u_k(\cdot)\|_\lambda^\alpha \\ &\le \rho\|\Delta u_k(\cdot)\|_\lambda^\alpha + k\frac{1-\alpha^{-(\lambda-1)(M-d)}}{\alpha^{\lambda-1}-1}\|\Delta u_k(\cdot)\|_\lambda^\alpha. \end{aligned} \tag{1.13}$$

Since $0 \le \rho < 1$ by assumption, it is possible to choose λ sufficiently large so that

$$\rho_0 = \rho + k\frac{1-\alpha^{-(\lambda-1)(M-d)}}{\alpha^{\lambda-1}-1} < 1.$$

From Eqn. (1.13)

$$\lim_{k\to\infty}\|\Delta u_k(\cdot)\|_\lambda^\alpha = 0.$$

From Eqns. (1.9) and (1.10), the following can be concluded.

$$\lim_{k\to\infty} y_k(mh + dh + \xi) = y_d(mh + dh + \xi), \forall m \in \{0, 1, \cdots, M - d\}$$

This completes the proof.

Theorem 1.2.1 implies that, if the bound h of estimation error of the delay time is known and the sufficient condition of Theorem 1.2.1 is satisfied, then the output trajectory exactly track the discrete points of the desired output trajectory $y_d(mh + \xi)$, which are spaced by the size of the uncertainty h.

Note that h can be considered as a measure of uncertainty of delay time. Thus, the smaller the h is, the better the estimation results. In implementation of the iterative learning controller using computer or μ-processor, the desired trajectory is mostly discretized, and stored in memory. Therefore, if h is smaller than the sampling interval, the output can exactly track the discretized trajectory. That is, if the bound of estimation error of the delay time is known to be within some prespecified bound, then a satisfied performance is achived in real applications.

As shown in the ILC algorithm (1.5) and Theorem 1.2.1, the convergence of the output is not guaranteed if the control input is not held at constant value over the time interval h.

If time interval $[0, T]$ is divided by h from $t = 0$ forward, i.e., a relation of $t = mh$ is satisfied at each discrete point, then Eqn. (1.5) is changed into the following equations.

Case 1 : $\tau + \xi < h$

$$u_{k+1}(t) = \begin{cases} u_k(0) + \Gamma e_k(dh), t \in [0, h - \xi] \\ u_k(mh) + \Gamma e_k(mh + dh), \\ \qquad \forall t \in [mh, mh + h), m \in \{1, 2, \cdots, M - d\} \end{cases}$$

Case 2 : $\tau + \xi \geq h$

$$u_{k+1}(t) = \begin{cases} u_k(0) + \Gamma e_k(dh + h), t \in [0, h - \xi] \\ u_k(mh) + \Gamma e_k(mh + dh + h), \\ \qquad \forall t \in [mh, mh + h), m \in \{1, 2, \cdots, M - d\} \end{cases}$$

That is, the control input is held over the time interval $[0, h - \xi]$ and over h after $t = h - \xi$. However, if τ is not known, it can not be determined which algorithm is applied.

It is remarked that the above result can be applied to linear systems with delays in the output or in the connection of subsystems. To be specific, let us consider the following system with output-delay.

$$\begin{aligned} \dot{x}(t) &= Ax(t) + Bu(t) \\ y(t) &= Cx(t - \tau') \end{aligned}$$

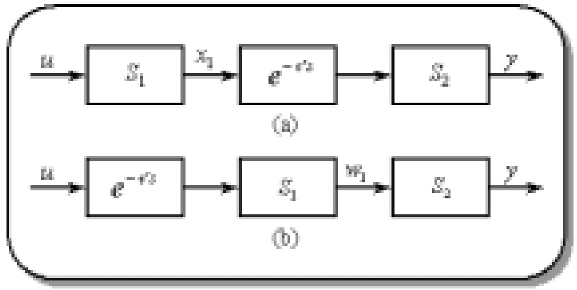


Figure 1.3 The time-delay system linked by cascade form

(a) The system with inner time-delay
(b) Rearrangement of (a)

In this case, if we change the state variable by $w(t) = x(t - \tau')$, then

$$\begin{aligned} \dot{w}(t) &= Aw(t) + Bu(t - \tau') \\ y(t) &= Cw(t). \end{aligned}$$

Since this is a form similar to the one in Eqn. (1.3), the case of a system with output-delay can be handled by the result of Theorem 1.2.1.

In case that there is time-delay between two subsystems linked by a cascaded form as shown in Fig. 1.3 (a), the system dynamics can be represented by the following Eqn. (1.14).

$$\begin{aligned} \dot{x}_1(t) &= A_1 x_1(t) + B_1 u(t) \\ \dot{x}_2(t) &= A_2 x_2(t) + B_2 C_1 x_1(t - \tau') \\ y(t) &= C_2 x_2(t) \end{aligned} \tag{1.14}$$

If we change the state variable by $w_1(t) = x_1(t - \tau')$, then it is rearranged as shown in Fig. 1.3 (b).

$$\begin{aligned} \dot{w}_1(t) &= A_1 w_1(t) + B_1 u(t - \tau') \\ \dot{x}_2(t) &= A_2 x_2(t) + B_2 C_1 w_1(t) \\ y(t) &= C_2 x_2(t). \end{aligned}$$

This form also can be taken care of by the result of Theorem 1.2.1.

1.3 THE EFFECT OF INITIAL STATE ERROR FOR THE SYSTEM WITH TIME-DELAY

In this section, the effect of initial state error is discussed when a modified ILC algorithm is applied. Consider the following problem.

Problem 1.3.1 *Suppose a desired trajectory $y_d(t)$, $t \in [0, T]$, is given and there exists an initial state error at each iteration, i.e., $x_k(0) = x_0 \neq x_d(0), k = 0, 1, 2, \cdots$. The problem is to find the effect of the initial state error when the following ILC algorithm Eqn. (1.15) is applied to the system Eqn. (1.3).*

$$\begin{aligned} u_{k+1}(t) &= u_k(mh) + \Gamma\left[e_k(mh + dh + \xi) - Re_k(mh + dh - h + \xi)\right], \\ & \forall t \in [mh, mh + h), m \in \{0, 1, \cdots, M - d\}. \qquad (1.15) \end{aligned}$$

Here

$$e_k(mh + dh + \xi) = y_d(mh + dh + \xi) - y_k(mh + dh + \xi)$$

and R is a $q \times q$ constant matrix.

Theorem 1.3.1 shows the effect of the initial state error when the ILC algorithm Eqn. (1.15) is applied. This law (Eqn. (1.15)) is a modified version of Eqn. (1.5).

Theorem 1.3.1 *Suppose that the update law Eqn. (1.15) is applied to the system Eqn. (1.3) and the initial state at each iteration can be different from the desired initial state, i.e., $x_k(0) = x_0 \neq x_d(0)$ for $k = 0, 1, 2, \cdots$. If*

$$\|I - \Gamma C \int_0^{h-\tau} e^{A\sigma} d\sigma B\|_\infty \leq \rho < 1, \qquad (1.16)$$

then

$$\begin{aligned} \lim_{k\to\infty} y_k(mh + dh + \xi) &= yd(mh + dh + \xi) + R^m C e^{A(dh+\xi)}(x_0 - x_d(0)), \\ & \forall m \in \{0, 1, \cdots, M - d\}. \qquad (1.17) \end{aligned}$$

Proof : Let $u_a(mh)$ be the control input that satisfies Eqn. (1.18).

$$y_a(mh + dh + \xi) = HF^{m+1}\begin{bmatrix} e^{A(dh-h+\xi)}x_0 \\ 0 \end{bmatrix} + H\sum_{j=0}^{m} F^{m-j}Gu_a(jh) \quad (1.18)$$

where

$$y_a(mh + dh + \xi) = y_d(mh + dh + \xi) + R^m C e^{A(dh+\xi)}(x_0 - x_d(0)).$$

Define

$$\begin{aligned} \Delta u_k(mh) &= u_a(mh) - u_k(mh) \\ \Delta y_k(mh) &= y_a(mh) - y_k(mh). \end{aligned}$$

Since

$$R^m C e^{A(dh+\xi)}(x_0 - x_d(0)) - RR^{m-1}Ce^{A(dh+\xi)}(x_0 - x_d(0)) = 0,$$

it follows from Eqns. (1.9), (1.15), and (1.18) that

$$\begin{aligned}\Delta u_{k+1}(mh) &= \Delta u_k(mh) - \Gamma\left[e_k(mh+dh+\xi) - Re_k(mh+dh-h+\xi)\right] \\ &= \Delta u_k(mh) - \Gamma\left[\Delta y_k(mh+dh+\xi) - R\Delta y_k(mh+dh-h+\xi)\right] \\ &= \left(I - \Gamma C\int_0^{h-\tau} e^{A\sigma}d\sigma B\right)\Delta u_k(mh) \qquad (1.19) \\ &\quad -\Gamma(HF-RH)\sum_{j=0}^{m-1} F^{m-j-1}G\Delta u_k(jh). \qquad (1.20)\end{aligned}$$

Taking the norm $\|\cdot\|_\infty$ on both sides of Eqn. (1.20), we have

$$\begin{aligned}\|\Delta u_{k+1}(mh)\|_\infty &\le \rho\|\Delta u_k(mh)\|_\infty \\ &\quad + k'\sum_{j=0}^{m-1}\left(\|F\|_\infty\right)^{m-j}\|\Delta u_k(jh)\|_\infty \qquad (1.21)\end{aligned}$$

where

$$k' = \|\Gamma(HF-RH)\|_\infty\|G\|_\infty\left(\|F\|_\infty\right)^{-1}.$$

Choosing a real number α such that $\alpha > \max\{1, \|F\|_\infty\}$ and multiplying $\alpha^{-\lambda m}$ on both sides of Eqn. (1.21), we have

$$\begin{aligned}\|\Delta u_{k+1}(\cdot)\|_\lambda^\alpha &\le \rho\|\Delta u_k(\cdot)\|_\lambda^\alpha + k'\sup_{0\le m\le M-d}\alpha^{-\lambda m}\sum_{j=0}^{m-1}(\|F\|_\infty)^{m-j}\alpha^{\lambda j}\|\Delta u_k(\cdot)\|_\lambda^\alpha \\ &< \rho\|\Delta u_k(\cdot)\|_\lambda^\alpha \\ &\quad + k'\sup_{0\le m\le M-d}\alpha^{-(\lambda-1)m}\frac{1-\alpha^{(\lambda-1)m}}{1-\alpha^{\lambda-1}}\|\Delta u_k(\cdot)\|_\lambda^\alpha \\ &\le \rho\|\Delta u_k(\cdot)\|_\lambda^\alpha + k'\frac{1-\alpha^{-(\lambda-1)(M-d)}}{\alpha^{\lambda-1}-1}\|\Delta u_k(\cdot)\|_\lambda^\alpha. \qquad (1.22)\end{aligned}$$

Since $0 \le \rho < 1$ by assumption, it is possible to choose λ sufficiently large so that

$$\rho_0 = \rho + k'\frac{1-\alpha^{-(\lambda-1)(M-d)}}{\alpha^{\lambda-1}-1} < 1.$$

From Eqn. (1.22)

$$\lim_{k\to\infty}\|\Delta u_k(\cdot)\|_\lambda^\alpha = 0.$$

From Eqns. (1.9) and (1.18), the following can be concluded.

$$\begin{aligned}\lim_{k\to\infty} y_k(mh+dh+\xi) &= y_d(mh+dh+\xi) + R^m Ce^{A(dh+\xi)}(x_0 - x_d(0)), \\ &\quad \forall m \in \{0, 1, \cdots, M-d\}\end{aligned}$$

This completes the proof.

The modified ILC algorithm Eqn. (1.15) has the form that the one step shifted error term is added to the ILC algorithm Eqn. (1.5). Theorem 1.3.1 tells that, if the initial state trajectory error is the same at each iteration, the output trajectory at the discrete points can be estimated from the desired output trajectory, the initial state error, and the learning controller parameter R as stated in (Lee and Bien, 1996). From Eqn. (1.17), it is obvious that if R is chosen such that all eigenvalues of R lies inside unit disk, the error decreases as time increases. Such a decreasing property is similar to the property that the error decreases by e^{Rt} in (Lee and Bien, 1996).

1.4 NUMERICAL EXAMPLES

To illustrate effectiveness of the proposed algorithms, two examples are presented.

Example 1 : Consider the following linear time-invariant dynamic system.

$$\begin{aligned} \dot{x} &= \begin{bmatrix} 0 & 1 \\ -1 & -2 \end{bmatrix} x(t) + \begin{bmatrix} 0 \\ 1 \end{bmatrix} u(t - 0.225) \\ y(t) &= \begin{bmatrix} 0 & 1 \end{bmatrix} x(t). \end{aligned} \tag{1.23}$$

The desired output trajectory is given as

$$y_d(t) = \begin{cases} 0, & 0 \leq t < 0.3 \\ 4(t-0.3) - 4(t-0.3)^2, & 0.3 \leq t \leq 1. \end{cases}$$

To confirm the undesirable phenomenon when a typical ILC algorithm is applied, suppose the D-type ILC algorithm(Arimoto *et al.*, 1984) is applied to the system (1.23) with some naive modificaion as in the following Eqn. (1.24) using estimated delay time 0.22.

$$u_{k+1}(t) = u_k(t) + \Gamma \dot{e_k}(t + 0.22) \tag{1.24}$$

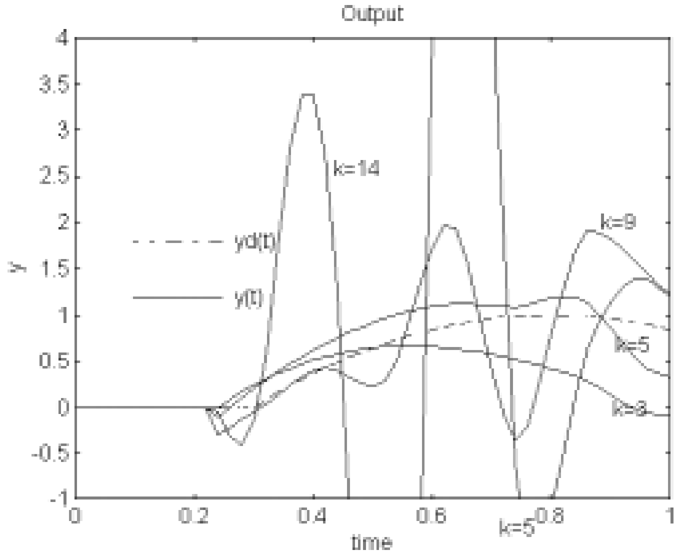
Then we find that as the iteration number k increases the control input becomes divergent as shown in Fig. 1.4.

Now, let us apply the proposed algorithm. For this, we consider two cases in which the delay time is estimated differently.

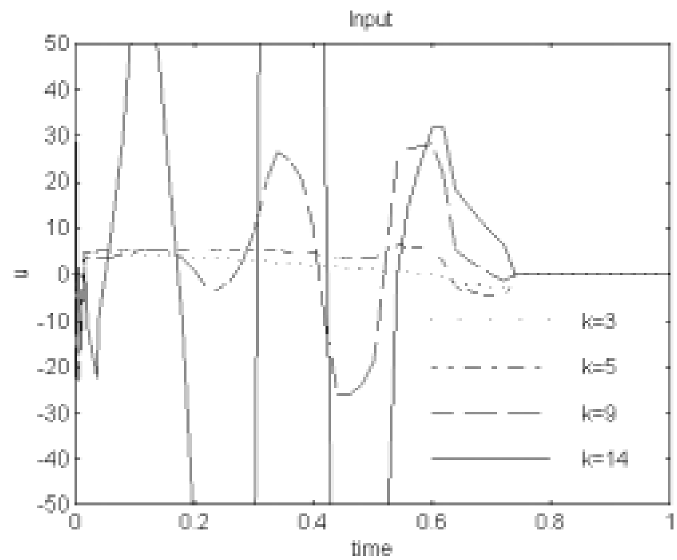
Suppose first the lower bound of delay time is 0.22 and the upper bound is 0.27, i.e., $0.22 \leq \tau' \leq 0.27$. The size of uncertainty h is 0.05. Then the delay time τ' can be represented as

$$\begin{aligned} \tau' &= (d-1)h + \tau + \xi \\ &= (5-1) * 0.05 + \tau + 0.02, \; 0 \leq \tau < 0.05. \end{aligned}$$

For another case, let the lower bound of the delay time be 0.22 and the upper bound be 0.32, i.e., $0.22 \leq \tau' \leq 0.32$. The size of uncertainty h is 0.1. Then



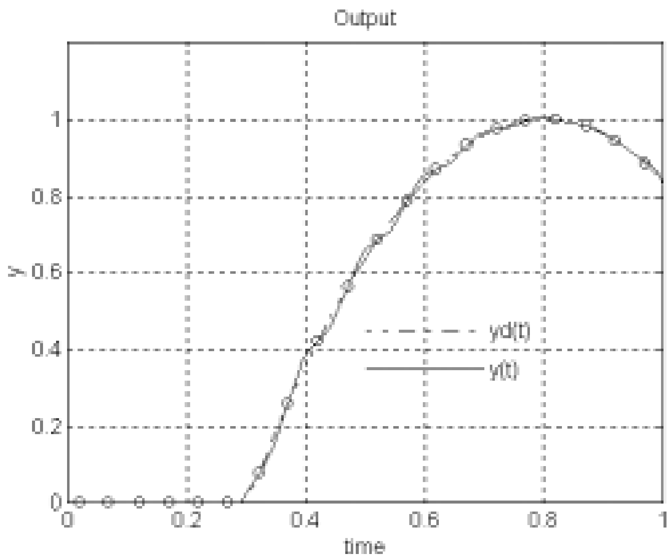
(a)



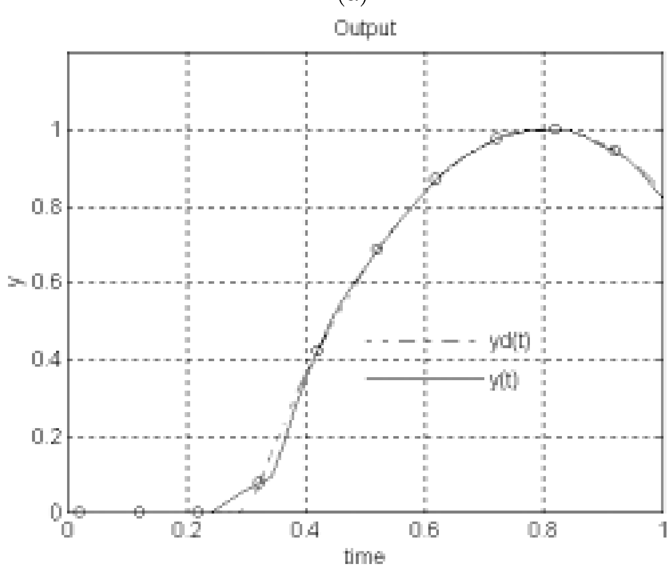
(b)

Figure 1.4 the diverged output $y(t)$ and input $u(t)$

(a) plant output
(b) control input



(a)



(b)

Figure 1.5 the desired output $y_d(t)$ and the plant output $y(t)$

(a) the case estimated by $0.22 < \tau' < 0.27$
(b) the case estimated by $0.22 < \tau' < 0.32$

the delay time τ' can be represented as

$$\tau' = (3-1)*0.1 + \tau + 0.02,\ 0 \leq \tau < 0.1.$$

The best choices of Γ from Theorem 1.2.1 are $\left(C\int_0^{h-\tau} e^{A\sigma} d\sigma B\right)^{-1} = 1/0.0241$ and $1/0.067$. $C\int_0^{h-\tau} e^{A\sigma} d\sigma B$ have been guessed 0.03 and 0.08 in each case assuming 20% uncertainty. Γ can be $1/0.03$ and $1/0.08$ from the guessed values. Fig. 1.5 (a) and (b) shows the convergent output trajectories after the 30th iteration at each case. The output $y(t)$ perfectly tracks discrete points of $y_d(t)$ at $t = mh + \xi, m \in \{d, d+1, \cdots, M\}$. Comparing (a) and (b), less h shows better performance.

Example 2 : Consider the following linear time-invariant dynamic system.

$$\begin{aligned} \dot{x} &= \begin{bmatrix} 0 & 1 \\ -1 & -2 \end{bmatrix} x(t) + \begin{bmatrix} 0 \\ 1 \end{bmatrix} u(t-0.225) \\ y(t) &= \begin{bmatrix} 0 & 1 \end{bmatrix} x(t). \end{aligned} \tag{1.25}$$

The desired output trajectory and initial state value are given as

$$\begin{aligned} y_d(t) &= \begin{cases} 0, & 0 \leq t < 0.25 \\ 4(t-0.25) - 4(t-0.25)^2, & 0.25 \leq t \leq 1. \end{cases} \\ x_k(0) &= \begin{bmatrix} 1 \\ 1 \end{bmatrix}. \end{aligned}$$

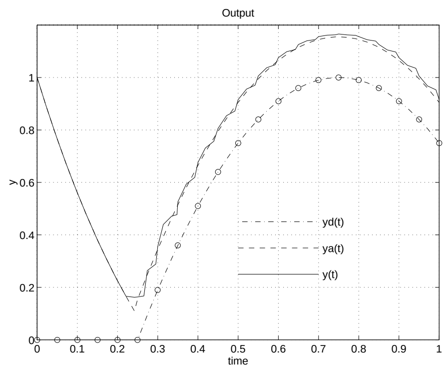
Suppose the lower bound of delay time is 0.2 and the upper bound is 0.25, i.e., $0.2 \leq \tau' \leq 0.25$. The size of uncertainty h is 0.05. Then the delay time τ' can be represented as

$$\begin{aligned} \tau' &= (d-1)h + \tau + \xi \\ &= (5-1)*0.05 + \tau,\ 0 \leq \tau < 0.05. \end{aligned}$$

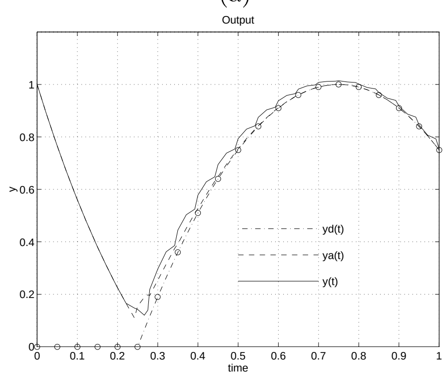
The best choices of Γ from Theorem 1.3.1 is $\left(C\int_0^{h-\tau} e^{A\sigma} d\sigma B\right)^{-1} = 1/0.0241$. $C\int_0^{h-\tau} e^{A\sigma} d\sigma B$ have been guessed 0.03 assuming 20% uncertainty. Γ can be $1/0.03$ from the guessed values. Fig. 1.6 (a), (b), and (c) show the convergent output trajectories after the 60th iteration at each case($R = 1.0$, $R = 0.8$, and $R = 0.5$), respectively. The output $y(t)$ perfectly tracks discrete points of $y_a(t)$ at $t = mh + dh + \xi, m \in \{0, 1, \cdots, M-d\}$. Comparing (a), (b), and (c), less R shows fast decrease.

1.5 CONCLUDING REMARKS

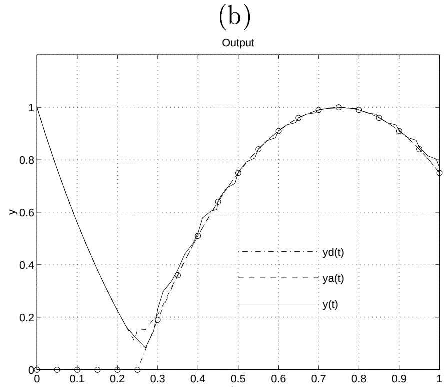
In this paper, the problem caused by estimation error of delay time was investigated when a typical ILC algorithm was applied, and a new ILC algorithm was proposed. If the new ILC algorithm is applied, the output of the plant can be convergent even if delay time estimation error exists, and tracks the



(a)



(b)



(c)

Figure 1.6 the desired output $y_d(t)$ and the plant output $y(t)$
(a) the case $R = 1.0$
(b) the case $R = 0.8$
(c) the case $R = 0.5$

discrete points of a given desired output trajectory. The effect of the initial state trajectory error is also discussed.

For the system with state-delay as described by Eqn. (1.26), the effect of delay time causes high complexity, and it is not known yet if any ILC algorithm is applicable for trajectory tracking.

$$\begin{aligned} \dot{x}(t) &= Ax(t-\tau') + Bu(t) \\ y(t) &= Cx(t) \end{aligned} \tag{1.26}$$

On the other hand, since the performance of the ILC mainly depends on the error bounds, an analysis of inter-sample behavior is important. Designing an ILC for a state-delay system and a rigorous analysis based on sampled data technique should be a challenging problem.

References

Arimoto, S. (1985) Mathematical theory of learning with applications to robot control. *Proc. of 4th Yale Workshop on Applications of Adaptive Systems*, New Haven, Connecticut, USA, 379–388.

Arimoto, S., Kawamura, S., and Miyazaki, F. (1984). Bettering operation of robots by learning. *Journal of Robotic Systems*, 1:123–140.

Bien, Z. and Huh, K. M. (1989). Higher-order iterative learning control algorithm. *IEE Proceedings - Part D*, 136:105–112.

Bondi, P., Casalino, G., and Gambardella, L. (1988). On the iterative learning control theory for robotic manipulators. *IEEE Journal of Robotics and Automation*, 4(1):14–21.

Hideg, L. M. (1996). Stability and convergence issues in iterative learning control : Part ii. *International Symposium on Intelligent Control*.

Hwang, D. H., Kim, B. K., and Bien, Z. (1993). Decentralized iterative learning control methods for large-scale linear dynamical systems. *International Journal of Systems Science*, 24(12):2239–2254.

Kawamura, S., Miyazaki, and Arimoto, S. (1988). Realization of robot motion based on a learning method. *IEEE Transaction on Systems, Man, and Cybernetics*, 18(1):126–134.

Koivo, A. J. and Guo, T. H. (1983). Adaptive linear controller for robotic manipulator. *IEEE Transaction on Automatic Control*, AC-28(2).

Lee, C. S. G., Chung, M. J., and Lee, B. H. (1984). An approach of adaptive control for robot manipulators. *Journal of Robotic Systems*, 1(1).

Lee, H. S. and Bien, Z. (1996). Study on robustness of iterative learning control with non-zero initial error. *International Journal of Control*, 64(3):345–359.

Lee, H. S. and Bien, Z. (1997). A Note on Convergence Property of Iterative Learning Control with Respect to Sup Norm. *Automatica*, 33(8):1591–1593.

Makowski, K. and Neustadt, L. W. (1974). Optimal problems with mixed control-phase variable equality and inequality constraints. *SIAM Journal of Control*, 12:184–228.

Togai, M. and Yamano, O. (1985). Analysis and design of an optimal learning control scheme for industrial robots : A discrete system approach. *Proc. of 24th IEEE Conference on Decision and Control*, 1399–1404.

Yi, S. (1993). *A Study on Batch Polymerization Reactor Control*. PhD thesis, Korea Advanced Institute of Science and Technology, Department of Chemical Engineering.