'88 KACC 1988. 10. 21~22

A Second-Order Iterative Learning Control Method

Zeungnam Bien and *Kyung Moo Huh*

Dept. of Electrical Eng., KAIST

P.O.Box 150, Cheongryang, Seoul, 130-650, Korea

Abstract--- For the trajectory control of dynamic systems with unidentified parameters, a second-order iterative learning control method is presented. In contrast to other known methods, the proposed learning control scheme can utilize more than one error history contained in the trajectories generated at prior iterations. A convergency proof is given and it is also shown that the convergence speed can be improved in compared to conventional methods. Examples are provided to show effectiveness of the algorithm, and, via simulation, it is demonstrated that the method yields a good performance even in the presence of disturbances.

1. Introduction

As a means of controlling a plant whose dynamics is not fully known in advance or changes in an unknown but slow manner, the approach of adaptive control has been intensively studied for more than a decade. There have been reported many examples of successful applications of the method, especially when the objective of control is regulation function. The class of systems for which the adaptive control method such as MRAC or STR can be applied is rather limited, however, and the scheme can be very complicated or inappropriate as a real-time control method for complex systems such as robot manipulators. In particular, the adaptive control systems can be helpless for the case when the output of the plant needs to be tightly controlled all the time along a prespecified path as in the task of robot trajectory tracking ; the method does not guarantee that the output at initial or intermediary time points remains within a specified error bound of tolerance.

Recently, a novel control method, called "learning control", is getting increasing attention as an alternative for controlling uncertain dynamic systems in a simple manner. It is a recursive on-line control method that relies on less calculation and requires less a priori knowledge about the system dynamics.

First of its kind to note is the learning control method proposed by Uchiyama [2] and elaborated as a more formal theory by Arimoto and his associates[3]. The idea is to use a simple algorithm repetitively to an unknown plant to achieve a perfect tracking. The algorithm is of the form

$$u_{k+1}(t) = u_k(t) + \Gamma \dot{e}_k(t) \tag{1}$$

where $u_k(t)$ is the control input at the k-th iteration and $e_k(t)$ denotes the error between the actual system output $y_k(t)$ as a response to $u_k(t)$ and the desired output $y_d(t)$,i.e.

$$e_k(t) = y_d(t) - y_k(t) .$$

Γ is a gain factor. When this simple algorithm is applied repeatedly for a class of not-necessarily time-invariant unknown systems, it was proved that the time derivative of the output converges to the time derivative of the desired trajectory of the system. This method was reported applicable, for example, for speed control of robotic manipulators with repetitive tasks. The original proposition was novel and attractive in concept : the algorithm is simple to calculate and apparently independent of the plant dynamics. This iterative control algorithm is called a kind of learning algorithm in the sense that the quality of control is improved in repetition and eventually the system learns to track the commanded trajectory.

However, this learning control in its initial form suffers from a few drawbacks such as :

- the algorithm utilizes a noncausal operator(i.e. differentiator) and
- the positional convergence is not guaranted : instead, the velocity $\dot{y}(t)$ was proved to converge to the desired velocity $\dot{y}_d(t)$.

The method has been refined and expanded in many ways. For example, Arimoto and his associates[5] showed the positional convergence in L_2-sense. Also, Craig in [6] proposed a modified algorithm for better performance of the form

$$u_{k+1}(t) = u_k(t) + p(t)^* e_k(t) \tag{2}$$

where p(t) is a given time function denoting the implulse response of a linear filter and * denotes the convolution integral operation.

As a means of avoiding noncausal operation such as the time differentiation in (1), Gu and Loh[7] proposed to use difference operation for digital realization in such a way that the algorithm becomes

$$u_{k+1}(i) = u_k(i) + \sum_{j=1} a_j e_k(i+j-1) \tag{3}$$

Recently, Oh,Bien and Suh[8] combined the concept of the above learning control algorithm with the structure of an adaptive system to introduce an algorithm of the form

$$u_{k+1}(t) = u_k(t) + [\tilde{B}'_k \tilde{B}_k]^{-1} \tilde{B}'_k (\dot{e}_k(t) - \tilde{A}_k e_k) \qquad (4)$$

where $\{\tilde{A}_k, \tilde{B}_k\}$ is a linear system model that is available at k-th iteration, and is updated by using some parameter estimator. We may also find a discrete-time approach of learning control theory in [9] and a robust learning algorithm presented in Furuta and Yamakita[10].

Note that, in all the algorithms in the above discussion, the control $u_{k+1}(t)$ is synthesized from only one single history data pair $(u_k(t), e_k(t))$.

In this paper, we propose that $u_{k+1}(t)$ be synthesized from multiple past history data pairs $(u_k(t), e_k(t))$, $(u_{k-1}(t), e_{k-1}(t))$ to enhance the convergence performance of the algorithm and at the same time make the system to be more robust to disturbances causing loss of information. For the sake of distinction, we shall call the iterative learning algorithm to be of first order if only one data pair $(u_k(t), e_k(t)), 0 \le t \le T$ is used to generate $u_{k+1}(t), 0 \le t \le T$ as in (1) while the algorithm is called a second-order algorithm if 2 consecutive pairs $(u_k(t), e_k(t)), (u_{k-1}(t), e_{k-1}(t))$ are used for synthesizing $u_{k+1}(t)$.

In the sequel, the notation convention is as follows :

For a matrix A, the transpose of A is denoted as A'.
For a finite dimensional vector x, $\|x\|$ denotes the Euclidean norm.
Also the following vector norm and the matrix norm are defined

$$\|f\|_\infty = \max_{1 \le i \le r} |f^{(i)}| \qquad when \ f = (f^{(1)}, \ldots, f^{(r)})$$

$$\|G\|_\infty = \max_{1 \le i \le r} \{\sum_{j=1}^{r} |g^{(ij)}|\} \qquad when \ G = (g^{(ij)})$$

2. Second-Order Iterative Learning Control Method for Linear Time-Invariant Systems

To present the theory more efficiently, we shall first examine in this section a 2nd-order iterative learning control algorithm for linear time-invariant systems.

Consider the linear time-invariant dynamical system

$$\dot{x}(t) = A\, x(t) + B\, u(t), \qquad x(0) = \xi^0$$

$$y(t) = C\, x(t) + D\, u(t) \qquad (5)$$

where x and u denote an nx1 state vector and a px1 control vector, respectively, and y is an mx1 output vector. Suppose the matrices A, B, C and D are not known.

Let $y_d(t), 0 \le t \le T$ be given the desired output trajectory and $\varepsilon^* > 0$ be given as a tolerance bound. We wish to find a control function $u(t), 0 \le t \le T$ such that the corresponding output trajectory $y(t)$ of the linear system in eq.(5) satisfies

$$E(y(t)) = \|y_d(t) - y(t)\| \le \varepsilon^*, \quad 0 \le t \le T, \qquad (6)$$

Since the dynamics is not known, the tracking problem that we consider is not trivial ; we will seek a solution by an iterative learning method.

For this, let the suffix k denote the iteration ordinal number of trial such that, for example, $y_k(t)$ is the value of the system output at time t, $0 \le t \le T$, at the k-th operation,etc. The notations $e_k(t)$, $x_k(t)$, and $u_k(t)$ will be similiarly defined.

As a solution method, we propose a new type of iterative learning control of the form

$$u_{k+1}(t) = P_1 u_k(t) + P_2 u_{k-1}(t) + Q_1 e_k(t) + Q_2 e_{k-1}(t) \qquad (7)$$

$$where \quad e_k(t) = y_d(t) - y_k(t) \qquad (8)$$

In Fig.1 is shown the algorithm schematically. We call this type of learning control algorithm as a "second order iterative learning control" because data stored at kth iteration and $(k-1)$th iteration are used to obtain the $(k+1)$th iteration control. It is observed that for fixed t, the algorithm in (7) is a 2nd-order difference equation in the iteration number k. In this context, the method in [3], for example, may be called as a first order iterative learning control.

As in [3], we now show that, under certain conditions, the algorithm in (7) results in convergence for output tracking.

Theorem 1.

Given the unknown LTI system equation (5) at known initial state $x(0) = \xi^0$ and the desired output $y_d(t), 0 \le t \le T$, suppose the algorithm (7) is applied under the following two conditions.

(A1) $P_1 + P_2 = I$

(A2) $\|P_1 - Q_1 D\|_\infty + \|P_2 - Q_2 D\|_\infty < 1$

If $u_0(t), 0 \le t \le T$ is chosen to be a continuous function, and the initial states at each iteration are reset to be $x_k(t) = \xi^0$, for each k, then, $y_k(t) \to y_d(t)$ on $t \in [0,T]$ as $k \to \infty$.

Proof.

Let $u_d(t)$, $x_d(t)$ be control input and system state respectively corresponding to the desired output $y_d(t)$. It follows from equations (5),(7),(8) and (A1) that

$$\begin{aligned} u_d(t) - u_{k+1}(t) &= P_1(u_d(t) - u_k(t)) + P_2(u_d(t) - u_{k-1}(t)) - Q_1\{Cx_d(t) + Du_d(t) \\ &\quad - Cx_k(t) - Du_k(t)\} - Q_2\{C(x_d(t) + Du_d(t) - Cx_{k-1}(t)) - Du_{k-1}(t)\} \\ &= (P_1 - Q_1 D)(u_d(t) - u_k(t)) + (P_2 - Q_2 D)(u_d(t) - u_{k-1}(t)) \\ &\quad - Q_1 C(x_d(t) - x_k(t)) - Q_2 C(x_d(t) - x_{k-1}(t)) \end{aligned} \qquad (9)$$

Taking norms gives

$$\begin{aligned} \|u_d(t) - u_{k+1}(t)\|_\infty &\le \|P_1 - Q_1 D\|_\infty \|u_d(t) - u_k(t)\|_\infty + \|P_2 - Q_2 D\|_\infty \\ &\quad \|u_d(t) - u_{k-1}(t)\|_\infty + \|Q_1 C\|_\infty \|x_d(t) - x_k(t)\|_\infty \\ &\quad + \|Q_2 C\|_\infty \|x_d(t) - x_{k-1}(t)\|_\infty \quad \text{for } all \ t \in [0,T]. \end{aligned} \qquad (10)$$

Now since $x_k(0) = x_d(0)$ for all k, we have, for $t \in [0,T]$,

$$\begin{aligned} \|x_d(t) - x_k(t)\|_\infty &= \|\int_0^t [\{Ax_d(\tau) + Bu_d(\tau)\} - \{Ax_k(\tau) + Bu_k(\tau)\}] d\tau\|_\infty \\ &\le \int_0^t \{\|A\|_\infty \|x_d(\tau) - x_k(\tau)\|_\infty + \|B\|_\infty \|u_d(\tau) - u_k(\tau)\|_\infty\} d\tau \end{aligned} \qquad (11)$$

Applying the Bellman-Gronwall Lemma [11], we get

$$||x_d(t)-x_k(t)||_\infty \le \int_0^t ||B||_\infty ||u_d(\tau)-u_k(\tau)||_\infty e^{a(t-\tau)} d\tau \tag{12}$$

$$\text{for } all \ t\in[0,t], \ \ where \ a = ||A||_\infty .$$

Therefore, combining (10) and (12), we see that, with $\delta u_k(t) = u_d(t)-u_k(t)$

$$||\delta u_{k+1}(t)||_\infty \le l_1 ||\delta u_k(t)||_\infty + l_2 ||\delta u_{k-1}(t)||_\infty$$

$$+ m_1 \int_0^t ||\delta u_k(\tau)||_\infty e^{a(t-\tau)} d\tau + m_2 \int_0^t ||\delta u_{k-1}(\tau)||_\infty e^{a(t-\tau)} d\tau \tag{13}$$

$$where \quad l_1 = ||P_1 - Q_1 D||_\infty, \ \ l_2 = ||P_2 - Q_2 D||_\infty$$

$$m_1 = ||Q_1 C||_\infty ||B||_\infty, \ \ m_2 = ||Q_2 C||_\infty ||B||_\infty$$

Now, define

$$||h(\cdot)||_\lambda \triangleq \sup_{t\in[0,T]} e^{-\lambda t} ||h(t)||_\infty .$$

Multiplying (13) by the positive function $e^{-\lambda t}$, we have

$$e^{-\lambda t} ||\delta u_{k+1}(t)||_\infty \le l_1 e^{-\lambda t} ||\delta u_k(t)||_\infty + l_2 e^{-\lambda t} ||\delta u_{k-1}(t)||_\infty$$

$$+ m_1 \int_0^t e^{-\lambda\tau} ||\delta u_k(\tau)||_\infty e^{(a-\lambda)(t-\tau)} d\tau + m_2 \int_0^t e^{-\lambda\tau} ||\delta u_{k-1}(\tau)||_\infty e^{(a-\lambda)(t-\tau)} d\tau$$

$$\le l_1 ||\delta u_k(\cdot)||_\lambda + l_2 ||\delta u_{k-1}(\cdot)||_\lambda$$

$$+ m_1 \int_0^t ||\delta u_k(\cdot)||_\lambda e^{(a-\lambda)(t-\tau)} d\tau + m_2 \int_0^t ||\delta u_{k-1}(\cdot)||_\lambda e^{(a-\lambda)(t-\tau)} d\tau$$

$$\text{for } all \ t\in[0,T] \tag{14}$$

Therefore,

$$||\delta u_{k+1}(\cdot)||_\lambda \le \left[l_1 + \frac{m_1}{\lambda - a}(1-e^{(a-\lambda)T}) \right] ||\delta u_k(\cdot)||_\lambda$$

$$+ \left[l_2 + \frac{m_2}{\lambda - a}(1-e^{(a-\lambda)T}) \right] ||\delta u_{k-1}(\cdot)||_\lambda \tag{15}$$

Now, it is not too difficult to show that $||\delta u_k(\cdot)||_\lambda \to 0 \ as \ k\to\infty$ if

$$\left[l_1 + \frac{m_1}{\lambda - a}(1-e^{(a-\lambda)T})\right] + \left[l_2 + \frac{m_2}{\lambda - a}(1-e^{(a-\lambda)T})\right] < 1$$

is satisfied. In fact, a nonnegative sequence $\{w_k\}, k=1,2,3,...$ with the property

$$w_{k+2} \le r\ w_{k+1} + s\ w_k \quad (r, s > 0)$$

converges to zero if $r+s<1$ holds.

Since $l_1 + l_2 < 1$ from the assumption (A2), it is possible to choose $\lambda>0$ large enough so that

$$l_1 + l_2 + \frac{m_1}{\lambda - a}(1-e^{(a-\lambda)T}) + \frac{m_2}{\lambda - a}(1-e^{(a-\lambda)T}) < 1 . \tag{16}$$

Thus $||\delta u_k(\cdot)||_\lambda \to 0 \ as \ k\to\infty$. By the definition of $||\ ||_\lambda$, we know that

$$\sup_{t\in[0,T]} ||\delta u_k(t)|| \le e^{\lambda T} ||\delta u_k(\cdot)||_\lambda .$$

Therefore, $\sup_{t\in[0,T]} ||\delta u_k(t)|| \to 0 \ as \ k\to\infty$ and this means

$$u_k(t) \to u_d(t) \ \ as \ k\to\infty \ \ on \ t\in[0,T] . \tag{17}$$

Furthermore, (12) implies

$$x_k(t) \to x_d(t) \ \ as \ k\to\infty \ \ on \ t\in[0,T] . \tag{18}$$

Thus, from (5),(17),(18)

$$y_k(t) \to y_d(t) \ \ as \ k\to\infty \ \ on \ t\in[0,T] .$$

[Q.E.D]

Remark 1 :

The proposed method can be proved to be applicable for a class of nonlinear dynamic systems.

Remark 2 :

In the proposed method, we can observe that if the controlled system is of the form (5) with nonzero D term, the method does not require any time-derivative operation. This fact makes the learning process to be effective even in the case where we can't measure the velocity or acceleration signal. This observation is in agreement with that of Sugie and Ono's in [12], where they reported D term plays a crucial role in the error convergence.

3. Examples with Comments

The proposed algorithm in the paper is less simple than most of the existing methods, e.g., that of Arimoto[3], but can be more effective in "learning and control", for there are more freedoms for adjustment for fast convergence. Further, our algorithm can be less sensitive to temporal disturbances injected in the current measurement $y_k(t)$ because the control action $u_{k+1}(t)$ utilizes previous measurements $y_j(t), j<k$, as well.

To be specific, consider the linear time-varying plant whose dynamics is described by

$$\begin{bmatrix} \dot{x}_1(t) \\ \dot{x}_2(t) \end{bmatrix} = \begin{bmatrix} 0 & 1 \\ -(2+5t) & -(3+2t) \end{bmatrix} \begin{bmatrix} x_1(t) \\ x_2(t) \end{bmatrix} + \begin{bmatrix} 0 \\ 1 \end{bmatrix} u(t) , \ \begin{bmatrix} x_1(0) \\ x_2(0) \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \end{bmatrix}, 0\le t\le 1$$

$$y(t) = [\ 0 \ \ 1\ ] \begin{bmatrix} x_1(t) \\ x_2(t) \end{bmatrix} \tag{19}$$

The plant (19) is assumed to be periodic in t with period $T=1$.

Let the desired output trajectory $y_d(t), t\in[0,1]$ to track be given by

$$y_d(t) = 12t^2(1-t) \ . \tag{20}$$

It is remarked that, as a formal notation for periodicity, the variable t of matrix $A(t)$ in eq.(19) and right-hand side of eq.(20) may be replaced by $t-[t]$, where $[t]$ is the Gaussian number of t, such that the system is periodic with $T=1$.

In three numerical examples in the following, we shall apply the 2nd-order algorithm of Section 2 for conciseness of presentation.

Example 1 : First, suppose the dynamics of the plant (19) is known. Instead of applying advanced control techniques such as LQ optimal controller, we apply the learning control algorithm repeatedly until satisfactory performance is obtained as in [3][8].

As in [8], let $\varepsilon^* = 0.06$, and let initial input $u_0(t)$, for $t\in[0,1]$ be given by

$$u_0(t) = 0, \quad 0\le t\le 1 . \tag{21}$$

It is noted that, since $\dot{y}(t) = C\ \dot{x}(t) = CA\ x(t) + CB\ u(t)$, we find

$$D = C\,B = [0\ 1] \begin{bmatrix} 0 \\ 1 \end{bmatrix} = 1 . \tag{22}$$

Then we can obtain $y_k(t)$ for the k-th iteration, as shown in Fig.2(a). Here, the control gains in (7) are chosen to be $P_1=1.1,\ P_2=-0.1,\ Q_1=1.4,\ Q_2=-0.15$. From this figure, we can observe that six iterative trials are sufficient to generate a trajectory which is within ε^*-bound of the desired trajectory $y_d(t)$.

To compare the effectiveness of our learning control method with Arimoto's method in [3], let's recall that their algorithm gives the convergence condition as follows :

$$||I-CB\Gamma|| < 1 . \tag{23}$$

The gain Γ was chosen to be unity to get a best convergence rate and via simulation, we found that eleven iterations were needed to generate a trajectory within ε^*-bound of the desired trajectory as shown in Fig.2(b).

Example 2 : Consider the case that the dynamics is unknown.

This time, let $\varepsilon^* = 0.1$. Since $D = CB$ is unknown, we shall replace D in (A2) with guessed model value $D_M = 0.5$ and choose P_1, P_2, Q_1 and Q_2 for a best convergence : thus we choose $P_1=0.8,\ P_2=0.2,\ Q_1=1.6,\ Q_2=0.4$. The resulted $y_k(t)$ for the k-th iteration is shown in Fig.3(a), in which we can observe that only eight iterative operations are sufficient to generate a trajectory within ε^*-bound of the desired trajectory.

In this case, if we apply the Arimoto's algorithm[3], the gain Γ is chosen to be $[D_M]^{-1}=\frac{1}{0.5}=2$ to render a best convergence rate for the assumed model. The resulting $y_k(t)$ for the k-th iteration is shown in Fig.3(b), in which we observe that nineteen iterative operations are needed to generate a trajectory within ε^*-bound of the desired trajectory.

Remark :

We have observed through examples that the 2nd-order learning algorithm shows a better convergence speed than a 1st-order learning control of [3]. This may be a generic property of higher-order learning control algorithms. A formal theory for general systems is yet to be established but for simple cases, one may show efficiency of convergence of a second-order algorithm by arguing as follows :

Let's consider the following simple linear time-invariant system.

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

$$y(t) = C\,x(t) \tag{24}$$

In Arimoto's method [3], the kth iteration error magnitude and the $(k+1)$th iteration error magnitude are related by :

$$||e_{k+1}(\cdot)||_\lambda \le ||I-CB\Gamma||_\infty ||e_k(\cdot)||_\lambda \tag{25}$$

$$where \quad e_k(t) = \dot{y}_d(t)-\dot{y}_k(t) \tag{26}$$

from which we may say that the convergence speed of Arimoto's is dictated by

$$|z| = ||I-CB\Gamma||_\infty . \tag{27}$$

In our 2nd-order method, the convergence of the algorithm was described by :

$$||\delta u_{k+1}(\cdot)||_\lambda \le ||P_1-Q_1D||_\infty ||\delta u_k(\cdot)||_\lambda + ||P_2-Q_2D||_\infty ||\delta u_{k-1}(\cdot)||_\lambda$$

$$where \quad D = CB . \tag{28}$$

The last equality $D = CB$ comes from the fact that

$$\dot{y}(t) = C\,\dot{x}(t) = CA\,x(t) + CB\,u(t) . \tag{29}$$

and the relation (26).

The convergence of this 2nd-order algorithm may well be said to be dictated by the magnitude of the roots of

$$z^2-||P_1-Q_1D||_\infty z-||P_2-Q_2D||_\infty = 0 \tag{30}$$

or

$$|z| = \frac{1}{2}\{(||P_1-Q_1D||_\infty^2+4||P_2-Q_2D||_\infty)^{\frac{1}{2}} \pm ||P_1-Q_1D||_\infty\} \tag{31}$$

For the fast convergence speed, we must choose the control gain Γ in (25) or P_1, P_2, Q_1, Q_2 in (28) such that $|z|$ is as small as possible under the condition that $D = CB$ is unknown.

To make a fair comparison, the system parameter D is modelled (or guessed) to be D_M and, assume that the relation $D_M = \alpha D$ ($\alpha>0$) holds between the modelled and real parameters. (For SISO system, this last relation is quite natural.)

Then, in Arimoto's method, we will choose $\Gamma = D_M^{-1}$ to make best convergence, and in this case the value of $|z|$ becomes

$$|z| = ||I-D\Gamma||_\infty = ||I-DD_M^{-1}||_\infty = ||I-\frac{1}{\alpha}I||_\infty = |1-\frac{1}{\alpha}|$$

In our 2nd-order method, we will choose $Q_2=P_2D_M^{-1}$ and $Q_1=P_1D_M^{-1}$ to make best convergence and the value of $|z|$ becomes

$$|z| = \frac{1}{2}\{(||P_1-P_1D_M^{-1}D||_\infty^2+4||P_2-P_2D_M^{-1}D||_\infty)^{\frac{1}{2}} \pm ||P_1-P_1D_M^{-1}D||_\infty\}$$

$$= |1-\frac{1}{\alpha}|\frac{1}{2}\{(||P_1||_\infty^2+\frac{4}{|1-\frac{1}{\alpha}|}||P_2||_\infty)^{\frac{1}{2}} \pm ||P_1||_\infty\} = \beta_{1,2}\,|1-\frac{1}{\alpha}|$$

So the proposed method yields the higher convergency speed than that of Arimoto's, if we choose the control gains P_1, P_2 such that $0<\beta_{1,2}<1$ holds.

Example 3 : The Effect of Disturbance

Under the settings of the control gains being the same as those of example 1, suppose a sequence of iterative learning operations have been carried out to be in a satisfactory state when a disturbance of size 1.0 is occurred at the output at $t=0.5$. As shown in Fig.4(a) and Fig.4(b), we can observe from the computer simulation that the tracking performance was recovered in 4 iterations when the proposed 2nd-order method was applied while a similar performance was achieved after 6 iterations for Arimoto's method. Obviously, our method is less sensitive than that of Arimoto's method in handling disturbances, showing quicker convergence.

4. Concluding Remarks

A second-order iterative learning control algorithm was proposed in which more historical data were used to better the output tracking performance. A sufficient condition for the convergency of the proposed algorithm was given, and via computer simulations, improved performance of the proposed algorithm was illustrated.

We remark that the proposed method is a kind of generalization of the conventional 1st-order method ; for example, Arimoto's method corresponds to the case of $P_1=I, P_2=0, Q_1=\Gamma, Q_2=0$ in the proposed method.

References

[1] CARBONNEL, J.G., MICHALSKI, R.S., and MITCHELL, T.M.: 'Machine Learning' (Springer Verlag, New York, 1984)

[2] UCHIYAMA, M.: 'Formulation of High-Speed Motion Pattern of a Mechanical Arm by Trial', Trans.Society of Instrument and Control Engineers of Japan, 1978, 14, (6), pp. 706-712

[3] ARIMOTO, S., KAWAMURA, S., and MIYAZAKI, F.: 'Bettering Operation of Robots by Learning', Journal of Robotic Systems, 1984, 1, (2), pp. 123-140

[4] KAWAMURA, S., MIYAZAKI, F., and ARIMOTO, S.: 'Realization of Robot Motion Based on a Learning Method', IEEE Trans. on Systems,Man,and Cybernetics, 1988, 18, (1), pp. 126-134

[5] KAWAMURA, S., MIYAZAKI, F., and ARIMOTO, S.: 'Applications of Learning Method for Dynamic Control of Robot Manipulators', Proc. of 24th IEEE Conference on Decision and Control, Ft. Lauderdale, Fl., 1985, pp. 1381-1386

[6] CRAIG, C.: 'Adaptive Control of Mechanical Manipulators' (Addison-Wesley,1988)

[7] GU, Y., and LOH, N.: 'Learning Control in Robotic Systems', Proc. IEEE International Symposium on Intelligent Control, 1987, Philadelphia,Pa. pp. 360-364

[8] OH, S.R., BIEN, Z., and SUH, I.H.: 'An Iterative Learning Control Method with Application for the Robot Manipulators', IEEE Journal of Robotics and Automation, to appear

[9] TOGAI, M., and YAMANO, O.: '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, Ft. Lauderdale, Fl., 1985, pp. 1399-1404

[10] FURUTA, K., and YAMAKITA, M.: 'The Design of a Learning Control System for Multivariable Systems', Proc.IEEE International Symposium on Intelligent Control, 1987, Philadelphia,Pa. pp. 371-376

[11] LAKSHMIKANTHAM, V., and LEELA, S.: 'Differential and Integral Inequalities' (Academic Press,1969, Volume 1, pp. 37-41)

[12] SUGIE, T.,and ONO, T.: 'On a learning Control Law', System and Control, Japan, 1987, 31, (2), pp. 129-135

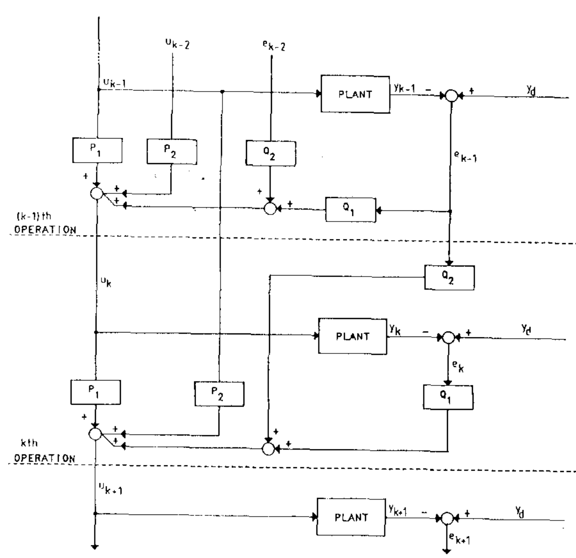

Fig.1 Structure of second-order iterative learning control scheme

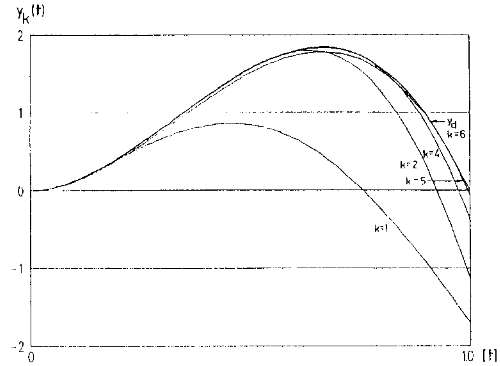

Fig 2(a) Output tracking performance for known plant (Example 1) by the proposed method

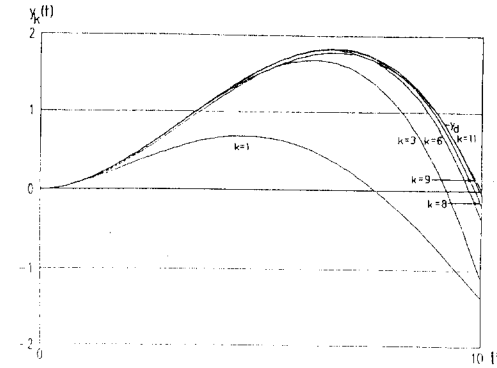

Fig 2(b) Output tracking performance for known plant (Example 1) by Arimoto's method

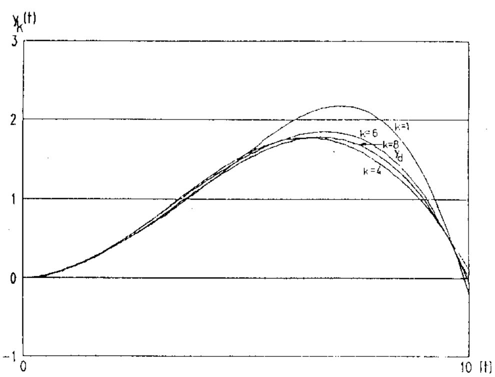

Fig 3(a) Output tracking performance for unknown plant (Example 2) by the proposed method

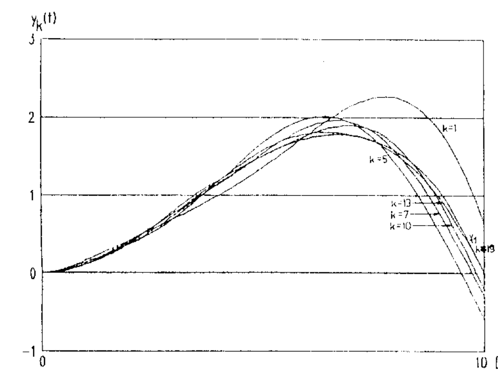

Fig 3(b) Output tracking performance for unknown plant (Example 2) by Arimoto's method

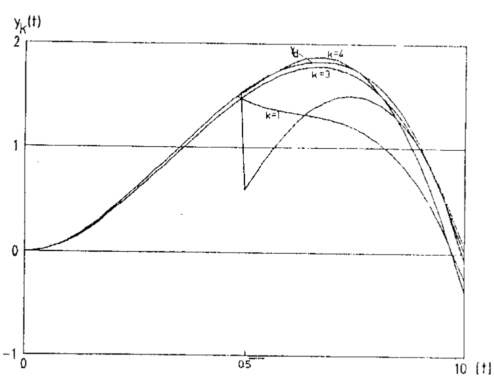

Fig 4(a) Recovery from disturbance (Example 3) by the proposed method

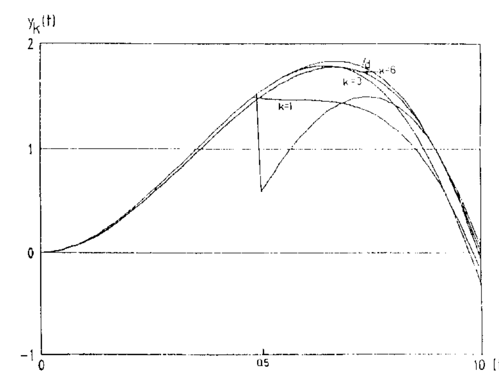

Fig 4(b) Recovery from disturbance (Example 3) by Arimoto's method