INTERNATIONAL JOURNAL OF ADAPTIVE CONTROL AND SIGNAL PROCESSING
Int. J. Adapt. Control Signal Process. 2002; **16**:515–538 (DOI: 10.1002/acs.707)

Closed-loop iterative learning control for non-linear systems with initial shifts

Mingxuan Sun and Danwei Wang*,†

School of Electrical and Electronic Engineering, Nanyang Technological University, Nanyang Avenue, Singapore 639798, Singapore

SUMMARY

This paper is concerned with the problem of the iterative learning control with current cycle feedback for a class of non-linear systems with well-defined relative degree. The tracking error caused by a non-zero initial shift is detected as extended D-type learning algorithm is applied. The defect is overcome by adding terms including the output error, its derivatives as well as integrals. Asymptotic tracking of the final output to the desired trajectory is guaranteed. As an alternative approach, an initial rectifying action is introduced in the extended D-type learning algorithm and shown effective to achieve the desired trajectory jointed smoothly with a transitional trajectory from the starting position. Also these algorithms with adjustable tracking interval ensure better robustness performance in the presence of initial shifts. Numerical simulation is conducted to demonstrate the theoretical results. Copyright © 2002 John Wiley & Sons, Ltd.

KEY WORDS: iterative learning control; feedback; initial condition problem; relative degree; non-linear systems

1. INTRODUCTION

Iterative learning control (ILC) systems, as a specialization of *repetitive systems*, perform a given task repeatedly. There are many practical applications of this kind of systems such as robotic manipulators, CNC machine tools, hard disk drive servos, rolling mills, long-wall coal cutting machines, chemical patch processes, etc. Rigorous stability theory for linear repetitive systems has been developed by Rogers and Owens [1], which provides one general method for solving the problem of linear ILC systems. ILC analysis and design methodologies for non-linear continuous-time systems have been developed since the midst of the 1980s. One fundamental design takes the form of

$$u_{k+1} = u_k + U(I_k) \tag{1}$$

where k is the number of operation cycle, $u_k(t)$ the control input and I_k the available information at the kth cycle, including the output error, its derivatives as well as integrals. $U(\cdot)$ represents

*Correspondence to: Professor D. Wang, School of Electrical and Electronic Engineering, Nanyang Technological University, Nanyang Avenue, Singapore 639798, Singapore.
† *E-mail address*: edwwang@ntu.edu.sg

Received 18 April 2000
Revised 10 October 2001
Accepted 10 October 2001
Published online 14 February 2002

the update function and is usually a linear map of their arguments. Applying the updating law, the last cycle's input is updated by the last cycle's information to generate the current cycle's input. The learning process is done in an open-loop configuration as shown in Figure 1. As one special type, the so-called D-type ILC captures the trend/direction information from the recorded data and ensures zero-error tracking over entire operation interval. The explicit sufficient conditions for guaranteeing the convergence were obtained by Arimoto *et al.* [2] and Hauser [3], and the convergence proofs are straightforward. The major characteristics have been further examined for systems with direct transmission term by Sugie and Ono [4], partial irregularity by Porter and Mohamed [5] and relative degree by Ahn *et al.* [6].

There exist early works that have suggested incorporating the feedforward input provided by learning algorithm in feedback configuration [7–10]. Useful insights on performance improvement benefited from usage of the feedforward/feedback control were presented, where the feedback control mainly aims to stabilize system so that the output trajectories are close to the desired one. Atkeson and McIntyre [7] utilized the system input, instead of feedforward input to form the learning algorithm. In Reference [11], a general form of such learning algorithm was presented and rigorously analysed, with control input saturation being taken into account. Bondi *et al.* [8] simply set a sufficient amount of a linear output feedback which was shown to be effective to ensure the tracking performance. This scheme assumes the presence of ideal acceleration sensors and the problem arisen from acceleration measurements was argued. Moore *et al.* [10] proposed adaptive gain adjustment technique which enables the system to adaptively choose the gains that yield convergence. The convergence proof depends on the obtained result in Reference [8]. Other profitable and important ways of incorporation with current cycle feedback include those proposed by Qu and Zhuang [12], Xu and Qu [13], Chen *et al.* [14], Moore [15] and Phan *et al.* [16].

To exploit the advantage of current cycle feedback and avoid the design task of feedforward control, many researchers have proposed the method of the last cycle's input being updated by only the measurement data from the current cycle, which can be formalized in the updating law of the form [17–20]

$$u_{k+1} = u_k + U(I_{k+1}) \tag{2}$$

The learning process is done in the way of dynamic feedback as shown in Figure 2. In this paper, this kind of learning algorithms is prefixed with *closed-loop*. It is already known that closed-loop

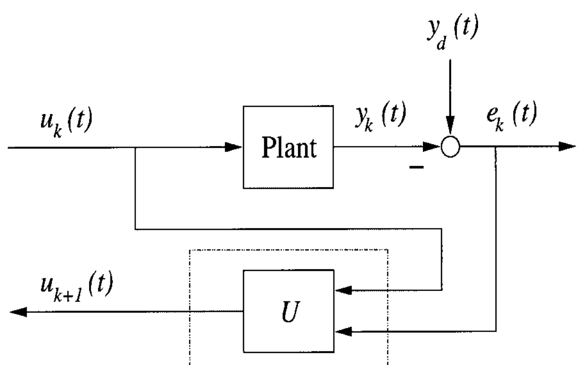

Fig. 1. Open-loop iterative learning control system.

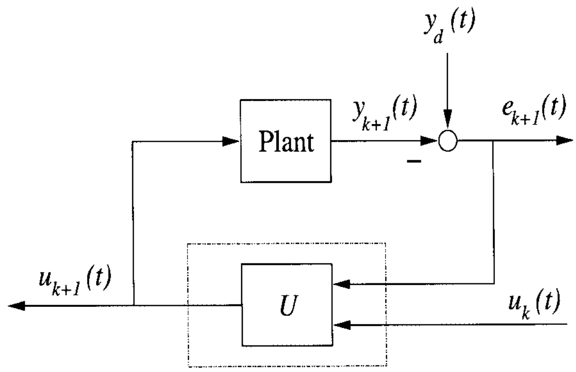

Fig. 2. Closed-loop iterative learning control system.

learning algorithms offer one way for the control design with features such as wide ranges in choosing learning gain and the improvement of convergence rate, in contrast to open-loop counterparts.

One obvious restriction in iterative learning control is with regard to initial condition of the controlled system. Repositioning operation should be done such that the initial condition at each cycle satisfies the requirement to ensure convergence. Amongst published literature, requirements on initial condition for different objective systems and learning algorithms are as follows [2, 21, 22]:

$$x_k(0) = x^0, \qquad y_k(0) = y_d(0) \tag{3}$$

$$x_{k+1}(0) = x_k(0) + \sum_{j=2}^{l+1} \sum_{i=j}^{l+1} A^{i-j} B K_i e_k^{j-1}(0), \quad y_0(0) = v(0) \tag{4}$$

$$x_{i+1}(0) = x_i(0) + B(t_0)L(t_0)e_i(t_0) \tag{5}$$

where $x_k(0)$ is the initial condition at the kth cycle. Equations (4) and (5) provide initial adjustment mechanism by which the desired initial condition can be assessed, but some system parameters are required to be known. One adjustment scheme of initial condition without requiring any precise system parameter was proposed in Reference [23], based on the relation between the variations of initial condition, input and output of linear discrete-time systems. One typical kind of the requirement is that the initial condition of the objective system at the beginning of each cycle is reset the same as the initial condition corresponding to the desired trajectory, i.e.

$$x_k(0) = x_d(0) \tag{6}$$

(3) is equivalent to (6) in case of relative degree 1. Another kind of requirement is that the initial condition at each cycle is reset the same but can be any given point, i.e.

$$x_k(0) = x_0 \tag{7}$$

where x_0 is fixed and finite. In Reference [4], it was shown that (7) is equivalent to (6) for systems with direct transmission term. However, the situation for systems without direct transmission term could be different and (7) is a relaxed requirement compared with (6). Benefiting from this relaxation, repositioning operation will become easy for some implementations. One example is that the system output is expected to track a step reference from the resetting zero position. Another example is for the cases with unknown desired initial conditions.

Due to practical repeatability precision, robustness with respect to perturbed initial conditions could be crucial due to the iteration nature of learning algorithms. It is necessary to certify that iterative learning control is technically sound even when existence of resetting errors is allowed to some extent at each cycle. Namely,

$$\|x_d(0) - x_k(0)\| \leqslant c_{x_{d0}} \tag{8}$$

instead of the restrict alignment (6). Arimoto *et al.* [24] demonstrated the robustness of a PID-type ILC for robots with the aid of linearization. Bondi *et al.* [8] established the boundedness of iterative trajectories through combining linear output feedback, also for robots. Heinzinger *et al.* [9] introduced a biased term with a forgetting factor into the D-type ILC [3] and presented systematic treatment of the robustness to the presence of resetting errors, state disturbances and measurement noises, for a class of non-linear systems. The same issue was also discussed for the PD-type and PID-type ILCs. Such a learning algorithm with a forgetting factor has also been applied to address other tracking problems, e.g. References [25, 26]. In Reference [26], the simultaneous motion and contact force tracking was solved for robotic manipulators whose end-effector is in contact with the constraint environment. In Reference [25], the tracking control was tackled for applications where the reference trajectory is allowed to vary slowly. In case of P-type ILCs with a forgetting factor, Arimoto *et al.* [27] investigated the robust properties by exploring the passivity of robot dynamics. In References [17, 28], the robustness was studied for a class of non-linear systems where boundedness of time derivative of the input–output coupling matrix was assumed. The existing results ensure boundedness of the error for iterative trajectories in the presence of resetting errors described by (8) and the bound continuously depends on the bound on the resetting errors. Recently, a comprehensive survey on ILC researches was made by Moore [15], in which the works concerning the properties of ILC to various types of environmental uncertainties were summarized.

Lee and Bien [29] characterized the effect of an initial shift described by (7) on the converged output trajectory as the D-type ILC [2] was applied. PD-type, PID-type and a generalized learning algorithm [29–31] were positively utilized to improve tracking performance, respectively. For better performance in the face of initial shifts

$$\|x_0 = x_k(0)\| \leqslant c_{x_0} \tag{9}$$

the method of 'iterative learning control with multi-model input', with the designed parameters, was shown effective to lower the size of the error bounds. It should be noted that the results in References [29–31] are only applicable for systems with relative degree 1. Without restriction on system relative degree, the technique of suitably reducing sampling rate of the controlled linear systems, presented by Hillenbrand and Pandit [32], was shown to be effective to achieve better tracking. Porter and Mohamed [5] considered partially irregular LTI systems and found that to eliminate the effect caused by a non-zero initial shift, an initial impulsive action is required in a learning algorithm. Such a learning algorithm enables trajectory zero-error tracking on entire tracking interval. However, the use of an impulsive action is not practical. Motivated by the work [6] on the iterative learning control of a class of non-linear systems with will-defined relative degree, an initial rectifying action is utilized to address the same problem in Reference

[33] and is shown to be effective to guarantee the uniform convergence of the system output to a desired trajectory with a smooth transient. The above-mentioned learning algorithms, however, are all in open-loop configuration. It is as yet not clear what happens to closed-loop learning algorithms confronting the same situation.

In this paper, firstly, we consider the case where the initial condition remains the same for each cycle but different from the desired initial condition. The tracking error caused by such a shift is detected as extended D-type learning algorithm is used. Secondly, we consider the case where the initial condition is reset to a given point or its neighbourhood. By adding terms to form the so-called extended PD-type and PID-type closed-loop learning algorithms, asymptotical tracking capability of the resultant output trajectory to the desired trajectory is shown. The derived convergence conditions are independent of the added terms. These terms, however, offer more flexibility for the resultant output trajectory. As an alternative method, an initial rectifying action is introduced in the extended D-type learning algorithm and is shown to be effective for achieving the desired trajectory with a smooth transience, which can be viewed as the closed-loop counterpart of the open-loop learning algorithm proposed in Reference [33]. Compared with the initial impulsive approach [5], the initial rectifying action is finite and implementable, even for a class of non-linear systems with well-defined relative degree. Finally, the limitation of extended D-type learning algorithm and effectiveness of the proposed methods in dealing with initial shift(s) are illustrated by numerical results.

2. PROBLEM FORMULATION AND PRELIMINARIES

Consider a class of non-linear continuous-time systems described by

$$\dot{x}(t) = f(x(t)) + B(x(t))u(t) \tag{10}$$

$$y(t) = g(x(t)) \tag{11}$$

where $x \in R^n$, $u = [u_1, \ldots, u_r]^T \in R^r$ and $y = [y_1, \ldots, y_m]^T \in R^m$ denote the state, the control input and the output of the system, respectively. Non-linear functions $f(\cdot) \in R^n$, $B(\cdot) = [b_1(\cdot), \ldots, b_r(\cdot)] \in R^{n \times r}$ and $g(\cdot) = [g_1(\cdot), \ldots, g_m(\cdot)]^T \in R^m$ are smooth in their domains of definition, which are known about certain properties. Every operation ends in a finite time T, i.e. $t \in [0, T]$. For each fixed $x(0)$, S denotes a mapping from $(x(0),\ u(t),\ t \in [0, T])$ to $(x(t), t \in [0, T])$ and O a mapping from $(x(0),\ u(t),\ t \in [0, T])$ to $(y(t),\ t \in [0, T])$. In these notations, $x(\cdot) = S(x(0), u(\cdot))$ and $y(\cdot) = O(x(0), u(\cdot))$.

Throughout this paper, the following derivative notations and definition are needed. The derivative of a scalar function $g(x)$ along a vector $f(x) \in R^n$ is denoted by, for $x \in R^n$,

$$L_f g(x) = \frac{\partial g(x)}{\partial x} f(x)$$

The repeated derivatives along the same vector are denoted by

$$L_f^j g(x) = L_f(L_f^{j-1} g(x)), \quad j = 1, 2, \ldots$$

and $L_f^0 g(x) = g(x)$. In addition, the derivative of $g(x)$ taken first along $f(x)$ and then along a vector $b(x) \in R^n$ is denoted by

$$L_b L_f g(x) = \frac{\partial (L_f g(x))}{\partial x} b(x)$$

Definition 2.1

The (vector) relative degree of non-linear continuous-time system (10)–(11) is associated with the vector $\mu = \{\mu_1, \ldots, \mu_m\}$ satisfying, for $x \in R^n$,

$$L_{b_p} L_f^j g_q(x) = 0,\ 0 \leqslant j \leqslant \mu_q - 2,\ 1 \leqslant p \leqslant r, 1 \leqslant q \leqslant m$$

and the $m \times r$ matrix

$$D(x) = \begin{bmatrix} L_{b_1} L_f^{\mu_1 - 1} g_1(x), & \ldots, & L_{b_r} L_f^{\mu_1 - 1} g_1(x) \\ & \vdots & \\ L_{b_1} L_f^{\mu_m - 1} g_m(x), & \ldots, & L_{b_r} L_f^{\mu_m - 1} g_m(x) \end{bmatrix}$$

being of full column rank.

Remark 2.1

If the system (10)–(11) has relative degree $\mu = \{\mu_1, \ldots, \mu_m\}$, the derivatives of the system output can be written as

$$y_q^{(j)} = L_f^j g_q(x), \quad 0 \leqslant j \leqslant \mu_g - 1 \tag{12}$$

$$y_q^{(\mu_q)} = L_f^{\mu_q} g_q(x) + [L_{b_1} L_f^{\mu_q - 1} g_q(x), \ \ldots, \ L_{b_r} L_f^{\mu_q - 1} g_q(x)] u \tag{13}$$

where μ_q is obviously the minimum order of time derivative of the qth output to which a direct transmission is established from at least one component of the control input u. Definition 2.1 allows the number of outputs greater than the number of inputs, in contrast to the definition in References [6, 34].

In the paper, the vector norm is defined as $\|a\| = \max_{1 \leqslant i \leqslant n} |a_i|$ for an n-dimensional vector $a = [a_1, \ \ldots, \ a_n]^{\mathrm{T}}$ and the matrix norm is defined as the induced norm by the vector norm, i.e. for a matrix $A = \{a_{ij}\} \in R^{m \times n}$, $\|A\| = \max_{1 \leqslant i \leqslant m} \sum_{j=1}^{n} |a_{ij}|$. The λ-norm for a vector-valued function $b(t) \in R^n$ is defined as $\|b(\cdot)\| \lambda = \sup_{t \in [0, T]} \{ e^{-\lambda t} \|b(t)\| \}, \ \lambda > 0$.

The control problem to be solved in this paper is now formulated as follows. We consider the case where the system under consideration does not reset the initial condition $x_k(0)$ to the desired one $x_d(0)$, the initial condition corresponding to the desired trajectory. Instead, there is an initial shift, i.e. $\|x_d(0) - x_k(0)\| \geqslant c_{x_{d0}}$, where a lasting offset is detected when applying extended D-type learning algorithm. We shall add certain terms to the learning algorithm to overcome the offset, including the output error, its derivatives as well as integrals. Our proposed algorithms ensure the final output to follow the desired trajectory as time goes. In addition, we introduce an initial rectifying action in the learning algorithm. The initial rectifying action will produce a smooth transition from the system resetting position to the desired trajectory. This smooth transition can be specified in the initial rectifying action. Analyses are given for more practical cases where the resetting is not perfect. That is, the initial condition $x_k(0)$ at each cycle is reset at a neighbourhood of x_0 and $\|x_0 - x_k(0)\| \leqslant c_{x_0}$. We shall analyse the effect due to the initial shifts on the final system output and show that the added terms and the initial rectifying action are effective to eliminate such effect. Furthermore, if c_{x_0} tends to zero, to control objective mentioned above can be achieved.

To make the problem more tractable, the following assumptions are imposed on the system (10)–(11):

(A1) The mappings S and O are one to one.
(A2) The system is with (vector) relative degree $\mu = \{\mu_1, \ldots, \mu_m\}$ for $x \in R^n$.
(A3) The functions $f(\cdot)$, $B(\cdot)$, $g(\cdot)$, $L_f^j g_q(\cdot)$, $1 \leqslant j \leqslant \mu_q$, $1 \leqslant q \leqslant m$ and $L_{b_p} L_f^{\mu_q - 1} g_q(\cdot)$, $1 \leqslant p \leqslant r$, $1 \leqslant q \leqslant m$, are globally Lipschitz in x for $x \in R^n$, i.e. $\|a(x') - a(x'')\| \leqslant l_a \|x' - x''\|$, $\|L_f^j g_q(x') - L_f^j g_q(x'')\| \leqslant l_1^q \|x' - x''\|$, $1 \leqslant j \leqslant \mu_q$ and $\|L_{b_p} L_f^{\mu_q - 1} g_q(x') - L_{b_p} L_f^{\mu_q - 1} g_q(x'')\| \leqslant l_2^{p,q} x' - x''\|$ for positive constants $l_a > 0$, $a \in \{f, B, g\}$, $l_1^q, l_2^{p,q}$ and for all $x', x'' \in R^n$.
(A4) The operator $B(\cdot)$ is bounded for $x \in R^n$.
(A5) For a desired trajectory $y_d(t) = [y_{1,d}(t), \ldots, y_{m,d}(t)]^{\mathrm{T}}$, $t \in [0, T], y_{i,d}(t)$ is μ_i times continuously differentiable.

3. EXTENDED D-TYPE ILC AND ITS TRACKING ERROR

The learning control design is reasonably based on the actions and their produced results. In view of (13), $\{u_k(t), y_{q,k}^{(\mu q)}(t), 1 \leqslant q \leqslant m\}$ is a pair of algebraically related cause and effect. In the open-loop case [6], if learning gain $\Gamma(y_k(t))$ is chosen properly, $u_k(t)$ plus update term $\Gamma(y_k(t))(y_d^{(\mu)}(t) - y_k^{(\mu)}(t))$ produces an improved input. One can use it as $u_{k+1}(t)$. On the other hand, $u_k(t)$ minus term $\Gamma(y_k(t))(y_d^{(\mu)}(t) - y_k^{(\mu)}(t))$ can be worse and we assume that it is given $u_{k-1}(t)$. This observation leads to the updating law

$$u_{k+1}(t) = u_k(t) + \Gamma(y_{k+1}(t))(y_d^{(\mu)}(t) - y_{k+1}^{(\mu)}(t)) \tag{14}$$

where k refers to the number of operation cycle, $y^{(\mu)}(t) = [y_1^{(\mu_1)}(t), \ldots, y_m^{(\mu_m)}(t)]^{\mathrm{T}}$, $\Gamma(y_k(t)) \in R^{r \times m}$ is the learning gain matrix bounded on R^m.

Remark 3.1

Updating law (14) belongs to the so-called D-type. In the early works [2, 3], its open-loop counterparts were investigated for the case of relative degree 1. To deal with systems with well-defined relative degree, > 1, extended D-type updating law was studied in References [6, 33]. The usage of current cycle feedback in our proposed schemes of this paper would provide a direct way in choosing learning gain and improving convergence rate, in contrast to their open-loop counterparts. This is the major feature appealing us. To the closed-loop ILCs in the case of mechanical systems, the presence of ideal acceleration sensors is assumed due to the usage of current cycle feedback. The same situation occurs in the ILC works [8, 10]. The intrinsic time delay of the sensors would cause tracking performance degradation due to the error between the actual acceleration and the measured one, which might be one limitation of the proposed methods. However, the convergence can be achieved theoretically provided the measurement errors are sufficiently small. Like the same argument by Bondi *et al.* [8], one condition is that an upper bound on the rate of change of the desired trajectory, which depends on the sensors used, should be set in the implementation so that the measurement errors are reduced.

The following theorem shows the effect of initial shift on the converged output:

Theorem 3.1

Given a desired trajectory $y_d(t), t \in [0, T]$, let the system (10)–(11) satisfy assumptions (A1)–(A5) and the updating law (14) be applied. If, at the beginning of each cycle,

$$x_k(0) = x_0, \quad k = 0, 1, 2, \dots \tag{15}$$

and the learning gain matrix is chosen such that

$$\|(I + \Gamma(g(x))D(x))^{-1}\| \leqslant \frac{1}{\rho} < 1, \quad x \in R^n \tag{16}$$

the system output $y_k(t)$ converges uniformly to $y^*(t)$ for $t \in [0, T]$ as $k \to \infty$, where $y^*(t) = [y_1^*(t), \dots, y_m^*(t)]^{\mathrm{T}}$, and for $1 \leqslant q \leqslant m$,

$$y_q^*(t) = y_{q,d}(t) - \sum_{j=0}^{\mu_q - 1} \frac{t^j}{j!}\left(y_{q,d}^{(j)}(0) - L_f^j g_q(x_0)\right) \tag{17}$$

Proof

For simplicity, the dependence of the variables on time is implied unless otherwise specified. Denote by u^* the control input satisfying

$$y^* = g(x^*) \tag{18}$$

where x^* is the corresponding state, the solution of the differential equation

$$\dot{x}^* = f(x^*) + B(x^*)u^*, \quad x^*(0) = x_0 \tag{19}$$

In view of y^* defined in (17), (14) can be written as

$$\begin{aligned} u_{k+1} &= u_k + \Gamma(y_{k+1})(y^{*(\mu)} - y_{k+1}^{(\mu)}) + \Gamma(y_{k+1})(y_d^{(\mu)} - y^{*(\mu)}) \\ &= u_k + \Gamma(y_{k+1})(y^{*(\mu)} - y_{k+1}^{(\mu)}) \end{aligned} \tag{20}$$

which leads to

$$(I + \Gamma(y_{k+1})D(x_{k+1}))\Delta u_{k+1}^* = \Delta u_k^* - \Gamma(y_{k+1})[c(x^*) - c(x_{k+1}) + (D(x^*) - D(x_{k+1}))u^*]$$

where $\Delta u_k^* = u^* - u_k$ and $c(x) = [L_f^{\mu_1} g_1(x), \dots, L_f^{\mu_m} g_m(x)]^{\mathrm{T}}$. Taking norms on both sides yields

$$\begin{aligned} \|\Delta u_{k+1}^*\| \leqslant \|(I + \Gamma(y_{k+1})D(x_{k+1}))^{-1}\| [\|\Delta u_k^*\| + \|\Gamma(y_{k+1})\|(\|c(x^*) - c(x_{k+1})\| \\ + \|D(x^*) - D(x_{k+1})\| \|u^*\|)] \end{aligned} \tag{21}$$

Applying the Lipschitz conditions results in

$$
\begin{aligned}
\|c(x^*) - c(x_{k+1})\| &= \left\| \begin{bmatrix} L_f^{\mu_1} g_1(x^*) - L_f^{\mu_1} g_1(x_{k+1}) \\ \vdots \\ L_f^{\mu_m} g_m(x^*) - L_f^{\mu_m} g_m(x_{k+1}) \end{bmatrix} \right\| \\
&\leqslant \left\| \begin{bmatrix} \| L_f^{\mu_1} g_1(x^*) - L_f^{\mu_1} g_1(x_{k+1}) \| \\ \vdots \\ \| L_f^{\mu_m} g_m(x^*) - L_f^{\mu_m} g_m(x_{k+1}) \| \end{bmatrix} \right\| \\
&\leqslant l_1 \| \|\Delta x_{k+1}^*\| \|
\end{aligned} \tag{22}
$$

and

$$
\begin{aligned}
&\|D(x^*) - D(x_{k+1})\| \\
&= \left\| \begin{bmatrix} L_{b_1} L_f^{\mu_1 - 1} g_1(x^*) - L_{b_1} L_f^{\mu_1 - 1} g_1(x_{k+1}), \ \ldots, \ L_{b_r} L_f^{\mu_1 - 1} g_1(x^*) - L_{b_r} L_f^{\mu_1 - 1} g_1(x_{k+1}) \\ \vdots \\ L_{b_1} L_f^{\mu_m - 1} g_m(x^*) - L_{b_1} L_f^{\mu_m - 1} g_m(x_{k+1}), \ \ldots, \ L_{b_r} L_f^{\mu_m - 1} g_m(x^*) - L_{b_r} L_f^{\mu_m - 1} g_m(x_{k+1}) \end{bmatrix} \right\|
\end{aligned} \tag{23}
$$

$$
\begin{aligned}
&\leqslant \left\| \begin{bmatrix} \left\| L_{b_1} L_f^{\mu_1 - 1} g_1(x^*) - L_{b_1} L_f^{\mu_1 - 1} g_1(x_{k+1}) \right\|, \ \ldots, \ \left\| L_{b_r} L_f^{\mu_1 - 1} g_1(x^*) - L_{b_r} L_f^{\mu_1 - 1} g_1(x_{k+1}) \right\| \\ \vdots \\ \left\| L_{b_1} L_f^{\mu_m - 1} g_m(x^*) - L_{b_1} L_f^{\mu_m - 1} g_m(x_{k+1}) \right\|, \ \ldots, \ \left\| L_{b_r} L_f^{\mu_m - 1} g_m(x^*) - L_{b_r} L_f^{\mu_m - 1} g_m(x_{k+1}) \right\| \end{bmatrix} \right\| \\
&\leqslant l_2 \|\Delta x_{k+1}^*\|
\end{aligned}
$$

where $\Delta x_{k+1}^* = x^* - x_{k+1}$, $l_1 = \max\{l_1^1, \cdots, l_1^m\}$ and $l_2 = \max\{l_2^{1,1} + \cdots + l_2^{r,1}, \cdots, l_2^{1,m} + \cdots + l_2^{r,m}\}$. Substituting (22) and (23) into (21) gives rise to, denoting c_Γ the norm bound for $\Gamma(\cdot)$, $c_1 = c_\Gamma(l_1 + l_2 c_{u^*})$ and $c_{u^*} = \sup_{t \in [0, T]} \|u^*(t)\|$,

$$
\rho \|\Delta u_{k+1}^*\| \leqslant \|\Delta u_k^*\| + c_1 \|\Delta x_{k+1}^*\| \tag{24}
$$

Now, we evaluate the state error on the right-hand side of (24). Integrating both sides of (10) and (19) produces

$$
\Delta x_{k+1}^* = \int_0^t [f(x^*) - f(x_{k+1}) + (B(x^*) - B(x_{k+1}))u^* + B(x_{k+1}) \Delta u_{k+1}^*] \, \mathrm{d}s
$$

Taking norms and using their properties give

$$
\|\Delta x_{k+1}^*\| \leqslant \int_0^t (c_2 \|\Delta x_{k+1}^*\| + c_B \|\Delta u_{k+1}^*\|) \, \mathrm{d}s \tag{25}
$$

where c_B is the norm bound for $B(\cdot)$ and $c_2 = l_f + l_B c_{u^*}$. Note the facts that, for $\lambda > 0$,

$$\begin{aligned}\sup_{t\in[0,\,T]} \int_0^t \mathrm{e}^{-\lambda t}\left|\left|\Delta x^*_{k+1}\right|\right| \mathrm{d}s \quad &\leqslant \quad \sup_{t\in[0,\,T]} \frac{1-\mathrm{e}^{-\lambda t}}{\lambda}\left|\left|\Delta x^*_{k+1}\right|\right|_\lambda \\ &= O\left(\frac{1}{\lambda}\right)\left|\left|\Delta x^*_{k+1}\right|\right|_\lambda\end{aligned}$$

and

$$\sup_{t\in[0,\,T]} \int_0^t \mathrm{e}^{-\lambda t}\left|\left|\Delta u^*_{k+1}\right|\right| \mathrm{d}s \leqslant O\left(\frac{1}{\lambda}\right)\left|\left|\Delta u^*_{k+1}\right|\right|_\lambda$$

Multiplying both sides of (25) by $\mathrm{e}^{-\lambda t}(\lambda > 0)$, and using the above, we get

$$\left|\left|\Delta x^*_{k+1}\right|\right|_\lambda \leqslant c_2\, O\left(\frac{1}{\lambda}\right)\left|\left|\Delta x^*_{k+1}\right|\right|_\lambda + c_B\, O\left(\frac{1}{\lambda}\right)\left|\left|\Delta u^*_{k+1}\right|\right|_\lambda$$

A $\lambda > 0$ is chosen such that

$$1 - c_2\, O\left(\frac{1}{\lambda}\right) > 0$$

which implies

$$\left|\left|\Delta x^*_{k+1}\right|\right|_\lambda \leqslant \frac{c_B\, O(1/\lambda)}{1 - c_2 O(1/\lambda)}\left|\left|\Delta u^*_{k+1}\right|\right|_\lambda \tag{26}$$

Then, taking the λ-norm for (24) and using (26), we obtain

$$\left(\rho - \frac{c_1 c_B\, O(1/\lambda)}{1 - c_2 O(1/\lambda)}\right)\left|\left|\Delta u^*_{k+1}\right|\right|_\lambda \leqslant \left|\left|\Delta u^*_k\right|\right|_\lambda \tag{27}$$

Since $\rho > 1$, it is possible to find a sufficiently large λ such that

$$\bar{\rho} = \rho - \frac{c_1 c_B\, O(1/\lambda)}{1 - c_2 O(1/\lambda)} > 1$$

Then, (27) is a contraction in $\left|\left|\Delta u^*_k\right|\right|_\lambda$. When the iterations increase, $k \to \infty$, we attain $\left|\left|\Delta u^*_k\right|\right|_\lambda \to 0$ so that $u_k \to u^*$ uniformly on $[0, T]$ as $k \to \infty$. If follows from (26) that $x_k \to x^*$ uniformly on $[0, T]$ as $k \to \infty$. Furthermore, by the assumption on $g(\cdot)$ in (A3), $y_k \to y^*$ uniformly on $[0, T]$ as $k \to \infty$. This completes the proof. □

Remarks 3.2

High gain feedback is known to offer a powerful way to improve the convergence rate. However, Theorem 3.1 shows that whatever the learning gain is chosen, the converged output trajectory $y^*(t)$ deviates from the desired trajectory $y_d(t)$. It is easy to see from Theorem 3.1 that

$$\begin{aligned}\lim_{k\to\infty}(y_d(t) - y_k(t)) &= y_d(t) - y^*(t) + \lim_{k\to\infty}(y^*(t) - y_k(t)) \\ &= y_d(t) - y^*(t)\end{aligned}$$

The offset given in (17) is invariable with the learning gain chosen based on (16), which implies that the convergence is ensured only if $L_f^j g_q(x_0) = y_{q,d}^{(j)}(0)$, $0 \leqslant j \leqslant \mu_q - 1$, $1 \leqslant q \leqslant m$. The boundedness of the tracking error can be argued in the presence of finite initial shift. It follows from (17) that

$$\left\|y_{q,d}(t) - y_q^*(t)\right\| = \left\|\sum_{j=0}^{\mu_q-1} \frac{t^j}{j!}\left(L_f^j g_q(x_d(0)) - L_f^j g_q(x_0)\right)\right\|$$

$$\leqslant \sum_{j=0}^{\mu_q-1} \frac{T^j}{j!} l_1^q \|x_d(0) - x_0\|$$

Therefore,

$$\left\|y_d(t) - y^*(t)\right\| \leqslant \max_{1 \leqslant q \leqslant m} \left(1 + \frac{T}{1!} + \cdots + \frac{T^{\mu_q-1}}{\mu_q - 1!}\right) l_1 \|x_d(0) - x_0\|$$

The term $\|x_d(0) - x_0)\|$ in the bound is dominant.

4. EXTENDED PD-TYPE AND PID-TYPE ILC

In this section, we shall show that the deviated convergence presented in Theorem 3.1 can be overcome by adding certain terms to the updating law (14) in the form of

$$u_{k+1}(t) = u_k(t) + \Gamma(y_{k+1}(t)) \begin{bmatrix} e_{1,k+1}^{(\mu_1)}(t) + \sum_{i=1}^{\mu_1} \gamma_{1,i} e_{1,k+1}^{(\mu_1-i)}(t) \\ \vdots \\ e_{m,k+1}^{(\mu_m)}(t) + \sum_{i=1}^{\mu_m} \gamma_{m,i} e_{m,k+1}^{(\mu_m-i)}(t) \end{bmatrix} \tag{28}$$

where $e_{q,k+1} = y_{q,d} - y_{q,k+1}$, $1 \leqslant q \leqslant m$ and $\gamma_{q,i}$, $1 \leqslant q \leqslant m$, $1 \leqslant i \leqslant \mu_q$ are the design parameters. This updating law is termed extended PD-type due to usage of the output error and its lower-order derivatives. Following theorem shows the effect of the designed parameters on the converged trajectory.

Theorem 4.1

Given a desired trajectory $y_d(t)$, $t \in [0, T]$, let the system (10)–(11) satisfy assumptions (A1)–(A5) and the updating law (28) be applied. If, at the beginning of each cycle,

$$\|x_0 - x_k(0)\| \leqslant c_{x_0}, \; k = 0, 1, 2, \ldots \tag{29}$$

and (16) holds, the asymptotic bound of error $y^*(t) - y_k(t)$ is proportional to c_{x_0} on [0, T] as $k \to \infty$, where $y^*(t) = [y_1^*(t), \ldots, y_m^*(t)]^T$ and for $1 \leqslant q \leqslant m$,

$$y_q^*(t) = y_{q,d}(t) - c e^{A_q t} z_q(0) \tag{30}$$

$$c = \begin{bmatrix} 1 & 0 & 0 & \ldots & 0 \end{bmatrix}$$

$$A_q = \begin{bmatrix} 0 & 1 & 0 & \cdots & 0 \\ 0 & 0 & 1 & \cdots & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & 0 & \cdots & 1 \\ -\gamma_{q,\mu_q} & -\gamma_{q,\mu_q-1} & -\gamma_{q,\mu_q-2} & \cdots & -\gamma_{q,1} \end{bmatrix},$$

$$z_q(0) = \begin{bmatrix} y_{q,d}^{(0)}(0) - L_f^0 g_q(x_0) \\ \vdots \\ y_{q,d}^{(\mu_q-1)}(0) - L_f^{\mu_q-1} g_q(x_0) \end{bmatrix}$$

Proof

As in the proof of Theorem 3.1, let u^* be the control input which generates the trajectory y^* and x^* is the corresponding state. Denote by $e_{q,k+1}^* = y_q^* - y_{q,k+1}$. In view of y^* defined in (30), the updating law (28) can be written as

$$u_{k+1} = u_k + \Gamma(y_{k+1}) \begin{bmatrix} e_{1,k+1}^{*(\mu_1)} + \sum_{i=1}^{\mu_1} \gamma_{1,i} e_{1,k+1}^{*(\mu_1-i)} \\ \vdots \\ e_{m,k+1}^{*(\mu_m)} + \sum_{i=1}^{\mu_m} \gamma_{m,i} e_{m,k+1}^{*(\mu_m-i)} \end{bmatrix}$$

$$+ \Gamma(y_{k+1}) \begin{bmatrix} y_{1,d}^{(\mu_1)} - y_1^{*(\mu_1)} + \sum_{i=1}^{\mu_1} \gamma_{1,i} \left(y_{1,d}^{(\mu_1-i)} - y_1^{*(\mu_1-i)} \right) \\ \vdots \\ y_{m,d}^{(\mu_m)} - y_m^{*(\mu_m)} + \sum_{i=1}^{\mu_m} \gamma_{m,i} \left(y_{m,d}^{(\mu_m-i)} - y_m^{*(\mu_m-i)} \right) \end{bmatrix}$$

$$= u_k + \Gamma(y_{k+1}) \begin{bmatrix} e_{1,k+1}^{*(\mu_1)} + \sum_{i=1}^{\mu_1} \gamma_{1,i} e_{1,k+1}^{*(\mu_1-i)} \\ \vdots \\ e_{m,k+1}^{*(\mu_m)} + \sum_{i=1}^{\mu_m} \gamma_{m,i} e_{m,k+1}^{*(\mu_m-i)} \end{bmatrix} \tag{31}$$

which leads to

$$(I + \Gamma(y_{k+1})D(x_{k+1}))\Delta u^*_{k+1} = \Delta u^*_k - \Gamma(y_{k+1})[c(x^*) - c(x_{k+1}) + (D(x^*) - D(x_{k+1}))u^*]$$

$$- \Gamma(y_{k+1})\begin{bmatrix} \sum_{i=1}^{\mu_1} \gamma_{1,\,i}\left(L_f^{\mu_1 - i} g_1(x^*) - L_f^{\mu_1 - i} g_1(x_{k+1})\right) \\ \vdots \\ \sum_{i=1}^{\mu_m} \gamma_{m,\,i}\left(L_f^{\mu_m - i} g_m(x^*) - L_f^{\mu_m - i} g_m(x_{k+1})\right) \end{bmatrix}$$

Taking norms and applying the bounds and the Lipschitz conditions give rise to

$$\rho||\Delta u^*_{k+1}|| \leqslant ||\Delta u^*_k|| + c_1||\Delta x^*_{k+1}|| \tag{32}$$

where c_Γ is the norm bound for $\Gamma(\cdot)$, $c_1 = c_\Gamma\left[l_1\left(1 + \max_{1\leqslant j\leqslant m} \quad \{\sum_{i=1}^{\mu_j} \gamma_{j,\,i}\}\right) + l_2 c_{u^*}\right]$, $l_1 = \max_{1\leqslant j\leqslant m}\{l_1^j\}$, $l_2 = \max_{1\leqslant j\leqslant m}\left\{\sum_{i=1}^{r} l_2^{i,\,j}\right\}$, and $c_{u^*} = \sup_{t\in[0,\,T]} \|u^*(t)\|$.

To evaluate the state error, we integrate the state equations and obtain

$$\Delta x^*_{k+1} = \int_0^t [f(x^*) - f(x_{k+1}) + (B(x^*) - B(x_{k+1}))u^* + B(x_{k+1})\Delta u^*_{k+1}]\,\mathrm{d}s + x_0 - x_{k+1}(0)$$

Taking norms and using their properties yield

$$||\Delta x^*_{k+1}|| \leqslant \int_0^t (c_2||\Delta x^*_{k+1}|| + c_B||\Delta u^*_{k+1}||)\,\mathrm{d}s + \|x_0 - x_{k+1}(0)\| \tag{33}$$

where c_B is the norm bound for $B(\cdot)$ and $c_2 = l_f + l_B c_{u^*}$. Multiplying both sides of (33) by $\mathrm{e}^{-\lambda t}(\lambda > 0)$ produces

$$||\Delta x^*_{k+1}||_\lambda \leqslant c_2\, O\left(\frac{1}{\lambda}\right)||\Delta x^*_{k+1}||_\lambda + c_B\, O\left(\frac{1}{\lambda}\right)||\Delta u^*_{k+1}||_\lambda + c_{x_0}$$

A $\lambda > 0$ is chosen such that

$$1 - c_2\, O\left(\frac{1}{\lambda}\right) > 0$$

which implies

$$||\Delta x^*_{k+1}||_\lambda \leqslant \frac{c_B O(1/\lambda)}{1 - c_2\, O(1/\lambda)}||\Delta u^*_{k+1}||_\lambda + \frac{1}{1 - c_2\, O(1/\lambda)}\, c_{x_0} \tag{34}$$

Then taking the λ-norm for (32) and using (34) result in

$$\left(\rho - \frac{c_1 c_B\, O(1/\lambda)}{1 - c_2\, O(1/\lambda)}\right)||\Delta u^*_{k+1}||_\lambda \leqslant ||\Delta u^*_k||_\lambda + \frac{c_1}{1 - c_2\, O(1/\lambda)}\, c_{x_0} \tag{35}$$

Since $\rho > 1$, it is possible to find a sufficiently large λ such that

$$\bar{\rho} = \rho - \frac{c_1 c_B\, O(1/\lambda)}{1 - c_2\, O(1/\lambda)} > 1$$

Then, (35) is a contraction in $\|\Delta u_k^*\|_\lambda$. When the iterations increase, $k \to \infty$,

$$\limsup_{k\to\infty} \|\Delta u_k^*\|_\lambda \leqslant \frac{1}{\bar{\rho}-1}\frac{c_1}{1-c_2\,O(1/\lambda)}c_{x_0} \tag{36}$$

The results for Δx_k^* and $y^* - y_k$ can be derived from (34) and the assumption on $g(\cdot)$ in (A3). The theorem follows. □

Remark 4.1

Theorem 4.1 implies that the suitable choice of $\Gamma(\cdot)$ is sufficient to guarantee the robustness but the derived condition is independent of parameters $\gamma_{q,i}$. These terms, however, offers more flexibility for the resultant output trajectory. If parameters $\gamma_{q,i}$ are chosen properly, e.g. the eigenvalue $\lambda(A_q) \leqslant 0$, $y^*(t)$ will follow $y_d(t)$ asymptotically as time increases. Thus, robustness performance can be improved by deducing the error between $y^*(t)$ and $y_d(t)$. Furthermore, when c_{x_0} tends to zero, the system will possess asymptotic tracking capability along time-axis.

Remark 4.2

If parameters $\gamma_{q,i} = 0$, the extended PD-type updating law becomes D-type one described by (14). Since

$$e^{A_q t} = \begin{bmatrix} 1 & t & \dots & \dfrac{t^{\mu_q-2}}{(\mu_q-2)!} & \dfrac{t^{\mu_q-1}}{(\mu_q-1)!} \\ 0 & 1 & \dots & \dfrac{t^{\mu_q-3}}{(\mu_q-3)!} & \dfrac{t^{\mu_q-2}}{(\mu_q-2)!} \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ 0 & 0 & \dots & 1 & t \\ 0 & 0 & \dots & 0 & 1 \end{bmatrix}$$

the trajectory (30) reduces to (17).

In the following, we shall show that more flexibility of the converged trajectory can be provided by extended PID-type learning algorithm in the form of

$$u_{k+1}(t) = u_k(t) + \Gamma(y_{k+1}(t))$$
$$\begin{bmatrix} e_{1,k+1}^{(\mu_1)}(t) + \sum_{i=1}^{\mu_1}\gamma_{1,i}e_{1,k+1}^{(\mu_1-i)}(t) & + \sum_{i=1}^{\eta_1}\gamma_{1,\mu_1+i}\int_0^t\int_0^{t_1}\dots\int_0^{t_{i-1}} e_{1,k+1}(t_i)\,dt_i\dots dt_1 \\ \vdots & \vdots \\ e_{m,k+1}^{(\mu_m)}(t) + \sum_{i=1}^{\mu_m}\gamma_{m,i}e_{m,k+1}^{(\mu_m-i)}(t) & + \sum_{i=1}^{\eta_m}\gamma_{m,\mu_m+i}\int_0^t\int_0^{t_1}\dots\int_0^{t_{i-1}} e_{m,k+1}(t_i)\,dt_i\dots dt_1 \end{bmatrix} \tag{37}$$

where $\gamma_{q,i}$, $1 \leqslant q \leqslant m$, $1 \leqslant i \leqslant \mu_q + \eta_q$ are the design parameters by which the final output trajectory is determined.

Theorem 4.2

Given a desired trajectory $y_d(t)$, $t \in [0, T]$, let the system (10)–(11) satisfy assumptions (A1)–(A5) and the updating law (37) be applied. If the initial condition satisfies (29) at each cycle and (16) holds, the asymptotic bound of error $y^*(t) - y_k(t)$ is proportional to c_{x_0} on $[0, T]$ as

$k \to \infty$, where $y^*(t) = [y_1^*(t), \ \ldots, \ y_m^*(t)]^{\mathrm{T}}$ and for $1 \leqslant q \leqslant m$,

$$y_q^*(t) = y_{q,\,d}(t) - c\mathrm{e}^{A_q t} z_q(0) \tag{38}$$

$$c = \begin{bmatrix} 1 & 0 & 0 & \ldots & 0 \end{bmatrix},$$

$$A_q = \begin{bmatrix} 0 & 1 & 0 & \ldots & 0 \\ 0 & 0 & 1 & \ldots & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & 0 & \ldots & 1 \\ -\gamma_{q,\,\mu_q+\eta_q} & -\gamma_{q,\,\mu_q+\eta_q-1} & -\gamma_{q,\,\mu_q+\eta_q-2} & \ldots & -\gamma_{q,\,1} \end{bmatrix},$$

$$z_q(0) = R_{q,\,\mu_q+\eta_q-1} R_{q,\,\mu_q+\eta_q-2} \ldots R_{q,\,\mu_q} \begin{bmatrix} y_{q,\,d}^{(0)}(0) - L_f^0 g_q(x_0) \\ \vdots \\ y_{q,\,d}^{(\mu_q-1)}(0) - L_f^{\mu_q-1} g_q(x_0) \end{bmatrix},$$

$$R_{q,i} = \begin{bmatrix} 1 & 0 & \ldots & 0 & 0 & \ldots & 0 \\ 0 & 1 & \ldots & 0 & 0 & \ldots & 0 \\ \vdots & \vdots & \ddots & \vdots & \vdots & \vdots & \vdots \\ 0 & 0 & \ldots & 1 & 0 & \ldots & 0 \\ -\gamma_{q,\,\mu_q} & -\gamma_{q,\,\mu_q-1} & \cdots & -\gamma_{q,\,1} & 0 & \ldots & 0 \\ -\gamma_{q,\,\mu_q+1} & -\gamma_{q,\,\mu_q} & \cdots & -\gamma_{q,\,2} & -\gamma_{q,\,1} & \ldots & 0 \\ \vdots & \vdots & \ldots & \vdots & \vdots & \ddots & \vdots \\ -\gamma_{q,\,i} & -\gamma_{q,\,i-1} & \cdots & -\gamma_{q,\,i-\mu_q+1} & -\gamma_{q,\,i-\mu_q} & \cdots & -\gamma_{q,\,1} \end{bmatrix}, \quad \mu_q \leqslant i \leqslant \mu_q + \eta_q - 1$$

Proof

Let u^* be the control input which generates the trajectory y^* and x^* is the corresponding state. In view of y^* defined in (38), the updating law (37) can be written as

$$u_{k+1} = u_k + \Gamma(y_{k+1}) \begin{bmatrix} \mathrm{e}_{1,\,k+1}^{*(\mu_1)}(t) + \sum_{i=1}^{\mu_1} \gamma_{1,\,i}\, \mathrm{e}_{1,\,k+1}^{*(\mu_1-i)}(t) & + \sum_{i=1}^{\eta_1} \gamma_{1,\,\mu_1+i} \int_0^t \int_0^{t_1} \cdots \int_0^{t_{i-1}} \mathrm{e}_{1,\,k+1}^*(t_i)\, \mathrm{d}t_i \cdots \mathrm{d}t_1 \\ \vdots & \vdots \\ \mathrm{e}_{m,\,k+1}^{*(\mu_m)}(t) + \sum_{i=1}^{\mu_m} \gamma_{m,\,i}\, \mathrm{e}_{m,\,k+1}^{*(\mu_m-i)}(t) & + \sum_{i=1}^{\eta_m} \gamma_{m,\,\mu_m+i} \int_0^t \int_0^{t_1} \cdots \int_0^{t_{i-1}} \mathrm{e}_{m,\,k+1}^*(t_i)\, \mathrm{d}t_i \cdots \mathrm{d}t_1 \end{bmatrix}$$

which leads to

$$
\begin{aligned}
&(I+\Gamma(y_{k+1})D(x_{k+1}))\Delta u_{k+1}^{*}\\
&=\Delta u_k^{*}-\Gamma(y_{k+1})[c(x^{*})-c(x_{k+1})+(D(x^{*})-D(x_{k+1}))u^{*}]\\
&-\Gamma(y_{k+1})\begin{bmatrix}\sum_{i=1}^{\mu_1}\gamma_{1,\,i}\left(L_f^{\mu_1-i}g_1(x^{*})-L_f^{\mu_1-i}g_1(x_{k+1})\right)\\ \vdots\\ \sum_{i=1}^{\mu_m}\gamma_{m,\,i}\left(L_f^{\mu_m-i}g_m(x^{*})-L_f^{\mu_m-i}g_m(x_{k+1})\right)\end{bmatrix}
\end{aligned}
$$

$$
-\Gamma(y_{k+1})\begin{bmatrix}\sum_{i=1}^{\eta_1}\gamma_{1,\,\mu_1+i}\int_0^t\int_0^{t_1}\cdots\int_0^{t_{i-1}}(L_f^0g_1(x^{*}(t_i))-L_f^0g_1(x_{k+1}(t_i)))\,\mathrm{d}t_i\cdots\mathrm{d}t_1\\ \vdots\\ \sum_{i=1}^{\eta_m}\gamma_{m,\,\mu_m+i}\int_0^t\int_0^{t_1}\cdots\int_0^{t_{i-1}}(L_f^0g_m(x^{*}(t_i))-L_f^0g_m(x_{k+1}(t_i)))\,\mathrm{d}t_i\cdots\mathrm{d}t_1\end{bmatrix}
$$

Taking norms and applying the bounds and the Lipschitz conditions give rise to

$$
\rho\|\Delta u_{k+1}^{*}\|\leqslant\|\Delta u_k^{*}\|+c_\Gamma\left[l_1\left(1+\max_{1\leqslant j\leqslant m}\left\{\sum_{i=1}^{\mu_j}\gamma_{j,\,i}\right\}\right)+l_2c_{u^{*}}\right]\|\Delta x_{k+1}^{*}\|
$$

$$
+c_\Gamma l_1\left\|\begin{bmatrix}\sum_{i=1}^{\eta_1}\gamma_{1,\,\mu_1+i}\int_0^t\int_0^{t_1}\cdots\int_0^{t_{i-1}}\|\Delta x_{k+1}^{*}(t_i)\|\,\mathrm{d}t_i\cdots\mathrm{d}t_1\\ \vdots\\ \sum_{i=1}^{\eta_m}\gamma_{m,\,\mu_m+i}\int_0^t\int_0^{t_1}\cdots\int_0^{t_{i-1}}\|\Delta x_{k+1}^{*}(t_i)\|\,\mathrm{d}t_i\cdots\mathrm{d}t_1\end{bmatrix}\right\|
$$

where c_Γ is the norm bound for $\Gamma(\cdot)$, $l_1=\max_{1\leqslant j\leqslant m}\{l_1^j\}$, $l_2=\max_{1\leqslant j\leqslant m}\{\sum_{i=1}^r l_2^{i,j}\}$, and $c_{u^{*}}=\sup_{t\in[0,\,T]}\|u^{*}(t)\|$.

Note the fact that

$$
\begin{aligned}
\mathrm{e}^{-\lambda t}\int_0^t\int_0^{t_1}\cdots\int_0^{t_{i-1}}\|\Delta x_{k+1}^{*}(t_i)\|\,\mathrm{d}t_i\cdots\mathrm{d}t_1&\leqslant\left(\frac{1-\mathrm{e}^{-\lambda T}}{\lambda}\right)^i\|\Delta x_{k+1}^{*}\|_\lambda\\
&=O\left(\left(\frac{1}{\lambda}\right)^i\right)\|\Delta x_{k+1}^{*}\|_\lambda
\end{aligned}
$$

which leads to

$$
\rho\|\Delta u_{k+1}^{*}\|_\lambda\leqslant\|\Delta u_k^{*}\|_\lambda+c_1\|\Delta x_{k+1}^{*}\|_\lambda
$$

where $c_1=c_\Gamma\left[l_1\left(1+\max_{1\leqslant j\leqslant m}\left\{\sum_{i=1}^{\mu_j}\gamma_{j,\,i}\right\}\right)+l_2c_{u^{*}}\right]+c_\Gamma l_1\max_{1\leqslant j\leqslant m}\left\{\sum_{i=1}^{\eta_j}\gamma_{j,\,\mu_j+i}\right\}\max\left\{1,\,O((1/\lambda)^{\bar\eta})\right\}$, and $\bar\eta=\max_{1\leqslant j\leqslant m}\{\eta_j\}$.

The rest is on exactly the same lines as that after (32) in the proof of Theorem 4.1. □

Remark 4.3

This section establishes the robustness of extended PD-type and PID-type closed-loop learning algorithms with respect to initial shifts. The results for their open-loop counterparts can be easily established and the proofs follow the lines similar to those for the theorems of this section. Park and Bien [30] investigated the same problem with the open-loop learning algorithm which allows a more general operator acting on the output error, for the case of relative degree 1. The open-loop counterparts of our proposed algorithms for the same system class are its special case. Note that our work, however, focuses on well-defined relative degree non-linear systems.

5. EXTENDED D-TYPE ILC WITH INITIAL RECTIFYING ACTION

In this section, we shall show that the deviated convergence presented in Theorem 3.1 can also be overcome with the assistance of initial rectifying action in the form of

$$u_{k+1}(t) = u_k(t) + \Gamma(y_{k+1}(t))(y_d^{(\mu)}(t) - y_{k+1}^{(\mu)}(t))$$

$$- \Gamma(y_{k+1}(t)) \begin{bmatrix} \sum_{j=0}^{\mu_1-1} \left(\frac{t^j}{j!}\int_t^h \theta_{\mu_1,h}(s)\,\mathrm{d}s\right)^{(\mu_1)} \left(y_{1,d}^{(j)}(0) - y_{1,0}^{(j)}(0)\right) \\ \vdots \\ \sum_{j=0}^{\mu_m-1} \left(\frac{t^j}{j!}\int_t^h \theta_{\mu_m,h}(s)\,\mathrm{d}s\right)^{(\mu_m)} \left(y_{m,d}^{(j)}(0) - y_{m,0}^{(j)}(0)\right) \end{bmatrix} \tag{39}$$

where $\theta_{\mu q,h}:[0,T]\rightarrow R$, $1\leqslant q\leqslant m$, is defined as

$$\theta_{\mu_q,h}(t) = \begin{cases} \frac{1}{h^{2\mu_q+1}}\frac{(2\mu_q+1)!}{\mu_q^{!2}}t^{\mu_q}(h-t)^{\mu_q}, & t\in[0,h] \\ 0, & t\in(h,T] \end{cases} \tag{40}$$

and satisfies

$$\int_0^h \theta_{\mu_q,h}(s)\,\mathrm{d}s = 1 \tag{41}$$

$$\left(\frac{t^j}{j!}\int_t^h \theta_{\mu_q,h}(\tau)\,\mathrm{d}\tau\right)^{(\mu_q)}|_{t=0} = \begin{cases} 1, & \mu_q = j \\ 0, & \mu_q \neq j \end{cases} \tag{42}$$

This added term will ensure that the system output converges to the desired trajectory uniformly except in the initial segment $[0,h]$. In the interval $[0,h]$, a smooth transition is generated from the resetting position to the desired trajectory. The merging of the transitional trajectory to the desired one occurs at the time $t=h$. The following theorem presents sufficient condition to guarantee such property of the converged output.

Theorem 5.1

Given a desired trajectory $y_d(t)$, $t\in[0,T]$, let the system (10)–(11) satisfy assumptions (A1)–(A5) and the updating law (39) be applied. If the initial condition at each cycle satisfies (29) and (16) holds, the asymptotic bound of error $y^*(t)-y_k(t)$ is proportional to c_{x_0} on $[0,T]$ as

$k \to \infty$, where $y^*(t) = [y_1^*(t), \ldots, y_m^*(t)]^{\mathrm{T}}$ and for $1 \leqslant q \leqslant m$,

$$y_q^*(t) = y_{q,d}(t) - \sum_{j=0}^{\mu_q - 1} \left(\frac{t^j}{j!} \int_t^h \theta_{\mu_q, h}(s)\, \mathrm{d}s \right) (y_{q,d}^{(j)}(0) - y_{q,0}^{(j)}(0)),\ t \in [0, T] \tag{43}$$

Proof

Denote by u^* the control input which generates the trajectory y^* and x^* the corresponding state. In view of y^* defined in (43), the updating law (39) can be written as

$$\begin{aligned} u_{k+1} &= u_k + \Gamma(y_{k+1})(y^{*(\mu)} - y_{k+1}^{(\mu)}) + \Gamma(y_{k+1})(y_d^{(\mu)} - y^{*(\mu)}) \\ &\quad - \Gamma(y_{k+1}) \begin{bmatrix} \sum_{j=0}^{\mu_1 - 1} \left(\frac{t^j}{j!} \int_t^h \theta_{\mu_1, h}(s)\, \mathrm{d}s \right)^{(\mu_1)} (y_{1,d}^{(j)}(0) - y_{1,0}^{(j)}(0)) \\ \vdots \\ \sum_{j=0}^{\mu_m - 1} \left(\frac{t^j}{j!} \int_t^h \theta_{\mu_m, h}(s)\, \mathrm{d}s \right)^{(\mu_m)} (y_{m,d}^{(j)}(0) - y_{m,0}^{(j)}(0)) \end{bmatrix} \\ &= u_k + \Gamma(y_{k+1})(y^{*(\mu)} - y_{k+1}^{(\mu)}) \end{aligned}$$

which leads to

$$(I + \Gamma(y_{k+1}) D(x_{k+1})) \Delta u_{k+1}^* = \Delta u_k^* - \Gamma(y_{k+1})[c(x^*) - c(x_{k+1}) + (D(x^*) - D(x_{k+1}))u^*]$$

Taking norms and applying the bounds and the Lipschitz conditions give rise to

$$\rho \|\Delta u_{k+1}^*\| \leqslant \|\Delta u_k^*\| + c_1 \|\Delta x_{k+1}^*\|$$

where c_Γ is the norm bound for $\Gamma(\cdot)$, $c_1 = c_\Gamma(l_1 + l_2 c_{u^*})$, $l_1 = \max\{l_1^1, \ldots, l_1^m\}$, $l_2 = \max\{l_2^{1,1} + \cdots + l_2^{r,1}, \ldots, l_2^{1,m} + \cdots + l_2^{r,m}\}$, and $c_{u^*} = \sup_{t \in [0,T]} \|u^*(t)\|$.

The rest is on exactly the same lines as that after (32) in the proof of Theorem 4.1. □

Remark 5.1

Theorem 5.1 shows that the error between $y^*(t)$ and $y_k(t)$ is proportional to c_{x_0} when the proposed initial rectifying action is used. The output error between $y_d(t)$ and $y_k(t)$ is thus largely reduced after $t \geqslant h$. The trajectory (43) is the same as that resulted by applying the open-loop learning algorithm [33]. But the difference between both learning algorithms lies in the derived sufficient conditions and the learning gain selection based on these conditions, as discussed in Remark 5.4.

Remark 5.2

Theorem 5.1 implies that if c_{x_0} tends to zero, a suitable choice of $\Gamma(\cdot)$ leads to uniform convergence of the system output to the trajectory $y^*(t)$ for all $t \in [0, T]$. From (43), it is seen that $y^*(t) = y_d(t)$, $t \in (h, T]$. Uniform convergence of the system output to the desired trajectory $y_d(t)$ is thus achieved on $(h, T]$, while the converged output trajectory on $[0, h]$ is governed by the initial rectifying action which is a smooth transition from initial position to the desired trajectory. This transitional trajectory joins the desired trajectory at $t = h$ moment which can be specified. Note that the open-loop learning algorithm based on terminal attractor, proposed in

Reference [30], also ensures uniform convergence over finite time interval. However, the result was still developed for systems with relative degree 1.

Remark 5.3

For the kth cycle, output trajectory $y_{q,k}(t)$, $1 \leqslant q \leqslant m$ should depend on the initial condition $y^{(j)}_{q,k}(0)$. However, from (43), the limit trajectory $y^*_q(t)$ only relies on $y^{(j)}_{q,0}(0)$ whenever c_{x_0} tends to zero. This is because the initial alignment requirement (15) leads to

$$\begin{aligned} y^{(j)}_{q,k+1}(0) &= L^j_f g_q(x_{k+1}(0)) \\ &= L^j_f g_q(x_k(0)) \\ &= y^{(j)}_{q,k}(0), \quad 0 \leqslant j \leqslant \mu_q - 1, \quad 1 \leqslant q \leqslant m \end{aligned}$$

Therefore,

$$y^{(j)}_{q,k}(0) = y^{(j)}_{q,0}(0), \quad 0 \leqslant j \leqslant \mu_q - 1, \quad 1 \leqslant q \leqslant m$$

Remark 5.4

The proposed closed-loop learning algorithms provide a great deal of freedom in choosing the learning gain matrix based on (16). Denote by $\bar{D} = [\hat{D}^{\mathrm{T}}\hat{D}]^{-1}\hat{D}^{\mathrm{T}}$ where $\hat{D}$ is an estimate of D. The learning gain is chosen as $\Gamma = \gamma\bar{D}$, $\gamma \neq 0$. If the matrix $\bar{D}D$ is invertible, and

$$1 + \frac{\lambda(\bar{D}D)^{-1}}{\gamma} \neq 0$$

where $\lambda((\bar{D}D)^{-1})$ is the eigenvalue of $(\bar{D}D)^{-1}$, then

$$\lim_{|\gamma|\to\infty} \left|\left|(I + \gamma\bar{D}D)^{-1}\right|\right| = \lim_{|\gamma|\to\infty} \left|\left|\frac{(\bar{D}D)^{-1}}{|\gamma|}\right|\right| \left|\left|\left[I + \frac{(\bar{D}D)^{-1}}{\gamma}\right]^{-1}\right|\right| = 0$$

Therefore, there exists a $|\gamma^*|$ large enough such that

$$\left|\left|(I + \gamma^*\bar{D}D)^{-1}\right|\right| \leqslant \frac{1}{\rho} < 1$$

In addition, one can choose $\Gamma = \gamma I$ if D is invertible. This observation indicates that the control design for the closed-loop learning algorithms allows larger model discrepancy.

6. AN ILLUSTRATIVE EXAMPLE

In this section, numerical simulation is carried out to demonstrate the significant improvement achieved by the proposed learning algorithms. A mechanism of DC-motor driving a single rigid link is selected as the illustrative example. The dynamics of the single link manipulator are described by

$$J\ddot{q} + v\dot{q} + (\tfrac{1}{2}m + M)gl\sin q = u \tag{44}$$

where q is the angular displacement of the manipulator and u the driving torque. The parameters m, M, l and v denote the mass, tip load, length and damping coefficient of the

Table 1 Physical parameters and estimation.

Parameter	True value	Estimation
m	1 kg	1.1 kg
M	2 kg	1.8 kg
l	0.5 m	0.55 m
v	3 kg m^2/s	
g	9.8 m/s^2	

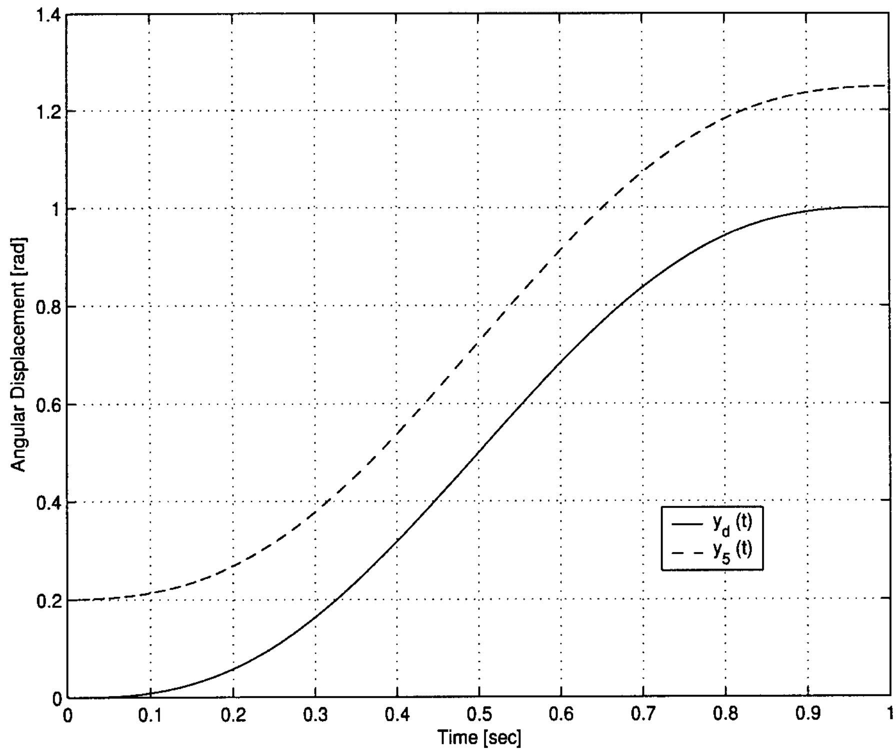

Fig. 3. Converged output trajectory by using extended D-type ILC in the presence of an initial shift.

manipulator, respectively. J represents the moment of inertia with respect to the joint and is defined as $J = Ml^2 + \frac{1}{3}ml^2$, and g the weight acceleration. The parameter values are given in Table 1 and the estimated ones have 10% errors from the true values. Let $y = q$ be the output variable and thus the system has relative degree 2. The desired trajectory for $t \in [0, 1]$ s is given to be

$$y_d(t) = 6t^5 - 15t^4 + 10t^3 \text{ rad} \tag{45}$$

Case 1: Three ILCs are used in the conducted simulations: (1) Extended D-type ILC (14) with $\Gamma = 20\hat{J}$; (2) Extended PD-type ILC (28), which reduces to

$$u_{k+1}(t) = u_k(t) + \Gamma(\ddot{e}_{k+1}(t) + \gamma_1 \dot{e}_{k+1}(t) + \gamma_2 e_{k+1}(t)) \tag{46}$$

with $\Gamma = 20\hat{J}$, $\gamma_1 = 20$ and $\gamma_2 = 100$; and (3) ILC with initial rectifying action (39) given as

$$u_{k+1}(t) = u_k(t) + \Gamma(\ddot{y}_d(t) - \ddot{y}_{k+1}(t)) + \Gamma_{k+1}(t)[\dot{\theta}_{2,h}(t)(y_d(0) - y_0(0)) + (t\dot{\theta}_{2,h}(t) + 2\theta_{2,h}(t))(\dot{y}_d(0) - \dot{y}_0(0))] \quad (47)$$

with $\Gamma = 20\hat{J}$, $h = 0.3$ and

$$\theta_{2,h}(t) = \begin{cases} \frac{30}{h^5}t^2(h-t)^2, & t \in [0, h] \\ 0, & t \in (h, T] \end{cases} \quad (48)$$

For all algorithms, initial control input is chosen as $u_0(t) = 0$, $t \in [0, 1]$.

In order to examine convergence performance in the presence of an initial shift, the initial condition at each cycle is reset to $(y_k(0), \dot{y}_k(0)) = (0.2, 0.05)$. By Theorem 3.1, the converged output trajectory by extended D-type ILC (14) is

$$\lim_{k\to\infty} y_k(t) = y_d(t) - (y_d(0) - y_0(0)) - t(\dot{y}_d(0) - \dot{y}_0(0)) \quad (49)$$

Figure 3 shows the converged trajectory at the fifth iteration where the output trajectory tracks the desired one with the lasting offset described by (49). This offset, according to Theorems 4.1 and 5.1, can be attenuated by applying learning algorithms (46) or (47). Applying

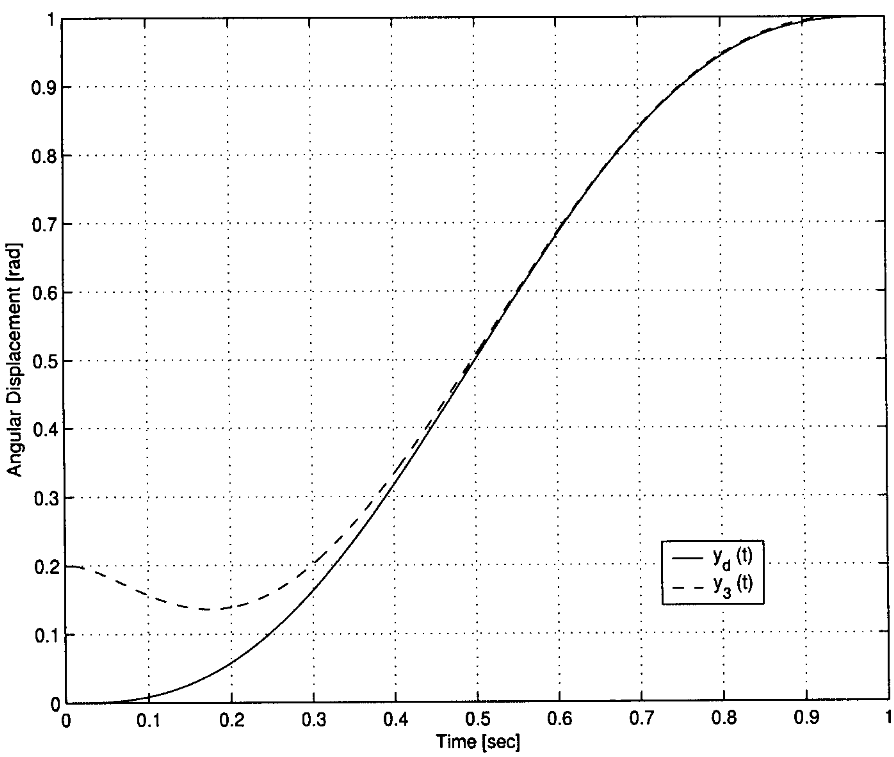

Fig. 4. Converged output trajectory by using extended PD-type ILC in the presence of an initial shift.

extended PD-type ILC (46), the limit trajectory is governed by the second-order characteristic equation, $s^2 + 20s + 100 = 0$. It is observed in Figure 4 that the output trajectory at the third cycle follows the desired trajectory asymptotically as time increases. Applying initial rectifying updating law (47), from Theorem 5.1, the resulted output trajectory is

$$\lim_{k\to\infty} y_k(t) = y_d(t) - \int_t^h \theta_{2,h}(s)\,\mathrm{d}s(y_d(0) - y_0(0)) - t\int_t^h \theta_{2,h}(s)\,\mathrm{d}s(\dot{y}_d(0) - \dot{y}_0(0)) \tag{50}$$

Define the performance index $J_k = \sup_{t\in[0.3,1]} \|y_d(t) - y_k(t)\|$. The iteration stops if $J_k < 0.001$. Via simulation, this performance requirement is achieved at the third iteration as shown in Figure 5. It is clearly seen that the output trajectory uniformly converge to the desired one on the interval [0.3, 1].

Case 2: To examine robustness performance of these learning algorithms in the presence of initial shifts, let the initial condition be $(y_k(0), \dot{y}_k(0)) = (0.2 + 0.01\ \text{randn},\ 0.05 + 0.01\ \text{randn})$. The randn is a generator of random scalar with normal distribution (mean = 0, and variance = 1) but bounded on $[-3, 3]$. The performance index is defined as $J_k = \sup_{t\in[0.3,1]} \|y_d(t) - y_k(t)\|$. Repetitions are done until $k = 100$. Figure 6 shows the tracking errors by learning algorithms (14), (46) and (47), respectively, where applying (46) and (47) gives satisfying robustness performance.

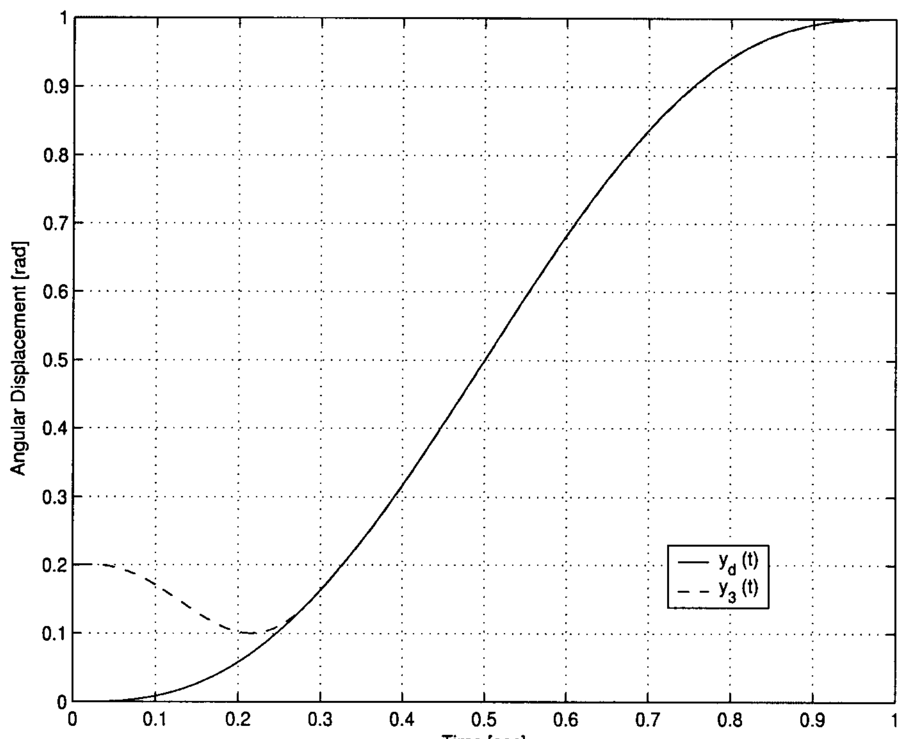

Fig. 5. Converged output trajectory by using ILC with initial rectifying action ($h = 0.3$) in the presence of an initial shift.

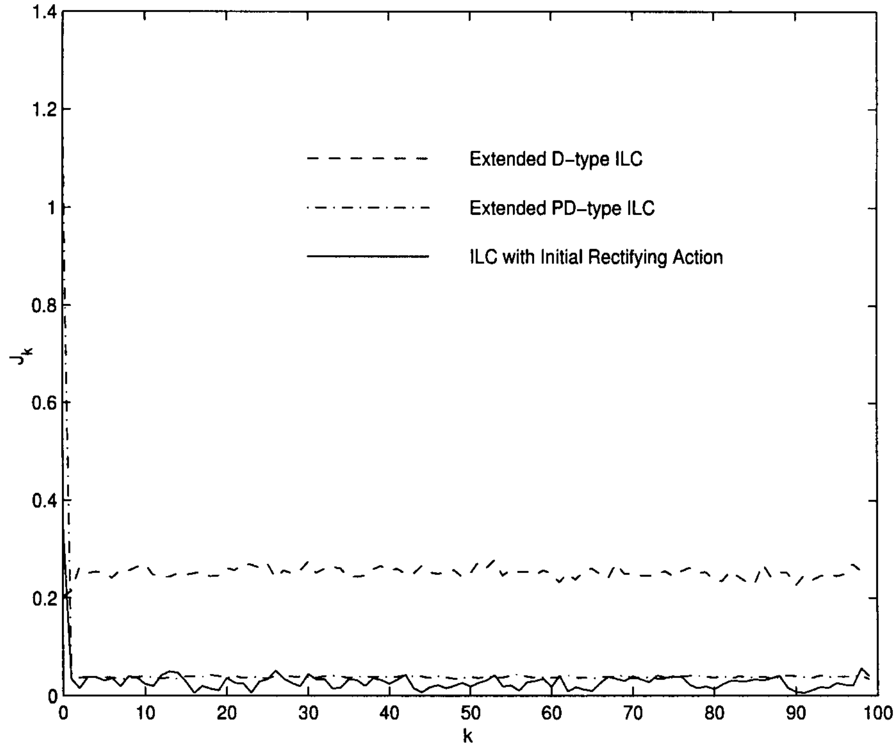

Fig. 6. Robustness performance comparison of learning algorithms in the presence of random initial shifts.

7. CONCLUSION

The defect in tracking performance caused by a non-zero initial shift is examined as extended D-type learning algorithm is applied, even though the learning process is done in closed-loop configuration. Extended PD-type (PID-type) learning algorithm and the initial rectifying learning algorithm are developed for performance improvement of systems with well-defined relative degree. The proposed learning algorithms are shown robust with respect to initial shifts. Furthermore, in the presence of an initial shift, it is proved that extended PD-type (PID-type) learning algorithm enables the system to possess asymptotic tracking capability, and the initial rectifying learning algorithm is able to achieve uniform convergence of the converged output to the desired one with a smooth transition.

REFERENCES

1. Rogers E, Owens DH. *Stability Analysis for Linear Repetitive Processes*. Springer: Berlin, 1992.
2. Arimoto S, Kawamura S, Miyazaki F. Bettering operation of robots by learning. *Journal of Robotic Systems* 1984; **1**(2): 123–140.
3. Hauser JE. Learning control for a class of nonlinear systems. In *Proceedings of the 26th IEEE Conference on Decision and Control*, Los Angeles, CA, U.S.A., December 1987; 859–860.
4. Sugie T, Ono T. An iterative learning control law for dynamical systems. *Automatica* 1991; **27**(4): 729–732.

5. Porter B, Mohamed SS. Iterative learning control of partially irregular multivariable plants with initial impulsive action. *International Journal of Systems Science* 1991; **22**(3): 447–454.
6. Ahn H-S, Choi C-H, Kim K-B. Iterative learning control for a class of nonlinear systems. *Automatica* 1993; **29**(6): 1575–1578.
7. Atkeson CG, McIntyre J. Robot trajectory learning through practice. In *Proceedings of the 1986 IEEE International Conference on Robotics and Automation*, San Francisco, CA, April 1986; 1737–1742.
8. Bondi P, Casalino G, Gambardella L. On the iterative learning control theory for robotic manipulators. *IEEE Journal of Robotics Automation* 1988; **4**: 14–22.
9. Heinzinger G, Fenwick D, Paden B, Miyazaki F. Stability of learning control with disturbances and uncertain initial conditions. *IEEE Transactions on Automatic Control* 1992; **37**(1): 110–114.
10. Moore KL, Dahleh M, Bhattacharyya SP. Iterative learning control: a survey and new results. *Journal of Robotic Systems* 1992; **9**(5): 563–594.
11. Jang T-J, Choi C-H, Ahn H-S. Iterative learning control in feedback systems. *Automatica* 1995; **31**(2): 243–245.
12. Qu Z, Zhuang H. Non-linear learning control of robot manipulators without requiring acceleration measurement. *International Journal of Adaptive Control and Signal Processing* 1993; **7**: 77–90.
13. Xu J-X, Qu Z. Robust iterative learning control for a class of nonlinear systems. *Automatica* 1998; **34**(8): 983–988.
14. Chen Y, Wen C, Sun M. A robust high-order P-type iterative learning controller using iteration tracking error. *International Journal of Control* 1997; **68**(2): 231–342.
15. Moore KL. Iterative learning control—an expository survey. *Applied and Computational Controls, Signal Processing, and Circuits* 1998; **1**: 151–214.
16. Phan MQ, Longman RW, Moore KL. Unified formulation of linear iterative learning control. In *AAS/AIAA Space Flight Mechanics Meeting*, Clearwater, FL, January 2000.
17. Chien CJ, Liu JS. A P-type iterative learning controller for robust output tracking of nonlinear time-varying systems. *International Journal of Control* 1996; **64**(2): 319–334.
18. Kuc T-Y, Lee JS, Nam K. An iterative learning control theory for a class of nonlinear dynamic systems. *Automatica* 1992; **28**(6): 1215–1221.
19. Owens DH. Universal iterative learning control using adaptive high-gain feedback. *International Journal of Adaptive Control and Signal Processing* 1993; **7**: 383–388.
20. Xu J-X, Wang XW, Lee TH. Analysis of continuous iterative learning control systems using current cycle feedback. In *Proceedings of the American Control Conference*, Seattle, Washington, U.S.A., June 1995; 4221–4225.
21. Chen Y, Wen C, Gong Z, Sun M. An iterative learning controller with initial state learning. *IEEE Transactions on Automatic Control* 1999; **44**(2): 371–376.
22. Porter B, Mohamed SS. Iterative learning control of partially irregular multivariable plants with initial state shifting. *International Journal of Systems Science* 1991; **22**(2): 229–235.
23. Choi C-H, Jeong G-M. Iterative learning control for linear nonminimum-phase systems based on initial state learning. In *Proceedings of the 3rd Asian Control Conference*, Shanghai, China, July 2000; 1872–1877.
24. Arimoto S, Kawamura S, Miyazaki F. Convergence, stability and robustness of learning control schemes for robot manipulators. *In Recent Trends in Robotics: Modelling, Control, and Education*, Jamishidi MJ, Luh LY, Shahinpoor M. (eds). Elsevier: NY, U.S.A., 1986; 307–316.
25. Saab SS, Vogt WG, Mickle MH. Learning control algorithms for tracking "slowly" varying trajectories. *IEEE Transactions of Systems, Man, and Cybernetics* 1997; **27**(4): 657–670.
26. Wang D, Soh YC, Cheah CC. Robust motion and force control of constrained manipulators by learning. *Automatica* 1995; **31**(2): 257–262.
27. Arimoto S, Naniwa T, Suzuki H. Selective learning with a forgetting factor for robotic motion control. In *Proceedings of the 1991 IEEE International Conference on Robotics and Automation*, Sacramento, CA, U.S.A., 1991; 728–733.
28. Saab SS. On the P-type learning control. *IEEE Transactions on Automatic Control* 1994; **39**(11): 2298–2302.
29. Lee H-S, Bien A. Study on robustness of iterative learning control with non-zero initial error. *International Journal of Control* 1996; **64**(3): 345–359.
30. Park K-H, Bien Z. A generalized iterative learning controller against initial state error. *International Journal of Control* 2000; **73**(10): 871–881.
31. Park K-H, Bien Z, Hwang D-H. A study on the robustness of a PID-type iterative learning controller against initial state error. *International Journal of Systems Science* 1999; **30**(1): 49–59.
32. Hillenbrand S, Pandit M. An iterative learning controller with reduced sampling rate for plants with variations of initial states. *International Journal of Control* 2000; **73**(10): 882–889.
33. Sun M, Wang D. Iterative learning control with initial rectifying action. *Automatica* 2002, to appear.
34. Isidori A. *Nonlinear Control Systems*. Springer: Berlin, 1995.

