International Journal of Electrical and Computer Engineering (IJECE)
Vol. 4, No. 6, December 2014, pp. 962~973
ISSN: 2088-8708

Adaptive PID Type Iterative Learning Control

Sara Zamiri*, Ali Madady, Hamid-Reza Reza-Alikhani
*Department of Control Engineering, Science and Research Branch, Islamic Azad University, Boroujerd, Iran

Article Info

Article history:

Received Jun 27, 2014
Revised Sep 10, 2014
Accepted Sep 30, 2014

Keyword:

Adaptive control
Iterative learning control
Monotonic convergence
PID type ILC

ABSTRACT

In this paper, an adaptive PID-type iterative learning control scheme is proposed for tracking problem in repetitive systems with unknown parameters. In this scheme, we use a combination of an optimal PID-type iterative learning controller and projection like adjusting algorithm that is based on tracking error which decreases by iterations increment. Layapunov method is used to convergence analysis of the presented scheme, and convergence condition is obtained in term of algorithm step size range. The effectiveness of proposed technique is illustrated by simulation results.

Corresponding Author:

Sara Zamiri
Department of Control Engineering, Science and Research Branch
Islamic Azad University, Boroujerd, Iran.
Email: s.zamiri1365@yahoo.com

1. INTRODUCTION

There are many industrial applications, that the system must periodically do a certain task over a finite trial length, such as in machine assembly by robot manipulators, chemical batch processes, and many other similar examples. Now, if human operators perform such this task repeatedly, they will learn to do their job better and better. This is because of human's learning and adaptive ability. this kind of learning is called iterative learning control (ILC) [1-3], which was first introduced by Arimoto et al. in 1984 [1]. The important characteristic of ILC is using information that are recorded at each iteration to adjust the control signal in an attempt to reduce the tracking error obtained during the next iteration, where by increasing the numbers iterations the tracking error will convergence to zero [4]. The operation of ILC in controlling repetitive systems with unknown parameters creates adaptive ILC algorithms. In [5], some adaptive some iterative learning control schemes for trajectory tracking of robot manipulators, with unknown parameters, is proposed. Note that many of the proposed adaptive ILC algorithms are combination of adaptive controllers and non-adaptive ILC algorithms. Accordingly in [6], by ILC algorithm a standard model reference scheme is expanded to continuous-time SISO linear time-invariant systems which perform repetitive tasks. In [7], a new adaptive switching learning control approach, which is called adaptive switching learning PD control law, was proposed that it has the ability of both learning and adaptive. A self-tuning iterative learning control approach in [8] was proposed for linear time-varying unknown systems. In [9], an adaptive PID learning controller was presented which composed of an adaptive PID feedback control scheme and a feed forward input learning scheme. Combines both concept of model reference adaptive control and ILC was proposed in [10] for unknown linear repeatable systems. An adaptive PI-type ILC scheme was presented in [11], without any prior knowledge of system parameters. Based on an estimation procedure using a Kalman filter and an optimization of a quadratic criterion is presented in [12], an adaptive Iterative Learning Control (ILC). A recent research [13] studied the optimal design of PID-type ILC for a discrete-time linear repetitive system.

By expanding the results of [13] to unknown system, a new control algorithm called adaptive PID-type iterative learning control that is the main debate of this paper.

The outline of the paper is as follows. In Section 2, some necessary definitions of the problem are given. A summary of the structure of PID type ILC and its parameter optimal design is presented in section 3. In section 4, an adaptive PID-Type ILC and its convergence operation is given. In section5, simulation results are presented to illustrate the effectiveness of the proposed method. The last section concludes the paper.

2. PROBLEM FORMULATION AN PRELIMINARIES

Let us intriduce subscript *'j'* and *'i'* as repetition (or operation/or iteration) and time during a given repetition of the system respectively where both *j* and *i* are integers, and $i \in [0, M]$. In this paper, we consider that the plant to be controlled is a discrete-time, linear, time-invariant, single-input single-output systems and described as follow:

$$\begin{cases} x_j(i+1) = Ax_j(i) + Bu_j(i), \\ y_j(i) = Cx_j(i), \\ x_j(0) = x_0, \\ \quad i = 0,1,\dots,M, \quad j = 0,1,\dots \end{cases} \tag{1}$$

Where $x \in \mathbb{R}^n$ is the state vector, $u \in \mathbb{R}^n$ and $y \in \mathbb{R}^n$ are input and output of the system respectively. *A*, *B*, and *C* are real-valued coefficients with appropriate dimensions. Also x_0 is the system initial condition. In this part, consider (1) and make the following reasonable assumptions:

(A1) The matrixes A, B and C are known.

(A2) The scalar *CB* is nonzero.

(A3) The system initial condition x_0 is unknown.

Under iterative learning control strategy, the error between the given desired output trajectory $y_d(i)$ and the system actual output $y_j(i)$ become smaller by increasing the numbers of repetition, so that following tracking can be establish:

$$\lim_{j\to\infty} \left(y_d(i) - y_j(i)\right) \qquad \text{for } 1 \le i \le M \tag{2}$$

Because only finite time intervals ($M < \infty$ samples) are comsidered output trajectory $y_d(i)$ form by building super vectors[1] $U(j)$ and $Y(j)$ form $u_j(i)$ and $y_j(i)$ as follows:

$$\begin{aligned} &U(j) = [u_j(0)\ u_j(1)\ u_j(2)\ \dots\ u_j(M-1)]^T \\ &Y(j) = [y_j(1)\ y_j(2)\ y_j(3)\ \dots\ y_j(M)]^T \end{aligned} \tag{3}$$

Where *T* denotes the transpose.

From (1) the following relation obtained easily:

$$Y(j) = H_p U(j) + H_x x_0 \tag{4}$$

Where H_p and H_x are the following matrixes:

$$H_x = \begin{bmatrix} CA \\ CA^2 \\ CA^3 \\ \vdots \\ CA^M \end{bmatrix}, \quad H_p = \begin{bmatrix} h_1 & 0 & 0 & & 0 \\ h_2 & h_1 & 0 & \cdots & 0 \\ h_3 & h_2 & h_1 & & 0 \\ & \vdots & & \ddots & \vdots \\ h_M & h_{M-1} & h_{M-2} & \cdots & h_1 \end{bmatrix} \tag{5}$$

Where h_k denotes the standard Markov parameters of the system (1), that is:

$$h_k = CA^k\text{-}1B \text{ for } k = 1, 2, \dots, M \tag{6}$$

[1] The super-vectors are marked by the elimination of the argument time.

Let us the operator T to map the vector h to a lower triangular Toeplitz matrix H_p, $H_p = T(h)$ that vector h is as follow:

$$h = [h_1 \quad h_2 \quad h_3 \quad \ldots \quad h_M]^T \tag{7}$$

Comment 1. We consider assumption (A2) is a standard assumption in ILC design which guarantees the existence of the learning gains. That is $h_1 = CB \neq 0$. This is not really a restriction because it can be satisfied by choosing a proper sampling period in discretizing the continuous-time systems.
Using (4) one can write:

$$Y(j+1) = y(j) + H_p Y(j) \qquad j = 0, 1, \ldots \tag{8}$$

Where:

$$V(j) = U(j+1) - U(j) \tag{9}$$

From (8) we can get:

$$Y_d - Y(j+1) = Y_d - Y(j) - H_p V(j) \tag{10}$$

The desired output trajectory y_d and the error $e_j(i) = y_d(i) - y_j(i)$ can be also written as below vectors:

$$\begin{aligned} Y(d) &= [y_d(1)\ y_d(2)\ y_d(3)\ \ldots\ y_d(M)]^T \\ E(j) &= [e_j(1)\ e_j(2)\ e_j(3)\ \ldots\ e_j(M)]^T \end{aligned} \tag{11}$$

Therefore relation (10) can be rewritten as follows:

$$E(j+1) = E(j) - H_p V(j) \qquad j = 0, 1, \ldots \tag{12}$$

The above relation is the dynamics of the error vector in the repetition domain.

3. PID TYPE ILC AND ITS PARAMETER OPTIMAL DESIGN

3.1. PID Type Iterative Learning Control

According to the [13] PID-type ILC is defined as follow:

$$U_{j+1}(i) = U_j(i) + k_p e_j(i+1) + k_i \sum_{m=1}^{i+1} e_j(m) + k_d \left(e_j(i+1) - e_j(i) \right), \quad i = 0, 1, \ldots, M\text{-}1, \quad j = 0, 1, \ldots \tag{13}$$

Where, k_p, k_i and k_d are PID learning gains (parameter/coefficient), which are called proportional, integration and derivative learning gains respectively.
Using vectors representation (9) and (11), we can rewrite the above relation like compact form of the following formula:

$$V(j) = k_p E(j) + k_i T_i E(j) + k_d T_d E(j) \tag{14}$$

Where:

$$T_1 = T([1\ 1\ 1\ \ldots\ 1]^T), \quad T_1 = \begin{bmatrix} 1 & 0 & 0 & \cdots & 0 \\ 1 & 1 & 0 & & 0 \\ 1 & 1 & 1 & & 0 \\ \vdots & & & & \vdots \\ 1 & 1 & 1 & \ldots & 1 \end{bmatrix} \tag{15}$$

$$T_d = T([1\ -1\ \ 0\ \dots\ 0]^T),\ T_d = \begin{bmatrix} 1 & 0 & 0 & \dots & 0 & 0 \\ -1 & 1 & 0 & \dots & 0 & 0 \\ 0 & -1 & \ddots & & & \vdots \\ \vdots & & \ddots & 1 & 0 & 0 \\ 0 & 0 & & -1 & 1 & 0 \\ 0 & 0 & \dots & 0 & -1 & 1 \end{bmatrix}$$

3.2. Convergence Analysis

The proposed ILCS is said to be convergent if the learning error approaches an infinitesimal value after sufficient learning iterations. Mathematically the following two definitions and theorems are given.

Definition 1. For proposed ILCS can be shown to converge in the sense that as $j \to \infty$ we have $y_j(i) \to y_d(i)$ for all $i \in [0,M]$, for arbitrary initial conditions, such that (2) holds, meaning:

$$\lim_{j\to\infty} E(j) = 0 \tag{16}$$

Theorem 1. ILCS is convergent if and only if learning gains k_p, k_i and k_d satisfy the inequality as follows:

$$\left|1 - h_1(k_p + k_i + k_d)\right| < 1 \tag{17}$$

Proof: see [13]

Comment 2. According to comment 1, since scalar $h_1 \triangleq CB$ is nonzero it can be find numerous real numbers for learning gains which they satisfy inequality (17).

Definition 2. The proposed ILCS is called monotonically convergent, if for any E(0) the following condition hold:

$$\|E(j+1)\|_\lambda \le \|E(j)\|_\lambda \tag{18}$$
$$\text{for } \lambda = 1, 2, \infty \quad \text{and} \quad j = 0, 1, 2, \dots$$

In particular, $\|E(j+1)\|_\lambda = \|E(j)\|_\lambda$ if and only if either $E(j) = 0$, that $\|\ \ \|_\lambda$ denotes the λ-norm.
In theorem 1 give us a sufficient and necessary condition for the presented learning process. Note that, this condition does not guarantee the convergence to monotonic. Thus, theorem 2 is presented for monotonic convergence. In this theorem, an optimal method is used for choosing k_p, k_i and k_d.

Theorem 2. The presented ILCS has monotonic convergence, with maximum desired convergence rate, if the learning gains k_p, k_i and k_d are chosen as follows:

$$K = \begin{bmatrix} k_p \\ k_i \\ k_d \end{bmatrix} = \begin{bmatrix} h^T h & h^T h_i & h^T h_d \\ h_i^T h & h_i^T h_i & h_i^T h_d \\ h_d^T h & h_d^T h_i & h_d^T h_d \end{bmatrix}^{-1} \begin{bmatrix} h_1 \\ h_1 \\ h_1 \end{bmatrix} = h_1 (H^T H)^{-1} \begin{matrix} 1 \\ 1 \\ 1 \end{matrix} \tag{19}$$

That, $K \in \mathbb{R}^3$ and also $H \in \mathbb{R}^{mx3}$ is defined as follow:

$$h_i = T_i h = \begin{bmatrix} h_1 \\ h_1 + h_2 \\ h_1 + h_2 + h_3 \\ \vdots \\ \sum_{i=1}^{M} h_1 \end{bmatrix}, h_d = T_d h = \begin{bmatrix} h_1 \\ h_2 - h_1 \\ h_3 - h_2 \\ \vdots \\ h_M - h_{M-1} \end{bmatrix}, H = [h\ \ h_1\ \ h_d]_{Mx3} \tag{20}$$

Proof: see [13].

4. ADAPTIVE PID TYPE ILC

In this part, we need to consider these conditions:

(B1) All the system parameters, namely the matrix A, B and C, are unknown.

(B2) The scalar CB is nonzero.

Here, according to (B1), Markov parameters of the system (1), that is $h = [h_1\ h_2\ h_3\ h_M]^T$ are unknown and the relation (19) is useless. So, in this case at first vector h should be estimated and then in order to determine learning gains, we use the relation as follow:

$$K(j) = \begin{bmatrix} k_p(j) \\ k_i(j) \\ k_d(j) \end{bmatrix} = \begin{bmatrix} \hat{h}^T\hat{h}(j) & \hat{h}^T\hat{h}_i(j) & \hat{h}^T\widehat{h_d}(j) \\ \hat{h}_i^T\hat{h}(j) & \hat{h}_i^T\hat{h}_i(j) & \hat{h}_i^T\widehat{h_d}(j) \\ \hat{h}_d^T\hat{h}(j) & \hat{h}_d^T\hat{h}_i(j) & \hat{h}_d^T\widehat{h_d}(j) \end{bmatrix}^{-1} \begin{bmatrix} \hat{h}_i(j) \\ \hat{h}_i(j) \\ \hat{h}_i(j) \end{bmatrix} = \hat{h}_i(j)(\widehat{H}(j)^T\widehat{H}_i(j))^{-1}\begin{bmatrix} 1 \\ 1 \\ 1 \end{bmatrix} \quad (21)$$

Hence, the control law (13), change to:

$$U_{j+1}(i) = U_j(i) + k_p e_j(i+1) + k_i(j)\sum_{m=1}^{i+1} e_j(m) + k_d(j)\left(e_j(i+1) - e_j(i)\right), \quad (22)$$

$i = 0, 1, \ldots, M\text{-}1, \quad j = 0, 1, \ldots$

Or:

$$V(j) = k_p(j)E(j) + k_i(j)T_iE(j) + k_d(j)T_dE(j)$$

Where $\hat{h}(j)$, $\hat{h}_1(j)$ and $\hat{h}_d(j)$ are, respectively, the estimations of h , h_i, and h_d in the j th iteration, that is:

$$\hat{h}(j) = [\hat{h}_1(j)\ \ \hat{h}_2(j)\ \ \hat{h}_3(j) \ldots \hat{h}_M(j)]^T \quad (23)$$

And:

$$\hat{h}_i(j) = T_i\hat{h}(j), \qquad \hat{h}_d(j) = T_d\hat{h}(j)$$

The $\hat{h}(j)$ is determined by a suitable method so that according to the assumption B2, following condition holds for all $j \in \{0,1,\ldots\}$

$$\hat{h}_1(j) \neq 0 \quad (24)$$

Until the learning gains $k_p(j)$, $k_i(j)$ and $k_d(j)$ always exist.
The next step is to establish an online adaptive algorithm for estimating h so that (24) hold. For this purpose let consider:

$$\hat{h}(j+1) = \hat{h}(j) + \Delta\hat{h}(j) \quad (25)$$

Where $\Delta\hat{h}(j)$ is a modifier term, which must be determined in a suitable method.
In order to determination of the modifier term, (12) is rewritten as following form:

$$E(j+1) = E(j) - W(j)h \quad (26)$$

Where:

$$W(j) = T\left([v_j(0)\ \ v_j(1)\ \ v_j(2)\ \ \ldots v_j(M-1)]^T\right) \quad (27)$$
$$v_j(i) = u_{j+1} - u_j(i) \qquad 0 \le i \le M-1$$

By using $\hat{h}_1(j)$ estimated $E(j+1)$ as the follow:

$$\hat{E}(j+1) = E(j) - W(j)\,\hat{h}(j) \quad (28)$$

From the difference of relation (26) and (28), we have:

$$\Psi(j) = W(j)(\hat{h}(j) - h) \quad (29)$$

Where $\Psi(j) \triangleq E(j+1) - \hat{E}(j+1)$.

Now, the purpose is determination of modifier term $\Delta\hat{h}_j$ in (25), so that value of vector $\Psi(j)$ decrease when the number of iteration increase, therefore, we define a quadratic cost function on $\Psi(j)$ as follow:

$$g(j) = \frac{1}{2}\Psi^T(j)P\Psi(j) \quad (30)$$

Where $P \in \mathbb{R}^{MxM}$ is a symmetric positive definite matrix.
Therefore, we rewrite the (25) as the following:

$$\hat{h}(j+1) = \hat{h}(j) + \mu(j)\left[-\frac{\partial g(j)}{\partial \hat{h}(j)}\right] \quad (31)$$

Where $\mu(j)$ is a positive scalar called algorithm step size, $\frac{\partial g(j)}{-\partial \hat{h}(j)}$ demonstrates the gradient of the $g(j)$ with respect to $\hat{h}(j)$.
Using (26) and (28) it is easy to derive that:

$$\frac{\partial g(j)}{\partial \hat{h}(j)} = W^T(j)P\Psi(\mathrm{j}) \quad (32)$$

So, from the (31) and (32) we can write the modifier term $\Delta\hat{h}(j)$ as follows:

$$\Delta\hat{h}(j) = \mu(j)Q(j) \quad (33)$$

Where:

$$Q(j) = W^T(j)P\,\Psi(j) \quad (34)$$

Finally, with considering the previous relations the adjusting algorithm (25) will become as follows:

$$\hat{h}(j+1) = \hat{h}(j) - \mu(j)Q(j) \quad (35)$$

In order to convergence analysis of the presented adaptive scheme, at first we examine the establishment of important condition (24), then, for this purpose the following steps are considered:

S1. In the choosing of the initial conditions for adjusting algorithm (35), we select $\hat{h}(0) \neq 0$.
S2. We provide some conditions so that from the following assumption
$\hat{h}_1(j) \neq 0$
The following result could be obtained:
$\hat{h}_1(j+1) \neq 0$
In order to provide the necessary conditions for step S2, we choose the step size of algorithm (35) that is $\mu(j)$ with considering the following constraint:

$$\mu(j) \neq \frac{\hat{h}_1(j)}{q_1(j)} \quad (36)$$

Where $q_1(j)$ is the first element of vector $Q(j)$.
Therefore by using the both previous steps and mathematical induction, condition (24) will be guaranteed for all $j \in \{0,1,...\}$.
The algebraic equations (21), control law (22), and the adjusting algorithm (35) are the main parts of the presented adaptive PID type ILC.
The convergence condition of the proposed adaptive PID type ILC is introduced in the theorem follows:
Theorem 3. The presented adaptive PID type ILC is convergent if the step size $\mu(j)$ in the algorithm (35) is chosen in the following interval:

$$0 < \mu(j) < \frac{2}{\lambda_{max}(P)\lambda_{max}(W(j)W^T(j))} \tag{37}$$

Where λ_{max} denotes the largest eigenvalue.

Proof of Theorem: let us consider the following Lyapunov function candidate:

$$F(j) = \hat{h}^T(j)h(j) \tag{38}$$

Where:

$$\hat{h}(j) = \hat{h}(j) - h \tag{39}$$

Now, the difference of the Lyapunov function (38) is given by

$$\Delta F(j) = F(j+1) - F(j) = -\Psi T(j)R(j)\Psi(j) \tag{40}$$

Where $R(j)$ is the following symmetric matrix:

$$R(j) = 2\mu(j)P - \mu^2(j)PW(j)W^T(j)P \tag{41}$$

It is easy to show that if $\mu(j)$ is in the interval (37), then the matrix $R(j)$ will be positive definite, it can be ensured that:

$$\Delta F(j) \leq 0 \tag{42}$$

That is $F(j)$ is a non-increasing function along j direction and hence $\hat{h}(j)$ will be bounded. Also since $F(j)$ is a nonnegative sequence, then from (42), we can obtain:

$$\lim_{j>\infty} \Delta F(j) = 0 \tag{43}$$

Since $R(j)$ is a symmetric and positive definite matrix, equation $\Delta F(j) = 0$ implies $\Psi(j) = 0$, then, from (43) we can show that:

$$\lim_{j>\infty} \Psi(j) = 0 \tag{44}$$

For sufficient large iteration, from (44), we have:

$$\Psi(j) = 0 \tag{45}$$

So, from algorithm (35) the constant values relative to iteration are obtained for $\hat{h}(j)$ like $h*$, that is:

$$\hat{h}(j) = h^* \text{for sufficiently large } j \tag{46}$$

In the basis of relation (21) the constant values are calculated for elements of vector $K(j)$, like k_p^* , k_i^* and k_d^*, as follows:

$$K(j) = \begin{bmatrix} k_p(j) \\ k_i(j) \\ k_d(j) \end{bmatrix} = \begin{bmatrix} k_p^* \\ k_i^* \\ k_d^* \end{bmatrix} = \begin{bmatrix} h^{*T}h^* & h^{*T}h_i^* & h^{*T}h_d^* \\ h_i^{*T}h^* & h_i^{*T}h_i^* & h_i^{*T}h_d^* \\ h_d^{*T}h^* & h_d^{*T}h_i^* & h_d^{*T}h_d^* \end{bmatrix} \quad \text{for sufficiently large } j \tag{47}$$
$$= h_1^*(H^{*T}H^*)^{-1}\begin{bmatrix} 1 \\ 1 \\ 1 \end{bmatrix}$$

Where:

$$h_1^* = T_1 h^* = \begin{bmatrix} h_1^* \\ h_1^* + h_2^* \\ h_1^* + h_2^* + h_3^* \\ \vdots \\ \sum_{l=1}^{M} h_l^* \end{bmatrix}, h_d^* = T_d h^* = \begin{bmatrix} h_1^* \\ h_2^* - h_1^* \\ h_3^* - h_2^* \\ \vdots \\ h_M^* - h_{M-1}^* \end{bmatrix}, H^* = [h^* h_i^* h_d^*] \tag{48}$$

From (29), (45) and (46) we have:

$$W(j)\hat{h}^* = 0 \tag{49}$$

Where:

$$\tilde{h}^* = h^* - h = [\tilde{h}_1^* \quad \tilde{h}_2^* \quad \dots \quad \tilde{h}_M^*] \tag{50}$$

We consider two different cases:
Case 1. The scalar $\tilde{h}_1^* = h_1^* - h_1$ is nonzero
In this case from (27) and (49) the following conclusions hold:

$$v_j(i) = 0 \text{ for } i = 0,1,\dots, M-1 \text{ and sufficiently large } j \tag{51}$$

By substituting for $k_p(j)$, $k_i(j)$ and $k_d(j)$ from (47) and for $v_j(i) = u_{j+1}(i) - u_j(i)$ from (51) into (22), we can obtain:

$$\begin{cases} (k_p^* + k_i^* + k_d^*)e_j(1) - k_d^* e_j(0) = 0 \\ (k_p^* + k_i^* + k_d^*)e_j(2) - k_i^* e_j(1) - k_d^* e_j(1) = 0 \\ \vdots \\ (k_p^* + k_i^* + k_d^*)e_j(M) - k_i^* \sum_{i=1}^{M-1} e_j(i) - k_d^* e_j(M-1) = 0 \\ e_j(0) \triangleq 0 \end{cases} \tag{52}$$

Which can be written in view of (47) as follows:

$$k_p^* + k_i^* + k_D^* = \frac{[((h^* - h_i^*)^T h_i^* h_d^{*T} - h^{*T} h_i^* h_i^{*T})(h^* - h_d^*)}{\det(H^{*T}H^*)} + \frac{((h^* - h_i^*)^T h_d^* h_d^{*T} - h^{*T} h_d^* h_i^{*T})(h_i^* - h^*)}{\det(H^{*T}H^*)} + \frac{((h^* - h_i^*)^T h^* h_d^{*T} - h^{*T} h^8 h_i^{*T})(h_d^* - h_i^*)}{\det(H^{*T}H^*)} \tag{53}$$

Where:

$$\det(H^{*T}H^*) = h^{*T}[h^* h_i^{*T} h_d^* - h_d^* h_d^{*T} h_i^*) - h_i^* h_i^{*T}(h^* h_d^{*T} h_d^* - h_d^* h_d^{*T} h^*) + h_d^* h_i^{*T}(h^* h_d^{*T} h_i^* - h_i^* h_d^{*T} h^*)]$$

Since h_i^* is the final value of $\hat{h}_i(j)$ and according to condition (24) the amounts of $\hat{h}_i(j)$ are nonzero for all $j \in \{0,1,\dots\}$, one can conclude that:

$$h_i^* \neq 0$$

Also, from (48) we have:

$$\begin{cases} h^* \neq h_i^* \\ h^* \neq h_d^* \end{cases} \Rightarrow h_i^* \neq h_d^* \tag{55}$$

Then, from (53), (54) and (55) we can result that:

$$k_p^* + k_i^* + k_d^* \neq 0 \quad (56)$$

From (56), based on (52), it can be ensured that $e_j(i) = 0$. Therefore,

$$\lim_{j\to\infty} e(j) = 0 \quad (57)$$

Then, we can say that in this part the proposed adaptive is convergence.
Case 2. The scalar $\tilde{h}_1^* = h_1^* - h_1$ is zero.
From (22) and (47) we will have:

$$V(j) = k_p^* E(j) + k_i^* T_i E(j) + k_d^* T_d E(j) \quad \text{for sufficiently large } j \quad (58)$$

By substituting for $V(j)$ from (58) into (12), we can get:

$$V(j) = k_p^* E(j) + k_i^* T_i E(j) + k_d^* T_d E(j) \quad \text{for sufficiently large } j \quad (59)$$

Where $I \in \mathbb{R}^{MxM}$ is identity matrix.
Where $H_e = I - k_p^* H_p - k_i^* H_p T_i - k_d^* H_p T_d$.
That H_e is a lower triangular Toeplitz matrix, then, we have $He = T(h_e)$.
By considering the vector $K(j)$ and matrix H^* from (47) and (48) respectively and also by definition of vector $\alpha = [1 \ \ 0 \ \ 0 \ \ \dots \ \ 0]^T \in \mathbb{R}^M$ we can write:

$$h_e = \alpha - [h^* h_i^* h_d^*][k_p^* k_i^* k_d^*] \quad (60)$$

Where:

$$\begin{cases} h_{ei} = 1 - h_1^*\left(k_p^* + k_i^* + k_d^*\right) \\ h_{ei} = -h_i^* k_p^* - (h_i^* + \sum_{m=2}^{i} h_m^*) - (h_i^* - h_{i-1}^*) k_d^* \\ i = 2,3,\dots,M \end{cases} \quad (61)$$

Considering the low triangular form of H_e, leads to the following characteristic polynomial for it:

$$\Delta_{H_e}(\lambda) = \det(\lambda I - H_e) = (\lambda - h_{e1})^M \quad (62)$$

By using (53), and considering in this case $h_1^* = h_1$ after some manipulation, we obtain:

$$|h_{e1}| = \left|1 - h_1\left(k_p^* + k_i^* + k_d^*\right)\right| < 1 \quad (63)$$

Clearly, all eigenvalues of H_e are absolutely less than one, so we can say that H_e is stable matrix and the learning process will converge, that means:

$$\lim_{j\to\infty} e(j) = 0 \quad (64)$$

Here the proof of the theorem is completed.
Comment 3. For choosing the $\mu(j)$, we should consider both (36) and (37) conditions, then if $\frac{\hat{h}_i(j)}{\hat{q}_i(j)}$ place in the interval of (37), the $\mu(j)$ we should choose it not equal to $\frac{\hat{h}_i(j)}{\hat{q}_i(j)}$.

5. SIMULATION RESULTS

In this Section an illustrative numerical example is given to demonstrate the effectiveness of the presented ILC algorithm.

Let us consider a DC motor, which rotates a mechanical load as Figure 1, where its field winding current is constant, but its armature supply is variable.

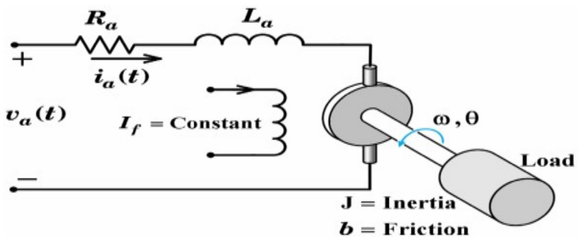

Figure 1. DC motor with constant field current

In this situation the block-diagram of the motor is as Figure 2 [14].

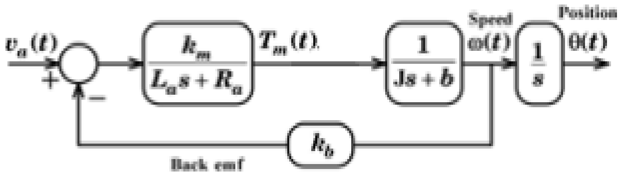

Figure 2. The motor block-diagram

Where R_a, L_a are the armature winding resistance and inductance respectively, k_m is the motor torque constant, J and b are the mechanical load inertia momentum and friction ratio respectively, k_b is the back EMF constant. Also $v_a(t)$, $i_a(t)$ are respectively the armature source voltage and current, ω(t) and θ(t) are the motor shaft rotational speed and angle respectively. Let us define the state variables and the output of the motor as follows:
State variables: $x(t) = [\theta(t) \quad \omega(t) \quad i_a(t)]^T$
Output: $y(t) = \theta(t)$
Now, by considering Figure 2 it is easy to obtain the state space equations of the motor as follows:

$$\begin{cases} \dot{x}(t) = Ax(t) + B_a v_a(t) \\ \quad y(t) = Cx(t) \end{cases}$$

Where $\dot{x}(t) = \triangleq \frac{dx}{dt}$, and:

$$A = \begin{bmatrix} 0 & 1 & 0 \\ 0 & -\frac{b}{j} & \frac{k_m}{j} \\ 0 & -\frac{k_b}{L_a} & -\frac{R_a}{L_a} \end{bmatrix}, B_a = \begin{bmatrix} 0 \\ 0 \\ \frac{1}{L_a} \end{bmatrix}$$
$$,C = [1 \quad 0 \quad 0]$$

It is desired to determine$v\ a\ (t\ )$, so that $y\ (t\ )$
Periodically tracks a given command signal
$y\ d\ (t\ )$ in time interval □0,$t\ f$ □ , such that as the iterations number
increases, the error between
$y\ (t\ )$ and
$y\ d\ (t\ )$ vanishes. The state equations of the motor should be discretized

in order to be ready for applying the presented ILC method. Let us choose the sampling period Ts □ 0.1 sec and the following parameters values:

$$R_a = 3.75\Omega, \quad L_a = 15x10^{-2}\text{H}, \quad k_m = k_b = 0.5\frac{\text{Nm}}{\text{A}}$$
$$b = 0.1\frac{\text{Nms}}{\text{rad}}, \quad \text{j} = 12x10^{-3}\text{kg.m}^2, \quad \text{t}_\text{f} = 10\text{sec}$$

The $x(0) = [\theta(0)\ \ \omega(0)\ \ i_a(0)]^T \triangleq x_0$ is selected as:

$x_0 = 0$

The obtained discrete state equations for the motor are as follows:

$$\begin{cases} x_j(i+1) = A_D x_j(i) + B_D v_{ai}(i) \\ y_j(i) = C_D x_j(i) \\ i = 0,1,\dots,M, \quad j = 0,1,\dots \end{cases}$$

Where:

$$M = \frac{t_f}{T_s} = 100$$
$$A_D = \begin{bmatrix} 1 & 0.0595 & 0.07121 \\ 0 & 0.2668 & 0.699 \\ 0 & -0.05592 & -0.0128 \end{bmatrix}$$
$$B_D = \begin{bmatrix} 0.02101 \\ 0.04747 \\ 0.2068 \end{bmatrix}, \quad C_D = [1\ \ 0\ \ 0]$$

The desired output trajectory, which is shown in Figure 3, is chosen as follows:

$$y_d(i) = i \sin(0.01 i \pi)$$

Motor input voltage at first iteration (say $j = 0$), that the controller has not any previous experience, is selected to be equal to 1. The matrix P is selected as identity matrix. The initial conditions and the step size of algorithm (35) are chosen as follows:

$$\hat{h}(0) = [0.5\ \ 0.5\ \ 0.5\ \ \dots 0.5]^T$$

$$\mu(j) = \frac{1.5}{\lambda_{max}(P)\lambda_{max}(W(j)W^T(j))}$$

The obtained trajectories for the motor rotational angle are shown in Figures 3 in various iterations. This figure shows that convergence speed is high and with increasing of the number of iteration, motor rotational angle rapidly convergence to the given desired output trajectory. Figure 4 shows that convergence is monotonic.

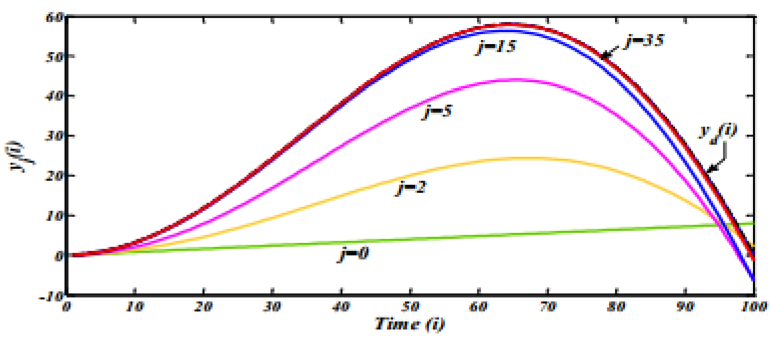

Figure 3. The desired output trajectory $y_d(i)$ and the motor shaft rotational angle

in the iterations j = 0, 2, 5, 15 and 35

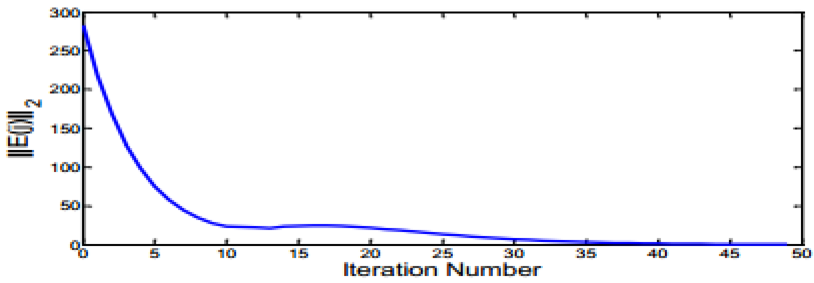

Figure 4. The norm 2 of error vector with respect to the iteration number j

6. CONCLUSION

In this paper an adaptive PID-type ILC is presented. In fact, it can be say that, this paper introduce a design method for optimal PID type iterative learning controller with iteration varying learning gains, for the repetitive systems with unknown parameters. In order to adjustment of learning gains an adjusting algorithm is proposed so that with using input-output data in iteration j, learning gains in the next iteration (iteration $j+1$) modified. Lyapunov method is proposed for convergence analysis and convergence condition is obtained in terms of adjusting algorithm step size. Finally, the result of computer simulation has demonstrated the effectiveness of the proposed adaptive scheme.

REFERENCES

[1] S. Arimoto, S. Kawamura, & F. Miyazaki, "Bettering operation of robots by learning", *Journal of Robotic Systems*, 1(2), 1984, 123–140.
[2] S. Arimoto, S. Kawamura, & F. Miyazaki, "Convergence, stability and robustness of learning control schemes for robot manipulators", in Recent Trends in Robotics: Modeling, Control, and Education (New York: Elsvier, 1986), 307–316.
[3] S. Kawamura, F. Miyazaki, & S. Arimoto, "Realization of robot motion based on a learning method", IEEE Transaction on Systems, Man and Cybernetics, 18 (1), 1988, 126–134.
[4] Xu, J. X. and Y. Tan, "*Linear and Nonlinear Iterative Learning Control*", Springer-Verlag: New York (2003).
[5] Tayebi, A. "Adaptive iterave learning control for robot manipulators". *Automatica*. 40(7), 1195C1203 (2004).
[6] Tayebi, A. "Model reference adaptive iterative learning control for linear system", *Int. J. Adapt. Control Sign. Proc.*, Vol. 20, pp. 475–489(2006).
[7] Ouyang, P. R., W. J. Zhang, and M. M. Gupta, "An adaptive swiching learning control method for trajectory tracking of robot manipulators", *Mechatronics*, Vol. 16, pp. 51–61 (2006).
[8] A. Madady, "A self-tuning iterative learning controller for time variant systems" *Asian Journal of Control*, vol. 10, no. 6, pp. 666-677, 2008.
[9] Kuc, T. Y., & Han, W. G, "An adaptive PID learning control of robot manipulators". *Automatica*, 36, 717–725, (2000).
[10] Madady, A., "A model reference adaptive iterative learning control system", *Proc. 32nd Annu. Conf. IEEE Ind. Electron. Soc.*, pp. 627–632, Paris, France (2006).
[11] A. Madady and H.-R. Reza-Alikhani, "Adaptive PI-type iterative learning control", in proceedings of the 5th IEEE International Conference on Intelligent systems, pp. 37-42, London, 2010.
[12] M. Norrlof, "An Adaptive Iterative Learning Control Algorithm with Experiments on an Industrial Robot" *IEEE Transactions on Robotics and Automation*, 18(2), pp.245-251, 2002.
[13] A. Madady, "PID type iterative learning control with optimal gains", *International Journal of Control Automation, and Systems*, vol. 6, no. 2, pp.194-203, 2008.
[14] R. C. Dorf, R. H. Bishop. "*Modern Control Systems*", (10th Edition, Prentice Hall, 2004).