Adaptive Iterative Learning Control Algorithm with Large Uncertainties in System Parameters

Habib Ghanbarpour Asl
Mechatronics Department, University of Turkish Aeronautical Association
Ankara, Turkey

Abstract

In this paper, an adaptive iterative learning control (AILC) algorithm has implemented by using the least squares approximation. A new method for calculating the learning gain of ILC algorithm has implemented. The ILC algorithm has been applied for a SISO linear time-invariant (LTI) dynamic system with unknown parameters, and a parameter identification algorithm is designed to optimize the accurate values of the unknown parameters and minimize the tracking error. A simulation study is used for testing the implemented method. Simulations show that AILC algorithm is suitable for linear systems that have unknown parameters, but the bounds of these parameters are limited. The controller is robust when the parameters have known bounds.

Keywords: *Iterative Learning Control, Parameter Identification, Unknown parameters, Repetitive system, Learning gain, Adaptive control system*.

1. Introduction

"Repetition is the mother of all learning", with this quotation, it can be deduced how a person gained his knowledge and experience, this concept has applied on the machines later to improve their performance with iteration progress. Iterative Learning Control (ILC) is classified as an intelligent control approach suitable for controlling systems that work in repetitive motion.

Even the iterative learning control idea started before the 1970 but the first publication was in 1974, and the first paper was in Japanese language (Formation of High-Speed Motion Pattern of Mechanical Arm by Trial) by Masaru Uchiyama in 1978 [1]. In the eighties, Arimoto et al. igorously formulated the Iterative Learning Control problem [2]. Since 1992, the study and researchers in ILC have progressed quickly. On one hand, important work has been shown and stated in the main area of developing and analyzing new algorithms of ILC. Researchers also have recognized that making integration between ILC and other control theories might give better controllers that show the desired performance, which is impossible to be shown by any individual approach [3].

Since the birth of the idea of ILC in early 1980's, the history of ILC can be separated into two periods. The first period was between the early 1980s' and early 1990's which represents the linearly increasing period of ILC, whether in terms of publications and reports in theory or in applications. The second period was from early 1990's so far, nevertheless, the activities of research in ILC undergo a nonlinear (exponential) increase [4].

The ILC is one of the hopeful algorithms for the control systems that working according to self-learning concept. It is an algorithm capable to track the required trajectory within a certain period of time [5]. The intelligent control has many branches, one of them is the iterative learning control (ILC), which can be defined as an effective control tool for improving the response of the repetitive motion performance of uncertain dynamic systems [6].

The main idea of ILC is taking the information of the past iteration and performing the current iteration by depending on this information, ILC will apply a simple algorithm repetitively to an unknown plant until reach to the perfect tracking [8], this is how ILC is learning from past experiences. To be specific, in the ILC, the desired trajectory could be tracked by the output of the dynamic system asymptotically along the iteration index because the control signal is updated iteratively using information generated from previous iterations [7].

The error is reduced from trail to trail by increasing the input signal until approach zero error in the output and the desired graph. It is similar to the human example, expressed before, as much repeat the activity as much gets more experience and rise up his physical performance. Impression and similarity are the qualities that enable a human to get his knowledge. In machines, the matter is not far away from this idea, where the initial setup, uniform sampling, fixed time point, repetitive desired trajectory and another setup could be taken similar to what mentioned in the case of humane qualities [6].

The fixed learning laws that the traditional ILC algorithm depend on are static and does not update themselves from the current iteration data, the only way to solve this problem is adjusting the learning law after each iteration [8], that issue make it has no big flexibility to deal with the dynamic systems that have unknown parameters [5], where the traditional ILC algorithm can reduce the error rate and get the convergence by depending on the information, from the previous iterations, but this algorithm has no ability to deal with big differences that may happen in the real parameters of the actual dynamic system.

That means, the traditional ILC algorithm to compute the error rate by depending on the real parameters of the dynamic system, the result of this computation will still accurate as long as the parameters of the real system are known and have no change, but in case of an existing one or more of unknown parameters the ILC algorithm will give wrong results which may lead to instability, because of that the ideas of finding adaptive iterative learning control (AILC) algorithms have appeared by merging the ILC algorithms with estimation or statistical theories.

Messer et al. (1990) used a new adaptive learning law, his algorithm depended on integral transforms [9]. French and Rogers (1998) decreased parameters by considering a system included adaptively estimated parameters in a finite time horizon [10]. Owens et al. (1998) implemented convergence/stability norms by using the current trail feedback for common adaptive learning control system and applied this algorithm for linear MIMO state space system by using high gain [11]. French et al. (1999) used of signal (BC) in the adaptation of the learning gain to provide a scheme of learning control [12]. Phan and Frueh (1999) used a reference model in a new ILC algorithm to be the leader of the learning process by enabling the controller to choose good knowledge from previous experience to achieve the desired properties [13]. Choi and Lee (2000) used a different advantage in the time domain, they have been able to estimate the uncertain parameters of AILC scheme by reducing the errors after recognizing the troubles with iteration progress [14]. Lee et al. (2002) applied the original adaptive control ideas after produced an algorithm by using the linear time-invariant system with no disturbances and by developing the learning laws with repetitive disturbances [15]. Chen and Jion (2002) obtained the convergence of the SISO nonlinear system by using a Nussbaum-type function to manipulate a fully unknown feedback high-frequency gain of the adaptive iterative learning control algorithm [16]. Owens and Feng (2003) they estimated the new parameters by using the quadratic performance index in a new ILC method [17]. Chiang et al. (2004) obtained a new output tracking error model to generate an output-based AILC algorithm by using filtered signals, these signals are the input and the output signals of the plan, and this method is applicable on the repetitively moving linear dynamic system with unknown parameters [18]. Ashraf et al. (2008) applied a merged algorithm on a practical simulation to show the robustness of it, the steepest descent approach has used this algorithm to find the values of the optimal gain matrix [5]. Stearns et al. (2009) used an identification technique to produce an iteration varying learning filter, they reduce the 2-norm error speedily by changing the learning filter within iteration [8]. Oh et al. (2015) introduced an AILC algorithm which can be applied to the discrete linear time-invariant LTI, and also it can be applied on the stochastic system with batch-varying reference trajectories BVRT [19].

In this paper, we made a combination of Arimoto's ILC law [2] and error output method of least square approximation [20] to implement an adaptive iterative learning control (AILC).

2. Problem Formulation

The learning algorithm that will be used in this paper is workable with the linear dynamic system in the discrete-time form. Consider the main continuous-time linear dynamic system model as equation (1):

$$\begin{aligned} \dot{x} &= Ax + Bu \\ y &= Cx \end{aligned} \tag{1}$$

The discretized form of the model in the equation (1) is:

$$\begin{aligned} x_k(i+1) &= \Phi x_k(i) + \Theta u_k(i) \\ y_k &= C x_k(i) \end{aligned} \tag{2}$$

Where $x_k(0) = x_0$ is the system initial state which it is unknown values, k and i are integers, $k = 0, 1, \ldots$ denotes iteration index, i is the time instance, $i \in \{0, 1, \cdots, N_p\}$, and N_p are the expected iteration length. $x_k(t) \in \mathbb{R}^n$, $u_k(t) \in \mathbb{R}^p$, and $y_k(t) \in \mathbb{R}^q$ refers to the state, input, and output, respectively. Φ, Θ, and C are matrices in discrete-time form with appropriate dimensions, where the matrix $C\Theta$ is invertible to guarantee the existence of the learning gain in the ILC, and the matrices Φ and Θ have unknown parameters. R is the given desired output trajectory.

The matrices Φ and Θ of the equation (2) are functions of the continuous-time system matrices A and B , and are implemented as:

$$\Phi = e^{A\Delta t} = \sum_{j=0}^{l} \frac{\Delta t^j}{j!} A^j \tag{3}$$

$$\Theta = A^{-1}(e^{A\Delta t} - I)B = \sum_{j=1}^{l} \frac{\Delta t^j}{j!} A^{j-1} B \tag{4}$$

Where, Φ is the state transition matrix, l represents the number of the terms of the Taylor series $2 < l < \infty$, the larger number of l causes the most accurate [21].

Adaptive iterative learning control AILC algorithm will adapt the unknown parameters and determine the control input of the system (2), so as the iteration number increasing, the error between $y_k(i)$ and $R(i)$ getting smaller so it can be expressed as:

$$\lim_{k\to\infty}(R(i) - y_k(i)) = 0, \text{ for } i = 1, 2, \cdots, N_p \qquad (5)$$

It is useful in the analysis to replace the linear dynamic system in (2) by a matrix model relating a vector of inputs to a vector of outputs for each trail, so that, the plant (2) could be written equivalently as:

$$Y_k = GU_k + dx_{k-1} \qquad (6)$$

Where U is the input vector, Y is the output vector, and G is the lifted plant model consists of the Markova parameters of the plant (2)

$$\begin{aligned} U_k &= [u_{k(i-1)} \quad u_{k(i)} \quad u_{k(i+1)} \quad \cdots \quad u_{k(i+N_p-1)}]^T \\ Y_k &= [y_{k(i)} \quad y_{k(i+1)} \quad y_{k(i+2)} \quad \cdots \quad y_{k(i+N_p)}]^T \\ d &= [C\Phi \quad C\Phi^2 \quad C\Phi^3 \quad \cdots \quad C\Phi^{N_p+1}]^T \\ R &= [r_{(i)} \quad r_{(i+1)} \quad r_{(i+2)} \quad \cdots \quad r_{(i+N_p)}]^T \end{aligned} \qquad (7)$$

The matrix G is defined as:

$$G = \begin{bmatrix} C\Theta & 0 & 0 & \cdots & 0 \\ C\Phi\Theta & C\Theta & 0 & \cdots & 0 \\ C\Phi^2\Theta & C\Phi\Theta & C\Theta & \cdots & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ C\Phi^{N_p}\Theta & C\Phi^{N_p-1}\Theta & C\Phi^{N_p-2}\Theta & \cdots & C\Theta \end{bmatrix} \qquad (8)$$

The tracking error at iteration k is defined as:

$$e_k(i) = r(i) - y_k(i) \qquad (9)$$

Where r is the reference signal.

The ILC algorithm is going to be used in this paper will be Arimoto's base algorithm as:

$$U_{k+1} = U_k + \Gamma e_k \qquad (10)$$

By substituting the equation (6) in (9) then plugging the result in it leads to the input update equation in (10):

$$U_{k+1} = U_k + \Gamma(R - GU_k - d) = f(U_k) \qquad (11)$$

Where Γ is a learning gain matrix responsible for ensuring the convergence of the iteration. In the below for designing of learning matrix, we will review some definition and theorems.

Definition 1 (Contraction mapping) [25]:

Suppose the vector pace X with metric d, be a complete metric space. The mapping $T: X \to X$ is a contraction mapping if $d(Tx, Ty) \le \alpha\, d(x, y)$, $0 \le \alpha < 1$. Here the x and y are vectors in the X. Consequently, in a contraction mapping, the distance between the mapped of two vectors is lower than the distance between vectors.

Definition (Fixed point) [25]:

The vector x is called the fixed point of the mapping $T: X \to X$ in the vector space X, if $x = Tx$. In other words, if the transformation of the vector x is x then the fixed point of that transformation is the vector x.

Theorem (Banach fixed point theorem) [25]:

Let $T: X \to X$ be a contraction mapping from a vector space X to X, then the transformation T has a unique fixed point.

Consequently, the sequence $x_k = Tx_{k-1}$ will converge to the fixed point when $x_0 \in X$ [25]. The proof of this theorem is presented in the reference [25].

Normed Space:

Let X be a vector space which possesses a norm denoted by $\|.\|$, then the vector space is called normed space.

Theorem:

Suppose $T: R^n \to R^n$ is a linear mapping from the normed vector space R^n to itself with Euclidean norm ℓ_2. The mapping T is considered as $T(U) = U + \Gamma(R - GU - d)$. Here U, R and d are vectors in R^n. The metric d is defined as: $d(x, y) = \|x - y\|$. Here x and y are vectors in the normed space R^n. If the maximum eigenvalue of the matrix $I - \Gamma G$ is lower than the one, the sequence equation (11) will coverage to a unique fixed point.

Proof:

Suppose the U_1 and U_2 are vectors in the normed space R^n. $T(U_1)$ and $T(U_2)$ are mapped vectors. Then the distance between mapped vectors is:

$$\begin{aligned} &\|T(U_1) - (TU_2)\| = \\ &\|(I - \Gamma G)U_1 + \Gamma(R - d) - (I - \Gamma G)U_2 + \Gamma(R - d)\| = \|(I - \Gamma G)(U_1 - U_2)\| \\ &\qquad \le \|(I - \Gamma G)\|\,\|U_1 - U_2\| \end{aligned}$$

When the norm $M = (I - \Gamma G)$ is lower than the one, then the mapping will be a contraction mapping, so due to Banach fixed point theorem will has a unique fixed point. If the norm of the vector is the Euclidean norm, then the norm of the matrix is the largest singular value of that matrix here will be presented by σ_{max} [25]. Consequently, the sequence of (11) will converge when the maximum singular value of the matrix $(I - \Gamma G)$ is lower than the one. The matrix $(I - \Gamma G)$ is a square matrix, it yields:

$$\sigma_{max} = \sqrt{eig(MM^T)} = \sqrt{eig(v^{-1}\Lambda v v^T \Lambda v^{-T})} \qquad (123)$$

Where the Λ, v and v^{-T} are eigenvalues, eigenvectors and transpose of eigenvector inverse matrix of M respectively. Consequently, when the eigenvector matrix v is a unit matrix and eigenvalues are lower than the one, then the maximum singular value of M always will remain lower than the one. Then the sequence (11) will converge, because after each iteration the distance between the U_k and real fixed point will be reduced.

In summary, the convergence will be assured when the maximum absolute eigenvalue of the learning gain matrix Γ be less than one.

$$\|I - \Gamma G\| = \max_{1 \le i \le N_p} |\lambda_i|, \; v = I \tag{14}$$

The eigenvalue's definition will be used to implement the learning gain matrix Γ with respect to eigenvalue.

$$(I - \Gamma G)v = \Lambda v \tag{135}$$

as the eigenvector matrix is unit matrix, then the equation **Error! Reference source not found.**5) in matrix form will be written as

$$\Gamma = (I - \Lambda)G^{-1} \tag{146}$$

The equation (16) is applicable only if the lower triangular matrix is G an invertible matrix. The matrix G looks like a lower triangular matrix, but it is not. We will use the box lower triangular matrix. Appendix A represents how by a recursive algorithm we will find the inverse of that matrix because finding the inverse of large matrices is a problem.

3. Output Error identification Method

Error output method is a method is a recursive solution for least square approximation method. That is applicable for linear and nonlinear systems, even it is a stochastic or deterministic system. The method has the potential for use as a heart of an identification algorithm. Researchers have used in several fields like signal processing, control, and trajectory tracking [22]. In this paper, Error Output Method has used to find the output error value, and Newton-Raphson method has used to minimize that error [23]. The method is dealing with the evaluating measured output and estimated output and propagating observed error through the Newton-Raphson method for tuning the unknown parameters and reducing the output error.
For the dynamic system (2), the error output estimation approach minimizes the next cost function:

$$J(\hat{P}) = \frac{1}{2} \sum_{k=1}^{N_p} (\tilde{Y}_k - \hat{Y}_k)^T R^{-1} (\tilde{Y}_k - \hat{Y}_k) \tag{157}$$

Where $\hat{Y}_k$ is the response of the mathematical system model at iteration *k*, whit the estimated unknown parameters up to iteration k-1, and $\tilde{Y}_k$ is the measured response of the system. To minimize the equation (1517) we will the Newton-Raphson algorithm as:

$$\hat{P}_{k+1} = \hat{P}_k - \left[\nabla^2_{\hat{P}} J(\hat{P})\right]^{-1} \nabla_{\hat{P}} J(\hat{P}) \tag{16}$$

Where $\hat{P}_{k+1}$, $\nabla_{\hat{P}} J(\hat{P})$, and $\nabla^2_{\hat{P}} J(\hat{P})$ *are* the new update of the estimated parameters, the Jacobian matrix and the Hessian matrices of the cost function respectively. The first and second derivatives of the cost function with respect to the unknown parameter are:

$$\nabla_{\hat{P}} J(\hat{P}) = 2 \sum_0^{N_p} (-e_i{}^T R^{-1} \gamma) \tag{17}$$

and

$$\nabla^2_{\hat{P}} J(\hat{P}) = 2 \sum_0^{N_p} (\gamma^T R^{-1} \gamma) \tag{20}$$

Where $e_i = y - \hat{y}$ and γ is the sensitivity of output explained in the equation (18)

$$\gamma = \hat{C}_{\hat{P}_k} \xi_k + \frac{\partial g}{\partial \hat{P}_k} \tag{18}$$

Where $\partial g / \partial \hat{P}_k$ is the partial derivative of the system output function $g(\hat{P}_k) = C x_k(i)$ in the system (2), and ξ is the sensitivity of the state explained in (19)

$$\xi_{k+1} = \hat{\Phi}_{\hat{P}_{k+1}} \xi_k + \frac{\partial f}{\partial \hat{P}_k} \tag{19}$$

Where $\partial f / \partial \hat{P}_k$ is the partial derivative of the dynamic system output function $f(\hat{P}_k) = \Phi x_k(i) + \Theta u_k(i)$ in the system (2).

4. Adaptive Iterative Learning Control Algorithm

In this part, we will talk about the AILC which it will be a combination of ILC algorithm (11) and error output method (15) as described in Figure 1.

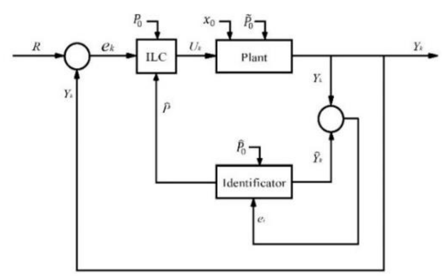

Figure 1: AILC algorithm

In this algorithm, ILC will calculate the first trail of the input U_k according to the initial main values of parameters P_i which have different is of parameters from of the measured values in the plant $\tilde{P}_i$, that will lead to an error in the output Y_k, the job of the identificator is to find an optimized parameters' values $\hat{P}_i$ and send it to ILC as a new values instead of the old ones so that will reduce the gap between P_i and $\tilde{P}_i$ then the errors will be reduced and the output Y_k will be enhanced.

IJCSI International Journal of Computer Science Issues, Volume 16, Issue 5, September 2019
ISSN (Print): 1694-0814 | ISSN (Online): 1694-0784
www.IJCSI.org https://doi.org/10.5281/zenodo.3465913

5. Simulation Results

In this part, an example is given to demonstrate the effectiveness of the (AILC) algorithm. By considering the variable k as the iteration number the obtained discrete state equations are as follows:

$$\left.\begin{array}{c} x_k(i+1) = \Phi x_k(i) + \Theta u_k(i) \\ y_k(t) = C x_k(i) \\ k = 0, 1, \ldots \\ i = 0, 1, \ldots, 1000 \end{array}\right\} \quad (20)$$

The dynamic system shown in Figure 2 will be used in this simulation which is a (spring-mass-damper) system

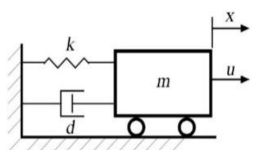

Figure 2: Spring-mass-damper dynamic system

The differential equation of the above system is

$$\ddot{x} + 2\zeta\,\omega_n \dot{x} + \omega_n{}^2 x = u \quad (21)$$

The state space model of this system in continuous-time form is

$$\begin{bmatrix} \dot{x} \\ \dot{v} \end{bmatrix} = \begin{bmatrix} 0 & 1 \\ -\omega_n^2 & -2\zeta\,\omega_n \end{bmatrix} \begin{bmatrix} x \\ v \end{bmatrix} + \begin{bmatrix} 0 \\ 1 \end{bmatrix} [u]$$
$$y = [1 \quad 0] \begin{bmatrix} x \\ v \end{bmatrix} \quad (22)$$

The discrete form matrices of the dynamic system after applying the equations (3) and (4) respectively, for $j = 2$ will be

$$\Phi = \begin{bmatrix} 1 & dt^2 + dt \\ -dt^2\omega_n^2 - dt\,\omega_n^2 & -2\omega_n\zeta dt^2 - 2\omega_n\zeta dt + 1 \end{bmatrix},$$
$$\Theta = \begin{bmatrix} dt^2 \\ -2\omega_n\zeta dt^2 + dt \end{bmatrix},$$

The initial value of the dynamic system $x_0 = 0$ and $v_0 = 0$ where x is the distance and v is the velocity of the mass m, lets $\Delta t = 0.001$, the unknown parameters are the dimension-less damping ratio ζ and the natural frequency of the system ω_n which will be considered in the vector form $P = [\zeta \quad \omega_n]^T$, let the actual values of the parameters $P = [0.10 \quad 0.20]^T$, the initial measured parameters $\tilde{P}_0 = [\tilde{\zeta} \quad \tilde{\omega}_n]^T$, where $\tilde{P}_0 = \hat{P}_0$, and the initial input $U_0 = 0$, the coefficient matrices are as below:

$$\Phi = \begin{bmatrix} 1.0000 & 0.0101 \\ -0.0001 & 0.9996 \end{bmatrix},$$
$$\Theta = \begin{bmatrix} 0.0001 \\ 0.0100 \end{bmatrix},$$
$$C = [1 \quad 0],$$

The desired output trajectory R shown in equation (23) will be applied to the dynamic system

$$R = \sum_{i=1}^{j} \frac{\sin(4.1i)t}{(i)!} \quad (23)$$

Where $j = 19$ and $i = 1, 3, \ldots j$, t is the time $0 < t < N$, and $N = 10\ sec$. The graph of the desired input is shown below in Figure 3.

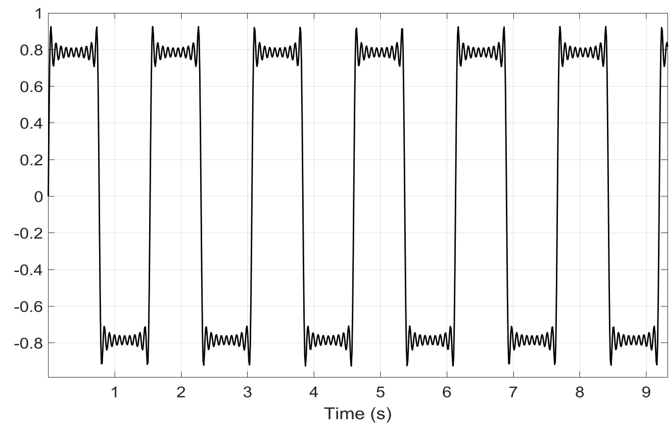

Figure 3: The desired output trajectory $R(i)$

The controller hasn't any previous experience, and the simulation is done according to the following cases.

Case 1: shows the response difference between ILC and (AILC) to the dynamic system when it has the parameters $\tilde{P}_0 = \hat{P}_0 = [5.0 \quad 5.0]^T$, the eigenvalue $\lambda = 0.99$, and the iteration number $k = 5$ shown in Figure 4.

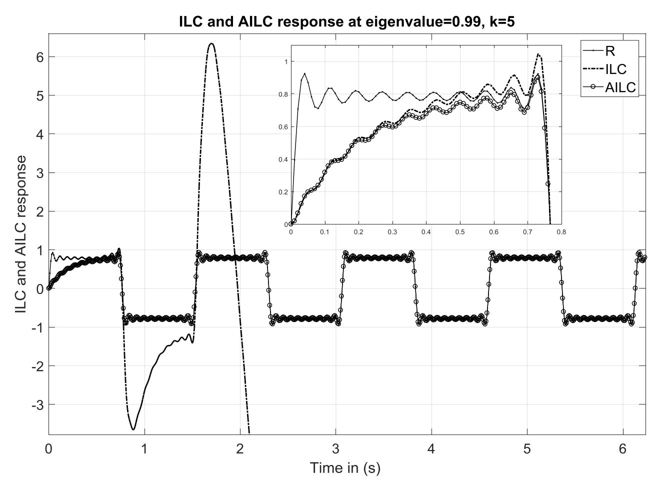

Figure 4: ILC vs AILC at $\lambda = 0.99$, k=5

IJCSI
www.IJCSI.org

Case 2: Shows the simulation of AILC to the dynamic system when the initial parameters $\hat{P}_0 = \hat{P}_0 = [0.08 \quad 0.22]^T$ at different eigenvalues for the equation (14) $\lambda = 0.99$, 0.95, 0.90 and at initial state values $x_0 = [0 \quad 0]$ and iteration number k=5.
The input signals u for this case at iterations $k = 5$ are shown in Figure 5

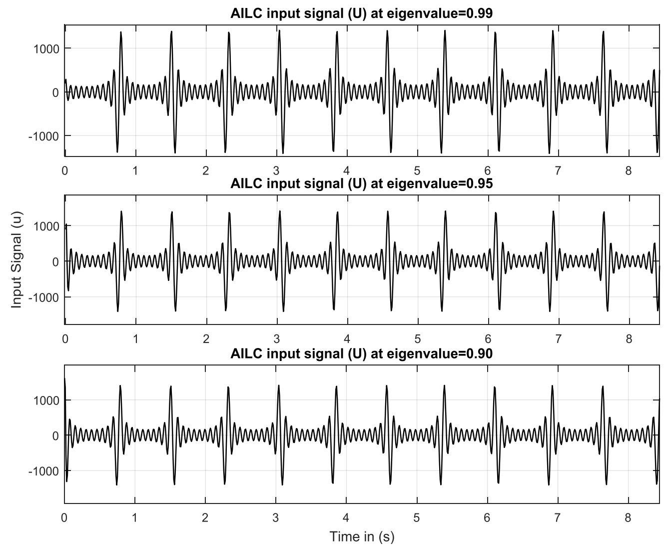

Figure 5: Input signals at $\lambda = 0.99, 0.95, 0.90$, and k=5

The response of AILC algorithm at the same conditions as shown in Figure 6

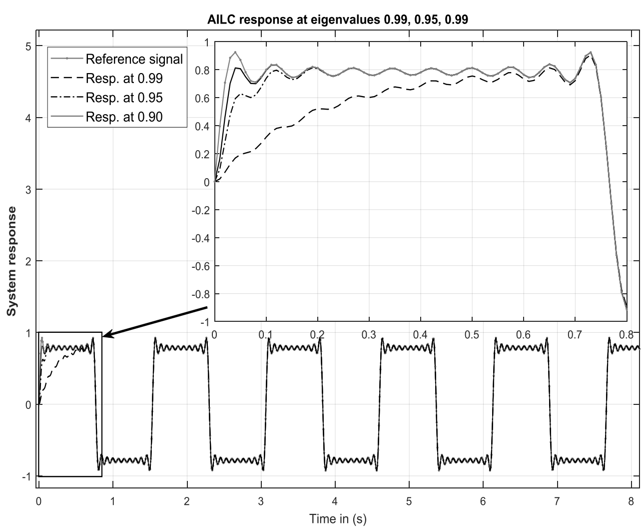

Figure 6: AILC response at $\lambda = 0.99, 0.95, 0.90$, and k=5

The output error e_k of the same condition at the iteration number $k = 5$ is shown in Figure 7

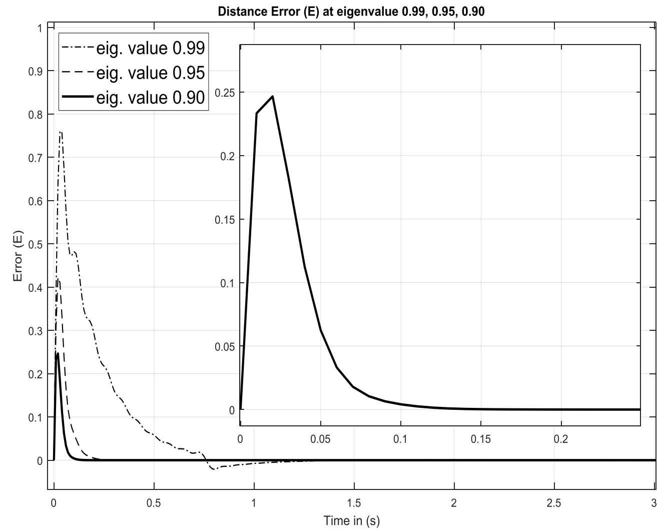

Figure 7: Output error at $\lambda = 0.99, 0.95, 0.90$, and k=5

The estimated parameters $k = 5$ are shown in Figure 8

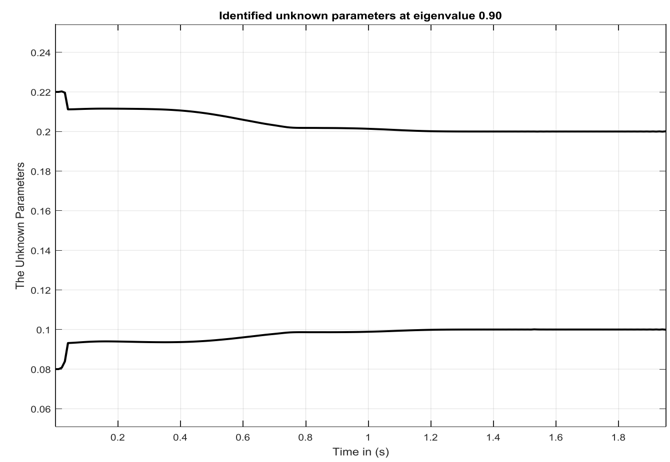

Figure 8: Estimated parameters at $\lambda = 0.90$, k=5

6. Conclusions

An adaptive method of the ILC algorithm by using the error output method is proposed. A new ILC learning gain (Γ) implementation by using eigenvalue law is presented, it is shown this method implementation the eigenvalue of the learning gain matrix (Γ), and how it is affecting the speed of the tracking convergence. This method gives a wide rate of adjustability and predictability of convergence rate. The application and the simulation results of AILC algorithm on a linear time-invariant SISO dynamic system are presented in different conditions.

Appendix A

Suppose a linear algebraic equation is given as:

$$\begin{bmatrix} z_1 \\ z_2 \\ z_3 \\ \vdots \\ z_{N_p} \end{bmatrix} = \begin{bmatrix} C\Theta & 0 & 0 & \cdots & 0 \\ C\Phi\Theta & C\Theta & 0 & \cdots & 0 \\ C\Phi^2\Theta & C\Phi\Theta & C\Theta & \cdots & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ C\Phi^{N_p}\Theta & C\Phi^{N_p-1}\Theta & C\Phi^{N_p-2}\Theta & \cdots & C\Theta \end{bmatrix} \begin{bmatrix} y_1 \\ y_2 \\ y_3 \\ \vdots \\ y_{N_p} \end{bmatrix}$$

Here, z_i and y_i, $i = 1 \dots N_p$, if the z_i are known then the vectors y_i will be calculated as:

$$y_1 = (C\Theta)^{-1} z_1$$
$$y_2 = (C\Theta)^{-1}(z_2 - C\Phi\Theta y_1)$$
$$\vdots$$
$$y_{N_p} = (C\Theta)^{-1}\left(z_{N_p} - C\Phi^{N_p}\Theta y_1 - C\Phi^{N_p-1}\Theta y_2 - \cdots - C\Phi\Theta y_{N_p-1}\right)$$

The inverse of the matrix G in the equation (8) will calculated sequentially.

$$G^{-1} = \begin{bmatrix} (C\Theta)^{-1} & 0 & 0 & \cdots & 0 \\ -(C\Theta)^{-1}C\Phi\Theta & (C\Theta)^{-1} & 0 & \cdots & 0 \\ . & \dots & (C\Theta)^{-1} & \cdots & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ \dots & \dots & \dots & \cdots & (C\Theta)^{-1} \end{bmatrix}$$

References

[1] M. Uchiyama, "(Formation of High-Speed Motion Pattern of Mechanical Arm by Trail)," Transactions of the Society of Instrument and Control Engineers, pp. vol. 14, issue 6, pp., 1978.

[2] S. Arimoto, S. Kawamura, and F. Miyazaki, "Bettering operation of dynamic systems by learning: A new control theory for servomechanism or mechatronics systems," in 23rd Conf. Decision Control, Las Vegas, Dec.1984.

[3] Z. Bien and J.-X. Xu, Iterative Learning Control Analysis, Design, Integration and Applications, New York: Kluwer Academic Publishers, 1998.

[4] J.-X. Xu and Y. Tan, Linear and Nonlinear Iterative Learning Control, Springer, March 3, 2003.

[5] S. Ashraf, E. Muhammad, and A. Al-Habaibeh, Self-learning control systems using identification-based adaptive iterative learning controller., Proceedings of the I MECH E Part C: Journal of Mechanical Engineering Science, 222, 1177–1187., 2008.

[6] H.-S. Ahn, Y. Q. Chen and K. L. Moore, "Iterative Learning Control: Brief Survey and Categorization," IEEE Transactions on Systems, Man and Cybernetics-Part C: Applications and Reviews, pp. VOL. 37, NO. 6, NOVEMBER 2007.

[7] S. Arimoto, S. Kawamura, and F. Miyazaki, "Bettering operation of Robots by learning," Journal of Robotic Systems, March 12, 1984.

[8] H. Stearns, B. Fine, and M. Tomizuka, Iterative Identification of Feedforward Controllers for Iterative Learning Control, California, USA: IFAC Proceedings Volumes (IFAC-PapersOnline), pp. 203-208, 2009.

[9] W. Messner, R. Horowitz, W. W. Kao, and M. Boals, A new adaptive learning rule, Robotics and Automation, 1990. Proceedings, 1990 IEEE Interna- tional Conference on, 1522–1527 vol.3, 1990.

[10] M. French and E. Rogers, Nonlinear iterative learning by an adaptive Lyapunov technique, Proc. 37th IEEE Conf. Decision Control, Tampa, FL, pp. 175–180, Dec.1998.

[11] D. H. Owens and G. S. Munde, Universal Adaptive Iterative Learning Control, Tampa, Florida USA: Proceedings of the 37th IEEE Conference on Decision & Control, Dec 1998.

[12] M. French, G. Munde, G. Rogers, and D. H. Owens, Recent developments in adaptive iterative learning control, Phoenix, Arizona USA: Proceedings of the 38* Conference on Decision & Control, December 1999.

[13] M. Q. Phan and J. A. Frueh, Model Reference Adaptive Learning Control with basic functions, Phoenix, AZ: Proc. 38th IEEE Conf. Decision Control, Dec. 1999, pp. 251–257, Dec. 1999.

[14] J. Y. Choi and J. S. Lee, Adaptive iterative learning control of uncertain robotic systems, 2000.

[15] S. C. Lee, R. W. Longman and M. Q. Phan, Direct model reference learning and repetitive control, Intell. Autom. Soft Comput. Vol. 8, no. 2, pp. 143-161, 2002.

[16] H. Chen and P. Jiang, Adaptive iterative feedback control for nonlinear system with unknown high-frequency gain, Proc. 4th World Congr. Intell. Control Autom. pp. 847–851, Jun. 2002.

[17] D. H. Owens, K. Feng and, Parameter Optimization in Iterative Learning Control, International Journal of Control, 2003.

[18] C.-J. Chien and C.-Y. Yao, An output-based adaptive iterative learning controller for high relative degree uncertain linear systems, Automatica vol.40, pp. 145-153, 2004.

[19] S.-K. Oh and J. M. Lee, Stochastic iterative learning control for discrete linear time-invariant system with batch-varying reference trajectories, Journal of Process Control, Vol. 36, pp. 64-78., 2015.

[20] L. Ljung, System Identification: Theory for the user, Sweden: University of Liköping, 1987.

[21] K. Ogata, Discrete-Time Systems 2nd edition, 1995.

[22] R. J. Harris, A Primer of Multivariate Statistics, 3rd ed., 2017.

[23] J. L. Crassidis and J. L. Junkins, Optimal Estimation of Dynamic Systems, CHAPMAN & HALL/CRC Press Company, 2004.

[24] Z. B. a. K. M. Huh, "Higher-order iterative control algorithm," IEEE Proc. Part D, Control Theory Appl.,, vol. vol. 136, pp. no. 3, 1227, May 1989.

[25] E. Kreyszig," Introductory Functional Analysis with Applications", Wiley, 1989, ISBN 0-471-50459-9.

Habib Ghanbarpour Asl received his B.S., M.S. and Ph.D. in aerospace engineering, flight dynamic, and control division, from the Amirkabir and Sharif Universities of Technology in 1996, 1999 and 2007, respectively. He is an Assistant Professor in the Department of Mechatronics Engineering at the University of Turkish Aeronautical Association. His research interests include Guidance, Control, Navigation, Robotics, Sensors, and Flying Robots.