An Optimal PD-Type Iterative Learning Control Design For Precise Position Controls

Hoang T.Hua , Nguyen M. Y

Abstract—In many industrial applications, the system needs to perform a motion repeatedly in which the conventional feedback controller cannot achieve the desired accuracy. This paper presents a PD-Type algorithm of Iterative Learning Control (ILC) the for repeated position tracking control problems of DC motors. In this scheme, the control action is computed considering not only the error of the system output and references but also the recognition of the previous performance. By this means, the control signal is updated continually which will mitigate the error to an acceptable range after some iterations, i.e., learning operations. The monotonic convergence is employed to formulate the cost function and determine the proper learning operators. The proposed control scheme is applied to control the trajectory of the position of DC motors. The result shows the advantages of ILC versus the conventional control techniques such as PID controllers, etc. After few iterations, the system output will be adjusted close to the reference in the whole trajectory of motions and the accuracy is accepted (10^{-3}).

Keywords—Iterative Learning Control, monotonic convergence, optimal design, precise controls.

I. INTRODUCTION

Although the control theory is developed quickly, the conventional PID controller is still the most widely cotroller in industry. This is due to the simple structure, effectiveness and its robustness. In many tasks, it is required the system to follow a trajectory repeatedly. Conventional control techniques cannot take advantages of the repetitiveness because it merely based on the real-time measurement of the system output and the reference value [1]. In this case, Iterative Learning Control (ILC) is introduced as an effective tool which is employed the concept of learning machines. The controller can learn itself from the previous executions (trials, motions, etc.) to improve the quality of performance such as reducing the error and/or time delay, etc. The control signal in the current trial is determined by coordinating the control signal and the error of the previous trial such that the system output will converge to the desired trajectory.

Since the ILC concept was first propose by S.Arimoto [2], there are many study efforts can be found in literature for the development of ILC [3-5]. It is recommended to improve the perfomance of the system with a single, periodic operation. ILC has been applied to many fields such as induction motor controls [6], industrial robots [7], water stage motion systems [8], autonomous vehicles [9], computer numerical control machine tools [10], antilock braking [11], semibatch chemical reactors [12], rapid thermal processing [13], cold rolling mills [14]. The design of tunable ILC based on PD-Type ILC is proposed. This is the simplest technique that does not require an extensive modeling and analysis. Some other designs of ILC algorithms use the parametric optimization approach [15]. H_∞ design technique is proposed to deal with the model uncertainties or noise [16]. These methods can be used to design a robust monotonically convergent ILC but require an intensive computation and performance. Model-based methods are presented in [17] in which a time-varying linear constrained systems with deterministic, stochastic disturbances and noises is employed to generalize the quadratic performance critera of ILC. Other approach, plant inversion method uses the inverted model of the system dynamics as the learning function [18]. This method can give the quick convergence, however, suffers from computation burden of system dynamics and sensitivity of modelling errors.

For that reasons, PID-based ILC technique has been proposed. The convergence property of P-Type ILC is studied in [19]; it is designed for a class of discrete-time linear system in [20]. A PD-Type ILC is introduced for the trajectory tracking of a pneumatic X-Y table in [21]. A PID-type ILC algorithm is presented for a robustness property against initial state errors in [22]. In the above-mentioned, PD-Type ILC becomes a powerful technique and widely used in industrial applications. The I-term can be neglected because ILC has a natural integration property to memorize one trial to the next.

This paper aims to design an optimal PD-Type ILC with the learning operators based on monotonic convergences. The PD operators will be formulated as a min-max problem with a cost function to minimize the convergent rate in order to archive the best convergent performance.

The rest of this paper is organized as follows. Section II presents the general learning algorithm. Section III solves the optimal design problem for the PD-Type ILC with monotonic convergences. The design is testified by a case study in both simulations and experiments in Section IV. The conclusion and discussion are given in Section V.

II. ITERATIVE LEARNING CONTROL OVERVIEW

A. System representations

Consider the discrete-time, linear time-invariant (LTI), SISO system:

$$\begin{cases} x_j(k+1) = Ax_j(k) + Bu_j(k) & (1) \\ y_j(k) \quad\;\;\, = Cx_j(k) & (2) \end{cases}$$

where k is the time index, j is the iteration index, y_j is the output, u_j is the control input.

With q is the forward time-shift operator qx(k) ≡ x(k + 1), and x(0) = x_0 for all j. The state space equation equivalent to:

$$y_j(k) = C(qI - A)^{-1}Bu_j(k) + CA^k x_0 \quad (3)$$

Take $P(q) \triangleq C(qI - A)^{-1}B$ and $d(k) = CA^k x_0$

The signal d(k) is the free response of the system to the initial condition x(0).

The plant P(q) is a proper rational function of q. To apply ILC, the plant P(q) is asymptotically stable. If P(q) is not asymptotically stable, we need to stablize P(q) by a feedback controller. P(q) has the delay of m.

Consider N-sample sequences of inputs and outputs:

$$u_j(k), k \in \{0,1,2, \dots, N\};$$
$$y_j(k), k \in \{m, m+1, \dots, N+m-1\};$$
$$d_j(k), k \in \{m, m+1, \dots, N+m-1\};$$

The desired outputs:

$$y_d(k), k \in \{m, m+1, \dots, N+m-1\};$$

For simplicity, we assume that the plant has the delay m = 1.

The error signal is defined by $e_j(k) = y_d(k) - y_j(k)$.

Dividing nummerators of P(q) in system (I) by its denominators to expand P(q) as an infinite power series:

$$P(q) = p_1 q^{-1} + p_2 q^{-2} + \cdots + \cdots, \quad (4)$$

where the coefficients are Markov parameters: $P_k = CA^kB$

The system (3) now can be written in the lifted form:

$$\begin{bmatrix} y_j(1) \\ y_j(2) \\ \vdots \\ y_j(N) \end{bmatrix} = \begin{bmatrix} p_1 & 0 & \cdots & 0 \\ p_2 & p_1 & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ p_N & p_{N-1} & \cdots & p1 \end{bmatrix} \begin{bmatrix} u_j(0) \\ u_j(1) \\ \vdots \\ u_j(N-1) \end{bmatrix} + \begin{bmatrix} d(1) \\ d(2) \\ \vdots \\ d(n) \end{bmatrix}$$

And

$$\begin{bmatrix} e_j(1) \\ e_j(2) \\ \vdots \\ e_j(N) \end{bmatrix} = \begin{bmatrix} y_d(1) \\ y_d(2) \\ \vdots \\ y_d(N) \end{bmatrix} - \begin{bmatrix} y_r(1) \\ y_r(2) \\ \vdots \\ y_r(N) \end{bmatrix} \quad (5)$$

Our goal is designing ILC method which determine the control input sequence of the system (3) such that with increase the number of iteration j the error between $y_d(k)$ and $y_j(k)$ decreases so that the tracking can be established:

$$\lim_{j\to\infty} \left(y_d(k) - y_j(k)\right) = 0 \quad \text{for k = 1,2,...,N+m-1.} \quad (6)$$

B. General Iterative Learning Control Algorithm

A widely used ILC algorithm has a form:

$$u_{j+1}(k) = Q(q)[u_j(k) + L(q)e_j(k+1)] \quad (7)$$

where Q(q) is the low pass filter, L(q) is the learning function.

In the lifted form:

$$\begin{bmatrix} u_{j+1}(0) \\ u_{j+1}(1) \\ \vdots \\ u_{j+1}(N-1) \end{bmatrix} = \begin{bmatrix} q_0 & q_{-1} & \cdots & q_{-(N-1)} \\ q_1 & q_0 & \cdots & q_{-(N-2)} \\ \vdots & \vdots & \ddots & \vdots \\ q_{N-1} & q_{N-2} & \cdots & q_0 \end{bmatrix} \left(\begin{bmatrix} u_j(0) \\ u_j(1) \\ \vdots \\ u_j(N-1) \end{bmatrix} + \begin{bmatrix} l_0 & l_{-1} & \cdots & l_{-(N-1)} \\ l_1 & l_0 & \cdots & l_{-(N-2)} \\ \vdots & \vdots & \ddots & \vdots \\ l_{N-1} & l_{N-2} & \cdots & l_0 \end{bmatrix} \begin{bmatrix} e_j(0) \\ e_j(1) \\ \vdots \\ e_j(N-1) \end{bmatrix} \right) \quad (8)$$

At the end of trial, the error is processed by L(q) added to the previous control signal and filtered through Q(q).This ILC algorithm is implemented as follow:

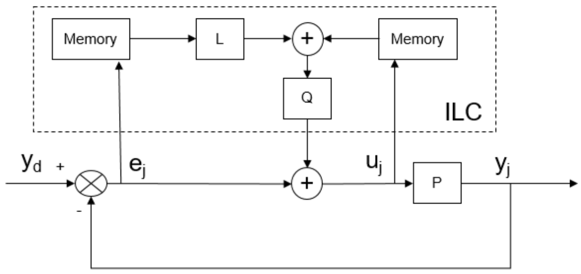

Figure 1. Iterative Learning Control.

Definition: The learning algorithm (7) is causal if $u_{j+1}(k)$ depends only on $u_j(h)$ and $e_j(h)$ for h ≤ k. It is non-causal if $u_{j+1}(k)$ is also a function of $u_j(h)$ or $e_j(h)$ for some h > k.

The advantage ILC algorithm has taken over the traditional feedback control is that ILC can anticipate and preemptively respond to repeated disturbances due to the non-causal algorithm. In practice, the non-causal algorithm is implementable because we have all the data sequence of the previous iteration.

ILC algorithm use feed-forward control action only, so it can not compensate the nonrepeating disturbances. Thus, in most real applications, a proper feedback controller is implemented in combination with ILC. There are 2 common architectures which are serial architechture and paralell architechture.

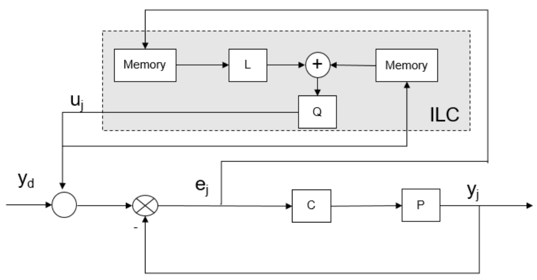

Figure 2. Serial architecture.

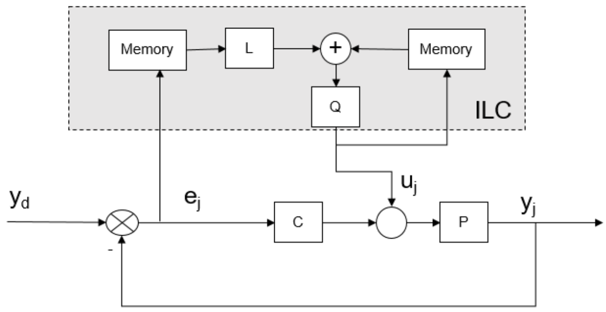

Figure 3. Paralell architecture.

In practice, The time duration in a trial is finite, although sometimes considering the time duration is infinite is useful to analysis and design in frequency domain. We consider the plant (3) and the learning function (6) in z-domain:

The z-transformation of the signal $\{x(k)\}_0^\infty$ is $X(z) = \sum_0^\infty x(k)z^{-1}$, and the transformation of a system is obtained by replacing q with z. To apply the z-transformation to the ILC, we must have N = ∞ because the calculations of the Fourier transform it is assumed that the time horizon is infinite [23]. In z-domain:

The plant is:

$$Y_j(z) = P(z)U_j(z) + D(z) \qquad (9)$$

The learning function is:

$$U_{j+1}(z) = Q(z)[U_j(z) + zL(z)E(z) \qquad (10)$$

C. *Analysis*

- **Stablity**

The the system (3) and (7) is asymptotically stable if there existes $\bar{u} \in R$ such that:

$|u_j(k)| \leq u$ for all k = {0,1,…,N-1} and j = {0,1,…,}

$\lim_{j\to\infty} u_j(k)$ exists

Substitute $e_j = y_d - y_j$ from (5) into learning algorithm (8) yields:

$$u_j = Q(I - LP)u_j + QL(y_d - d)$$

Let $\rho(A) = max_i(\lambda_i)(A)$ be the spectral radius of the matrix A, and $\lambda_i(A)$ the ith eigenvalue of A.

Theorem 1: The ILC system is asymptotically stable if:

$$\rho(Q(I - LP) < 1 \qquad (11)$$

For z-domain:

Theorem 2: If $||Q(z)[1 - zL(z)P(z)||_\infty < 1$, the system (3) and (7) is asymtotically stable for N = ∞.

- **Performance**

The system is asymptotical stable, so the asymtotic error [8] is defined:

$$\begin{aligned} e_\infty &= \lim_{j\to\infty} e_j \\ &= \lim_{j\to\infty} (y_d(k) - P(q)u_j(k) - d(k)) \\ &= y_d - P(q)u_j(k) - d(k) \end{aligned}$$

Replacing j by ∞ in (5) and (8), we got:

$$u_\infty = Q[(I - LP)u_\infty + L(y_d - d)] \qquad (12)$$

$$y_\infty = Pu_\infty + d \qquad (13)$$

Now e_∞ is:

$$e_\infty = (y_d - d)[I - P[I - Q(I - LP)]^{-1}QL \qquad (14)$$

Similar for the frequency-domain representation we get:

$$E_\infty(z) = \frac{1 - Q(z)}{1 - Q(z)[1 - zL(z)P(z)]}[Y_d(z) - D(z)]$$

Theorem 3: Suppose P and L are not identically zero. Then, for the ILC system (3), (7), $e_\infty(k) = 0$ for all k and for all y_d and d, if and only if the system is AS and Q(q) = 1. Q(s) is chosen as a low pass filter with a cut-off frequency ω_c, where unity gain for low frequencies in range $[0,\omega_c]$ and zero gain for higher frequencies.

- **Learning Behaviour**

An example of large transient growth in ILC system has been presented in [3]. The error of e_j increase for several iterations at the beginning before it decreases. Transient growth is problematic in practice because it can effect the stability conditions. Monotonic convergence is considered to void large learning transients.

The system (3), (7) is monotonically convergent under a given norm ||•|| if:

$$|| e_\infty - e_{j+1} || < \gamma|| e_\infty - e_j ||$$

Form (5) and (8), we obtain:

$$e_\infty - e_{j+1} = Q(I - LP)(e_\infty - e_j)$$

For z-domain:

$$E_\infty(z) - E_{j+1}(z) = Q(z)[I - L(z)P(z)][E_\infty(z) - E_j(z)]$$

Theorem 4:

For time-domain: If the ILC system (3), (7) satisfies:

$$\gamma_1 \triangleq \sigma(PQ(I - LP)P^{-1}) < 1$$

So,

$$|| e_\infty - e_{j+1} ||_2 < \gamma_1|| e_\infty - e_j ||_2$$

For z-domain: If the ILC system (3), (7) satisfies:

$$\gamma_2 \triangleq ||Q(z)[1 - zL(z)P(z)]||_\infty < 1$$

So,

$$||E_\infty(z) - E_{j+1}||_\infty < \gamma_2||E_\infty(z) - E_j(z)||_\infty$$

III. OPTIMAL PD-TYPE ILC

A. *PD-Type ILC*

Although a lot of advanced techniques have been proposed to solve the control problems, the classical control technique is still dominating in the industrial. The limited affanity with the advanced mathematical tools needed to perform the advanced technique is an important obstacle for control engineer in industry. Therefore, we propose the design of PD-type ILC because its simple structure and less the mathematical tools needed.

PD-Type learning function consists of propotional and derivative gain on error. In discrete time, the PD-type learning function can be written as follow:

$$u_{j+1} = u_j + k_p e_j(k+1) + k_d \frac{e_j(k+1) - e_j(k)}{T}$$

where T is sampling interval. According to the Theorem 3, filter Q has the unity gain in order to get the perfect tracking.

The learning algorithm is illustrated in the figure below.

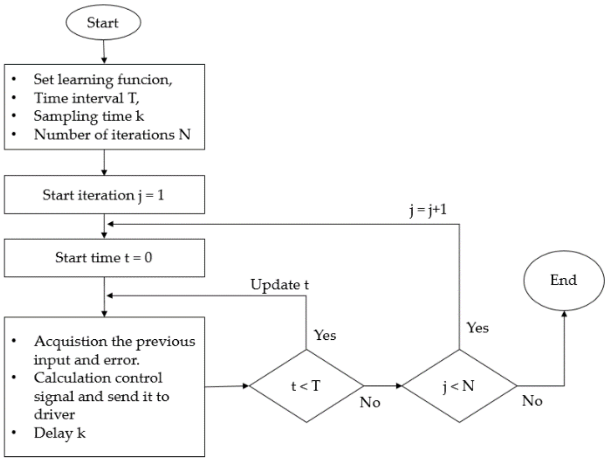

Figure 4. Flowchart of ILC.

B. Optimal Design

We have the schematic of PD-Type ILC is in the Figure 5 below.

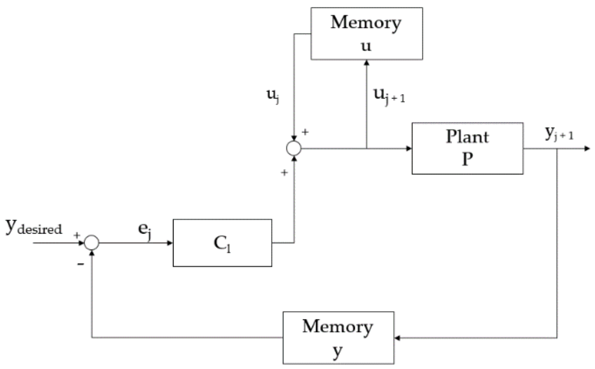

Figure 5. Schematic of PD-Type ILC.

The compensator C_l is in the form of discretized PD controller.

$$C_l(z) = k_p + k_d \frac{1}{T_s} \frac{z-1}{1}$$

With kp, kd, Ts are the propotional gain, derivative gain and the sampling time respectively.

From Theorem 4, the monotonically convergent condition is:

$$||E_\infty(z) - E_{j+1}||_\infty < \gamma ||E_\infty(z) - E_j(z)||_\infty$$
$$=> \ ||E_{j+1}||_\infty < \gamma ||E_j(z)||_\infty$$

Clearly, as far as the tracking error signals of the first iteration, E_0, is finite, then $||E_j|| = \gamma^j ||E_0|| \rightarrow 0$ as $j \rightarrow \infty$. See the smaller γ the faster the algorithm converges.

We have:

$$\gamma \triangleq ||Q(z)[1 - zL(z)P(z)]||_\infty$$

Take Q(z) has the unity gain for perfect tracking. Subtituting the learning function L(z) by $C_l(z)$. We get:

$$\gamma \triangleq ||[1 - zC_l(z)P(z)]||_\infty$$

In order to get the best convergence, we need to determine the cofficient k_p, k_d in C_l which satisfy the cost function J.

Cost function:

$$J_{opt} = \min_{kp,kd>0} ||1 - zC_l(z)P(z)||_\infty \qquad (15)$$

It is equavalent to:

$$J_{opt} = \min_{kp,kd>0} ||1 - e^{j\omega t}(k_p + k_d \frac{e^{j\omega t} - 1}{T_s e^{j\omega t}})P(e^{j\omega t})||_\infty$$
(16)

IV. Case Study

In this section, the optimal PD-Type ILC is implemented and compare to conventional feedback PID controller. The algorithms are implemented on the real model that has diagram in Figure 6. Considering DC motor as the plant. The output is the number of pulses recorded by an encoder.

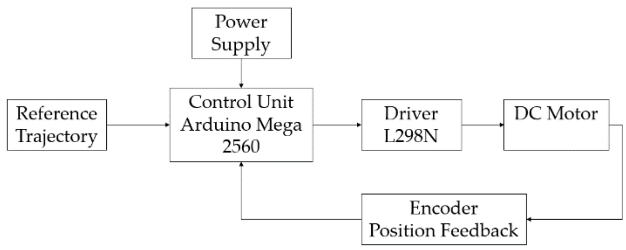

Figure 6. Block diagram of DC motor control system.

A. System Modeling

Ignoring the electrical dynamic, the relation between the linear motion of DC motor and the motor input voltage can be approximated as a second-order nominal model.

$$P(s) = \frac{a}{bs^2 + cs}$$

Using Identification Toolbox in Matlab, we obtain the transfer function:

$$P(s) = \frac{4120}{s^2 + 25.55s}$$

The sampling period is set to be 0.01 second. The discretized transfer function:

$$P(z) = \frac{0.1895z + 0.1741}{z^2 - 1.775z + 0.7745}$$

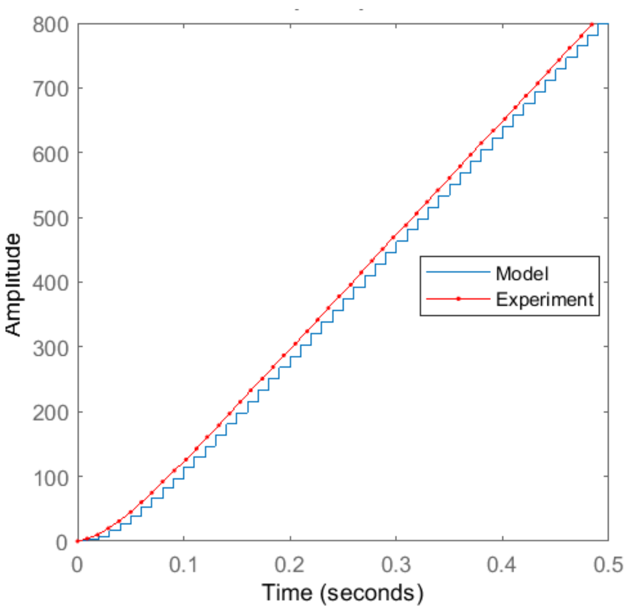

Figure 7. Model valdiation.

We first design the feedback controller to ensure the stablity and conpensate the non-repeated disturbance. The PID parameters are obtained by PID-Tuner Toolbox. New dynamic model of the plant after implemented PID controller is:

$$P(z) = \frac{0.05903z^3 - 0.04243z^2 - 0.05058z + 0.03506}{1.059z^3 - 2.817z^2 + 2.498z - 0.7395} \quad (17)$$

B. Optimal Design

The reference motion of DC motor is described by:

$$y_d(t) = [1 - \cos(4t)] * 187 \qquad (t \in [0, 1.56])$$

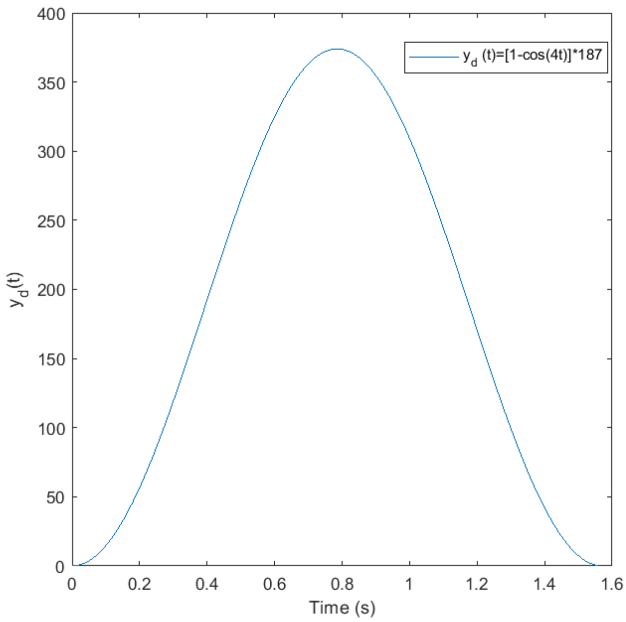

Figure 7. Reference output.

Subtitute (17) into (16) and sovle for k_p and k_d with Ts = 0.01 (s), ω = 4 (rad/s).

$$J_{opt} = \min_{kp,kd>0} ||1 - e^{j\omega t}(k_p + k_d \frac{e^{j\omega t} - 1}{T_s e^{j\omega t}})P(e^{j\omega t})||_\infty$$

Solving J_{opt} numerically using Casadi [24], we get

k_p = 0.730924;

k_d = 0.0993292;

J_{min} = 4.0227x10-7.

In simulation environment, the output trajectory at iteration 1 and iteration 10 are shown in the Figure 8 and Figure 9 respectively.

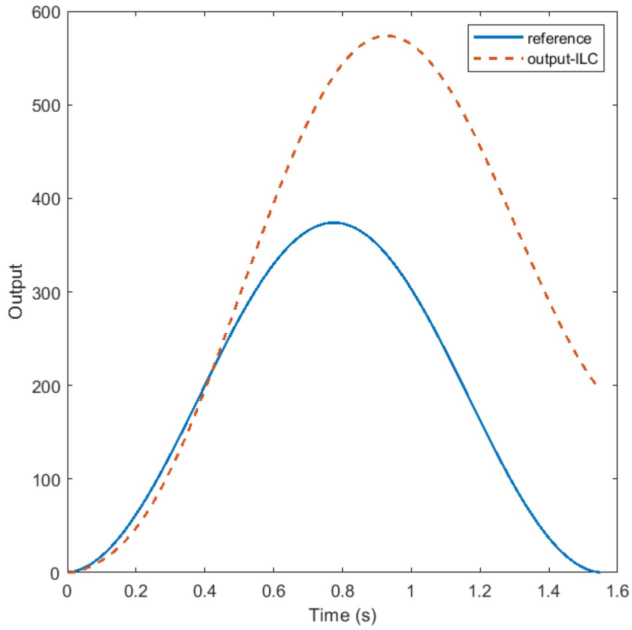

Figure 8. Output at Iteration 1.

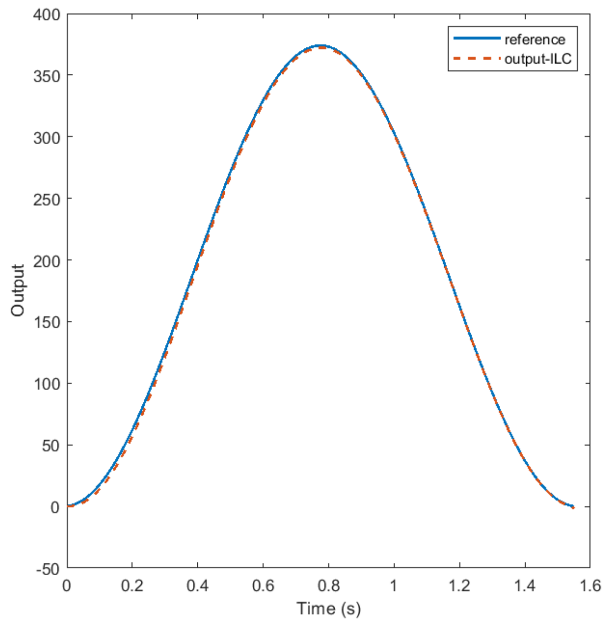

Figure 9. Output at Iteration 10.

In Figure 10, output of system only use PID controller is shown. The output remains the same for all iterations with no improvement.

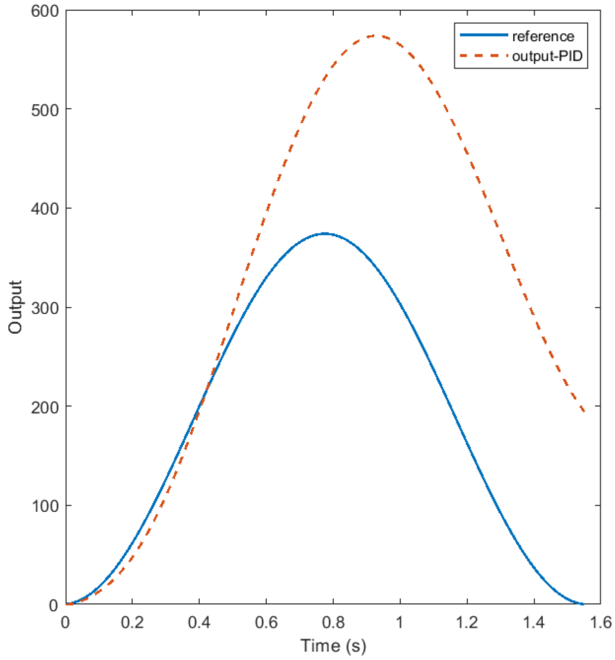

Figure 10. Output of system only use PID.

Define the error-norm as:

$$err = \sum_{k=0}^{156} \sqrt{(y_d[k])^2 - (y_{out}[k])^2}$$

The error-norm value of trajectory error in both case using ILC and using PID only is shown below.

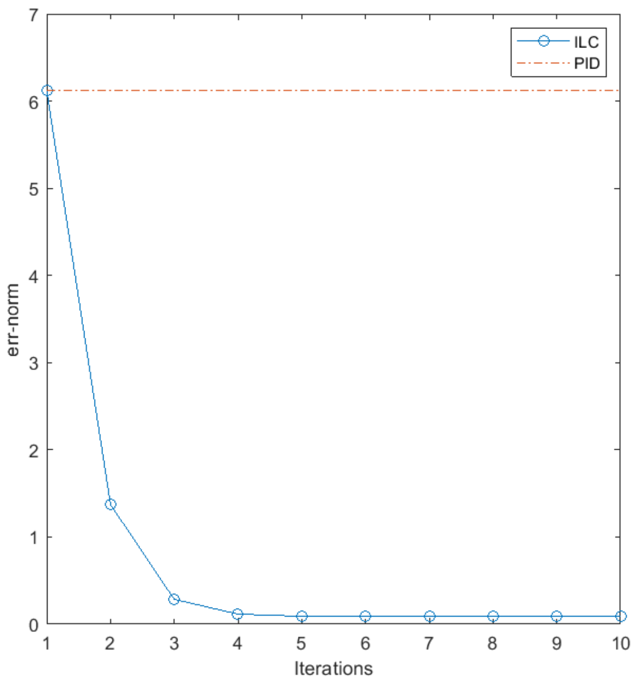

Figure 11. Error-norm value.

It can be seen that the performance of system is enhanced throuh ierations with apllying ILC.

The algorithm is implemented on the real model. The results are in the Figures below.

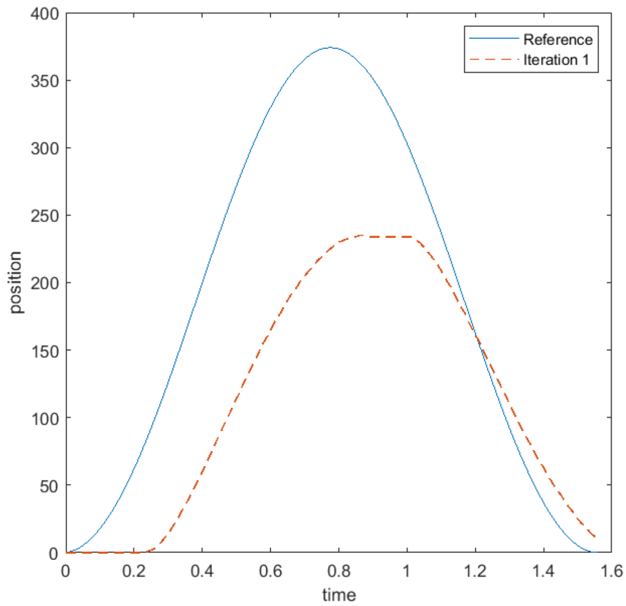

Figure 12. Output at Iteration 1.

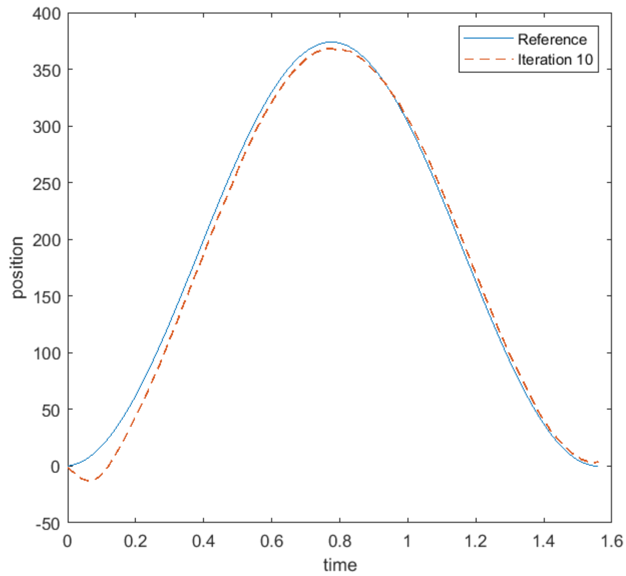

Figure 13. Output at Iteration 10.

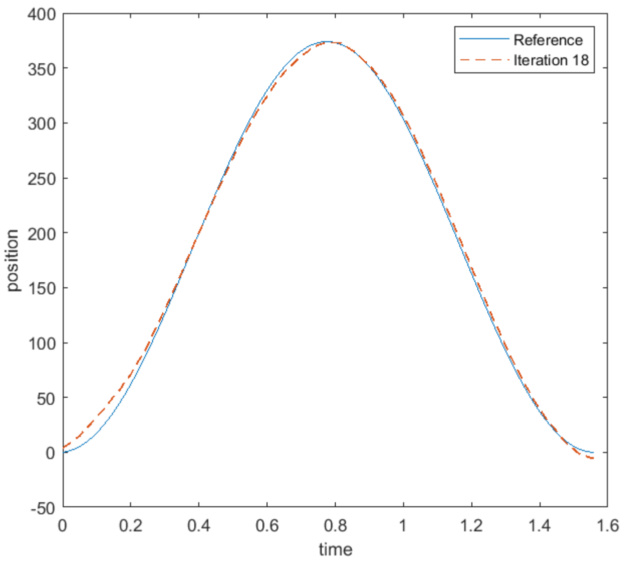

Figure 14. Output at Iteration 18.

V. CONCLUSION

This paper aims to design and implement PD-Type ILC to sovle the position control problem with periodic reference signal. At first, it analyzes the essential criteria of ILC technique. The design of PD-Type ILC using monotonic convergence conditions is presented. The design is implemented in both simulations and practical experiments. The results show that ILC has a great advantage comparing to conventional PID controller in case the reference signal is periodic. Although the real-word result is not the same as simulation as expected, it shows that using the ILC could improve the performance in the practical environment. In future work, it is extended to deal with other tunning methods, ILC design technique and more complex systems as improved applications of embedded field.

REFERENCE

1. Kevin L. Moore, "Iterative Learning Control for Deterministic Systems (Advances in Industrial Control)", Iterative Learning Control for Deterministic Systems,Springer-Verlag Series on Advances in Industrial Control, Springer-Verlag, London, January 1993.
2. S. Arimoto, S. Kawamura, and F. Miyazaki, "Bettering operation of robots by learning," Journal of Robotic Systems, vol. 1, no. 2, pp. 123-140, 1984.
3. K. L. Moore, "Iterative learning control – an expository overview," Applied & Computational Controls, Signal Processing, and Circuits, vol. 1, no. 1, pp.151–241, 1999.
4. Douglas A. Bristow, Marina Tharayil and Andrew G. Alleyne, "A survey of Iterative learning control, a learning method for high-performance tracking control", IEEE control systems magazine, June 2006.
5. Hyo-Sung Ahn, YangQuan Chen, Kevin L. Moore, "Iterative Learning Control: Brief Survey and Categorization", IEEE Transactions on Systems, Man, and Cybernetics, Part C (Applications and Reviews) ,Volume: 37 , Issue: 6 , Nov. 2007.
6. S.A. Saab, "A stochastic iterative learning control algorithm with application to an induction motor," Int. J. Contr., vol. 77, no. 2, pp. 144–163, 2004.
7. M. Norrlof, "An adaptive iterative learning control algorithm with experiments on an industrial robot," IEEE Trans. Robot. Automat., vol. 18, no. 2, pp. 245–251, 2002.
8. D. de Roover and O.H. Bosgra, "Synthesis of robust multivariable iterative learning controllers with application to a wafer stage motion system," Int. J. Contr., vol. 73, no. 10, pp. 968–979, 2000.
9. Y.Q. Chen and K.L. Moore, "A practical iterative learning pathfollowing control of an omni-directional vehicle," Asian J. Contr., vol. 4, no. 1, pp. 90–98, 2002.
10. D.-I. Kim and S. Kim, "An iterative learning control method with application for CNC machine tools," IEEE Trans. Ind. Applicat., vol. 32, no. 1, pp. 66–72, 1996.
11. C. Mi, H. Lin, and Y. Zhang, "Iterative learning control of antilock braking of electric and hybrid vehicles," IEEE Trans. Veh. Technol., vol. 54, no. 2, pp. 486–494, 2005.
12. C. Mi, H. Lin, and Y. Zhang, "Iterative learning control of antilock braking of electric and hybrid vehicles," IEEE Trans. Veh. Technol., vol. 54, no. 2, pp. 486–494, 2005.
13. D.R. Yang, K.S. Lee, H.J. Ahn, and J.H. Lee, "Experimental application of a quadratic optimal iterative learning control method for control of wafer temperature uniformity in rapid thermal processing," IEEE Trans. Semiconduct. Manufact., vol. 16, no. 1, pp. 36–44, 2003.
14. S. Garimella and K. Srinivasan, "Application of iterative learning control to coil-to-coil control in rolling," IEEE Trans. Contr. Syst. Technol., vol. 6, no. 2, pp. 281–293, 1998.
15. J. Hatonen, D. H. Owens, and K. Feng, "Basis functions and parameter optimization in high order iterative learning control," Automatica, vol. 42, no. 2, pp. 287-294, 2006.
16. K. L. Moore, Y. Chen, and H.-S. Ahn, "Algebraic H_∞ design of higher order iterative learning controllers," in Proc. 2005 IEEE Int. Symp. Intelligent Control, Limassol, Cyprus, 2005, pp. 1207–1212.
17. H. Lee, K. S. Lee, and W. C. Kim, "Modelbased iterative learning control with a quadratic criterion for time-varying linear systems," Automatica, vol. 36, no. 5, pp. 641-657, 2000.
18. K. Kinosita, T. Sogo, and N. Adachi, "Iterative learning control using adjoint systems and stable inversion," Asian J. Contr., vol. 4, no. 1, pp. 60–67,2002.
19. L. Moore, "An observation about monotonic convergence in discrete-time, P-type iterative learning control," Proc. of the IEEE International Symposium on Intelligent Control, pp. 45-49, MX, USA, 2001.
20. K. L. Moore, Y. Q. Chen, and V. Bahl, "Feedback controller design to ensure monotonic convergence in discrete-time, P-type iterative learning control," Proc. of the 4th Asian Control Conference, pp. 440-445, Singapore, September 2002.
21. C. K. Chen and J. Hwang, "PD-type iterative learning control for trajectory tracking of a pneumatic X-Y table with disturbances," JSME International Journal, Series C, vol. 49, no. 2, pp. 520-526, 2006.
22. K.-H. Park, Z. Bien, and D.-H. Hwang, "A study on the robustness of a PID-type iterative learning controller against initial state error," International Journal of Systems Science, vol. 30, no. 1, pp 49-59, 1999.
23. M. Norrlof and S. Gunnarsson, "Time and frequency domain convergence properties in iterative learning control," Int. J. Contr., vol. 75, no. 14, pp. 1114–1126, 2002.
24. Joel A. E. Andersson, Joris Gillis, Greg Horn, James B. Rawlings, Moritz Diehl, "CasADi: a software framework for

nonlinear optimization and optimal control", Mathematical Programming Computation March 2018.