PI CONTROLLER DESIGN BASED ON ITERATIVE LEARNING CONTROL

Hesham W. Gomma

Electronics, Communications and Computer Engineering Department
Faculty of Engineering, Helwan University

This paper presents a technique that can be used in designing an equivalent PI controller to different classes of the known Iterative Learning Control (ILC) namely P-type and high order ILCs. The equivalent PI controller can be explicitly represented in the z domain in contrast to the time domain based ILC, which gives better potential for stability analysis. Moreover, the derived PI controller combines the ease of tuning with the learning feature from past processes, which is the base of Iterative Learning Control (ILC). The results show that the ability of proposed approach to provide an equivalent PI controller that provides similar performances to the ILC when applied to various systems with no restrictions on the system order or type.

1. INTRODUCTION

In the mid eighties a new control technique called Iterative Learning Control (ILC) was developed [1]. The main observation that led to this strategy is that when a process performs the same task over and over again, the resultant error is the same. Then it must be possible to gradually reduce this error by experience. The fact that several analyses, simulations and experiments have proven that is could be true, made people think that ILC is essentially something new.

Although, the development of ILC stemmed originally from the robotics area where repetitive motions show up naturally in many applications [3]–[8]; later on, this technique has shown significant ability to deal with other different systems such food processing plants, wafer steppers, hard disk servo controllers and chemical batch reactors [9]. This is because its anticipatory character and ability to ensure the compensation for repetitive external disturbances by learning based on previous iterations without further modeling burdens. Moreover, ILC does not require knowing the variations of reference and disturbance with the ability to ensure the control system robustness under some well-stated conditions [10]. In addition, learning functions leads to rapid convergence (for example P-type) and allow controller tuning without requiring the detailed mathematical model of the controlled plant. Furthermore, in some well-stated conditions ILC ensure the control system robustness with respect to process modeling uncertainties. Learning controller's stability analysis with disturbances, uncertain initial conditions were discussed in reference [11]. Some of the nonlinear robust ILC algorithms [12-14] are also addressed for uncertain systems.

However, the main drawback of general ILCs is that, there is no mathematical model for the entire learning control system can describe both the dynamics of ILC along the time and iteration axes [15]. Also, the ability to be tuned and formalizing the

connection between robustness and dynamic and steady-state control systems performance and ensuring the best of these requirements simultaneously, treating the situations in which the reference and disturbance inputs do not have repetitive variations. Much of the work on ILC has focused on converged performance. In [2] it was shown that, under ideal circumstances, the P-type ILC can be used to obtain zero error tracking for an LTI discrete-time system. Later different learning algorithms have been developed with proven monotonic convergence such high order ILC [16] and [17]. Monotonic convergence does not only ensure that performance improves at each iteration, but it can also be easily related to a convergence rate that indicates how quickly the ILC will effectively converge.

On the other hand, PI control is common and popular controller in industry although it has number of limitations such as the difficulty of finding suitable PI parameters. Accordingly, a considerable amount of research has been conducted to improve the performance of the PI controllers and PI parameters auto tuning schemes.

The main objective of this paper is to introduce a technique that is able to design a PI controller based on the ILC controller. This concept is based on designing a PI controller that is able to provide similar response to that obtained by ILC after achieving a reasonable response. This will be conducted by using identification algorithm such as recursive least square (RLS). The proposed PI has all advantages of the ILC controller. Furthermore, it can be used as a self tuning PI controller.

2. P-TYPE ILC

ILC as a control idea is to refine the input signal to a system that operates repeatedly. This can be explained by considering a system in an initial state to which a fixed length input signal is applied. Then the system is returned to its initial state when the end of input signal is reached. Consequently, the output trajectory that resulted from the applied input is compared to a desired trajectory. The error is used to construct a new input signal (of the same length) to be applied the next time the system operates. This process is then repeated. The goal of the ILC algorithm is to properly refine the input sequence from trial to trial so that as more and more trials are executed the actual output will approach the desired output at all points in time along the trajectory.

This is different from conventional control, where tracking problems attempt to converge to the desired trajectory in the limit as time increases. The basic idea of ILC is illustrated in Fig. 1 [2], [3]. Standard assumptions are that the plant has stable dynamics i.e., the system returns to the same initial conditions at the start of each trial and then the trial lasts for a fixed time *Tf* , and that each trial has the same length. However, in case of unstable systems, it is suggested to stabilize the system first with a suitable control technique then the ILC can be introduced. This section introduces the following P-type (Arimoto-type ILC) considering the architecture shown in Fig. 1.

$$u_{k+1}(t) = u_k(t) + \gamma e_k(t+1) \qquad\qquad (1)$$

where, $u_k(t)$ is the system input, $e_k(t) = y_d(t) - y_k(t)$ is the error and γ is the learning gain with $y_k(t)$ the system output and $y_d(t)$ the desired response, "*k*" is the iteration index. According to the above algorithm, the learning controller's goal is to derive an

optimal input $u(t)$, for $t \in [1, N-1]$ by calculating the error $e_k(t) = y_d(t) - y_k(t)$ on the interval $t \in [1, N]$. This is achieved by adjusting the input from the current trial u_k to a new input u_{k+1} for the next trial. It is clear that, the main feature of ILC is that the algorithm depends on past data; the fact that the initial conditions are reset at the beginning of each trial allows ILC to do "non-causal" processing on the errors from the previous trial.
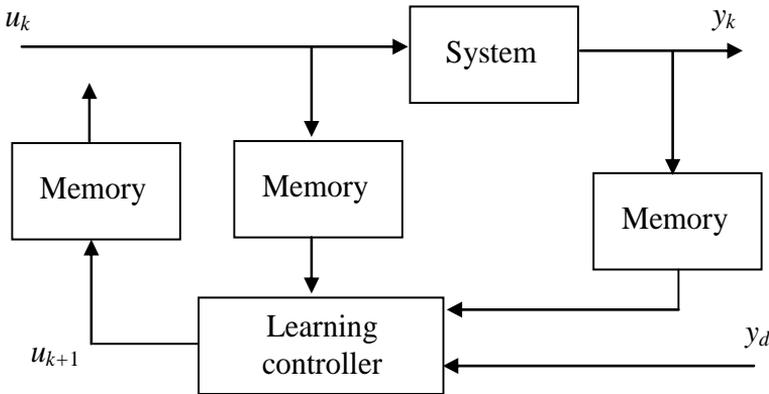

Figure 1: Iterative learning control architecture

3. CONVERGENCE

It is natural to argue that the ILC design objectives should be first focused on the monotonic convergence issue. The convergence properties of the Arimoto-type ILC algorithm can briefly discussed in this section. Consider a discrete-time, linear, time-invariant system of relative degree one:

$$y_k(z) = (h_1 z^{-1} + h_2 z^{-2} + \cdots) u_k(z)$$
$$y_k(z) = H(z) u_k(z) \tag{2}$$

where z^{-1} is the standard delay operator in time, $y_k(z)$ and $u_k(z)$ are the z-transforms of the system's output and input sequences, $y_k(t)$ and $u_k(t)$ respectively, "t" is the time index and satisfies $t \in [0, N]$, and "k" denotes the iteration index. This system can be written as

$$Y_k = H U_k \tag{3}$$

where

$$U_k := \begin{bmatrix} u_k(0) & u_k(1) & \cdots & u_k(N-1) \end{bmatrix}^T$$
$$Y_k := \begin{bmatrix} y_k(1) & y_k(2) & \cdots & y_k(N) \end{bmatrix}^T$$

and *H* is the matrix of Markov parameters of the plant given by

$$H \overset{\Delta}{=} \begin{bmatrix} h_1 & 0 & 0 & \cdots & 0 \\ h_2 & h_1 & 0 & \cdots & 0 \\ h_3 & h_2 & h_1 & \cdots & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ h_N & h_{N-1} & h_{N-2} & \cdots & h_1 \end{bmatrix} \tag{4}$$

The variables h_i are the standard Markov parameters of the system $H(z)$. The convergence properties of the Arimoto-type ILC algorithm have been well-established in the literature. It is well-known that the combination of Equation 1 with Equation 2 converges in a given norm topology if the induced operator norm satisfies

$$\|I - \gamma H\|_i < 1 \tag{5}$$

Note that this sufficient condition ensures monotone convergence in the sense of the relevant norm topology. Moreover, if

$$|1 - \gamma h_1| < 1 \tag{6}$$

then this is the necessary and sufficient condition for convergence [2]. The latter condition does not guarantee monotone convergence as observed in [2]. Therefore, to guarantee monotone convergence in addition to the necessary and sufficient condition for convergence (6), the following condition can be considered

$$|h_1| > \sum_{i=2}^{N} |h_i| \tag{7}$$

However, this is just a sufficient condition which may be too restrictive since it does not relate to the learning gain.

4. PI CONTROL DESIGN BASED ON ILC

Proportional-integral (PI) controllers have been used extensively in the process industries since they are simple with highly acceptable tuning capability and often effective. Moreover, there are different arrangements exist for the PI design, which allow designers to choose the most suitable for achieving their goals. The used PI control law in this paper is represented in the following form

$$u(t) = \left(K_p + \frac{K_I}{\Delta} \right) (y_d(t) - y(t)) \tag{8}$$

where $y_d(t)$ denotes the reference trajectory, $y(t)$ is the system output, K_p and K_I are the proportional and integral gains respectively, and is $\Delta(=1 - z^{-1})$ is difference operator.

4.1 PI Design Algorithm

The main concept of designing an equivalent PI to the converged ILC controller is based on estimating the PI control parameters. This will be achieved after the certain *N* trials that achieve acceptable performance by the designer. Then the design of the equivalent PI is conducted based on a the selected control signal

$$u_N(t) = u_{N-1}(t) + \gamma(y_d(t+1) - y_{N-1}(t+1)) \tag{9}$$

where $u_N(t)$ is *Nth* trail of the control signal, $u_{N-1}(t)$ (*N-1*)*th* trail of the control signal, and $y_{N-1}(t)$ is the (*N-1*)*th* trail of the output signal. However, introducing $u_{N-1}(t)$ to the system leads to $e_{N-1}(t)$, by holding both $e_{N-1}(t)$ and $u_{N-1}(t)$ then considering Equation (9), this leads to $u_N(t)$. The idea here is to find a PI controller that is able to produce the same response for the ILC. The proposed *PI* controller should produce a control signal u_N according to an input signal $e^*(t) := e_{N-1}(t)$. Consequently the PI controller can be represented as

$$u^*(t) := \left(K_p^{ILC} + \frac{K_I^{ILC}}{\Delta} \right) e^*(t) \tag{10}$$

where $u^*(t)$ is the equivalent PI control signal, and K_p^{ILC} and K_I^{ILC} are the estimated proportional and integral gains of the equivalent PI. The new algorithm can be defined as follows by assuming *U* and *Y* be vector spaces and the equivalent PI is to develop an input u^*, such that with $y^* = Hu^*$, where

- $u^* \in U$ and there exists $u_k, \tilde{u} \in U$, developed by the ILC, such that $\lim_{k \to N} u_k = \tilde{u}$, provided that $\left\| u^* - \tilde{u} \right\| \le \varepsilon_1$ with $\varepsilon_1 > 0$ and sufficiently small.
- $y^* \in Y$ and there exists $y_k, \tilde{y} \in Y$, obtained by the ILC, such that $\lim_{k \to N} y_k = \tilde{y}$, provided that $\left\| y^* - \tilde{y} \right\| \le \varepsilon_2$, with $\varepsilon_2 > 0$ and sufficiently small.

4.2 Identification Scheme

The above assumptions can be achieved by using a suitable identification tool, such as Recursive Least Square (RLS) method. The RLS algorithm [18] is based on representing the system to be identified as

$$A(z^{-1})y(k) = B(z^{-1})u(k-1) + \eta(k) \tag{11}$$

where $y(k)$ is the output of the system, $u(k)$ is the input, and $\eta(k)$ is a zero-mean white Gaussian noise term, while $a_1, \cdots a_n$, $b_1, \cdots b_n$ are the system parameters that constitute the following polynomials

$$A(z^{-1}) = 1 + a_1 z^{-1} + a_2 z^{-2} + \cdots + a_{na} z^{-na}$$

$$B(z^{-1}) = b_0 + b_1 z^{-1} + b_2 z^{-2} + \cdots + b_{nb} z^{-nb}$$

A concise vector expression for equation (11) is given by

$$y(k) = \phi^T(k)\theta + \eta(k) \tag{12}$$

where ϕ is the vector of regression that includes measured values of input and output

$$\phi(k) = [-y(k-1) - y(k-2) \cdots - y(k-n_a) \; u(k-1) \cdots u(k-n_b-1)]^T$$

and θ, is the unknown system parameter vector,

$$\theta = [a_1 \; a_2 \cdots a_{na} \; b_1 \; b_2 \cdots b_{nb}]^T$$

Deriving RLS algorithm, yield to the following

$$\hat{\theta}(k) = \hat{\theta}(k-1) - \frac{S(k-1)\phi(k)\left(\phi^T(k)\hat{\theta}(k-1) - y(k)\right)}{\lambda I + \phi^T(k)S(k-1)\phi(k)}$$

$$S(k) = \frac{1}{\lambda}\left(S(k-1) - \frac{S(k-1)\phi(k)\phi^T(k)S(k-1)}{\lambda I + \phi^T(k)S(k-1)\phi(k)}\right) \tag{13}$$

where λ is the forgetting factor given by $0 < \lambda \leq 1$, $S(k)$ is the covariance matrix, and $\hat{\theta}$ is a vector contains the latest parameters estimates.

4.3 The Design Scheme

This section presents a new design scheme in Figure 2 that can be used for obtaining the equivalent PI controller to a pre-designed ILC. Similar design concept was introduced in [19], where PID controllers were designed equivalent to GPC controllers. However, the difficulty of developing this new scheme is mainly regarded to the nature of the ILC which does not have an explicit representation in the z-domain. Considering Figure 2, it can be seen that there is an estimator, namely RLS which is deployed to estimate the parameters K_p^{ILC} and K_I^{ILC} of the PI controller simultaneously by considering the error $e(t)$ and the control signal $u(t)$ as input signals to the RLS. The estimation process will be conducted offline/online after achieving an acceptable performance, in a selected iteration N, by the designed ILC. As soon as the RLS converges to certain parameters K_p^{ILC} and K_I^{ILC}, the estimated PI controller is said to be obtained and equivalent to the pre designed ILC for a certain value of the learning gain γ. Accordingly, the PI can be applied to the system and replace the existing ILC when needed. The new PI controller would incorporate the advantage of being easy to tune and the aforementioned advantages of the ILC when introduced to different classes of systems.

In addition, this technique has the potential to be used as an adaptive technique whenever an adaptive ILC is considered. In this case different values of γ can be introduced and accordingly the PI parameters can be estimated. The major advantage of the proposed techniques, that it is not limited to certain ILC algorithms, it can be

easily extended to other types such as high order ILC (see section 5), PID-type, and PD type. In order to show the similarity between the ILC and its equivalent PI controllers, the following examples are presented.
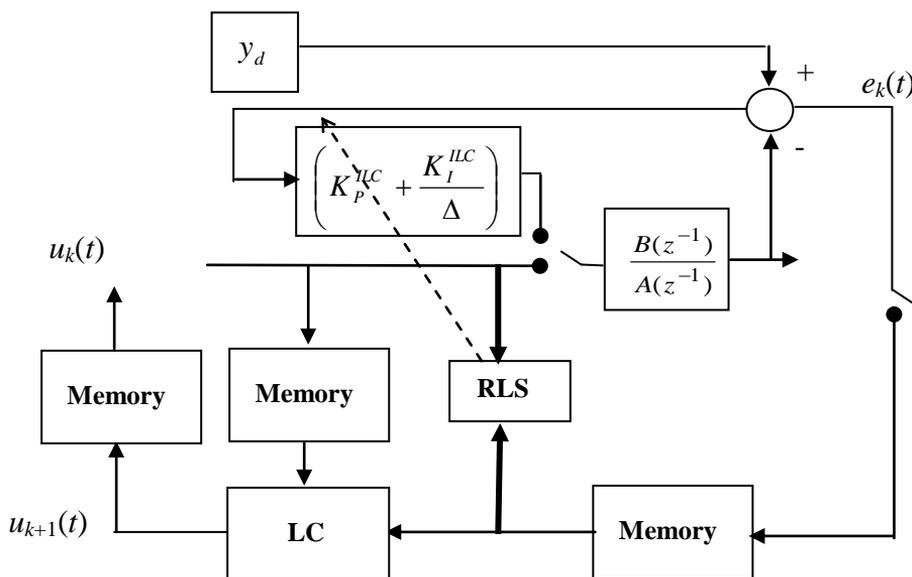

Figure 2: Block diagram for the proposed technique

4.4 Simulation Examples

Example 1

This example is to show how the proposed algorithm (see: Figure 2) will give similar results to that given by Iterative Learning Control (ILC). For the sake of clarity the controller will be designed for a second order model.

$$G_1(z) = \frac{0.13617z^{-1} + 0.11152z^{-2}}{1 - 1.6457z^{-1} + 0.67032z^{-2}} \tag{14}$$

where the sampling was chosen to be $T_s = 0.1$. The simulation was conducted by selecting the learning gain ($\gamma = 0.3$). Figures 3 and 4 illustrate the simulation of the model when the ILC is applied considering a sequence of iterations. It is clear that in iteration 6, the ILC is able to achieve reasonable performance. Holding the data from the last iteration and applying the identification algorithm (see: Figure 2) the parameters for the equivalent PI, are obtained and found to be $K_P^{ILC} = 1.1318$ and $K_I^{ILC} = 0.1027$. Then applying the PI controller to the system can lead to the response which is given in Figure 4. It is clear that the PI still able to stabilize the system with mostly similar response to the one obtained by ILC. One of the main advantages of the designed PI controller is the potential to re-tune the identified parameters whenever different responses are required.

Example 2

One major advantage of the ILC is its ability to deal with different systems despite its order. Therefore, considering this advantage will have a good impact on designing an equivalent PI controller. This example is dealing with the following third order model

$$G_2(z) = \frac{0.094889z^{-1} - 0.13519z^{-2} + 0.047073z^{-3}}{1 - 2.316z^{-1} + 1.7734z^{-2} - 0.44933z^{-3}} \tag{15}$$

where the sampling time was chosen to be $T_s = 0.1$ and the learning gain $\gamma = 0.4$. Figures 6 and 7, show the system response when it is controlled by the ILC. It can be easily observed that the system response improves by iterations and it can be seen that the steady state error and the tracking performance in the final iteration (number 13) is satisfactory. Again, by holding the final iteration, the design technique are used to estimate the PI controller parameters which are found to be $K_P^{ILC} = 1.3532$ and $K_I^{ILC} = 0.1383$. Applying the derived controller to the same system results in the response shown in Figure 8. It is clear that the PI is able to provide satisfactory response, in terms of similarity, when it compared with the ILC. Again, this PI controller can be easily manipulated by most operators in different industrial applications.

5. High order ILC

In this section an alternative ILC is introduced to show the ability of the proposed design technique to deal with different ILC schemes [21]

$$u_{k+1}(t) = u_k(t) + \gamma(e_k(t+1) - \beta e_k(t)) \tag{16}$$

For some different values $\beta > 0$ for a selected learning gain γ, the convergence of this scheme can be achieved by verifying the same conditions. Accordingly, tuning can make the ILC convergence monotonic. In the simple time domain high order scheme demonstrated the learning matrix Q is given as [21].

$$Q = \gamma \begin{bmatrix} 1 & 0 & 0 & \cdots & 0 & 0 \\ \beta & 1 & 0 & \cdots & 0 & 0 \\ 0 & \beta & 1 & \cdots & 0 & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots & \vdots \\ 0 & 0 & 0 & \cdots & \beta & 1 \end{bmatrix} \tag{17}$$

The convergence condition is

$$\|I - HQ\|_i < 1 \tag{18}$$

The matrix Q can take different values to satisfy Equation (18), and H, is the matrix of Markov parameters.

Example 3

In order to ensure the ability of the proposed technique in designing equivalent PI controllers to different types of ILC, the above controller (Equation 16) is introduced to following second order model

$$G_1(z) = \frac{0.13617z^{-1} + 0.1115z^{-2}}{1 - 1.6457z^{-1} + 0.67032z^{-2}} \tag{19}$$

where β is chosen to be 0.1 and learning gain γ is selected to be 0.1. Repeating the same procedures in the above examples has shown that the ILC has provided a good tracking result at iteration 11 (see: Figure 9). Then applying the proposed technique to this iteration an estimating the PI controller parameters has given $K_P^{ILC} = 0.1288$, $K_I^{ILC} = 1.2216$. The response of the PI controller is given in Figure 9. It is clear that the equivalent PI is able to provide similar response to that obtained by the ILC. This confirms the ability of the proposed technique to provide equivalent PI controller to different classes of ILC.
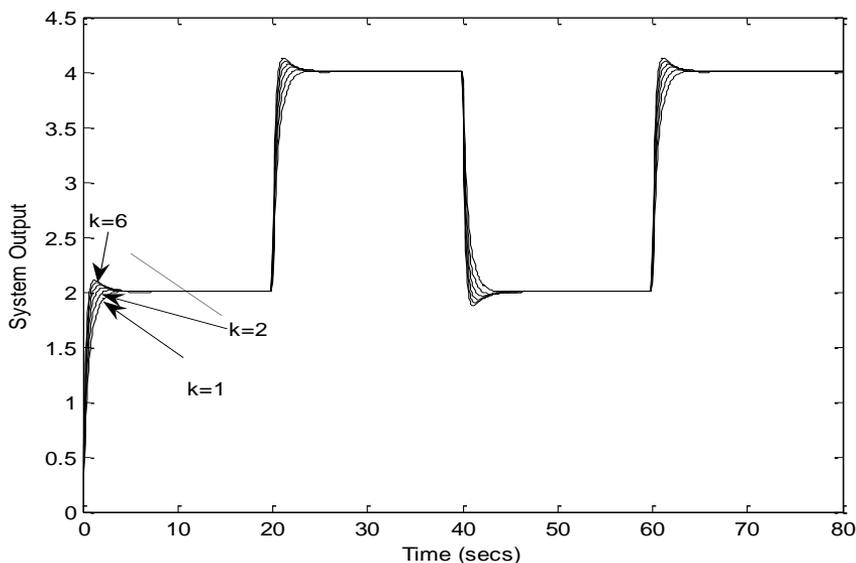

Fig 3: Second order response using ILC.

6. CONCLUSION

This paper has introduced a new technique to design an equivalent explicit PI controller (in the z-domain) to the ILC P-type and high order ILC. The equivalent controller has shown good ability in providing similar performance to the original ILC, when tested with the same systems. This has been accomplished without introducing any restrictions on the type, order and structure of the chosen controlled systems. In addition, it has the advantage of providing the designer with preliminary PI control parameters that can be easily tuned whenever needed. This can facilitate the designer role while selecting the control parameters from a wide pool of selection. The new

design framework is strongly recommended to be used as an equivalent approach to the ILC control design. Furthermore, the advantage of this work that it can be generalized to other forms of ILC approaches whenever an explicit PI controller is needed.
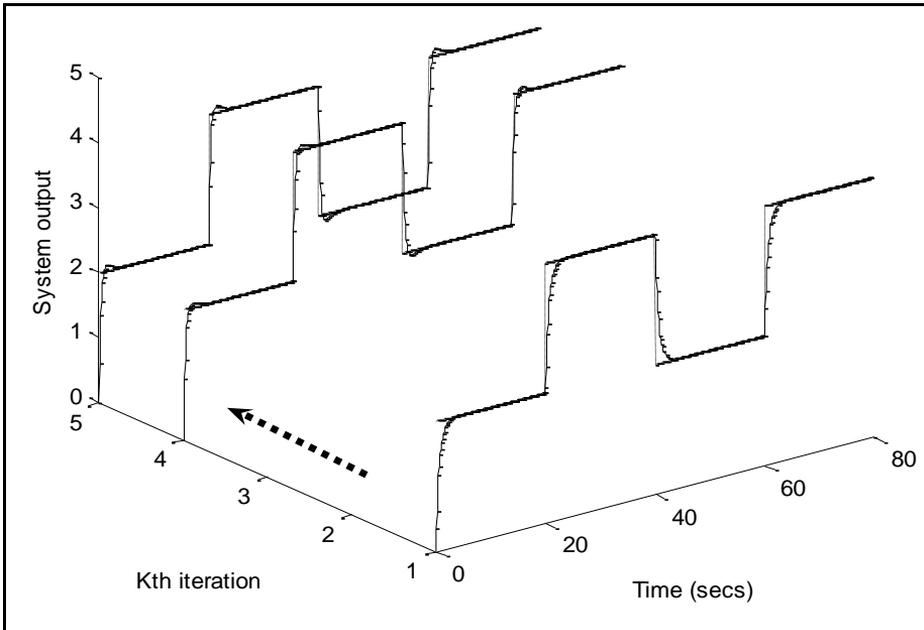

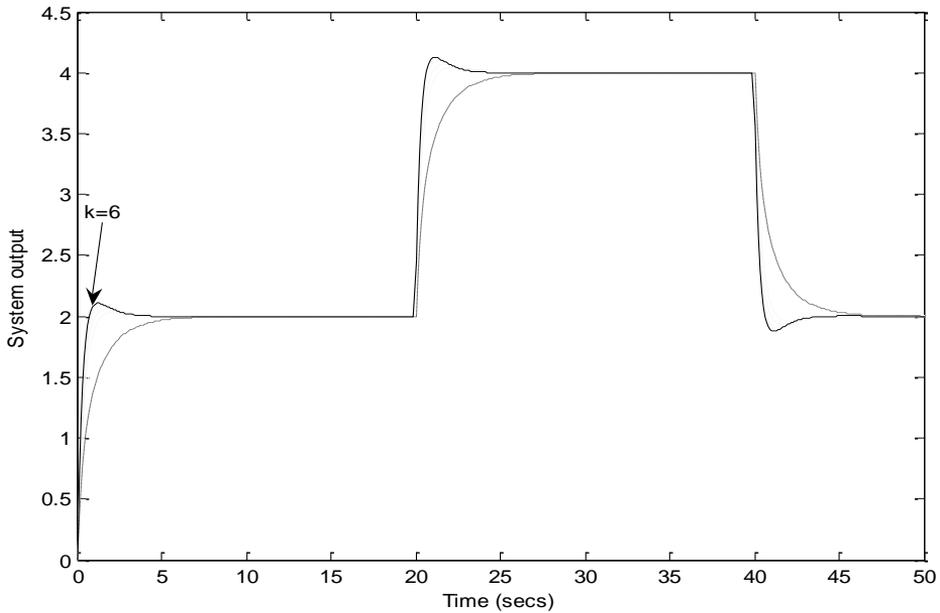Figure 4: Second order response using ILC
(___ ILC response, ---- Reference)

Figure 5: Second order response using PI and ILC
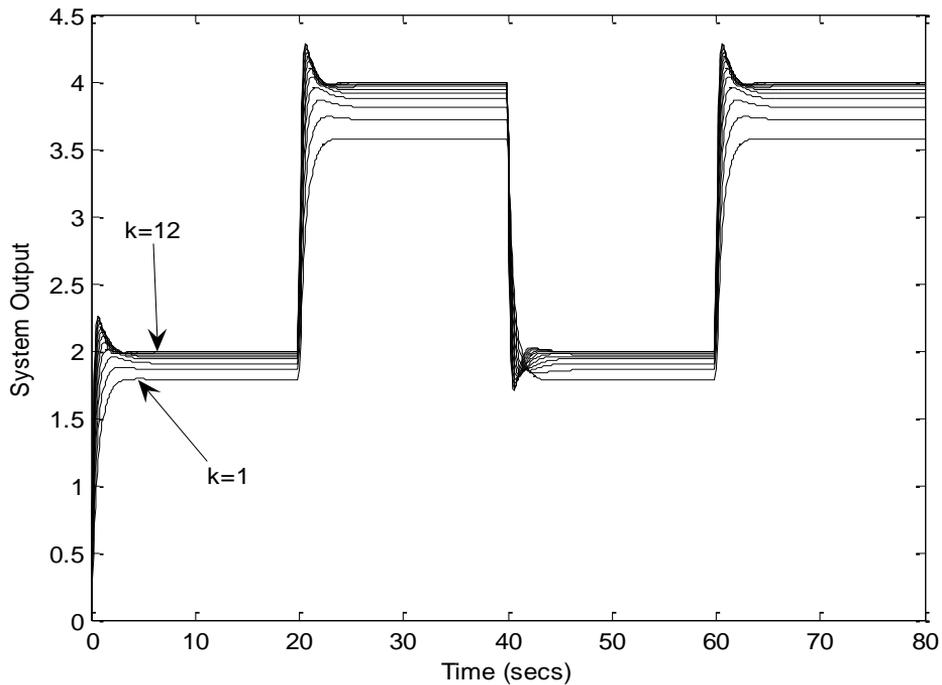(___ ILC (iteration 6), ---- PI controller)

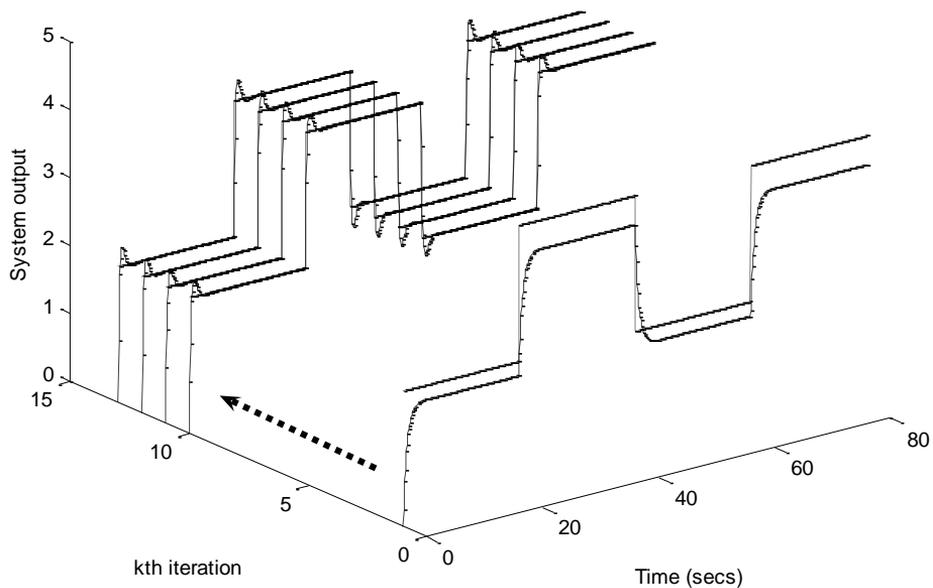Figure 6: Third order response using ILC

Figure 7: Second order response using ILC
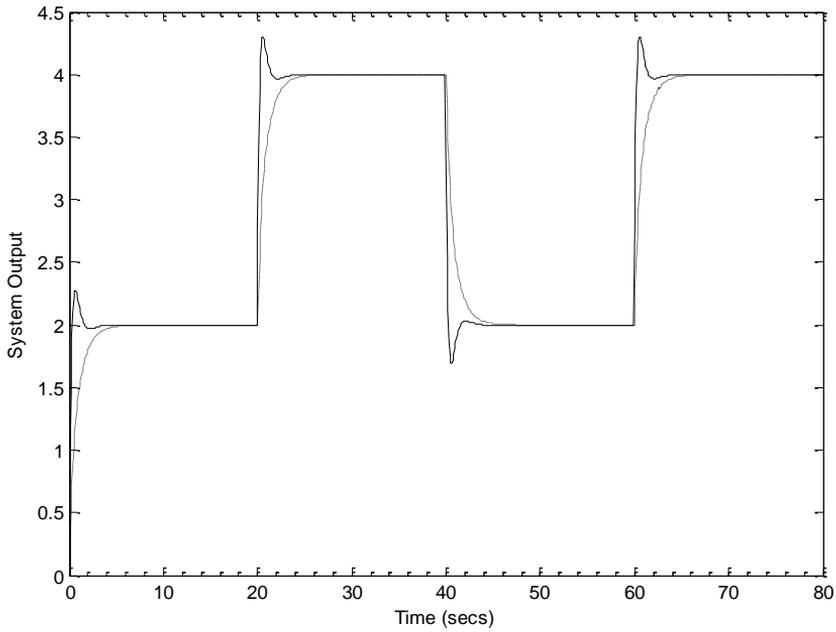(___ ILC response, ---- Reference)

Figure 8: Third order response using ILC and PI
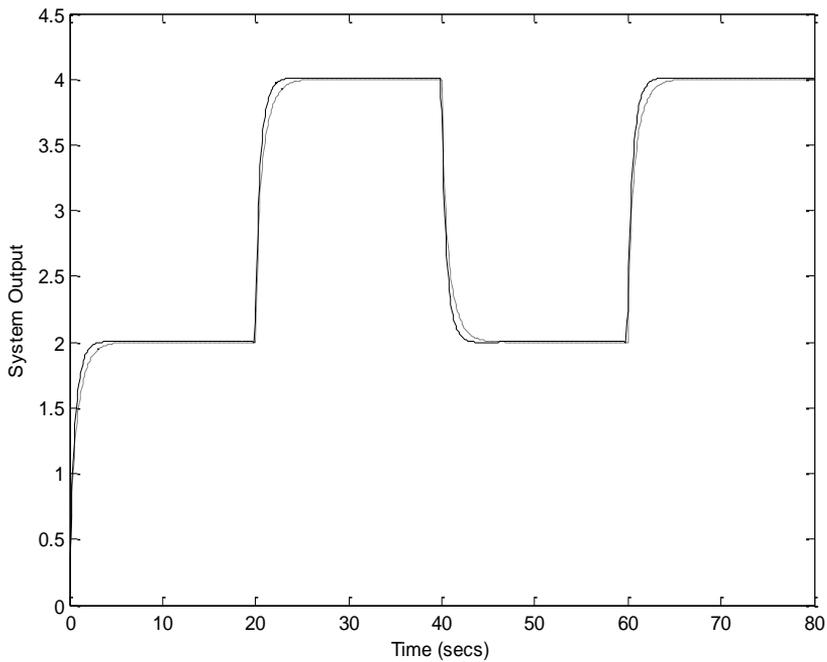(___ ILC (iteration 13), ---- PI controller)

Figure 9: High order ILC and PI responses
(___ ILC (iteration 11), ---- PI controller)

REFERENCES

1. S. Arimoto, S. Kawamura, and F. Miyazaki, "Bettering operation of robots by learning," J. Robot. Syst., Vol. 1, No. 2, pp. 123–140, 1984.
2. K. Moore, "Multi-loop control approach to designing iterative learning controllers," in Proceedings of 37th IEEE Conference on Decision and Control, (Tampa, Florida), pp. 666—671, December 1998.
3. G. Casalino and G. Bartolini, "A learning procedure for the control of movements of robotic manipulators", IEEE transactions on robotics and automation, Vol. 18, No. 4, August 2002.
4. J. J. Craig, "Adaptive control of manipulators through repeated trials," in Proc. ACC, San Diego, CA, pp. 1566–1572, June 1984
5. P. Bondi, G. Casalino, and L. Gambardella, "On the iterative learning control theory for robotic manipulators," IEEE J. Robot. Automat., Vol. 4, pp. 14–22, Feb. 1988.
6. K. Guglielmo and N. Sadegh, "Theory and implementation of a repetitive robot controller with Cartesian trajectory description," J. Dynam. Syst., Meas., Contr., Vol. 118, pp. 15–21, March 1996.
7. R. Horowitz, W. Messner, and J. B. Moore, "Exponential convergence of a learning controller for robot manipulators," IEEE Trans. Automat. Contr., Vol. 36, No. 4, pp. 890–894, July 1991.
8. F. Lange and G. Hirzinger, "Learning accurate path control of industrial robots with joint elasticity," in Proc. IEEE Conf. Robotics and Automation, Detroit, MI, pp. 2084–2089, May 1999.
9. J.R. Cueli, and C. Bordons, "Application of iterative nonlinear model predictive control to a batch pilot reactor", IFAC congress, Prague, 2005.
10. R. Precup, S. Preitl, "On the use of iterative learning control in fuzzy control system structures", 7th International Symposium of Hungarian Researchers on Computational Intelligence, pp.69-82, Budapest, 2006.
11. G. Heinzinger, D. Fenwick, B. Paden and F. Miyazaki, "Stability of learning control with disturbances and uncertain initial Conditions", *IEEE Trans. of Automatic Control*, Vol. 37, No. 1, 1992.
12. J. X. Xu, Z. Qu, "Robust iterative learning control for a class of nonlinear systems", *Automatica*, Vol. 34, No. 8, pp. 983-988, 1998.
13. J. H. Moon, T. Y. Doh, M.J. Chung, "A robust approach to iterative learning control design for uncertain systems", *Automatica*, Vol. 34, No. 8, pp. 1001-1004, 1998.
14. A. Tayebi, M. B. Zaremba, "Robust iterative learning control design is straightforward for uncertain LTI systems satisfying the robust performance condition", *IEEE Transactions on Automatic Control*, Vol. 48, No. 1, pp. 101-106, 2003.
15. S. Gopinath, Kar, I. N., and Bhatt, R. K. P. "Iterative learning control design for discrete-time systems using 2-d system theory", Proceedings of the International Conference on Advances in Control and Optimization of Dynamical Systems, pp.23-28 2007.
16. Y. Chen, C. Wen, J.-X. Xu and M. Sun, "High-order iterative learning identification of projectile's aerodynamic drag coefficient curve from radar

measured velocity data", *IEEE Trans. on Contr. Sys. Tech.,* Vol. 6, No. 4, pp. 563–570, 1998.

17. Y., Chen, Z. Gong and C. Wen, "Analysis of a high order iterative learning control algorithm for uncertain nonlinear systems with state delay". *Automatica,* Vol. 34, No. 3, pp. 345–353, 1998.

18. H. W. Gomma, "Adaptive PID Control Design Based on Generalized Predictive Control (GPC)" IEEE CCA, Taipei, Taiwan, pp. 1685-1690, 2004.

19. L. Ljung, "System Identification: Theory for the User", (2nd Edition), Prentice-Hall, Inc., New Jersey, 1999.

20. K. L. Moore and Chen, Y. Q., "On monotonic convergence of high order iterative learning update laws", IFAC 15th Triennial World Congress, Barcelona, Spain, 2002.

21. K. L. Moore and Chen, Y. Q., "A separative high-order framework for monotonic convergent Iterative Learning Controller Design", Proceedings of the American Control Conference Denver, pp.3644-3649, Colorado June 4-6, US, 2003.

تصميم نظام تحكم تناسبي تكاملي مكافئ لنظم التحكم المتعلمة بالتكرار

يقدم هذا البحث تقنية يمكن استخدامها لتصميم نظام التحكم التناسبي – التكاملي (PI) مكافئ للعديد من أنظمة التحكم المتعلمة بالتكرار وسيتم تمثيل نظام التحكم التناسبي – التكاملي (PI) على هيئة معاملات "Z" عل النقيض لأنظمة التحكم المتعلقة بالتكرار الممثلة في الزمن فقط مما يعطيها سهولة في عملية تحليل الاتزان وسوف يجمع ال (PI) المصمم بمميزات سهولة ضبطه مع قدرته التعليمة و قد أظهرت النتائج قدرة التقنية المقترحة على تصميم نظم تحكم (PI) لها قدرة على إعطاء نفس الأداء عند مقارنتها بالأنظمة المستعملة بالتكرار عند تطبيقها على العديد من النظم بدون أي شروط مسبقة على طبيعة النظم المتحكم فيها من حيث النوع أو الدرجة