

PAPER • OPEN ACCESS

Iterative Learning Control for a Repetitive 2D Process

To cite this article: S Muthumari *et al* 2020 *IOP Conf. Ser.: Mater. Sci. Eng.* **995** 012003

View the [article online](#) for updates and enhancements.

A promotional banner for the 240th ECS Meeting. The banner features a colorful striped border at the top. On the left, the ECS logo is displayed in a green circle. To its right, the text reads "240th ECS Meeting" in large blue font, followed by "Oct 10-14, 2021, Orlando, Florida" in a smaller blue font. Below this, it says "Register early and save up to 20% on registration costs" in bold black text, and "Early registration deadline Sep 13" in a smaller black font. At the bottom left, there is a red "REGISTER NOW" button. On the right side of the banner, there is a photograph of a diverse group of people in a professional setting, with a man in a white shirt and tie clapping and smiling in the foreground.

ECS **240th ECS Meeting**
Oct 10-14, 2021, Orlando, Florida
**Register early and save
up to 20% on registration costs**
Early registration deadline Sep 13
REGISTER NOW

Iterative Learning Control for a Repetitive 2D Process

S Muthumari¹, V Gomathi^{1*}, R Srimathi², Kishore Eswaran³, V Kamatchi Kannan⁴

1 Tamilnadu Electricity Board, Tamilnadu, India.

1* Assistant Professor-III, SASTRA Deemed University, Thanjavur, Tamilnadu, India.

2 Assistant Professor-SR, Vellore Institute of Technology, Chennai, Tamilnadu, India.

3 Developer, KPIT, Bangalore, Karnataka, India.

4 Associate Professor, Bannari Amman Institute of Technology, Sathyamangalam, Tamilnadu, India.

muthumarikannan@gmail.com, gomathi@eie.sastra.edu, srimathi.r@vit.ac.in,
kishoreeswaran23@gmail.com, kamatchikannanv@bitsathy.ac.in

Abstract: To control the systems which operate in a repetitive mode is known to be a difficult task. For example, systems like robotic manipulators are used for doing the repetitive assigned task. This can be overcome by the iterative learning control (ILC) method. In this article, the DC motor model is discussed for controlling the speed of the motor by the design of an ILC algorithm, and its performance is compared with the conventional PID algorithm. The ILC algorithm is implemented for the minimum and non-minimum phase systems. The resulting response, absolute and integral absolute errors are plotted. The Z-N tuning method is used to implement PID control. The performance of the DC motor has been evaluated by comparing the Integral Square Error (ISE), Integral Absolute Error (IAE), and Integral Time Absolute Error (ITAE) with the conventional PID controller.

Keywords: DC motor, Iterative Learning Control (ILC), Z-N tuning, Integral Absolute Error (IAE), Integral Square Error (ISE), Integral Time Absolute Error (ITAE).



1. Introduction

The performance of a system that executes the same work many times can be improved by learning from the past execution (iteration). The improvement in performance can be achieved by connecting the control for subsequent iterations with error information. This technique is used to control the systems operating in a repetitive or trial-to-trial mode with the requirement that a reference trajectory $y_d(t)$ is defined over a finite interval $0 \leq t \leq \alpha$, where α denotes the trial length, is followed to high precision.

A typical control law is given in (1) the following form:

$$u_{k+1}(p) = u_k(p) + \Delta u_{k+1}(p), \quad k \geq 0 \quad (1)$$

The general block diagram for an ILC controller is given in the Figure 1.

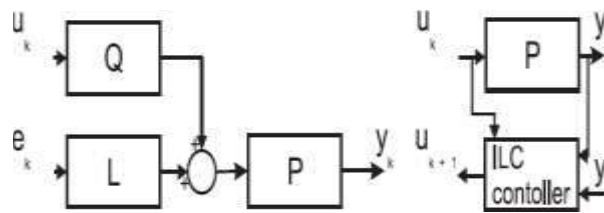


Figure 1. Block Diagram for an ILC controller

Two-dimensional process-based iterative learning control using noncausal finite-time interval data was explained by Błażej.C, Krzysztof.G, and Eric Rogers [1]. M. Rezaei and Kerman [2] have discussed PID parameter selection based on iterative learning control. Iterative learning control law design for an experimentally supported 2D process for error convergence and performance improvement was proposed by Lukasz. Hi and Krzysztof.G [3]. A review of repetitive control, run-to-run control, and ILC was proposed by Y.Wang, Furong.G, and Francis J. Doyle III [4]. Kevin L. Moore, Yang Q.C, and Hyo-Sung A [5] were explained about iterative learning control. Bristow, D.A.Tharayil, and M.Alleyne, A.G. [6] detailed ILC in a survey of iterative learning control paper. Owens D.H. [7],[8] presented optimization in iterative learning control. Jian-Xin Xu; Ying Tan[9] deals with both linear, nonlinear ILC. P. J. Schutyser [10] presented an approach to noncausal iterative learning Control. R.W. Longman [11] has discussed the ILC and repetitive control. Linear iterative learning control schemes were analyzed and explained elaborately by N. Amann, D. H. Owens, E. Rogers [12]. The paper is discussed in the following manner: Section 2 discusses the DC motor model. Section 3 explains the design of the ILC algorithm. Section 4 compares ILC and conventional algorithms. Section 5 includes the simulation results of the DC motor, and the performance criteria were evaluated. Finally, Section 6 concludes how the ILC is better than the conventional PID controller.

2. Modelling of A DC Motor

An electromechanical DC motor system is considered in this work. From the electrical system with field and armature circuit, only the armature is considered for the analysis as the field is excited by a constant voltage. The rotating part of the motor and load connected to the shaft on the mechanical system side. The DC motor speed control system with armature is shown in Figure 2.

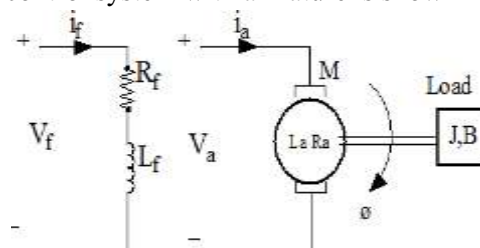


Figure 2. DC Motor Circuit

Where,

Ra is the Armature (Electric) resistance (Ω)

La is the Armature (Electric) inductance (H)

ia is Armature current (A)

Va is Armature voltage (V)

eb is the Back emf (V)

Tq is the torque developed by the motor (kg.cm)

ω is Angular displacement (rad/sec)

J is the Moment of inertia of motor (kg.m²)

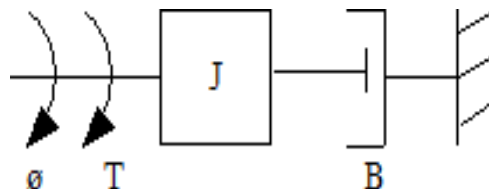


Figure 3. Mechanical Equivalent of the DC Motor

$$J \frac{d^2\theta}{dt^2} + B \frac{d\theta}{dt} = KI \tag{2}$$

$$Js^2\theta(s) + Bs\theta(s) = KI(s) \tag{3}$$

Where,

J is Motor's moment of inertia (kg.m²)

B is Motor and load's frictional coefficient (Nm.s)

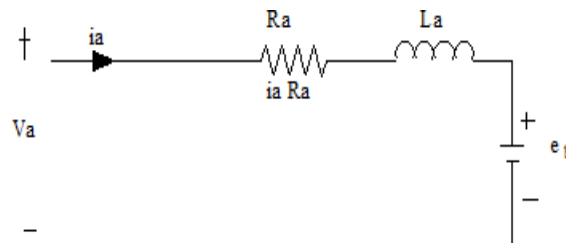


Figure 4. Electrical Equivalent of the DC Motor

$$e_b = K\omega = K \frac{d\theta}{dt} \tag{4}$$

eb is the back emf

Applying Kirchoff's voltage law

$$L \frac{di}{dt} + Ri = V - e_b \tag{5}$$

$$LsI(s) + RI(s) = V - Ks\theta(s) \tag{6}$$

$$I(s) = \frac{V(s) - Ks\theta(s)}{R + Ls} \tag{7}$$

Using the equations (2)-(7) obtained from the mechanical equivalent shown in Figure 3 and electrical equivalent shown in Figure 4, we can arrive at the transfer function of the DC motor.

$$G(s) = \frac{\omega(s)}{V(s)} = \frac{K}{[(R+S)(JS+B)+K^2]} \tag{8}$$

3. Design of ILC Algorithm

3.1 Design Guidelines

The closed-loop system transfer function is represented by P(s), and the controller is represented by C(s), and ILC is shown in Figure 5. The tracking error ek is an error at kth iteration and is given as input to the controller. It is found by comparing measured position yk with reference position

y_d . All iterations are of same finite time length, $t \in [t_0, t_{end}]$. The repetitive part of the error signal is reduced by ILC during the iterations. From one iteration to another, the feedforward signal is calculated by tracking error, and by using the learning filter L , the error is filtered and included in the forward path.

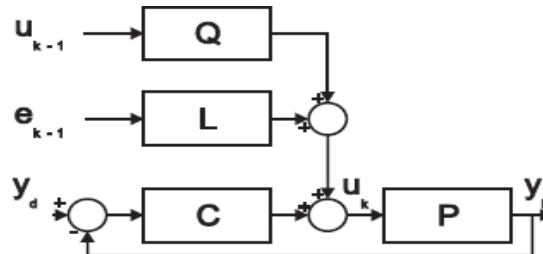


Figure 5. Block diagram of ILC Architecture

L , Q is ILC parameters to be chosen in accordance with the convergence criteria, where P is the process's transfer function, and SP is the sensitivity function of the same.

The control law (9) linear time-invariant (LTI) system $y_k = P*u_k$ is as

$$u_{k+1} = Q*u_k + L*(y_d - y_k) \tag{9}$$

A sufficient condition for convergence of the above equation for all y_d and all u_0 is given in (10)-(15) :

$$|Q - SP*L| < 1 \tag{10}$$

Proof

$$e_k = y_d - y_k, \text{ and } y_k = P*u_k \tag{11}$$

$$u_{k+1} = Q*u_k + L*e_k \tag{12}$$

$$e_{k+1} = y_d - P*Q*u_k - P*L*e_k \tag{13}$$

$$e_{k+1} = y_d - Q*y_k - P*L*e_k \tag{14}$$

$$e_{k+1} = (1 - Q)y_d + (Q - P*L)e_k \tag{15}$$

The error will decrease when Q and L satisfy the criteria equation of convergence. The fixed point error will be reached after one trial when $(Q - SP*L) = 0$. The error will increase every trial, and finally, the output won't be bounded when $Q - SP*L > 1$ then. The error will be constant as in (16) after a number of trials because it converges to a fixed point.

$$\lim_{k \rightarrow \infty} e_{k+1} = \lim_{k \rightarrow \infty} e_k = e^* \tag{16}$$

Then the asymptotic error e^* is given by equation (17)

$$e^* = \frac{(1-Q)}{1-Q+P*L} y_d \tag{17}$$

And the asymptotic input is in (18) – (20)

$$u^* = \frac{1}{1-Q+P*L} y_d \tag{18}$$

$$u^* = Q u^* + L e^* \tag{19}$$

$$u^* = \frac{1}{1-Q} e^* \tag{20}$$

3.2 Learning filter Design (Design of L)

The selection of Q and L is necessary for the design of the ILC controller for an LTI process for achieving good performance. From the asymptotic error equation, it can be observed that Q must be chosen close to unity to get a minimum error. From the convergence criteria, the learning filter should

obey the convergence rule and can be obtained as $(SP)^{-1}$ for fast convergence.

Using the closed-loop transfer from the feedforward to the error signal, i.e., the process sensitivity $SP = P/(1+P*C)$. The propagation of the error from iteration to iteration can be written as $e_{k+1} = (Q - SP*L) e_k$.

3.3 Design of Q

Q should be designed in such a way that L becomes a proper transfer function when multiplied with Q.

Example given in (21)-(23)

For first order Q,

$$Q = 1/(\epsilon s + 1) \tag{21}$$

Second order Q,

$$Q = 1/(\epsilon s + 1)^2 \tag{22}$$

Third order Q,

$$Q = 1/(\epsilon s + 1)^3 \tag{23}$$

The value of ϵ should be appropriately chosen. As ϵ value is greater, the settling time for the response is greater. For small values of ϵ , the settling time is minimum.

4. Comparison of ILC And Conventional Algorithm

The ILC algorithm has been implemented for a minimum phase system, and various values of ϵ in Q design; the response plots are shown in Figures 6, 7, and 8.

4.1 Simulation Results for Example 1

$$G(s) = \frac{1}{s^3 + 3s^2 + 3s + 1}$$

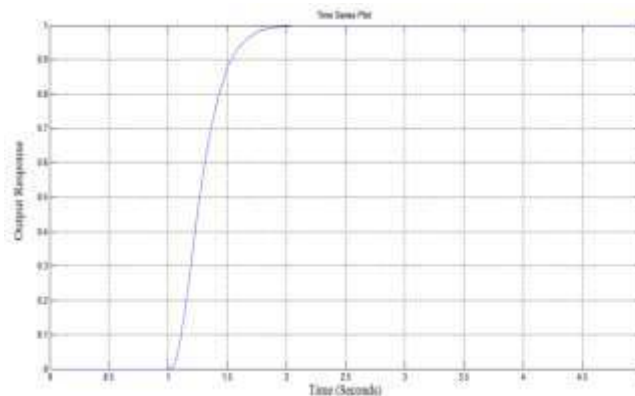


Figure 6. Output response when $\epsilon = 0.1$

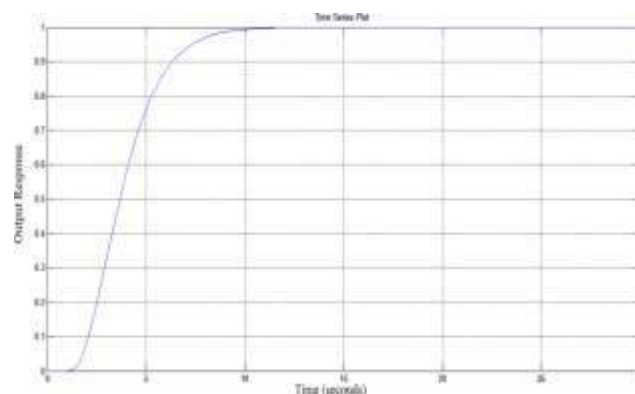


Figure 7. Output response when $\epsilon = 1$

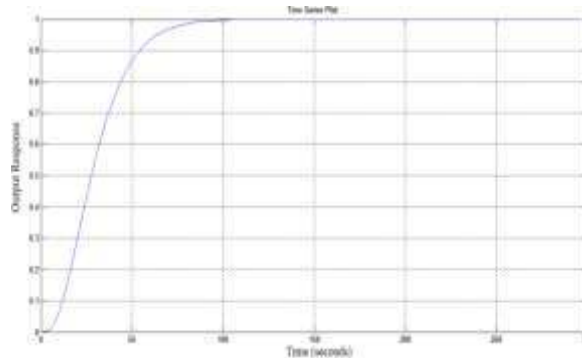


Figure 8. Output response when $\varepsilon = 1$

The PID controller is designed for DC motor velocity control. The mathematical representation of the PID controller is given in (24)

$$C(S) = K_p + \frac{K_i}{S} + K_d S \tag{24}$$

The Z-N tuning method can be used to evaluate K_p , K_i , and K_d .

The Table 1 and 2 show the performance indices ISE, IAE, ITAE for the systems designed under the ILC and Conventional algorithms. The simulations of the algorithms are done for three types of systems, namely: Minimum Phase System, Non-minimum phase system, and Delayed system.

Table 1. Performance Indices of Systems Design Under ILC Algorithm

Systems	Performance Indices
Minimum Phase System $\frac{1}{S^3 + 3S^2 + 3S + 1}$	ISE : 0.2062
	IAE : 0.3
	ITAE : 0.36
Non-Minimum Phase System $\frac{-S+1}{S^2+S+1}$	ISE : 0.5
	IAE : 1.0
	ITAE : 2
Delayed System $\frac{1}{S^2 + 3S + 1}$	ISE : 1.25
	IAE : 2
	ITAE : 5

Table 2. Performance Indices of Systems Design Under Conventional Algorithm

Systems	Performance Indices
Minimum Phase System $\frac{1}{S^3 + 3S^2 + 3S + 1}$	ISE : 0.81
	IAE : 0.725
	ITAE : 0.712
Non-Minimum Phase System $\frac{-S+1}{S^2+S+1}$	ISE : 2.076
	IAE : 2.22
	ITAE : 5.602
Delayed System $\frac{1}{S^2 + 3S + 1}$	ISE : 0.541
	IAE : 0.451
	ITAE : 2.41

4.2 Uncertainty Handling

For the minimum phase system discussed above, the ILC algorithm is executed when the system has 10% uncertainty. The output response of a step signal in the presence of 10% uncertainty is shown in Figure 9.

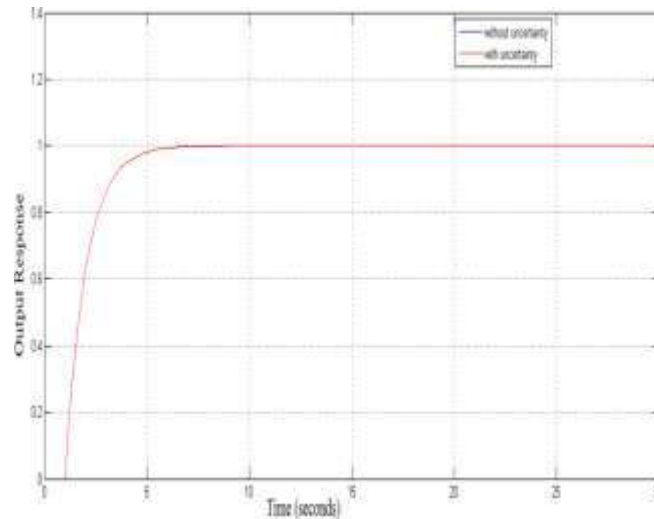


Figure 9. Output Response of ILC under 10% uncertainties

For the same minimum phase system, when handled by a conventional algorithm, the output response of a step signal in the presence of 10% uncertainty is shown in Figure 10.

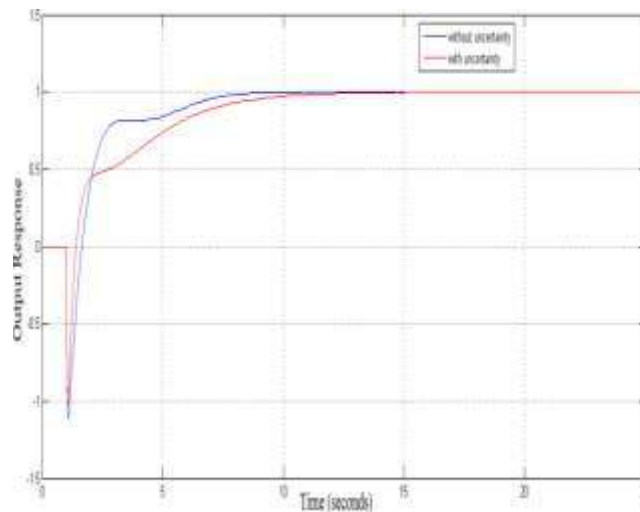


Figure 10. Output Response of conventional algorithm under 10% uncertainties

5. Results and Discussion

The proposed ILC controller is designed for the DC motor through simulation. The same is done using the conventional method. This is done for comparison purposes.

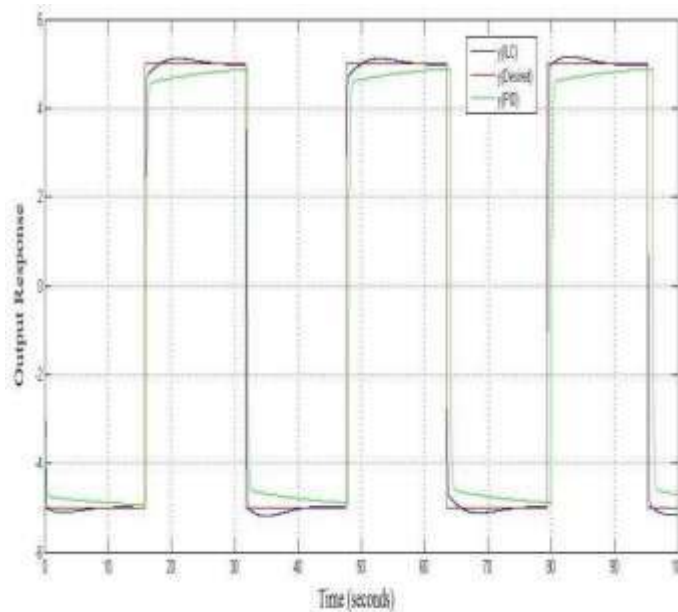


Figure 11. Output responses of ILC & Conventional Algorithms

Figure 11 shows the output response plot of both ILC and conventional algorithms. From the figure, it is clear that the rise time of an ILC algorithm is less when compared to a conventional algorithm. And there is an offset present in the conventional algorithm, and it is eliminated by the ILC algorithm. The Absolute Error versus Time plot is shown in Figure 12, and it reveals that the area under the graph is greater for the conventional algorithm than that of the ILC algorithm.

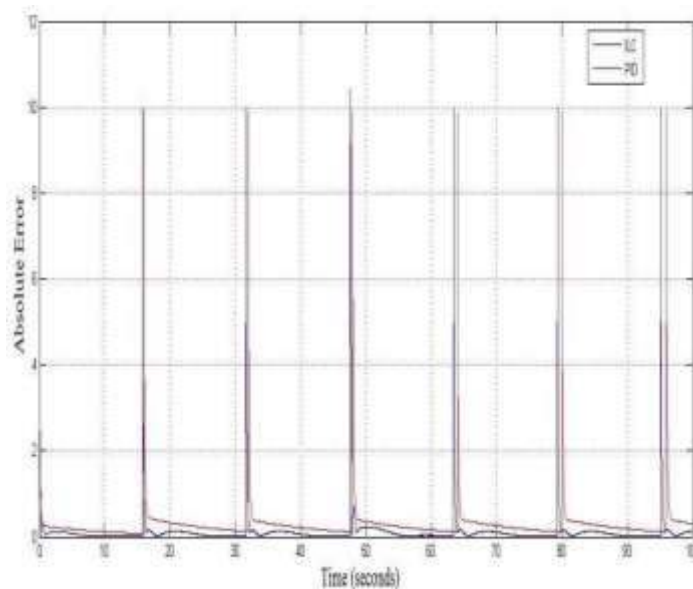


Figure 12. Plot of absolute error using ILC & PID for the DC Motor

The Integral Absolute Error plot is shown in the Figure 13 from that it is observe that the error value for conventional algorithm keeps increasing by large number in comparison with that of the ILC algorithm.

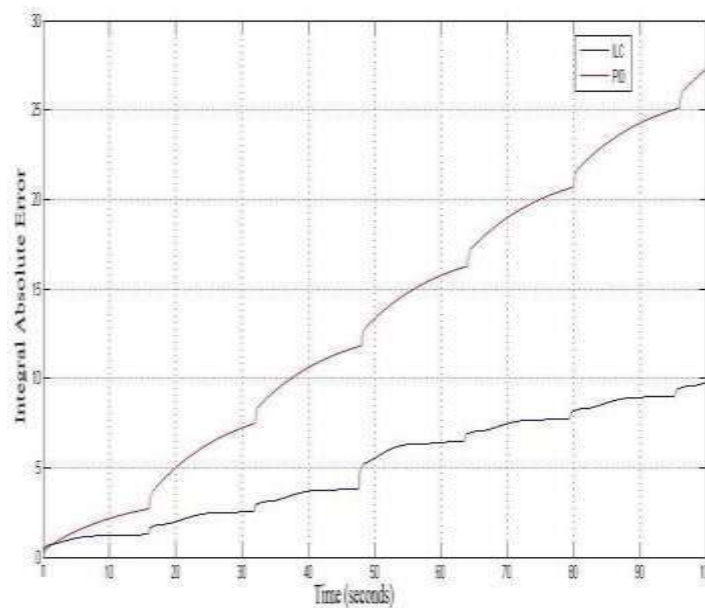


Figure 13. Performance measure of the DC Motor using ILC & PID Algorithms

6. Conclusion

The application-based implementation, as well as the test case systems of the ILC algorithm, are the evidence to validate the superiority of the ILC algorithm over the conventional algorithm. The ILC algorithm offers better speed control than the conventional control algorithm. The rise time is reduced, and the settling time is also lower, thus offering better efficiency. Thus the objective is validated, and hence we can conclude that the ILC algorithm offers better performance.

References

- [1] Błażej C Krzysztof G Eric Rogers 2014 2D systems based robust iterative learning control using noncausal finite-time interval data, *Systems & Control Letters* pp 36–42.
- [2] Rezaei M Kerman 2011 PID Parameter Selection Based on Iterative Learning Control *Electrical Engineering Department University of Kerman Contemporary Engineering Sciences Vol 4* pp 201 – 220,
- [3] Lukasz Hi Krzysztof G Zhonglun C Eric R Chris Freeman T Paul L 2010 Experimentally supported 2D systems based iterative learning control law design for error convergence and performance *Control Engineering Practice Vol 18* pp 339–348.
- [4] Wang Y Furong G Francis J Doyle III 2009 Survey on iterative learning control repetitive control and run-to-run control *Journal of Process Control* pp 1589–1600.
- [5] Kevin L Moore Yang Q C and Hyo-Sung A 2006 Iterative Learning Control A Tutorial and Big Picture View, *Tutorial Sessions - Industry and Applications IEEE CDC*.
- [6] Bristow DA Tharayil M Alleyne AG 2006 A Survey of Iterative Learning Control A learning based method for high performance tracking control *IEEE control systems magazine Vol 26* pp 96-144.
- [7] Owens DH Hatonen J 2005 Iterative learning control- An optimization paradigm *Annual Review in Control Vol 29 No 1* pp 57-70.
- [8] Owens DH Feng K 2003 Parameter optimization in iterative learning control *International Journal of Control Vol 76 No 11* pp 1059-1069.
- [9] Jian-Xin Xu Ying Tan 2003 *Linear and Nonlinear Iterative Learning Control Springer-Verla* pp 177 ISBN 3-540-40173-3.

- [10] Schutyser PJ 2002 Noncausal Iterative Learning Control Control Laboratory Faculty of Electrical Engineering University of Twente.
- [11] Longman RW 2000 Iterative Learning Control and Repetitive Control for Engineering Practice International Journal of Control Vol 73 pp 930–954.
- [12] Amann N Owens DH Rogers E 2000 Analysis of Linear Iterative Learning Control Schemes - A 2D Systems/Repetitive Processes Approach Multidimensional Systems and Signal Processing Vol 11 pp 125–177.