

# Point to Point ILC with Initial State Learning using Neural Networks

Haris Anwaar

University of Science and  
Technology, Beijing, China

Yin YiXin

University of Science and  
Technology, Beijing, China

Muhammad Ammar

Ashraf

University of Science and  
Technology, Beijing, China

Salman Ijaz

Beihang University, Beijing, China

## ABSTRACT

Point to Point ILC involves the tracking of specific points during motion in a repetitive manner. Point to point ILC makes the assumption that initial starting position of each trial remains same. In this paper, initial starting position of point to point motion in each trial is learned using neural networks. The proposed algorithm can also track the points which are changing in respective trials. The algorithm is checked for three points tracking during a trial, which are changing in sinusoidal manner. The results are shown by simulations in the end.

## Keywords

Point to Point Iterative learning control, Adaptive ILC, Iterative Learning control

## 1. INTRODUCTION

Many control systems in practice perform finite duration tasks repetitively, the system resets to the initial condition after the completion of each task, and the task is required to be repeated again. In these special systems, the control systems need to follow a specific trajectory, which can involve the tracking of entire trajectory, or the intermediate points or only the last point or form. Control technique to deal with these problems is classified as iterative learning control (ILC), first introduced by Arimoto. Based on the tracking requirements described above, iterative learning control (ILC) is termed as point to point ILC for intermediate points tracking or specific points tracking as given by [1]-[4], terminal ILC for final point tracking as given by [5]-[9]. The strength of ILC is to utilize the process information comprising of historical data and to improve the tracking performance on every trial. Since classical iterative learning control algorithm was introduced by Arimoto, the study of ILC is receiving extensive attention both for the improvement in its own algorithm i.e., theoretical domain and its application areas.

Iterative learning control (ILC) is regarded as an intelligent control technique, and is used to improve the transient tracking performance for the systems which work in repetitive manner. ILC still finds importance for numerous control processes due to the existence of un modeled dynamics, the parametric uncertainties, or the disturbances and measurement noise emerging during actual system operation or due to the lack of suitable model design techniques. Especially in case of nonlinear system, where traditional control methods are not enough. So, for those systems that perform the tasks repetitively, ILC is a technique that can help in overcoming the limitations of other conventional controllers, hence making it

likely to realize the perfect tracking performance when there are uncertainties in the model. ILC deals well in case of repetitive disturbances, whereas, in order to improve the robustness against non-repetitive disturbance, ILC is combined with feedback control.

Point to point ILC or Terminal ILC can be described as the special cases of traditional ILC, i.e., entire trajectory tracking. So, there is a question that why we still need to define point to point ILC and terminal ILC. To emphasize the need of this classification, a few reasons are elaborated in the following:

Point to point ILC and terminal ILC are required because of the non-availability of measurements on the entire operation trajectory. However, there are cases where measurements are available for the entire trajectory, but the tracking of some points is more important, standard ILC can be applied by designing reference passing through these points [10][11], [2]. Same way PTP ILC can be applied for TILC as it is a special case of PTP ILC. Even in these cases PTP ILC and TILC is still preferable because of the difficulty of selecting an optimal trajectory passing through the required points in case of PTP ILC, and in case of change in the plant, the optimal reference trajectory will no longer be optimal.

The size of memory and hence the computations can be reduced. As in case of storing measurements for general ILC, greater memory is required because it needs to store all the measurements, while in point to point ILC only fewer measurements at specified points need to be stored, same way for TILC only one point measurements storage is required.

Performance demands can be achieved in a better way while using PTP ILC and TILC. In these cases, as described above, the algorithms they target specific points, where unnecessary constraints are removed, which can result in less control effort and better convergence speed.

Point to point control needs the motion profile to be generated in advance, and the tracking controller is needed to be designed. Whereas ILC provides the capability of learning from previous trials. So ILC can handle this problem simply by the use of any reference connecting the desired critical points [12]. An extra freedom is obtained because of the removal of the unnecessary constraints which the plant is following a predefined output between points in case of traditional ILC.

In this paper, a technique is proposed which calculates the effect of initial states on the output of the system, and an adaptive point to point ILC algorithm is proposed, which tracks the specific points at specific intervals during the whole

working iteration. The proposed technique provides fast convergence to the desired points even in case of variable initial states. Hence, the proposed algorithm can deal with wide number of ILC areas.

In section II, point to point ILC problem and its representation is described. In section III the mathematical representation for the initial state learning is given. Section IV describes the proposed technique in detail. Section V describes the simulation results which shows the usability of the proposed technique. In section VI the conclusion is given.

## 2. POINT TO POINT ILC

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

$$\begin{aligned} x_k(t+1) &= Ax_k(t) + Bu_k(t) + w(t) \\ y_k(t) &= Cx_k(t) + v(t) \end{aligned} \quad (1)$$

where  $t \in [0, T]$ . Here  $x(t) \in R^n, y(t) \in R^m$  and  $u \in R^m$  represent states, output and input variables of system respectively. Whereas  $w(t)$  and  $v(t)$  represent system and measurement noise. Here  $t$  and  $k$  represents time and iteration index.

Next, the  $N$ -sample input and output terms can be expressed as

$$\begin{aligned} u_k(t), t \in \{0, 1, \dots, T-1\}, \\ y_k(t), t \in \{n, n+1, \dots, T+n-1\}, \end{aligned}$$

and desired system output is expressed as

$$y_d(t), t \in \{n, n+1, \dots, T+n-1\}.$$

Here  $T$  is time duration of a trial and  $n$  represents the order of the system. The error signal for above sequence is defined by  $e_k(t) = y_d(t) - y_k(t)$ . In practice,  $T$  is always finite, however for analysis and design, it's useful sometimes to consider trial length as infinite. The objective of ILC is the production of a series of inputs  $u_k(t)$ , so that the output and the desired system output  $y_k(t)$  tracks efficiently the reference output  $y_d(t)$  or as  $t \rightarrow \infty, e_k(t) \rightarrow 0$ .

Point to Point ILC, contrary to traditional ILC tracks specific points. In this case the reference is not defined for the entire trajectory, i.e., we don't have the measurements of the entire trajectory. So the tracking of points other than specified by the reference is meaningless or irrelevant in this case. Its example can be given by, a robotic manipulator executing a pickup and placing task, here we are only concerned about tracking performance at the picking place and at the position of placing down whereas the behavior within these positions is of less concern. This also involves certain constraints, which will be explained later on. It can also be termed as intermediate point ILC which means that some points inside the trajectory are more important to track.

Consider the system represented by equation (1) again, to deal with the point to point ILC, the output normally is written in the lifted matrix form with relative degree 1 i.e.,  $CB \neq 0$

$$y_k = Gu_k + d \quad (2)$$

Here  $G$  is the  $N \times N$  matrix

$$G = \begin{bmatrix} CB & 0 & \dots & 0 & 0 \\ CAB & CB & \dots & 0 & 0 \\ CA^2B & CAB & \dots & 0 & 0 \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ CA^{N-1}B & \dots & CAB & CB \end{bmatrix} \quad (3)$$

Next the  $N$ -sample terms of inputs and outputs can be expressed as under

$$u_k = [u_k(t_1) \quad u_k(t_2) \quad \dots \quad u_k(t_N)] \quad (4)$$

$$y_k = [y_k(t_1) \quad y_k(t_2) \quad \dots \quad y_k(t_N)] \quad (5)$$

and the desired system output

$$y_k^e = [y_k(t_1) \quad y_k(t_2) \quad \dots \quad y_k(t_M)] \quad (6)$$

Hence,  $y_k^e$  can be written as

$$y_k^e = G_e u_k + d^e \quad (7)$$

Where  $G_e, d^e$  are  $M \times N$  matrix and  $M \times 1$  vector obtained by extracting the  $t_i$ th rows  $1 \leq i \leq M$  from  $G$  and  $d$ , respectfully.

Hence in this case the tracking error vector is written as

$$e_k^e = r^e - y_k^e \quad (8)$$

Hence the problem in point to point ILC is to find the input updating control law so that the tracking error at the  $M$  intermediate points i.e.,  $0 \leq t_i \leq t_M$  asymptotically follows  $\lim_{k \rightarrow \infty} e_k^e \rightarrow 0$  and  $\lim_{k \rightarrow \infty} u_k = u^*$ .

Reference signal to be followed in case of Point to Point ILC is a subset of the entire trial duration, i.e., the reference is only defined on  $0 \leq t_i \leq t_M$  intermediate points where  $M < N$ . In this case reference in vector form can be represented as follows:

$$r^e = [r(t_1) \quad r(t_2) \quad \dots \quad r(t_M)]^T \quad (9)$$

In order to reach to the specific points, matrix  $G$  is used and multiplied with another matrix  $J$ .  $J$  matrix can be defined as

$$J_{i,j} = \begin{cases} 1 & i = 1, 2, \dots, M, j = I \\ 0 & \text{otherwise} \end{cases} \quad (10)$$

Where  $i$  represents the position of the points to be tracked. For an example, consider  $T_s$  as sampling point, and at 3 time instants  $T_1, T_2$  and  $T_3$  and  $J_1 = T_1/T_s, J_2 = T_2/T_s$  and  $J_3 = T_3/T_s$  and here matrix  $J_{i,j}$  is written as under

$$J = \begin{bmatrix} 0 & \dots & 0 & 1 & 0 & \dots & \dots & \dots & 0 \\ 0 & \dots & \dots & \dots & 0 & 1 & 0 & \dots & 0 \\ 0 & \dots & \dots & \dots & \dots & \dots & \dots & 0 & 1 \end{bmatrix} \quad (11)$$

And after multiplying it with  $G$  matrix defined above we get the matrix of the form shown below

$$\begin{bmatrix} CA^{a-1}B & \dots & CAB & CB & 0 & \dots & \dots & \dots & 0 \\ CA^{b-1}B & \dots & \dots & \dots & CAB & CB & 0 & \dots & 0 \\ CA^{c-1}B & \dots & \dots & \dots & \dots & \dots & \dots & CAB & CB \end{bmatrix} \quad (12)$$

Where  $a, b$  and  $c$  represent the position of the points to track. The above matrix describes the weights to be multiplied with the input vector  $u$ , at specific iterations to give the output value which can be described as under

$$y_k = J * G * u_k \quad (13)$$

Where  $y_k$  represents the plant output at trial  $k$  and represents the value of output on the points which we want to track.

Now, the objective is to find  $u_k$  such that following condition is satisfied

$$\lim_{k \rightarrow \infty} \|y_r - JGu_k\| = 0 \quad (14)$$

Now the standard ILC law can be written as

$$u_{k+1} = u_k + L(y_r - Jy_k) \quad (15)$$

So that the error evolution between the points is written in the following form

$$J e_{k+1} = (I_M - JGL) J e_k \quad (16)$$

And the convergence condition is given as

$$\rho(I_M - JGL) < 1 \quad (17)$$

This condition guarantees point-to-point error of reaching zero value.

### 3. PROBLEM FORMULATION FOR INITIAL CONDITION TRACKING

Consider eq (1) again, here the relation between the initial conditions and the output at specific instants of the trial can be written as follows

$$\begin{aligned} y_k(p_1) &= CA^{p_1}x_k(0) + C \sum_{t=0}^{p_1-1} A^{p_1-t-1}B u_k \\ y_k(p_2) &= CA^{p_2}x_k(0) + C \sum_{t=0}^{p_2-1} A^{p_2-t-1}B u_k \\ y_k(p_3) &= CA^{p_3}x_k(0) + C \sum_{t=0}^{p_3-1} A^{p_3-t-1}B u_k \\ &\vdots \\ y_k(p_N) &= CA^{p_N}x_k(0) + C \sum_{t=0}^{p_N-1} A^{p_N-t-1}B u_k \end{aligned} \quad (18)$$

Here  $p_1, p_2, p_3$  and  $p_N$  represents the points to be tracked during the trial.

The above formulation can also be written in the form

$$\begin{aligned} y_k(p_1) &= f_{p_1}(x_k(0)) + B_{p_1}^* u_k \\ y_k(p_2) &= f_{p_2}(x_k(0)) + B_{p_2}^* u_k \\ y_k(p_3) &= f_{p_3}(x_k(0)) + B_{p_3}^* u_k \\ &\vdots \\ y_k(p_N) &= f_{p_N}(x_k(0)) + B_{p_N}^* u_k \end{aligned} \quad (19)$$

Here  $f_{p_N}(x_k(0)) = CA^{p_N}x_k(0)$  and  $B_{p_N}^* = C \sum_{t=0}^{p_N-1} A^{p_N-t-1}B$ .

In order to approximate the relation between the initial states and the output at specific instants, RBF neural networks has been introduced. Which is written as

$$f_{p_N}(x_k(0)) = W_{p_N}^T \varphi(x_k(0)) \quad (20)$$

Here  $W_{p_N}^T \in R^{n \times O}$  represents an unknown ideal weight matrix,  $O$  represents the number of neurons in the hidden layer.  $f_{p_N}(\cdot)$  greatly depends on the structure of neural network and the number of neurons.  $\varphi(x_k(0)) \in R^O$  represents the output of the neural network hidden layer.

Hence, the system's output at specific instants of time in terms of above RBF network can be written as follows

$$\begin{aligned} y_k^{NN}(p_1) &= f_{p_1}(x_k(0), W) + B_{p_1}^* u_k = W^T \varphi(x_k(0)) + B_{p_1}^* u_k \\ y_k^{NN}(p_2) &= f_{p_2}(x_k(0), W) + B_{p_2}^* u_k = W^T \varphi(x_k(0)) + B_{p_2}^* u_k \\ y_k^{NN}(p_3) &= f_{p_3}(x_k(0), W) + B_{p_3}^* u_k = W^T \varphi(x_k(0)) + B_{p_3}^* u_k \\ &\vdots \\ y_k^{NN}(p_N) &= f_{p_N}(x_k(0), W) + B_{p_N}^* u_k = W^T \varphi(x_k(0)) + B_{p_N}^* u_k \end{aligned} \quad (21)$$

Here, Let  $\Theta = [W_{p_1}^T, W_{p_2}^T, W_{p_3}^T, \dots, W_{p_N}^T, B_{p_1}^*, B_{p_2}^*, B_{p_3}^*, \dots, B_{p_N}^*]$ ,  $\psi_k = [\varphi(x_k(0))^T, u_k^T]$

Which can be written in compact form as under,

$$y_k^{NN}(p_N) = \Theta \psi_k \quad (23)$$

**Assumption 1.** The system is both controllable, and observable,  $n \leq m$ , and  $B_{p_N}^* \in R^{n \times m}$  is full rank.

**Assumption 2.** Initial state  $x_k(0)$  is accessible in every iteration.

### 4. CONTROLLER DESIGN

The control update law for the proposed technique is given by the following equation:

$$u_k = (\hat{B}_k^*)^{-1} [y_{d,k}(N) - \hat{W}_k^T \varphi(x_k(0))] \quad (24)$$

Here  $\hat{B}_k^*$  and  $\hat{W}_k^T$  are the estimates of  $B$  and  $W$  for the  $k$ -th iteration.

$u_k$  for every iteration is applied on the plant and the neural network approximation function.

The estimated plant output is

$$\begin{aligned} \hat{y}_k(N) &= \hat{f}(x_k(0), \hat{W}_k) + \hat{B}_k^* u_k \\ &= \hat{W}_k^T \varphi(x_k(0)) + \hat{B}_k^* u_k \end{aligned} \quad (25)$$

The estimate error is

$$\hat{e}_k(N) = \hat{y}_k(N) - y_k(N) \quad (26)$$

Let  $\hat{\Theta}_k = [\hat{W}_k^T, \hat{B}_k^*]$ ,  $\hat{\theta}_{i,k}$  represent the  $i$ -th row vector of  $\hat{\Theta}_k$

And  $\hat{\omega}_{i,k}^T$  and  $\hat{b}_{i,k}^*$  denote the  $i$ -th row vectors of  $\hat{W}_k^T$  and  $\hat{B}_k^*$  respectively. Then,  $\hat{\theta}_{i,k} = [\hat{\omega}_{i,k}^T, \hat{b}_{i,k}^*]$ ,  $i = 1, 2, \dots, n$ . Hence the estimated model is

$$\hat{y}_k(N) = \hat{\omega}_{i,k}^T \varphi(x_k(0)) + \hat{b}_{i,k}^* u_k = \hat{\theta}_{i,k} \Gamma_k \quad (27)$$

And the neural network updating law is

$$\hat{\theta}_{i,k+1} = \hat{\theta}_{i,k} - \frac{\Gamma_k^T}{1 + \Gamma_k^T \Gamma_k} \quad (28)$$

### 5. CASE STUDY

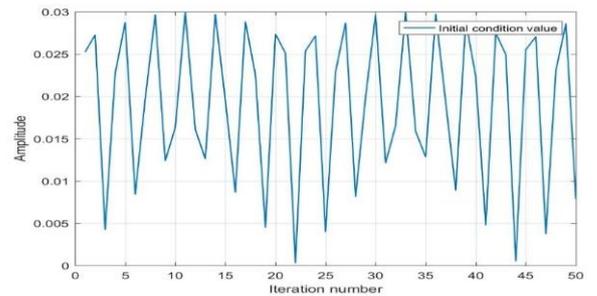
In order to validate the proposed methodology, the following discrete-time SISO system is used

$$x_k(t+1) = \begin{bmatrix} 0.5 & 0.035 & 0.025 \\ 0.0255 & 0.6 & -0.99 \\ 0.2 & 0.2 & 0.0 \end{bmatrix} x_k(t) + [0.2 \ 0.2 \ 0.0]^T u_k \quad (29)$$

The interval is  $[0, 20]$ .

The initial starting points are changing randomly within the range  $[0, 0.1]$  for every iteration. The tracking reference points can be changing, for the simulation three points are chosen to be tracked and these three points are changing per iteration.

For the simulation the value of  $d_0 = 0.001$ ,  $\theta = [5 \ 5 \ 0.5]$  and  $u_0 = 0$ . The error propagation for three different points tracking within the iteration are given in the figure below. From the figures it can be seen that although the initial conditions are varying but the proposed algorithm track the iteration variant reference for every points within one trial.



**Figure 1** Random initial conditions for respective iteration number

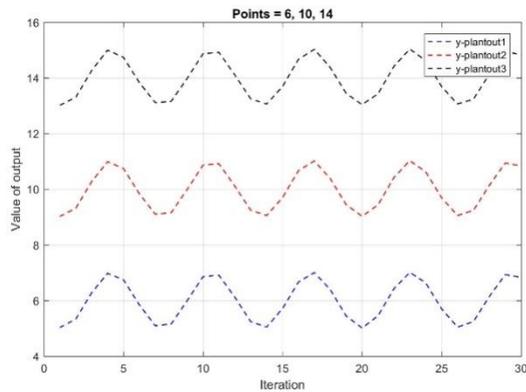


Figure 2 Points tracking performance

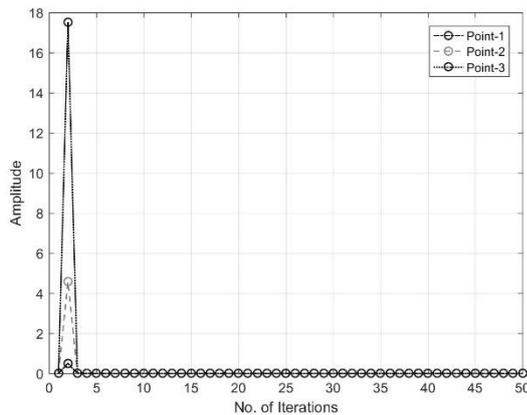


Figure 3 Error propagation per iteration

## 6. CONCLUSION

A point to point ILC algorithm is proposed which introduces the capability of tracking the iteration varying initial start position changes. Moreover, the proposed algorithm has the ability to track iteration varying reference points within the iterations. This algorithm can be used in number of applications providing adaptivity to the initial condition changes and changing reference points. Good convergence is achieved using neural networks learning.

## 7. REFERENCES

- [1] C. T. Freeman and Y. Tan, "Iterative learning control with mixed constraints for point-to-point tracking," *IEEE Trans. Control Syst. Technol.*, vol. 21, no. 3, pp. 604–616, 2013.
- [2] H. Ding and J. Wu, "Point-to-point motion control for a high-acceleration positioning table via cascaded learning schemes," *IEEE Trans. Ind. Electron.*, vol. 54, no. 5, pp. 2735–2744, 2007.
- [3] C. Chien and Y. Shu, "Study of a Class of Sampled-Data ILC from the Point of Performance Improvement and Memory Capacity," *Control and Decision Conference (CCDC)*, Chinese, pp. 4197–4202, 2016.
- [4] R. Chi, Z. Hou, B. Huang, and S. Jin, "A Unified Data-driven Design Framework of Optimality-based Generalized Iterative Learning Control," *Comput. Chem. Eng.*, vol. 77, pp. 10–23, 2015.
- [5] R. Chi, Z. Hou, S. Jin, C. J. Chien, and D. Wang,

"Terminal ILC design and analysis via a dynamical predictive model," *IEEE Int. Conf. Control Autom. ICCA*, vol. 639798, no. 1, pp. 1156–1161, 2014.

- [6] J. Xu, Y. Chen, T. H. Lee, and S. Yamamoto, "Terminal iterative learning control with an application to RTPCVD thickness control," *Automatica*, vol. 35, pp. 1535–1542, 1999.
- [7] G. Gauthier, M. Ieee, B. Boulet, and S. Member, "Robust Design of Terminal ILC with an Internal Model Control Using  $\mu$ -analysis and a Genetic Algorithm Approach," *American Control Conference*, pp. 2069–2075, 2010.
- [8] G. Gauthier and B. Boulet, "Terminal Iterative Learning Control design with singular value decomposition decoupling for thermoforming ovens," *2009 Am. Control Conf.*, no. 1, pp. 1640–1645, 2009.
- [9] Y. Liu, R. Chi, and Z. Hou, "Terminal ILC for Tracking Iteration-varying Target Points," *Asian Journal of Control*, vol. 12, no. June, pp. 266–272, 2015.
- [10] K.-H. Park, "A study on the robustness of a PID-type iterative learning controller against initial state error," *Int. J. Syst. Sci.*, vol. 30, no. 1, pp. 49–59, 1999.
- [11] J. Van De Wijdeven and O. Bosgra, "Hankel iterative learning control for residual vibration suppression with MIMO flexible structure experiments," *Proc. Am. Control Conf.*, vol. 1, pp. 4993–4998, 2007.
- [12] C. T. Freeman, "Constrained Point-to-Point Iterative Learning Control," *IFAC Proceedings Volumes*, vol. 0, no. 7, pp. 3611–3616, 2011.
- [13] L. Wang, C. T. Freeman, and E. Rogers, "Predictive iterative learning control with experimental validation," *Control Eng. Pract.*, vol. 53, pp. 24–34, 2016.
- [14] B. Wahlberg, "System Identification Using Laguerre Models," *IEEE Trans. Automat. Contr.*, vol. 36, no. 5, pp. 551–562, 1991.
- [15] P. M. Van den Hof and O. H. Bosgra, "A Generalized Orthonormal Basis for Linear Dynamical Systems," *IEEE Trans. Automat. Contr.*, vol. 40, no. 3, pp. 451–465, 1995.
- [16] L. Wang, "Discrete model predictive controller design using Laguerre functions," *J. Process Control*, vol. 14, no. 2, pp. 131–142, 2004.
- [17] H. Ahn, Y. Chen, and K. L. Moore, "Iterative Learning Control: Brief Survey and Categorization," *IEEE Transactions on Systems, Man, and Cybernetics, Part C (Applications and Reviews)*, vol. 37, no. 6, pp. 1099–1121, 2007.
- [18] D. A. Bristow and M. Tharayil, "A learning-based method for high-performance tracking control," *IEEE Control Systems*, June, pp. 96–114, 2006.
- [19] C. T. Freeman, "Constrained point-to-point iterative learning control with experimental verification," *Control Eng. Pract.*, vol. 20, no. 5, pp. 489–498, 2012.
- [20] B. Chu, C. T. Freeman, and D. H. Owens, "A Novel Design Framework for Point-to-Point ILC Using Successive Projection," *IEEE Transactions on Control Systems Technology*, vol. 23, no. 3, pp. 1156–1163, 2015.