

Point to Point ILC with Receding Horizon Optimization Approach

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

Iterative learning control is a control technique used for the tracking of a finite duration trajectory. Iterative learning control (ILC) with focus on speed of tracking specific points and tracking error on these points is analyzed in this paper. A technique is introduced which employs the receding horizon optimization to track the points along with the iterative learning control is introduced. In order to increase the efficiency of optimization, use of Laguerre functions is introduced which gives more freedom in parameterizing the optimization trajectory and in tuning the optimization parameters. Hence the technique can be efficiently used to track points within the trajectory with good performance.

Keywords

Point to Point Iterative learning control, receding horizon control, Iterative Learning control

1. INTRODUCTION

We Many control systems in practice performs 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 needs 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 unmodeled 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 combines the benefits of receding horizon optimization and the iterative learning control to track the points within the entire trajectory. Receding horizon optimization have the tendency to reach the set point quickly and without overshoots. Moreover, input and

output constraints can also be dealt by using receding horizon approach. Laguerre functions are introduced for parameterizing the optimization trajectory for fast sample data applications.

In section II, we have briefly described point to point ILC problem and its representation. In Section III receding horizon control with integral action is described mathematically, Section IV describes our proposed technique, which combines receding horizon with point to point ILC, Section V describes the simulation results which shows the usability of the proposed technique.

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 meaning less 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 (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*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 $M * N$ matrix and $M * 1$ 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.

3. RECEDING HORIZON OPTIMIZATION

While using receding horizon control approach, an optimal control problem using a finite horizon is solved, which gives open-loop states and control trajectories. The control trajectory obtained is then applied to the plant/system for a

segment of the horizon window. Hence, this optimization process is then performed again, which results in a sampled data feedback law.

$$\begin{aligned} x(t_i + 1 / t_i) &= Ax(t_i) + B\Delta u(t_i) \\ x(t_i + 2 / t_i) &= Ax(t_i + 1 / t_i) + B\Delta u(t_i + 1) \\ &= A^2x(t_i) + AB\Delta u(t_i) + B\Delta u(t_i + 1) \\ &\quad \vdots \\ x(t_i + N_p / t_i) &= A^{N_p}x(t_i) + A^{N_p-1}B\Delta u(t_i) \\ &\quad + A^{N_p-2}B\Delta u(t_i + 1) \\ &\quad + \dots + A^{N_p-N_c}B\Delta u(t_i + N_c - 1) \end{aligned} \quad (14)$$

From the above mentioned predicted state variables, the corresponding predicted output variables are represented as

$$\begin{aligned} y(t_i + 1 / t_i) &= CAx(t_i) + CB\Delta u(t_i) \\ y(t_i + 2 / t_i) &= CAx(t_i + 1 / t_i) + CB\Delta u(t_i + 1) \\ &= CA^2x(t_i) + CAB\Delta u(t_i) + CB\Delta u(t_i + 1) \\ &\quad \vdots \\ y(t_i + N_p / t_i) &= CA^{N_p}x(t_i) + CA^{N_p-1}B\Delta u(t_i) \\ &\quad + CA^{N_p-2}B\Delta u(t_i + 1) \\ &\quad + \dots + CA^{N_p-N_c}B\Delta u(t_i + N_c - 1) \end{aligned} \quad (15)$$

Here all the predicted variables have been written in terms of present state variable $x(t_i)$ and the future control movement $\Delta u(t_i + j)$, here $j = 0, 1, \dots, N_c - 1$.

In vector form, it is written as

$$Y = [y(t_i + 1/t_i) \ y(t_i + 2/t_i) \ \dots \ y(t_i + N_p/t_i)]^T \quad (16)$$

$$\Delta U = [\Delta u(t_i + 1) \ \Delta u(t_i + 2) \ \dots \ \Delta u(t_i + N_c)]^T \quad (17)$$

Where the dimension of vector Y is N_p and the dimension for ΔU is N_c . Above mentioned equations in compact matrix form is expressed as

$$Y = F x(t_i) + \varphi \Delta U \quad (18)$$

Here

$$F = \begin{bmatrix} CA \\ CA^2 \\ CA^3 \\ \vdots \\ CA^{N_p} \end{bmatrix}; \quad (19)$$

$$\varphi = \begin{bmatrix} CB & 0 & \dots & 0 \\ CAB & CB & \dots & 0 \\ CA^2B & CAB & \dots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ CA^{N_p-1}B & CA^{N_p-2}B & \dots & CA^{N_p-N_c}B \end{bmatrix} \quad (20)$$

Here the objective is to derive the predicted output variable as near as possible to the desired set-point $r(t_i)$ at time t_i , within a prediction horizon, here we are assuming that the set point signal holds one constant value in the optimization window. Hence, the objective now is to transform it into a setting such that the ideal control parameter vector ΔU is obtained which minimize the error between the set point and the predicted output.

If we assume that the vector containing the desired set point is given by

$$R_s^T = \{1 \ 1 \ \dots \ 1\}r(t_i) = \bar{R}_s^T r(t_i) \quad (21)$$

Then the cost function is given as

$$J = (R_s - Y)^T (R_s - Y) + \Delta U^T \bar{R} \Delta U \quad (22)$$

Here the first term in eq. (18) represents the difference between the set point and the real output and \bar{R} in the second term represents the diagonal weighting matrix for desired closed loop performance.

Here in this case, our problem is to find optimal ΔU , which is found by minimizing the following cost function

$$J = (R_s - Fx(t_i))^T (R_s - Fx(t_i)) + 2\Delta U^T \varphi^T (R_s - Fx(t_i)) + \Delta U^T (\varphi^T \varphi + \bar{R}) \Delta U \quad (23)$$

By differentiating J w.r.t ΔU , and equating it to zero, we get the relation for minimum ΔU

$$\Delta U = (\varphi^T \varphi + \bar{R})^{-1} \varphi^T (R_s - Fx(t_i)) \quad (24)$$

With the condition that $(\varphi^T \varphi + \bar{R})^{-1}$ exists. Here using eq (17) we can rewrite equation (20) as

$$\Delta U = (\varphi^T \varphi + \bar{R})^{-1} \varphi^T (\bar{R}_s r(t_i) - Fx(t_i)) \quad (25)$$

Here in our case of point to point ILC the reference signal $r(t_i)$ defines the points to be tracked. When the first iteration starts with the specific points to be tracked the new reference $r(t_i)$ is assigned the desired values. And first iteration uses the receding horizon algorithm to get the weights of desired input which is used by ILC in the second iteration. This way accuracy of ILC is improved and convergence to desired points to be tracked is achieved more quickly.

4. PROPOSED TECHNIQUE

Here a hybrid technique is proposed which uses the benefits of two techniques i.e., receding horizon control and iterative learning control. Here we used the benefits of the receding horizon control which optimize the trajectory of the system every time when it has to reach some set-point. In point to point ILC the output has to follow the specific points, so to reach to these point more quickly receding horizon optimization and iterative learning control both are used. Hence for tracking a point we first use the receding horizon control input and its value adjusts the weights for the next trial of the input for ILC. Benefit of using this technique is that when the specific points during the entire trajectory are changing receding horizon technique provides a fast reaching of these points.

5. USE OF LAGUERRE FUNCTIONS

One drawback of using receding horizontal method is that it needs to optimize a large number of parameters, so it needs a lot of computations and requires more memory to store the parameters. To avoid this condition Laguerre functions can be used to reduce the requirement of number of parameters which are to be optimized on-line, here scaling factor can be employed to control the time scale of predictive control system as applied in [13].

Laguerre functions have been introduced for system identification long ago, as in [14], [15] and has been applied on receding horizon control in [16].

The idea is to use a set of discrete orthogonal functions to represent $u_k(n)$, the future control input signal, in a moving horizon window, here $0 \leq n \leq N_p$. If N represents the number of terms, and $l_i(n)$, where $0 \leq i \leq N$ represents set of Laguerre functions. Then the input can be written in terms of Laguerre functions as

$$u^k(n) \approx \sum_{m=1}^N c_m l_m(n) \quad (26)$$

Which can also be written as

$$u^k(n) \approx L^T(i)\eta \quad (27)$$

Here $L(i) = [l_m(n)l_m(n) \dots l_m(n)]^T$ and η represents the Laguerre coefficients i.e., $\eta = [c_1c_2 \dots c_N]^T$.

When the control trajectory is represented in terms of Laguerre polynomials, the predicted state variable vector is represented as

$$x(t_i + N_p / t_i) = A^{N_p}x(t_i) + \sum_{j=0}^{N_p-1} A^{N_p-j-1}B L^T(N_p)\eta \quad (28)$$

6. POINT TO POINT ILC FORMULATION

The criteria to minimize in the Point to Point ILC in comparison to traditional ILC is to solve the following minimizing quadratic function

$$J_{k+1}(u_{k+1}) = \|e_{k+1}^e\|_Q^2 + \|u_{k+1} - u_k\|_R^2 \quad (29)$$

Here e_{k+1}^e represents the error at intermediate points i.e., $0 \leq t_i \dots \leq t_M$.

Hence the input update law for point to point ILC is given by

$$u_{k+1} = u_k + Le_{k+1}^e \quad (30)$$

Where L represents the learning function.

7. Stable systems

We have used the lifted system representation, and in case of lifted form of ILC based on system model, the way out is to find a feedback control law which stabilizes the system, and then ILC can be applied on them, as mentioned in [17],[18]. The system can be stabilized by using infinite time algebraic riccati equation, and then we use the gain obtained by solving the algebraic riccati equation to form a feedback law, using A,B and C of the system dynamics, Hence we get

$$K = (R + B^T P B)^{-1} B^T P A \quad (31)$$

Here P is obtained by solving the algebraic riccati (ARE) equation given as below

$$A^T P A - A^T P (R + B^T P B)^{-1} B^T P A + Q - P = 0 \quad (32)$$

We know by discrete quadratic regulator theory that all the eigenvalues of $A - BK$ exist within the unit circle.

8. OBSERVER

The above mentioned algorithm is suitable in the cases when we have the information of $x(t)$ i.e., states at all the times t , but in reality, in many of the applications the information of $x(t)$ is not available or not measurable, in these cases we can use observer to estimate the unknown states of the system. Kalman filter is also a good choice for the state estimation of the system.

9. CASE STUDY

We consider a plant having the transfer function

$$G_p(s) = \frac{0.5(s+1)}{s^2-2s+1} \quad (33)$$

And the reference signal is defined as $at t_1 = 3, t_2 = 6, t_3 = 10$ having the values $r(t_1) = 6, r(t_2) = 10, r(t_3) = 3$. The value of L as 0.7.

We applied our proposed algorithm and we get the following results, Note here that the trajectory in practice cannot have so steep points, but here we have shown these steep curves just to elaborate the speed of the tracking.

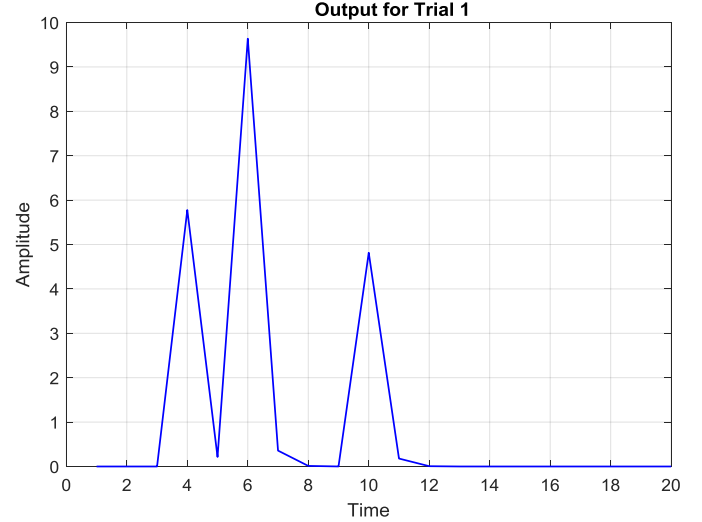


Fig 1 Result of using RHC in first iteration

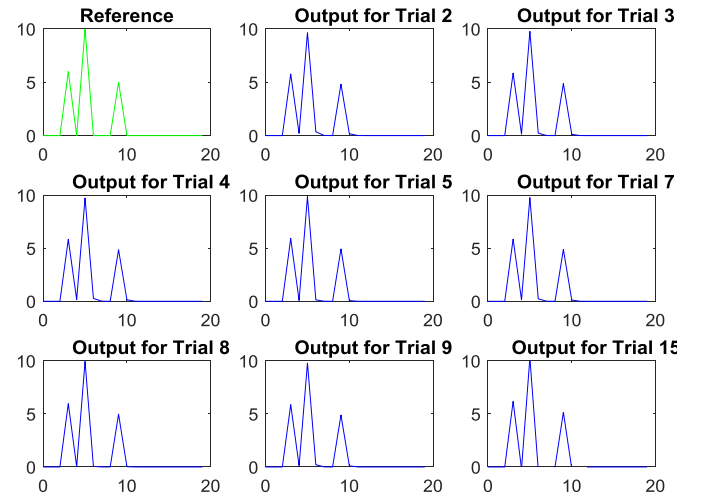


Fig 2. Variation of Output with respect to iterations

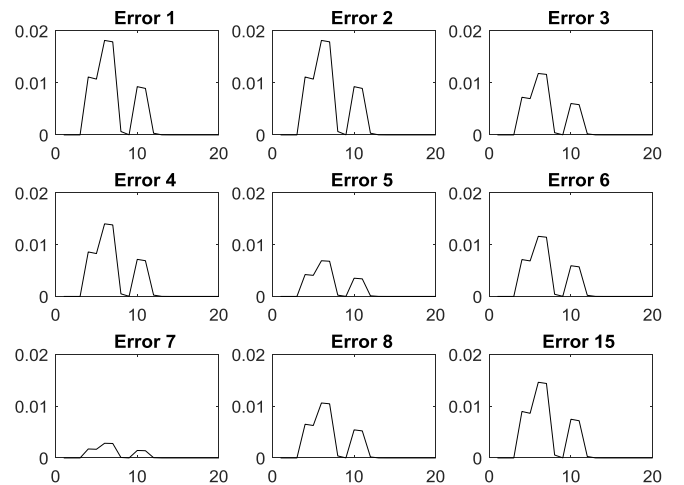


Fig 3. Error evolution with respect to iterations

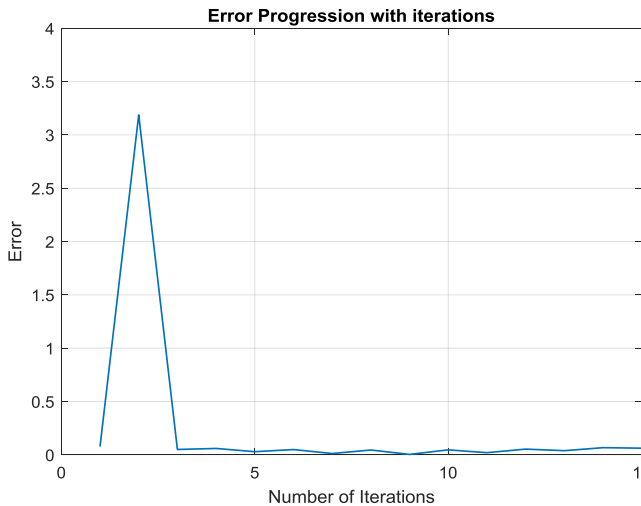


Fig 4. Propagation of error along the iteration axis

From fig.1 we can observe that the output is following the required points and with the use of ILC the error decreases further with the iterations. Fig 2 represents the reference and the output values for consecutive iterations, and shows how the output tracks the points. Fig 3 represents the error evolutions with respect to iterations. Here, we can observe that the error remains within a very small value. Fig 4 represents the propagation of error with respect to the iteration.

10. CONSTRAINTS

Point to point ILC faces a problem of vibrations after the reaching the point to be tracked or between the points to be tracked. This problem has been dealt by different proposed techniques i.e., Input shaping as introduced in [2] for high-acceleration positioning table. This can also be achieved by tracking reference output using normal feedback, whereas ILC is used for improving the input shape to update the parameters. This can also be solved by using ILC both for reference tracking and reducing the vibrations between the points.

Other major constraints to be embedded in point to point ILC framework are the constraints comprising of input, output and system states.

These constraints like Input saturation and amplitude constraints can be dealt efficiently by a barrier method in the condition of single point to point ILC [19], [1]. But it requires a large number of trials to be performed and requires the manual tuning of the parameters, an incorrect choice results in violation of constraints permanently. To address this issue, a successive projections based algorithm is proposed in [20] which incorporates wider range of input energy constraints, actuator amplitude constraints and saturation limits. This method is far simpler, and includes wider class of applicability.

Point to point ILC can start with the design of initial reference as a first step, which remains static over all trials. Or the reference is updated between trials to increase the performance and robustness of a broad class of linear ILC law when applying to the point to point control problem [12].

11. CONCLUSION

A hybrid technique is proposed which takes advantage of receding horizon control and iterative learning control to track the points within the trajectory. Laguerre functions are introduced for the efficient use of receding horizon control, Laguerre functions provide the option of tuning the performance by controlling the number of parameters. In the future this technique will be applied on a robot manipulator and the constraints will be addressed.

12. 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 μ -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.