2011 50th IEEE Conference on Decision and Control and
European Control Conference (CDC-ECC)
Orlando, FL, USA, December 12-15, 2011

Iterative Learning Control for Optimal Multiple-Point Tracking

Tong Duy Son, Dinh Hoa Nguyen, and Hyo-Sung Ahn†

Abstract— **This paper presents a new optimization-based iterative learning control (ILC) framework for multiple-point tracking control. Conventionally, one demand prior to designing ILC algorithms for such problems is to build a reference trajectory that passes through all given points at given times. In this paper, we produce output curves that pass close to the desired points without considering the reference trajectory. Here, the control signals are generated by solving an optimal ILC problem with respect to the points. As such, the whole process becomes simpler; key advantages include significantly decreasing the computational cost and improving performance. Our work is then examined in both continuous and discrete systems.**

I. INTRODUCTION

In control theory, control schemes to achieve outputs that pass through specified terminal points have been divided into two steps: trajectory planning and tracking control. In these schemes, the trajectory planner attempts to generate an optimal reference trajectory from information of the given set of points; the main focus of research in this area pertains to interpolation techniques. On the other hand, the controller—which is designed to track the desired outputs—focuses on the system dynamics. Here, the improved accuracy in trajectory tracking results has led to the development of various control schemes, such as proportional integral derivative (PID) control, feedback control, adaptive control, and iterative learning control (ILC).

ILC is a control methodology for tracking a desired trajectory in repetitive systems, such as those found in applications such as robotics, semiconductors, and chemical processes. The ILC algorithm refines input sequences through the experiences of previous iterations so that the output converges to a reference trajectory trial-to-trial. A number of publications [1]– [4] have shown that the ILC algorithm guarantees the convergence of the output to the desired trajectory in the iteration domain. In ILC research, terminal iterative learning control (TILC) is derived to generate inputs such that outputs track given desired terminal points. In addition, a number of applications showed that the performance of tracking predefined points could be improved using ILC theory [5]–[8]. However, although these works demonstrated the effectiveness of ILC theory in dealing with terminal control problems, they only addressed these issues with one terminal point. Thus, the common approach is to build an ILC algorithm that produces an initial input that only considers the errors of the endpoint from previous iterations. Recently, there has been some research that considers a TILC problem in which the system has multiple desired terminal points [9]–[11]. Specifically, in [9], an ILC framework was developed in which the reference trajectory is updated in the frequency domain between trials; the work in [10] investigates an interpolation technique for iteratively updating the reference trajectory. On the other hand, TILC was developed to directly specify points, rather than to determine a trajectory. In [11], the monotonic convergence of errors at all points can be ensured; however, the performance is dependent on the sampling time.

In most publications, ILC theory follows the direction of tracking a prior identified trajectory. Thus, the ILC law forms the basis for the TILC problem by defining the reference trajectory that goes through the established set of points. However, dividing the TILC problem into trajectory planning and trajectory tracking shows drawbacks under certain circumstances. First, most trajectory planning algorithms face difficulties in generating an optimal reference trajectory. In particular, the existence of a large number of points can lead to a significant increase in the computational analysis and memory requirements. Second, ILC theory [1]– [3] has shown that the system performance and rate of convergence depend on both the system dynamics and the reference trajectory. Consequently, even if an optimal trajectory is chosen, the ILC controller could be unsatisfactory. And the last reason is that the existence of errors in both stages can result in deficient performance as an effect of the indirect method. Therefore, these reasons motivate our study to combine two stages into one ILC controller such that it improves the performance and optimizes the computational cost.

In this paper, we attempt to design a controller subject to both assigned points and a dynamic system capable of tracking multiple points in repetitive systems. Even though this problem has been previously considered in terms of the optimal control for single-input single-output (SISO) systems [12], our goal is to propose an ILC theory for multi-input multi-output (MIMO) systems. The proposed ILC scheme is then applied to investigate the repetitive nature of these systems. Furthermore, the optimal ILC controllers and analyses are shown to consider limitations in actuator demands. Another limitation we are attempting to overcome

T.D. Son and H.-S. Ahn are with the Distributed Control and Autonomous Systems Laboratory, Department of Mechatronics, Gwangju Institute of Science and Technology (GIST), Korea; (Email: hyosung@gist.ac.kr). D.H. Nguyen (Hoa_Dinh_Nguyen@ipc.i.u-tokyo.ac.jp) is with the Department of Information Physics and Computing, Graduate School of Information Science and Technology, The University of Tokyo, Japan.

is the requirement to pass over all given points, which is overly restrictive in many systems since the data is noise contaminated. Moreover, the relationship of the control signal and system performance is also presented as an effect of our algorithm.

The remainder of this paper is organized as follows. In Section II, we provide the problem formulation of TILC. Section III then considers our work with continuous systems, while Section IV presents the problem in discrete systems. Simulation results are given in Section V, and Section VI concludes this work.

II. MULTIPLE-POINT TRACKING WITH ILC

In a multiple terminal points problem, there are specified time instants in system operation $t_1, t_2, \ldots, t_M$, where $0 \leq t_1 < t_2 < \ldots < t_M \leq T$. Let us define the desired outputs at these points as

$$y_d(t_1), y_d(t_2), \ldots, y_d(t_M).$$

The control task is to then construct a control law that drives the outputs through or close to these points. In conventional control schemes, the trajectory planner builds a reference trajectory y_{ref} such that y_{ref} passes the desired points at $t_1, t_2, \ldots, t_M$. Note that the trajectory is usually chosen from an optimal strategy; for example, minimizing the total passing time. Then, from the system model, we design a controller to track the given trajectory. One traditional solution is the use a PID control and feedback control to generate the correction signal $u(t)$.

The intelligent control technique ILC can be applied to repetitive systems that operate over an interval $[0, T]$ to track the reference trajectory y_{ref}. In this case, the learning algorithm utilizes output errors and control inputs from previous iterations to compute updated control inputs as

$$u_{k+1} = T_u u_k + T_e e_k$$

where the error at the k-th iteration e_k is calculated from

$$e_k = y_{ref} - y_k.$$

For the linear time invariant plant $y_k = T_s u_k$, the algorithm satisfies the convergence of error, $\lim_{k\to\infty} e_k = e^*$, if

$$\rho(T_u - T_e T_s) < 1,$$

where $\rho(A)$ is the spectral radius of the matrix A.

As discussed, the ILC theory is typically built from the reference trajectory y_{ref}; in contrast, we propose an optimal ILC approach to work directly with multiple points $y_d(t_1), y_d(t_2), \ldots, y_d(t_M)$. To generate the optimal control signal, we consider a performance index that adopts the errors at multiple points, such that

$$J = \sum_{i=1}^{m} \|e_{k+1}(t_i)\|_{q_i} + \left\|u_{k+1} - u_k\right\|_R + \left\|u_{k+1}\right\|_S$$

where $e_k(t_i)$ is the error at the terminal time instant t_i in the k-th iteration, i.e.,

$$e_k(t_i) = y_d(t_i) - y_k(t_i).$$

In the cost function, we consider the norms of errors at multiple points, the control signal, and its rate of change. It is notable that the cost function approach was previously investigated in a norm optimal ILC [13] for treatment with a desired trajectory rather than the specific data points. By minimizing J, a sequence of optimal control signals in the iteration domain is produced. Moreover, by driving the outputs close to the desired prespecified points, it leads to a trade-off between the control energy and the system performance.

III. OPTIMAL ILC FOR CONTINUOUS SYSTEMS

In this section, we consider an ILC algorithm capable of tracking multiple terminal points for a continuous system. In this case, a linear time invariant system operates on an interval $t \in [0, T]$, such that

$$\begin{aligned} \dot{x}_k(t) &= Ax_k(t) + Bu_k(t) \\ y_k(t) &= Cx_k(t) \end{aligned} \tag{1}$$

where k is the iteration index. The system is a MIMO system that has matrices A, B, and C with appropriate dimensions. In this paper, we assume that the system is both controllable and observable.

The primary control task is then to achieve the desired output of the terminal points through an ILC algorithm trial to trial. From the linear system theory, we can find output of the system at the $i-$th sample time in the k-th iteration as

$$y_k(t_i) = Ce^{At_i}x_k(0) + C\int_0^{t_i} e^{A(t_i - t)}Bu_k(t)dt.$$

As a result, the error is computed as

$$e_k(t_i) = y_d(t_i) - Ce^{At_i}x_k(0) - C\int_0^{t_i} e^{A(t_i - t)}Bu_k(t)dt.$$

Obviously, without loss of generality, it is possible to replace $y_d(t_i)$ with $y_d(t_i) - Ce^{At_i}x_k(0)$; or just assume that $x_k(0) = 0$. Furthermore, the initial state condition is assumed to be identical in all iterations.

By defining

$$p_i(t) = \begin{cases} Ce^{A(t_i - t)}B & \text{if } t \leq t_i \\ 0 & \text{if } t > t_i \end{cases},$$

we can rewrite the terminal point errors at the time instant t_i as

$$e_k(t_i) = y_d(t_i) - \int_0^T p_i(t)u_k(t)dt.$$

Then, the super vector frameworks with respect to the given time instants of outputs and errors are given as

$$\begin{aligned} \mathbf{y}_d &= \begin{bmatrix} y_d^T(t_1) & y_d^T(t_2) & \ldots & y_d^T(t_M) \end{bmatrix}^T \\ \mathbf{e}_k &= \begin{bmatrix} e_k^T(t_1) & e_k^T(t_2) & \ldots & e_k^T(t_M) \end{bmatrix}^T. \end{aligned}$$

Similarly,

$$\mathbf{P}(t) = \begin{bmatrix} p_1^T(t) & p_2^T(t) & \ldots & p_M^T(t) \end{bmatrix}^T.$$

And the multiple errors in the super vector forms are:

$$\mathbf{e}_{k+1} = \mathbf{y}_d - \int_0^T \mathbf{P}(t) u_{k+1}(t) dt.$$

Next, we consider the following performance index:

$$J = \sum_{i=1}^{M} e_{k+1}^T(t_i)\, \mathbf{q}_i e_{k+1}(t_i) + \int_0^T u_{k+1}^T(t)\, \mathbf{S} u_{k+1}(t)\, dt + \int_0^T \left(u_{k+1}(t) - u_k(t)\right)^T \mathbf{R} \left(u_{k+1}(t) - u_k(t)\right) dt \tag{2}$$

where $\mathbf{R}, \mathbf{S}$, and $\mathbf{q}_i$ are diagonal positive definite matrices with $\mathbf{R}, \mathbf{S} = (r\mathbf{I}, s\mathbf{I})$ and $\mathbf{q}_i$ is the weighting matrix for the error at the time instant t_i.

We can then rewrite (2) to incorporate the vector form of multiple errors as

$$J = \mathbf{e}_{k+1}^T \mathbf{Q} \mathbf{e}_{k+1} + \int_0^T u_{k+1}^T(t)\, \mathbf{S} u_{k+1}(t)\, dt + \int_0^T \left(u_{k+1}(t) - u_k(t)\right)^T \mathbf{R} \left(u_{k+1}(t) - u_k(t)\right) dt$$

where $\mathbf{Q}$ is a symmetric positive definite weight matrix.

A. ILC Controller

To obtain the optimal input at the $(k+1)$-th iteration, differentiating J with respect to $u_{k+1}(t) \in L^2[0,T]$ then setting this derivative to vanish yields

$$-\mathbf{P}^T(t)\, \mathbf{Q} \left(\mathbf{y}_d - \int_0^T \mathbf{P}(t)\, u_{k+1}(t)\, dt\right) + (\mathbf{R} + \mathbf{S})\, u_{k+1}(t) = \mathbf{R} u_k(t).$$

Here, we introduce a new variable z_k such that

$$u_k(t) = \mathbf{P}^T(t) z_k \tag{3}$$

with respect to the control signal at the k-th iteration; we can then rewrite (3) as

$$-\mathbf{P}^T(t) \mathbf{Q} \left(\mathbf{y}_d - \int_0^T \mathbf{P}(t)\, \mathbf{P}^T(t)\, z_{k+1} dt\right) + (\mathbf{R} + \mathbf{S})\, \mathbf{P}^T(t)\, z_{k+1} = \mathbf{R} \mathbf{P}^T(t)\, z_k.$$

With the chosen $\mathbf{R}, \mathbf{S} = (r\mathbf{I}, s\mathbf{I})$, the following equation is derived:

$$\left((r+s)\mathbf{I} + \mathbf{Q} \int_0^T \mathbf{P}(t)\, \mathbf{P}^T(t)\, dt\right) z_{k+1} = r z_k + \mathbf{Q} \mathbf{y}_d. \tag{4}$$

The new algorithm is built on the basis of vector z_k, and the control inputs follow from the sequence $\{z_k\}$ by trials. This derivation significantly decreases the computational cost in the ILC algorithm since the dimensions of system matrices are optimized into the number of desired terminal times.

In the next section, we will show the convergence property of the z_k updating equation. Accordingly, the convergence properties of the control input and errors are evaluated.

B. Convergence

Given

$$\mathbf{W} = \int_0^T \mathbf{P}(t) \mathbf{P}^T(t) dt,$$

since different $p_i(t)$ vanish at different times, the set of functions $p_i(t)$ with $i = 1, 2, \ldots, M$ are linearly independent. Therefore, $\mathbf{W}$ is a symmetric positive definite matrix. Thus, (4) can be rewritten as

$$((r+s)\, \mathbf{I} + \mathbf{Q}\mathbf{W})\, z_{k+1} = (r\mathbf{I} + \mathbf{Q}\mathbf{W})\, z_k + \mathbf{Q}\mathbf{e}_k, \tag{5}$$

where $\mathbf{e}_k = \mathbf{y}_d - \mathbf{W} z_k$.

Lemma 3.1: The iterative learning equation (5) is convergent if $\mathbf{Q}, \mathbf{R}$, and $\mathbf{S}$ are chosen such that

$$\rho\left(((r+s)\, \mathbf{I} + \mathbf{Q}\mathbf{W})^{-1} r\right) < 1. \tag{6}$$

Proof: First, we prove the non-singularity of $[(r+s)\mathbf{I} + \mathbf{Q}\mathbf{W}]$. It can be examined easily that the following equalities always hold with appropriate dimensions of K, L, X, Y and that K, L is invertible.

$$\begin{bmatrix} K & 0 \\ 0 & L + YK^{-1}X \end{bmatrix} = \begin{bmatrix} I & 0 \\ -YK^{-1} & I \end{bmatrix} \begin{bmatrix} K & -X \\ Y & L \end{bmatrix} \begin{bmatrix} I & K^{-1}X \\ 0 & I \end{bmatrix},$$

$$\begin{bmatrix} K + XL^{-1}Y & 0 \\ 0 & L \end{bmatrix} = \begin{bmatrix} I & XL^{-1} \\ 0 & I \end{bmatrix} \begin{bmatrix} K & -X \\ Y & L \end{bmatrix} \begin{bmatrix} I & 0 \\ -L^{-1}Y & I \end{bmatrix}.$$

Then, using the product property for determining matrices in the above equalities, we obtain

$$\det \begin{bmatrix} K & -X \\ Y & L \end{bmatrix} = \det K \det\left(L + YK^{-1}X\right) = \det L \det\left(K + XL^{-1}Y\right)$$

Therefore, we see that the non-singularity of $L + YK^{-1}X$ is equivalent to the non-singularity of $K + XL^{-1}Y$. Now, substituting $K = (r+s)I, X = Y = W^{\frac{1}{2}}$, and $L = Q^{-1}$, and noting that $(r+s)I + W^{\frac{1}{2}} Q W^{\frac{1}{2}}$ is nonsingular since $W^{\frac{1}{2}} Q W^{\frac{1}{2}} > 0$ and $(r+s)I > 0$, we have

$$\begin{aligned} & (r+s)I + W^{\frac{1}{2}} Q W^{\frac{1}{2}} \text{ is nonsingular} \\ \Leftrightarrow\ & Q^{-1} + W(r+s)^{-1} I \text{ is nonsingular} \\ \Leftrightarrow\ & (r+s)I + QW \text{ is nonsingular.} \end{aligned}$$

Consequently, the sequence of z_k is obtained from

$$z_{k+1} = \mathbf{T}_z z_k + \mathbf{T}_e \mathbf{e}_k$$

with $\mathbf{T}_z$ and $\mathbf{T}_e$ are defined as

$$\begin{aligned} \mathbf{T}_z &= ((r+s)\, \mathbf{I} + \mathbf{Q}\mathbf{W})^{-1} (r\mathbf{I} + \mathbf{Q}\mathbf{W}), \\ \mathbf{T}_e &= ((r+s)\, \mathbf{I} + \mathbf{Q}\mathbf{W})^{-1} \mathbf{Q}. \end{aligned}$$

Thus, this results in the condition for convergence of the iterative learning algorithm as

$$\rho\left(\mathbf{T}_z - \mathbf{T}_e \mathbf{W}\right) < 1,$$

where $\mathbf{T}_z - \mathbf{T}_e \mathbf{W} = ((r+s)\, \mathbf{I} + \mathbf{Q}\mathbf{W})^{-1} r$. ■

Moreover, note that if $\mathbf{Q} = q\mathbf{I}$, where q is real positive, the algorithm achieves monotonic convergence. The reason is that now $((r+s)\mathbf{I} + \mathbf{QW})^{-1} r$ is a symmetric positive definite matrix which has the largest singular value equals its spectral radius, and $\rho\left(((r+s)\mathbf{I} + \mathbf{QW})^{-1} r\right) < 1$ [11]. As such, from the result of Lemma (3.1), we obtain the following convergence property of the control input.

Theorem 3.1: For the linear continuous system (1), the following ILC system

$$\begin{aligned} u_k(t) &= \mathbf{P}^T(t) z_k \\ z_{k+1} &= \mathbf{T}_z z_k + \mathbf{T}_e \mathbf{e}_k \end{aligned}$$

drives the system outputs close to the desired terminal points. Moreover, the control input converges to a fixed point $u_\infty(t)$ as

$$u_\infty(t) = \mathbf{P}^T(t)(s\mathbf{I} + \mathbf{QW})^{-1}\mathbf{Q}\mathbf{y}_d.$$

Proof: First, we define the L_2-norm of the control signal as

$$\|u(t)\|^2 = \int_0^T u^T(t)u(t)dt.$$

Then,

$$\begin{aligned} \|u_k(t)\|^2 &= \int_0^T z_k^T \mathbf{P}(t)\mathbf{P}^T(t) z_k dt \\ &= z_k^T \mathbf{W} z_k. \end{aligned}$$

Since $\mathbf{W}$ is a positive definite matrix, $\mathbf{W} = \mathbf{V}^T\mathbf{V}$, in which $\mathbf{V}$ has independent columns, leads to $z_k^T \mathbf{W} z_k = \|\mathbf{V} z_k\|^2$. Therefore,

$$\|u_k(t)\| \le \|\mathbf{V}\| \|z_k\|.$$

Consequently, the convergence of the control signal is guaranteed from the convergence of the z_k learning algorithm, as in Lemma (3.1). In this case, the converged vector of z_k is achieved from (5),

$$((r+s)\mathbf{I} + \mathbf{QW}) z_\infty = r z_\infty + \mathbf{Q}\mathbf{y}_d,$$

or equivalently,

$$z_\infty = (s\mathbf{I} + \mathbf{QW})^{-1}\mathbf{Q}\mathbf{y}_d.$$

Hence, the converged input is

$$u_\infty(t) = \mathbf{P}^T(t)(s\mathbf{I} + \mathbf{QW})^{-1}\mathbf{Q}\mathbf{y}_d.$$

■

C. Control Performance

The performance of the controller depends on the steady state value of error $\mathbf{e}_\infty$, such that

$$\begin{aligned} \mathbf{e}_\infty &= \mathbf{y}_d - \int_0^T \mathbf{P}(t) u_\infty(t) dt \\ &= \mathbf{y}_d - \mathbf{W}(s\mathbf{I} + \mathbf{QW})^{-1}\mathbf{Q}\mathbf{y}_d. \end{aligned} \quad (7)$$

From (7), we can conclude that the steady state error does not depend on the parameter $\mathbf{R}$; i.e., the performance of the controller and the rate of convergence are unrelated. Moreover, the weighting matrices $\mathbf{Q}$ and $\mathbf{S}$ determine the performance of the tracking technique, where the entries of the matrix $\mathbf{Q}$ determine how the different performance the points are achieved; in practical applications, there is always the case in which the importances of particular points are different. Additionally, from (7), the smallest possible error at all terminal points $\mathbf{e}_\infty = 0$ requires that $s = 0$, with positive definite matrices $\mathbf{Q}$ and $\mathbf{W}$.

IV. OPTIMAL ILC FOR DISCRETE-TIME SYSTEMS

In this section, we analyze the point tracking control problem in discrete-time systems. Our motivation is the fact that many practical implementations will result in a discrete-time ILC algorithm. Let us first consider the linear discrete-time invariant system

$$\begin{aligned} x_k(t+1) &= Ax_k(t) + Bu_k(t) \\ y_k(t) &= Cx_k(t) \end{aligned} \quad (8)$$

where $x_k(t) \in \mathbb{R}^p$, $u_k(t) \in \mathbb{R}^m$, and $y_k(t) \in \mathbb{R}^n$, and k is the iteration index. In addition, the system operates on a time interval $t = 0, 1, 2, \ldots, N-1$, and matrices A, B, and C are system matrices with appropriate dimensions.

In the k-th iteration, the output of the system at the i-th sample time is calculated as

$$y_k(t_i) = CA^{t_i} x_k(0) + C\sum_{j=0}^{t_i-1} A^{t_i-j-1} B u_k(j).$$

Here, if we assume $x_k(0) = 0$, the errors are computed as

$$e_k(t_i) = y_d(t_i) - C\sum_{j=0}^{t_i-1} A^{t_i-j-1} B u_k(j).$$

Then, formulating the N-sample sequence of inputs in a super-vector framework:

$$\mathbf{u}_k = \begin{bmatrix} u_k^T(0) & u_k^T(1) & \ldots & u_k^T(N-1) \end{bmatrix}^T,$$

and by introducing $g_i(t)$

$$g_i(t) = \begin{cases} CA^{t_i-t-1}B & \text{if } t < t_i \\ 0 & \text{if } t \ge t_i \end{cases},$$

the output at the i-th time instant is expressed as

$$\begin{aligned} y_k(t_i) &= \sum_{t=0}^{N-1} g_i(t) u_k(t) \\ &= \mathbf{g}_i^T \mathbf{u}_k \end{aligned}$$

where $\mathbf{g}_i$ is defined by

$$\mathbf{g}_i = \begin{bmatrix} g_i(0) & g_i(1) & \ldots & g_i(N-1) \end{bmatrix}^T.$$

As a result, the cost function for the problem of tracking multiple terminal points $t_1, t_2, ..., t_M$ in the discrete time model is given as

$$\begin{aligned} J = &\sum_{i=1}^{M} \left(y_d(t_i) - \mathbf{g}_i^T \mathbf{u}_{k+1}\right)^T \mathbf{q}_i \left(y_d(t_i) - \mathbf{g}_i^T \mathbf{u}_{k+1}\right) \\ &+ \mathbf{u}_{k+1}^T \mathbf{S} \mathbf{u}_{k+1} + \left(\mathbf{u}_{k+1} - \mathbf{u}_k\right)^T \mathbf{R} \left(\mathbf{u}_{k+1} - \mathbf{u}_k\right) \end{aligned} \quad (9)$$

where $\mathbf{R}, \mathbf{S} = (r\mathbf{I}, s\mathbf{I})$, $\mathbf{q}_i$ are positive definite diagonal matrices.

Similar to the previous section, we define

$$\begin{aligned} \mathbf{y}_d &= \left[\begin{array}{cccc} y_d^T(t_1) & y_d^T(t_2) & \ldots & y_d^T(t_M) \end{array}\right]^T \\ \mathbf{G} &= \left[\begin{array}{cccc} \mathbf{g}_1^T & \mathbf{g}_2^T & \ldots & \mathbf{g}_M^T \end{array}\right]^T. \end{aligned}$$

Then, the cost function (9) can be rewritten as

$$\begin{aligned} J &= \left(\mathbf{y}_d - \mathbf{G}\mathbf{u}_{k+1}\right)^T \mathbf{Q} \left(\mathbf{y}_d - \mathbf{G}\mathbf{u}_{k+1}\right) \\ &\quad + \mathbf{u}_{k+1}^T \mathbf{S} \mathbf{u}_{k+1} + \left(\mathbf{u}_{k+1} - \mathbf{u}_k\right)^T \mathbf{R} \left(\mathbf{u}_{k+1} - \mathbf{u}_k\right). \end{aligned} \tag{10}$$

Note that the controller in the $(k+1)$-th trial is achieved from the required stationary condition $\delta J/\delta \mathbf{u}_{k+1} = 0$, or

$$-\mathbf{G}^T\mathbf{Q}\left(\mathbf{y}_d - \mathbf{G}\mathbf{u}_{k+1}\right) + \mathbf{R}\left(\mathbf{u}_{k+1} - \mathbf{u}_k\right) + \mathbf{S}\mathbf{u}_{k+1} = 0 \tag{11}$$

Setting

$$\mathbf{u}_k = \mathbf{G}^T\mathbf{z}_k,$$

the equation (11) is derived as

$$-\mathbf{Q}\left(\mathbf{y}_d - \mathbf{G}\mathbf{G}^T\mathbf{z}_{k+1}\right) + r\left(\mathbf{z}_{k+1} - \mathbf{z}_k\right) + s\mathbf{z}_{k+1} = 0. \tag{12}$$

Hence, (12) is an iterative learning algorithm, i.e.,

$$\left((r+s)\mathbf{I} + \mathbf{Q}\mathbf{W}_d\right)\mathbf{z}_{k+1} = \left(r\mathbf{I} + \mathbf{Q}\mathbf{W}_d\right)\mathbf{z}_k + \mathbf{Q}\mathbf{e}_k, \tag{13}$$

where $\mathbf{W}_d = \mathbf{G}\mathbf{G}^T$ is a symmetric positive definite matrix. From Lemma (3.1), matrix $\left((r+s)\mathbf{I} + \mathbf{Q}\mathbf{W}_d\right)$ is positive definite; therefore, by defining

$$\begin{aligned} \mathbf{L}_z &= \left((r+s)\mathbf{I} + \mathbf{Q}\mathbf{W}_d\right)^{-1}\left(r\mathbf{I} + \mathbf{Q}\mathbf{W}_d\right), \\ \mathbf{L}_e &= \left((r+s)\mathbf{I} + \mathbf{Q}\mathbf{W}_d\right)^{-1}\mathbf{Q}, \end{aligned}$$

and from (13), leads to the following theorem regarding the ILC control algorithm for discrete-time systems.

Theorem 4.1: For the linear discrete-time system (8), the ILC system

$$\begin{aligned} \mathbf{u}_k &= \mathbf{G}^T\mathbf{z}_k \\ \mathbf{z}_{k+1} &= \mathbf{L}_z\mathbf{z}_k + \mathbf{L}_e\mathbf{e}_k \end{aligned}$$

drives the system outputs close to the desired terminal points. Moreover, the control input converges to a fixed point $\mathbf{u}_\infty$ as

$$\mathbf{u}_\infty = \mathbf{G}^T(s\mathbf{I} + \mathbf{Q}\mathbf{W}_d)^{-1}\mathbf{Q}\mathbf{y}_d$$

and the error $\mathbf{e}_\infty$ is defined as

$$\mathbf{e}_\infty = \mathbf{y}_d - \mathbf{W}_d(s\mathbf{I} + \mathbf{Q}\mathbf{W}_d)^{-1}\mathbf{Q}\mathbf{y}_d.$$

Proof: The results of Theorem (4.1) are obtained in the same manner as for Lemma (3.1) and Theorem (3.1). ■

In the case of a discrete time system, we can more clearly see a significant decrease of the computational analyses. In our learning algorithm, vector $z_k \in \mathbb{R}^M$, and $\mathbf{L}_z$, $\mathbf{L}_e$ are $mM \times mM$ matrices where M is the number of terminal points. In comparison, the conventional ILC algorithm updates the input with the system matrix $mN \times mN$. As the length of iteration increases $(N > 1000)$, which is common in many applications such as robotics with a high sampling rate, the requirement of memory and time dramatically increases.

V. NUMERICAL EXAMPLE

In this section, we present an example of tracking multiple points with a linear continuous system model. The simulation illustrates the convergences of error and inputs under our proposed ILC approach. Accordingly, based on suitable chosen weighting matrices, the ILC algorithm produces well-behaved output curves that go through, or very close to, desired multiple terminal points after some iterations. The results are then compared to different weighting matrices to demonstrate the trade-off between the error and energy of the control signal.

Here, the continuous system is chosen as

$$\begin{aligned} \dot{x} &= \begin{pmatrix} 0 & 1 & 0 \\ 0 & 0 & 1 \\ -0.2 & -0.3 & -1 \end{pmatrix} x + \begin{pmatrix} 0 \\ 0 \\ 1 \end{pmatrix} u \\ y &= \begin{pmatrix} 0 & 0 & 0.1 \end{pmatrix} x \end{aligned}$$

which operates on interval $t \in [0, 1]$. We select 10 points in the interval as desired points.

For the first case, weighting matrices are $\mathbf{Q} = 5\mathbf{I}$, $\mathbf{R} = 5.10^{-3}\mathbf{I}$, and $\mathbf{S} = 10^{-3}\mathbf{I}$. In Fig. 1, the results show the fast convergence of control input signal and error through iterations. Hence, we could achieve a very good performance without creating a trajectory. Fig. 2 contains output curves that are generated from different iterations. It can be seen that after 20 iterations, the output curve passes almost exactly through all terminal points.

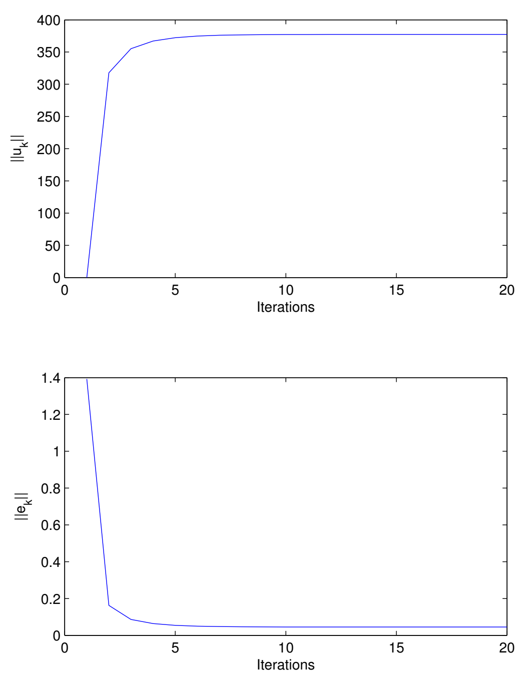

Fig. 1. Convergence of input and error sequences.

Next, to show the effect of parameters, we change the weighting matrix $\mathbf{Q}$ to $\mathbf{Q} = \mathbf{I}$. By comparing the output curves obtained in Figs. 2 and 3, we can see the difference in system performance. In the same iteration, the errors at the terminal points are larger than the ones obtained from $\mathbf{Q} = 5\mathbf{I}$. However, the control signal expends less energy; in this example, the energy of the control signal at the 20th iteration in the two cases is calculated as 380 and 300, respectively.

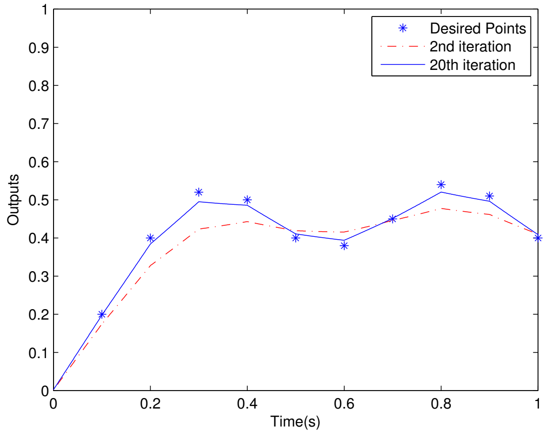

Fig. 2. Output curves with $\mathbf{Q} = 5\mathbf{I}$.

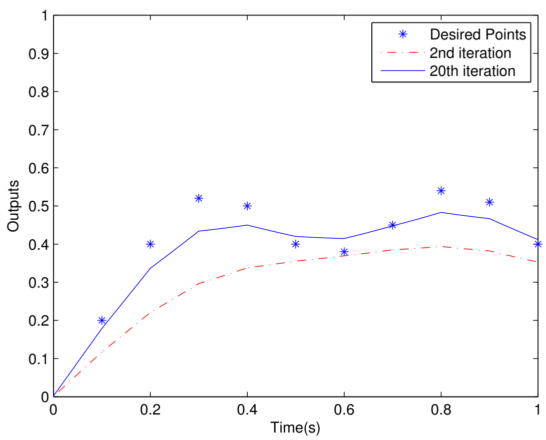

Fig. 3. Output curves with $\mathbf{Q} = \mathbf{I}$.

VI. CONCLUSION

The concept of learning through the experience of ILC to track a desired trajectory has been extensively analyzed in the area of control. However, when there is a mass data point, these ILC approaches have trouble in generating an optimal trajectory, performance, and rate of convergence. Moreover, most ILC algorithms formulate system models in a lift-system representation; thus, the computational cost and time increases whenever the length of the operation time increases. Our approach overcomes these drawbacks by utilizing only the essential information of data points without building the desired trajectory. In this paper, we have shown that the ILC approach that investigates critical points can successfully obtain the convergence of error and control inputs. By manipulating these parameters, a very good performance is achieved.

This paper makes two key contributions to this research field. The first is to present an analysis of the optimal tracking of multiple points problem based on ILC theory. The results improve upon those obtained by a traditional ILC, being significantly more direct and simple. The second contribution relates to a new class of application: path scheduling. For example, when we design an optimal path for an autonomous vehicle, we may have to impose restrictions on particular points and control signals, which may require learning to deliver a suitable path. The proposed ILC theory is appropriate for use in this case. Future work will extend the theory to more generic scenarios where we consider the path scheduling problem for multiple vehicles in a particular context such as conflict avoidance.

VII. ACKNOWLEDGMENTS

This research was supported by the MKE (The Ministry of Knowledge Economy), Korea, under the ITRC (Information Technology Research Center) support program supervised by the NIPA (National IT Industry Promotion Agency) (NIPA-2011-C1090 -1131- 0006).

REFERENCES

[1] H.S. Ahn, Y.Q. 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, 2007, pp. 1099–1121.

[2] D.A. Bristow, M. Tharayil, and A.G. Alleyne, “A survey of iterative learning control: A learning-based method for high-performance tracking control”, *IEEE Control Systems Magazine*, vol. 26, no. 3, 2006, pp. 96–114.

[3] K.L. Moore, “Iterative Learning Control: An Expository Overview”, *Applied and Computational Control, Signals, and Circuits*, vol. 1, no. 1, 1999, pp. 151–214.

[4] R.W. Longman, “Iterative learning control and repetitive control for engineering practice”, *Automatica*, vol. 73, no. 10, 2000 pp. 930–954.

[5] J.X. Xu, Y. Chen, L.Tong Heng, and S. Yamamoto, “Terminal iterative learning control with an application to RTPCVD thickness control”, *Automatica*, vol. 35, no. 9, 1999, pp. 1535–1542.

[6] G. Gauthier and B. Boulet, “Terminal iterative learning control design with singular value decomposition decoupling for thermoforming ovens”, *in Proc. American Control Conf.*, 2009, pp. 1640–1645.

[7] J.X. Xu and D. Huang, “Initial state iterative learning for final state control in motion systems”, *Automatica*, vol. 44, no. 12, 2008, pp. 3162–3169.

[8] S. Arimoto, M. Sekimoto, and S. Kawamura, “Iterative Learning of Specified Motions in Task-Space for Redundant Multi-Joint Hand-Arm Robots”, *in IEEE International Conf. on Robotics and Automation*, 2007, pp. 2867–2873.

[9] C.T. Freeman, Z. Cai, E. Rogers, and P.L. Lewin, “Iterative Learning Control for Multiple Point-to-Point Tracking Application”, *IEEE Transactions on Control Systems Technology*, 2010.

[10] T.D. Son, D.H. Nguyen and H.S. Ahn, “An Interpolation Method for Multiple Terminal Iterative Learning Control”, *in Proc. IEEE Multi-Conference on Systems and Control Conf.*, 2011, submitted.

[11] T.D. Son, and H.S. Ahn, “Terminal Iterative Learning Control with Multiple Pass Points”, *in Proc. American Control Conf.*, 2011.

[12] Shan Sun, M. Egerstedt, C.F.Martin, “Control Theoretic Smoothing Spline”, *IEEE Transactions on Automatic Control*, vol. 45, no. 12, 2000, pp. 2271–2279.

[13] K.L. Barton and A.G. Alleyne, “A Norm Optimal Approach to Time-Varying ILC With Application to a Multi-Axis Robotic Testbed”, *IEEE Transactions on Control Systems Technology*, vol. 19, no. 1, 2011, pp. 166–180.