Journal of the Korean Society of Marine Engineering, Vol. 38, No. 10 pp. 1281~1286, 2014
J. Korean Soc. of Marine Engineering (JKOSME)
http://dx.doi.org/10.5916/jkosme.2014.38.10.1281

ISSN 2234-7925 (Print)
ISSN 2234-8352 (Online)
Original Paper

Improvement of trajectory tracking control performance by using ILC

Dang-Khanh Le[1] · Taek-Kun Nam†

(Received September 2, 2014 ; Revised October 28, 2014 ; Accepted November 20, 2014)

Abstract: This paper presents an iterative learning control (ILC) approach for tracking problems with specified data points that are desired points at certain time instants. To design ILC systems for such problems, unlike traditional ILC approaches, an algorithm which updates not only the control signal but also the reference trajectory at each trial will be developed. The relationship between the reference trajectory and ILC control in tracking problems where there are specified data points through which the system should pass is investigated as the rate of convergence. In traditional ILC, the desired data is stored in a tracking profile file. Due to the huge size of the data file containing the target points, it is important to reduce the computational cost. Finally, simulation results of the presented technique are mentioned and compared to other related works to confirm the effectiveness of proposed scheme.

Keywords: Iterative learning control, Specified data points, Trajectory update, Convergence rate, Tracking performance.

1. Introduction

Control schemes for tracking problems can be divided into two steps: trajectory planning and tracking control. In these schemes, the trajectory planner attempts to generate an optimal reference trajectory from a given set of points in the motion profile (i.e., specified data points that are desired points at some specified time instants). Then the controller, which is designed to track the reference trajectory, focuses on the system dynamics to generate a sequence of inputs. To improve the accuracy in trajectory tracking, various control schemes such as feedback control, robust control, and iterative learning control have been developed.

Iterative learning control (ILC) is a control methodology for tracking a reference trajectory in repetitive systems, that are found in applications such as robotics, semiconductors, and chemical processes. The prime strategy of ILC algorithms is to refine the input sequence from one trial in order to improve the performance of the system on the next trial. A number of surveys have effectively covered the novel ideas and development of ILC methodology and we refer the reader to these references for more information on the basic ideas of ILC **[1]-[6]**.

In this paper, we investigate the relationship between trajectory planning and ILC control in tracking problems where there are specified data points through which the system should pass, while the trajectory between these points is less important. The desired data is stored in a tracking profile file. Due to the huge size of the data file containing the target points, it is important to reduce the computational cost. These problems lead us to study ILC controller with specified data points. As mentioned, most ILC algorithms focus on tracking a fixed reference trajectory. There have been fewer studies on the connection between the trajectory planning and ILC control algorithms, although there is increasing interest in this topic. In terminal ILC, which considers only one end point as a desired output point, Xu and Huang **[7]** and Gauthier and Boulet **[8]** have addressed the relationship between a point in the motion profile and the control update. Recently, there has been a series of works that consider the tracking problem with multiple specified data points **[9][10]**, with a particular application to stroke rehabilitation to assist point-to-point movements of patients **[11]**. These works have shown that, the performance of multiple points tracking problems could be improved based on ILC techniques. Specifically, in Freeman et al. **[9]** the authors developed an ILC framework in which the reference trajectory is up-dated in the frequency domain, and in Freeman and Tan **[10]** a fixed reference trajectory with multiple pass points has been studied. These techniques were developed in discrete-time settings and were not interpolation-based. Tracking of nonidentical trajectories also has

† Corresponding Author (ORCID: http://orcid.org/0000-0003-0432-3504): Division of Marine Engineering, Mokpo National Maritime University 530-729, Korea, E-mail: tknam@mmu.ac.kr, Tel: 061-240-7225
1 Department of Marine Engineering, Mokpo National Maritime University, E-mail: ledangkhanhmtb@gmail.com,

been studied in Xu et al. **[12]**, but without a reference update in the iteration domain. Similar to Freeman et al. **[9]**, in this paper, tracking problems with multiple pass points at specified data points is also considered. An ILC approach for improving tracking performance is proposed. In proposed approach, a reference trajectory update method, interpolation technique in the time domain is applied, while in Freeman et al. **[9]** and Freeman et al. **[13][14]**, they use discrete Fourier transform (DFT) technique in the frequency domain.

The remainder of this paper is organized as follows. In Section 2, we provide the formulation of tracking problem, and summarize the conventional approach to solving the problem. Section 3 considers the ILC algorithm where the reference trajectory is updated iteratively. Simulation results are given in Section 4, and Section 5 concludes this work.

2. Tracking problem with specified data points

Consider a linear, continuous-time system that operates on an interval t ∈ [0, T] given as **Equation (1)**.

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

where k is the iteration index. The system is a MIMO system that has state $x_k(t) \in R^{n_x}$, control signal $u_k(t) \in R^{n_u}$ and output $y_k(t) \in R^{n_y}$. The matrices A, B, and C have appropriate dimensions. In the tracking problem the system trajectory must pass through, or close to, a limited number of specified points, at prescribed times. The specified time instants in the system operation are defined from the set $T^s = \{t_1^s, t_2^s, ..., t_M^s\}$, where $0 \le t_1^s < t_2^s < ... < t_M^s \le T$, and the desired outputs at these points are given by $y_d(t_1^s), y_d(t_2^s)...y_d(t_M^s)$. As a result, the control task is to construct a control law that drives the system output to go through the desired outputs as closely as possible:

$$y_k(t_i^s) = y_d(t_i^s), i = 1, 2, ..., M. \tag{2}$$

In typical tracking schemes, a trajectory planner can generate a reference trajectory *r(t)* such that it passes the desired points at $t_1^s, t_2^s, ..., t_M^s$. Then, a tracking controller, such as an ILC controller, can be designed to ensure the tracking the generated reference trajectory with an improved performance. In the trajectory planning stage, one of the common techniques to find a reference trajectory from the given points is "interpolating spline", which is a tool in numerical analysis. The generated function might be a spline if it exactly interpolates the given points **[15]**. However, the reference trajectory may not be feasible for the system since it may vary too fast to allow the dynamics to follow the trajectory or it may vary too slowly, which gives a conservative control performance. Thus, a control theoretic interpolating spline was introduced **[16]**. The suggested approach addresses these problems by considering auxiliary system dynamics, rather than the spline functions.

In a typical problem, after a trajectory having been generated, a tracking control is applied. To illustrate, consider the case where a linear discrete-time system operates repetitively, from the same initial condition at the start of each trial and suppose an ILC controller is applied to drive the system output to go as close as possible to the reference trajectory. In ILC, the learning algorithm utilizes output errors and control inputs from the previous iterations to compute an updated control signal **[1][4]**. To describe this, let the discrete-time system be described in the lifted system framework as

$$y_k = Pu_k, \tag{3}$$

where the input signal, output signal, and reference trajectory are written in the super-vector forms as

$$\begin{aligned} u_k &= [u_k^T(0), u_k^T(1), ..., u_k^T(N-1)]^T, \\ y_k &= [y_k^T(1), y_k^T(2), ..., y_k^T(N)]^T, \\ y_d &= [y_d^T(1), y_d^T(2), ..., y_d^T(N)]^T \end{aligned} \tag{4}$$

and the system matrix $P = [p_{ij}]$ is a Markov matrix which is a lower triangular Toeplitz matrix. For this system we then adopt the following ILC algorithm:

$$u_{k+1} = T_u u_k + T_e(y_d - y_k). \tag{5}$$

It is well known that, this ILC algorithm is convergent if $\rho(T_u - T_e P) < 1$, where $\rho(A)$ is the spectral radius of the matrix A. The algorithm guarantees a monotonic convergence if the condition $\bar{\rho}(T_u - T_e P) < 1$, where $\bar{\rho}(A)$ is the largest singular value of the matrix A, is satisfied **[1][4]**. The track-ing problem we are interested in, then, is to force the continuous-time system of **Equation (1)** or the discrete-time system of **Equation (3)** to pass through (or close) to the desired points of **Equation (2)** by using an algorithm such as **Equation (5)**. In the next section an approach is proposed to solve this problem.

3. Reference trajectory update-based ILC

In this section, we propose a ILC scheme updates not only the control signal but also the reference trajectory at each trial. Firstly, the connection between trajectory planning and the ILC controller is analyzed, and then this relationship is investigated for improving the convergence rate.

3.1 Reference trajectory update law

The addressed problem can be formulated as finding a new reference trajectory at every iteration such that, it brings a better performance than that of traditional ILC with a fixed reference trajectory throughout iterations. In other words, the reference trajectory is planned iteratively; but the new reference trajectory $r_k(t), t \in [0, T]$, always satisfies the condition to have desired values at specified data points:

$$r_k(t_i^s) = y_d(t_i^s), \tag{6}$$

where $i = 1, 2, ..., M$. Here, a class or sequence of reference trajectories that always go through the specified outputs (6) needs to define. Consider any given initial reference trajectory $r_0(t)$ satisfying $r_0(t_i^s) = y_d(t_i^s)$. It is obvious that $r_k(t)$ is obtained as follows:

$$r_k(t) = r_0(t) + h(t) f(t), \ k \geq 1, \tag{7}$$

where, $h(t) = (t - t_1^s)(t - t_2^s)...(t - t_M^s)$, $f(t)$ is an arbitrary continuous function, and $r_k(t)$ satisfies the condition to have desired values at specified data points. Plainly, since $h(t_i^s) = 0$ at $i = 1, 2, ..., M$, the trajectory $r_k(t)$ guarantees the condition (6).

Denote $f(t)$ and the reference trajectory $r_k(t)$ in the *k-th* iteration in the super vector form as follows:

$$\begin{aligned} f_k &= [f_k^T(1), f_k^T(2), ..., f_k^T(N)]^T, \\ r_k &= [r_k^T(1), r_k^T(2), ..., r_k^T(N)]^T. \end{aligned} \tag{8}$$

Then, with the given initial reference trajectory $r_0(t)$ and the function $h(t)$, the trajectory is updated as guarantees the condition $r_k(t_i^s) = y_d(t_i^s)$. Thus, the updated trajectory at the specified data points remains constant. In contrast, the new reference trajectory at other time instants is dependent on f_{k+1}. Therefore, f_{k+1} can be considered as a learning factor in the learning algorithm.

3.2 ILC Controller

In this section, the ILC controller is designed where both the reference trajectory and the input are iteratively computed. Firstly, the condition on the selection of the learning factor f_{k+1} is analyzed. Then, the monotonic convergence of the controller is addressed, by considering the convergence rate.

As increasing the convergence rate of the ILC algorithm compared to the ILC controller with the given pre-planned reference trajectory, f_{k+1} should be chosen such that

$$\| r_{k+1} - y_k \| \leq \| r_k - y_k \| . \tag{9}$$

Consider the learning update algorithm.

$$f_{k+1} = F(r_k - y_k). \tag{10}$$

Theorem 1.

If the matrix $F = diag(\gamma(1), \gamma(2), ..., \gamma(N))$ is chosen at $\forall t \in [0, T]$ rather than $t \in T^s = \{t_1^s, t_2^s, ..., t_M^s\}$, then $(1 + h(t)\gamma(t)) < 1$, and $\| I + FH \| = 1$. Conclusively, condition (9) is guaranteed.

Proof. From (9) and (10),

$$\begin{aligned} r_{k+1} &= r_k + HF(r_k - y_k) - y_k \\ &= (I + FH)(r_k - y_k). \end{aligned} \tag{11}$$

Hence, we obtain $\| r_{k+1} - y_k \| \leq \| I + HF \| \, \| r_k - y_k \|$. Since $\| I + FH \| = 1$, then $\| r_{k+1} - y_k \| \leq \| r_k - y_k \|$.

Note that, the diagonal matrix H is determined on the basis of the given specified data points. Hence, it is always possible to find suitable $\gamma(t)$ analytically such that $(1 + h(t)\gamma(t)) < 1$ at $\forall t \in [0, T]$, other than $t \in T^s = \{t_1^s, t_2^s, ..., t_M^s\}$, $\| I + FH \| = 1$. For example, $\gamma(t)$ could be chosen as

$$\gamma(t) = -\alpha \frac{sgn(h(t))}{h_{\max}}, 0 < \alpha < 1 \tag{12}$$

with $h_{\max} = \max(|h(t)|)$.

From these settings, $I + FH$ is a diagonal matrix as $I + FH = diag(1 + h(1)\gamma(1), 1 + h(2)\gamma(2), ..., 1 + h(N)\gamma(N))$, and its norm is defined by $\| I + FH \| = \max_{t = 1, 2, ..., N} |1 + h(t)\gamma(t)|$. Hence, $(1 + h(t)\gamma(t)) < 1$ at $\forall t \in [0, T]$, other than $t \in T^s$. Thus, $\| I + FH \| = 1$.

Thus far, the update law for the reference trajectory has been added; then next, the monotonic convergence property of the new ILC controller will be shown.

Consider the linear system (3) and the ILC controller:

$$\begin{aligned} u_{k+1} &= u_k + L(r_{k+1} - y_k) \\ r_{k+1} &= r_k + HF(r_k - y_k) \end{aligned}, \tag{13}$$

where L is the learning gain matrix and F is chosen as in the previous part.

Theorem 1.

The ILC system is monotonic convergent and r_k is bounded if

L is chosen such that $\| I-PL \| < 1$ and the matrix $(I-PL)(I+HF)$ is diagonalizable.

Proof. The error at the *(k + 1)-th* iteration is derived as

$$\begin{aligned} r_{k+1}-y_{k+1} &= r_{k+1}-P(u_k+L(r_{k+1}-y_k)) \\ &= r_{k+1}-y_k-PL(r_{k+1}-y_k) \\ &= (I-PL)(I+HF)(r_k-y_k). \end{aligned} \tag{14}$$

Then, it leads to the inequality

$$\| e_{k+1} \| \leq \| I-PL \| \; \| I+HF \| \; \| e_k \| . \tag{15}$$

Consequently, the monotonic convergence of error is guaranteed if $\| I-PL \| < 1$. Next, the boundedness of r_k is mentioned. From (13) and (14), following equality is derived

$$r_{k+1} = HF\left[\sum_{i=1}^{k}((I-PL)(I+HF))^i\right]e_0+r_0. \tag{16}$$

Since, $\| (I-PL)(I+HF) \| < 1$ the spectral radius of $(I-PL)(I+HF)$ is less than 1. Thus, since it is diagonalizable, the sequence $\sum_{i=1}^{k}((I-PL)(I+HF))^i$ is convergent to a constant matrix as $k \rightarrow \infty$, which implies that r_k is bounded.

To show the effectiveness of the proposed approach, the rate of convergence of the proposed algorithm is compared to the other ILC algorithms that track a fixed reference trajectory. Consider the following typical ILC algorithm, which uses the given reference trajectory r:

$$\overline{x_{k+1}} = \overline{x_k}+L(r-\overline{y_k}). \tag{17}$$

As a result, the error at the $(k+1)th$ iteration is

$$\begin{aligned} \overline{e_{k+1}} &= r-P(\overline{u_k}+L(r-\overline{y_k})) \\ &= (I-PL)\overline{e_k}. \end{aligned} \tag{18}$$

Then, from (14) and (18), the ILC algorithm with the updated trajectory has a faster rate of convergence than fixed case since $\| (I+HF)e_k \| < \| e_k \|$.

4. Simulation results

The performance of the proposed techniques is illustrated by the following linear discrete-time system model:

$$\begin{aligned} x(t+1) &= \begin{pmatrix} 0 & 1.2 & 0 \\ 0 & 0 & 1.1 \\ -0.1 & -0.3 & -0.2 \end{pmatrix} x(t) + \begin{pmatrix} 0.2 \\ 0.5 \\ 0 \end{pmatrix} u(t) \\ y(t) &= (1\ 0\ 0)x(t) \end{aligned} \tag{19}$$

which operates on t ∈ {0, 1,⋯, 79}. There are 6 points of (1, 0), (10, 4), (26, -9), (40, 8), (60, -1), (80, 1.4) was selected in the interval as desired specified points in the motion profile. In the approach, H is selected randomly; but with the key requirement of $h(1) = h(10) = h(26) = h(40) = h(60). = h(80) = 0$. Simulation results are as shown in **Figure 1** ~ **Figure 4**.

Figure 1 shows the specified points(blue points) in the motion profile and fixed interpolated trajectory(blue line). Red line and green line means the updated trajectories of 3rd iteration and 50th iteration respectively. The fixed interpolated trajectory is obtained by applying spline technique from selected points. The 3rd and 50th reference trajectories are generated by using proposed approach.

Figure 2 illustrates achieved actual outputs of given system by appling proposed ILC controller. Accordingly, based on suitable chosen weighting matrices, the controller produces superior performance; specifically, output signals close to and go through the desired given data points after some iteration.

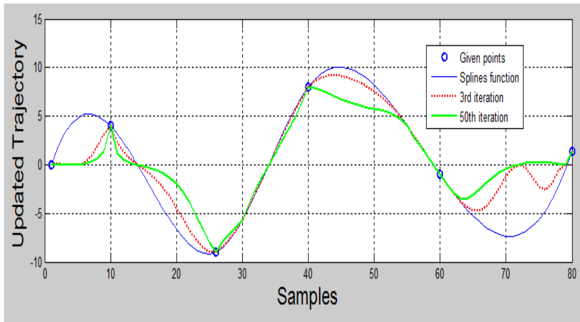

Figure 1: Specified points, fixed-trajectory & updated trajectories

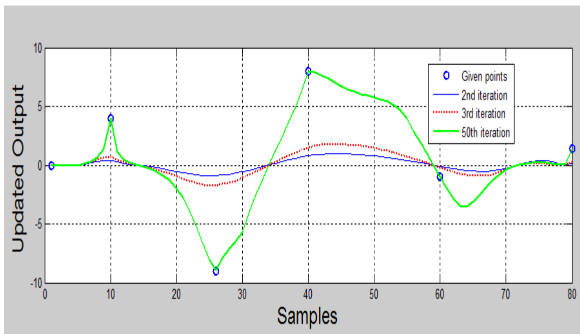

Figure 2: Updated outputs and achieved actual outputs

Figure 3 depicts the errors of the proposed approach and the method suggested by Freeman et al. **[10]**. It shows the norm of errors in linear scale versus trial number. As shown in these plots, the proposed approach shows just a slightly better performance than the method of Freeman et al. **[10]**.

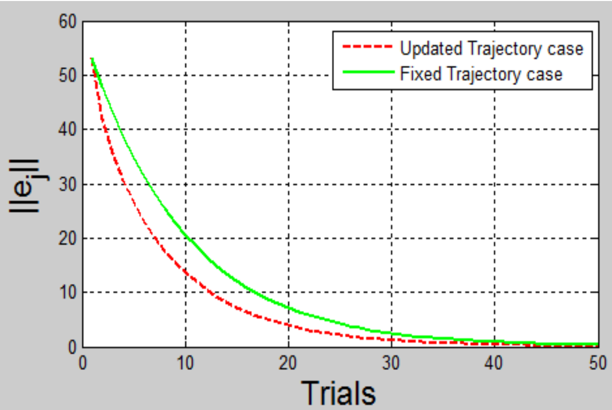

Figure 3: Errors of proposed approach and Freeman's method (fixed trajectory case)

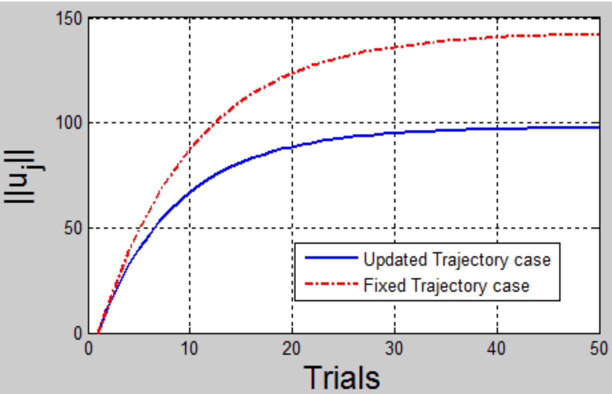

Figure 4: Control efforts of proposed approach and Freeman's method

However, from **Figure 4**, which depicts the control efforts (i.e., the amount of control signals used in ILC updates) of the proposed approach and Freeman et al. **[10]**, it is obvious to see that the proposed approach requires lesser control efforts than the method of Freeman et al. **[10]**. It is noticeable that the computational cost of the proposed approach is much less than that of Freeman et al. **[10]** since the former uses only the specified data points.

5. Conclusion

This paper presented optimal tracking strategies for problems where the system must pass through or close to desired data points at given specified time instants, for the case of systems that repeat their operation from the same initial conditions at each trial. Taking an iterative learning control approach, we also proposed a methods which bridge trajectory planning with ILC tracking control, using a time-domain based approach where the reference trajectory is updated iteratively, and at the same time the control input is updated. The relationship between the reference trajectory and ILC tracking control was investigated for improving the rate of convergence. Simulation were accomplished and their results were analyzed for verifying effectiveness of the proposed algorithm. Future work will consider the application of the proposed ILC to the nonlinear systems as well as linear systems with model uncertainties.

Acknowledgements

The research was supported by the Green Marine Equipment RIS Center Program through the Ministry of Trade, Industry & Energy (MOTIE).

References

[1] N. Amann, D. H. Owens, and E. Rogers, “Iterative learning control using optimal feed back and feed forward actions,” International Journal of Control, vol. 65, no. 2, pp. 277-293, 1996.

[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, pp. 96-114, 2006.

[3] 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, pp. 1099-1121, 2007.

[4] K. L. Moore, “Iterative learning control: An expository overview,” Journal of Applied and Computational Control, Signals, and Circuits, vol. 1, no. 1, pp. 151-214, 1999.

[5] D. K. Le and T. K. Nam, “Optimal iterative learning control with model uncertainty,” Journal of the Korean Society of Marine Engineering, vol. 37, no. 7, pp. 743-751, 2013.

[6] D. K Le and T. K. Nam, “Performance improvement of trajectory tracking control by using ILC,” Proceedings of the 38th KOSME Spring Conference, pp. 194, 2014.

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

[8] G. Gauthier and B. Boulet, “Terminal iterative learning control design with singular value decomposition decoupling for thermoforming ovens,” Proceedings of the American Control Conference, pp. 1640-1645, 2009.

[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, vol. 19, no. 3, pp. 590-600, 2011.

[10] C. T. Freeman and Y. Tan, “Point-to-point iterative learning control with mixed constraints,” Proceedings of the American Control Conference, pp. 3657-3662, 2011.

[11] C. T. Freeman, E. Rogers, A. M. Hughes, J. H. Burridge, and K. L. Meadmore, “Iterative learning control in healthcare electrical stimulation and robotic-assisted upper limb stroke rehabilitation,” IEEE Control Systems Magazine, vol. 32, no. 1, pp. 18-43, 2012.

[12] J. X. Xu and J. Xu, “Iterative learning control for non-uniform trajectory tracking problems,” Proceedings of the 15th IFAC World Congress, pp. 1048-1048, 2002.

[13] C. T. Freeman, C. Zhonglun, P. L. Lewin, and E. Rogers, “Iterative learning control for multiple point-to-point tracking,” Proceedings of the 48th IEEE Conference on Decision and Control, pp. 3288-3293, 2009a.

[14] C. T. Freeman, C. Zhonglun, P. L. Lewin, and E. Rogers, “Objective-driven ilc for point-to-point movement tasks,” Proceedings of the American Control Conference, pp. 252-257, 2009b.

[15] A. D. Luca, L. Lanari, and G. A. Oriolo, “Sensitivity approach to optimal spline robot trajectories,” Journal of Automatica, vol. 27, no. 3, pp. 535-539, 1991.

[16] S. Sun, M. Egerstedt, and C. F. Martin, “Control theoretic smoothing spline,” IEEE Transactions on Automatic Control, vol. 45, no. 12, pp. 2271-2279, 2000.