Automatica 89 (2018) 135–143

Contents lists available at ScienceDirect

Automatica

journal homepage: www.elsevier.com/locate/automatica

Point-to-point iterative learning model predictive control☆

Se-Kyu Oh [1], Byung Jun Park, Jong Min Lee *

School of Chemical and Biological Engineering, Institute of Chemical Processes, Seoul National University, 1 Gwanak-ro, Gwanak-gu, Seoul, 08826, Republic of Korea

A R T I C L E I N F O

Article history:
Received 18 October 2016
Received in revised form 12 July 2017
Accepted 21 September 2017

Keywords:
Iterative learning control
Model predictive control
Point-to-point control

A B S T R A C T

Iterative learning model predictive control (ILMPC) is a technique that combines iterative learning control (ILC) and model predictive control (MPC). The objective is to track a reference trajectory of repetitive processes on a finite time interval while rejecting real-time disturbances. In many repetitive processes, the output is not required to track all the points of a reference trajectory. In this study, we propose a point-to-point ILMPC (PTP ILMPC) technique considering only the desired reference points, and not an entire reference trajectory. In this method, an arbitrary reference trajectory passing through the desired reference values need not be generated. Numerical examples are provided to demonstrate the performances of the suggested approach in terms of PTP tracking, iterative learning, constraint handling, and real-time disturbance rejection.

1. Introduction

Iterative learning control (ILC) is an effective technique for controlling repetitive, cyclic, batch, or iterative processes wherein the same task is repeated for a finite time interval. ILC can achieve iteration-wise asymptotic convergence along the iteration axis under model uncertainty and iteration-invariant disturbances by learning from previous iterations. The technique was originally introduced for robot manipulators (Arimoto, Kawamura, & Miyazaki, 1984) and has since been applied to many industrial processes (Ahn, Chen, & Moore, 2007). Because the conventional ILC law is updated using the error obtained in the previous iteration, the controller is said to be in an open-loop configuration, and thus, cannot be used to reject real-time disturbances.

Several studies have been conducted to incorporate a feedback controller in ILC. An ILC using current cycle feedback (CCF) was proposed to combine ILC with the conventional feedback control wherein the error obtained in the current cycle is simply added to the conventional ILC algorithm (Hashimoto, Xu, Kang, & Harashima, 1987; Xu, Wang, & Heng, 1995). Amann et al. proposed a norm-optimal ILC based on an optimization problem wherein the norm of the tracking error is minimized (Amann, Owens, & Rogers, 1996). Several closed-loop ILC methods based on two-dimensional (2D) systems have been proposed for robust tracking of a set-point profile against uncertainties and real-time disturbances (Liu, Gao, & Wang, 2010; Liu & Wang, 2012). Recently, ILC methods combined with model predictive control (MPC), called iterative learning model predictive control (ILMPC), have attracted much attention because of their applicability for complex constrained multivariable control problems observed in process industries (Lee, Chin, Lee, & Lee, 1999; Oh & Lee, 2016; Wang, Zhou, & Gao, 2008). A major drawback of applying MPC to iterative processes is that it cannot track an entire reference trajectory under model-plant mismatch and shows identical tracking performance for all iterations because it does not use the information, e.g., tracking errors, inputs, and outputs of previous runs. The main objective of the ILC method, along with the feedback controller, is to track an entire reference trajectory at all time steps while rejecting real-time disturbances.

However, it is not necessary for the output to track an entire reference trajectory in many applications such as robotic “pick-and-place” tasks, crane control, rapid thermal processes, and chemical batch reactors (Freeman & Tan, 2013; Xu, Chen, Lee, & Yamamoto, 1999). The errors only at particular points are critical. Several ILC techniques have been developed to solve the point-to-point (PTP) tracking problem without generating an arbitrary reference trajectory, which passes through the desired points. Terminal ILC (TILC) is a control technique used to track only a terminal target point (Chi, Wang, Hou, & Jin, 2012; Hou, Wang, Yin, & Tang, 2011). TILC was first proposed for rapid thermal processing in chemical vapor

☆ This research was supported by Basic Science Research Program through the National Research Foundation of Korea (NRF) funded by the Ministry of Science, ICT & Future Planning (MSIP) (NRF-2016R1A5A1009592). The material in this paper was not presented at any conference. This paper was recommended for publication in revised form by Associate Editor Abdelhamid Tayebi under the direction of Editor Thomas Parisini.

* Corresponding author.
E-mail addresses: sekyu@snu.ac.kr (S.-K. Oh), qazws93@snu.ac.kr (B.J. Park), jongmin@snu.ac.kr (J.M. Lee).
[1] This author is currently a Postdoctoral Fellow at the Super Ultra Low Energy and Emission Vehicle Center of Korea University, Seoul, Korea.

https://doi.org/10.1016/j.automatica.2017.11.010
0005-1098/© 2017 Elsevier Ltd. All rights reserved.

decomposition system, wherein the main objective is to control the terminal deposition thickness (Xu et al., 1999). However, in the TILC techniques, only the terminal point is considered; thus, ILC techniques that can consider multiple points, called PTP ILC, have been proposed. Freeman, Cai, Rogers, and Lewin (2011) proposed a PTP ILC approach wherein the reference trajectory is updated between iterations. Unlike the reference-trajectory update method, several PTP ILC techniques were proposed without updating reference trajectory (Chu, Freeman, & Owens, 2015; Freeman & Tan, 2013; Son, Ahn, & Moore, 2013). Whereas these methods can handle constraints, they do not contain the feedback mechanism within an iteration and thus cannot reject real-time disturbance. Unlike the general tracking problems, the PTP tracking problems have output errors only at the desired points. Thus, a conventional feedback controller cannot be combined with PTP ILC because the conventional feedback controller requires output feedback at every time step. In addition, combining PTP ILC with the real-time feedback control necessitates state feedback at each time step for minimizing the output errors at future reference points.

The main objective of this study is to propose a framework that can accomplish both the asymptotic PTP tracking along the iteration axis and real-time disturbance rejection along the time axis. For the real-time feedback control of the PTP tracking problem, we propose a PTP ILC method combined with MPC, called point-to-point iterative learning model predictive control (PTP ILMPC). In the proposed PTP ILMPC, the state variables of each time step are employed as feedback signals and only the output errors of future reference points are minimized. Thus, the method only requires desired reference points without a reference trajectory, which is often arbitrarily designed to pass through the desired reference points. Furthermore, the proposed PTP ILMPC algorithm can reject real-time disturbances via the feedback mechanism and can consider both input and output constraints. Output constraints are implemented as soft constraints by adding a slack variable to the constraints to avoid infeasibility and potential conflict between input and output constraints. To ensure the convergence of tracking error, the suggested approach requires that the errors between the measured and estimated outputs tend to zero for all time points as the number of iterations tends to infinity. However, neither the classical observer nor the Kalman filter can be used to ensure that the estimation error converges to zero for all time steps. To overcome this issue, iterative learning observer (ILO) is incorporated into the algorithm, thus ensuring that the estimation error tends to zero for all time steps as the number of iterations tends to infinity (Hätönen & Moore, 2007). The nominal stability and robustness are analyzed in this study. In Section 5.1, a comparison between the proposed technique and the existing PTP ILC (Son et al., 2013) is presented.

The rest of this paper is organized as follows: In Section 2, the prediction model is derived using the double-incremental model. Section 3 introduces the extraction matrix and presents the main algorithm of the PTP ILMPC including ILO. Section 4 presents three results of the convergence analysis. The first result is the nominal stability of the input sequences, and subsequently, a robust convergence condition is provided. Finally, we show the convergence of the error. In Section 5, numerical illustrations are provided for a single-input single-output (SISO) linear system and a multiple-input multiple-output (MIMO) nonlinear process. Finally, concluding remarks are provided in Section 6.

2. Problem formulation

We consider the linear discrete time-invariant system which operates on an interval $t \in [0,\ N]$:

$$\bar{x}_k(t+1) = \bar{A}\bar{x}_k(t) + \bar{B}u_k(t),\ \ y_k(t) = \bar{C}\bar{x}_k(t) \tag{1}$$

where t is the time index; k is the iteration or batch index; $\bar{x}_k(t) \in \mathbb{R}^{n_x}$; $u_k(t) \in \mathbb{R}^{n_u}$; $y_k(t) \in \mathbb{R}^{n_y}$; $\bar{A}$, $\bar{B}$, and $\bar{C}$ are matrices of appropriate dimensions. The state–space model is augmented using $y_k(t+1) = \bar{C}\bar{A}\delta\bar{x}_k(t) + y_k(t) + \bar{C}\bar{B}\delta u_k(t)$ where δ is the time-increment operator, i.e., $\delta\bar{x}_k(t) = \bar{x}_k(t) - \bar{x}_k(t-1)$ and $\delta u_k(t) = u_k(t) - u_k(t-1)$.

$$\begin{aligned} \begin{bmatrix} \delta\bar{x}_k(t+1) \\ y_k(t+1) \end{bmatrix} &= \overbrace{\begin{bmatrix} \bar{A} & 0 \\ \bar{C}\bar{A} & I \end{bmatrix}}^{A} \overbrace{\begin{bmatrix} \delta\bar{x}_k(t) \\ y_k(t) \end{bmatrix}}^{x_k(t)} + \overbrace{\begin{bmatrix} \bar{B} \\ \bar{C}\bar{B} \end{bmatrix}}^{B} \delta u_k(t) \\ y_k(t) &= \underbrace{\begin{bmatrix} 0 & I \end{bmatrix}}_{C} \begin{bmatrix} \delta\bar{x}_k(t) \\ y_k(t) \end{bmatrix}. \end{aligned} \tag{2}$$

To define a new state vector, the incremental model takes the following general form:

$$x_k(t+1) = Ax_k(t) + B\delta u_k(t),\ \ y_k(t) = Cx_k(t). \tag{3}$$

Eq. (3) is referred to as an incremental state–space model or linear velocity-form model (Wang, 2004). A well-posed optimization problem with the velocity-form model ensures that the offset can be eliminated, which is defined as the steady state error between the controlled outputs and the desired set-points (Betti, Farina, & Scattolini, 2013). The offset-free tracking is necessary to reject real-time disturbances. Subsequently, system (3) can be rewritten in the form of the following iteration incremental model using the batch-increment operator Δ.

$$\Delta x_k(t+1) = A\Delta x_k(t) + B\Delta\delta u_k(t),\ \ \Delta y_k(t) = C\Delta x_k(t) \tag{4}$$

where $\Delta x_k(t) = x_k(t) - x_{k-1}(t)$, $\Delta y_k(t) = y_k(t) - y_{k-1}(t)$, and $\Delta\delta u_k(t) = \delta u_k(t) - \delta u_{k-1}(t)$. The prediction model for general ILMPC can be represented using the double-incremental model (4) as follows:

$$\hat{\mathbf{y}}_k^p(t+1|t) = \mathbf{y}_{k-1}^p(t+1) + \mathbf{G}\Delta\delta\mathbf{u}_k^m(t) + \mathbf{F}\Delta\hat{x}_k(t|t) \tag{5}$$

$$\mathbf{G} \triangleq \begin{bmatrix} CB & 0 & \cdots & 0 \\ CAB & CB & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ CA^{p-1}B & CA^{p-2}B & \cdots & CA^{p-m}B \end{bmatrix},\ \mathbf{F} \triangleq \begin{bmatrix} CA \\ CA^2 \\ \vdots \\ CA^p \end{bmatrix} \tag{6}$$

$$\begin{aligned} \hat{\mathbf{y}}_k^p(t+1|t) &\triangleq \left[\hat{y}_k(t+1|t)^T \ \cdots \ \hat{y}_k(t+p|t)^T\right]^T \\ \Delta\delta\mathbf{u}_k^m(t) &\triangleq \left[\Delta\delta u_k(t)^T \ \cdots \ \Delta\delta u_k(t+m-1)^T\right]^T \end{aligned} \tag{7}$$

where p is the prediction horizon; m is the control horizon; $\hat{y}_k(t+1|t)$ denotes output estimates of $y_k(t+1)$ based on the information available at time t of the kth iteration; $\Delta\hat{x}_k(t|t)$ denotes state estimates. In case of ILMPC, the algorithm should have a reference trajectory for all time t. The vector of reference trajectory for ILMPC is defined as $\mathbf{r} \triangleq \left[r(1)^T \quad r(2)^T \quad \cdots \quad r(N)^T\right]^T$, and $\mathbf{r}^p(t) \triangleq \left[r(t)^T \quad r(t+1)^T \quad \cdots \quad r(t+p)^T\right]^T$.

In the ILMPC algorithm, the prediction and control horizons should not exceed remaining time points. Thus, we introduce the concept of shrinking horizons (Joseph & Hanratty, 1993), and the horizons are updated as

$$\begin{aligned} p &= \begin{cases} p_0 & \text{, if } p_0 \le N-t \\ N-t & \text{, otherwise} \end{cases} \\ m &= \begin{cases} m_0 & \text{, if } m_0 \le N-t \\ N-t & \text{, otherwise} \end{cases} \end{aligned} \tag{8}$$

where p_0 and m_0 are the initial prediction horizon and initial control horizon, respectively.