Iterative Learning Control Integrated with Model Predictive Control for Real-Time Disturbance Rejection of Batch Processes

Se-Kyu Oh and Jong Min Lee
School of Chemical and Biological Engineering, Institute of Chemical Processes, Seoul National University, 1 Gwanak-ro, Gwanak-gu, Seoul, 08826, Korea

Keywords: Iterative Learning Control, Model Predictive Control, Disturbance Rejection, Non-zero Convergence

In the present paper, iterative learning control (ILC) is integrated with a model predictive control (MPC) technique to reject real-time disturbances. The proposed scheme is called iterative learning model predictive control (ILMPC). ILC is an effective control technique for batch processes, but it is not a real-time feedback controller. Thus, it should be combined with MPC for real-time disturbance rejection. The existing ILMPC techniques make the error converge to zero. However, if the error converges to zero, an impractical input trajectory may be calculated. We use a generalized objective function to independently tune weighting factors of manipulated variable change with respect to both the time index and batch horizons. If the generalized objective function is used, output error converges to non-zero values. We provide convergence analysis for both cases of zero convergence and non-zero convergence.

Introduction

Iterative learning control (ILC) is an effective control technique for the reference trajectory tracking problem of batch processes. It was originally proposed for robot manipulator (Arimoto et al., 1984) and has been studied for many industrial processes (Bristow et al., 2006; Ahn et al., 2007). In the field of process control, many studies have been conducted for chemical batch reactor (Mezghani et al., 2001; Lee and Lee, 2003) and semiconductor processes (Choi and Do, 2001; Yu et al., 2014). General ILC can achieve exponential or monotonic convergence along the batch index; however, it cannot reject real-time disturbances because ILC is basically open-loop control within a batch. Model predictive control (MPC) is a popular optimization-based control strategy based on a prediction model. MPC can reject real-time disturbances; however, it cannot show convergence along the batch index. In other words, MPC shows the same tracking performance for all batches. Therefore, ILC should be combined with MPC to reject real-time disturbances of multivariable batch processes. The control technique of combining ILC and MPC is often referred to as iterative learning model predictive control (ILMPC) (Oh and Lee, 2016).

In most studies of ILMPC, the state vector consists of the entire error sequences of a batch (Lee et al., 2000; Xiong et al., 2005; Liu and Kong, 2013), and a prediction horizon is fixed as an entire batch horizon; therefore, the control calculation might not be performed within a sample time. Several studies combine a time-wise feedback controller and batch-wise feed-forward controller separately (Chin et al., 2004; Lu et al., 2015). This two-stage approach requires two optimization steps, and thus has difficulties in system analysis and parameter tuning. In addition, all the papers described above cannot tune the weighting factor of the rate of input change vector with respect to the time index independently. In this case, an input trajectory for perfect tracking is calculated when convergence is complete. It can lead to impractical input trajectory.

In the present paper, we propose a generalized ILMPC scheme which can independently tune weighting factors of the rate of input change vector with respect to both the time index and batch horizons. The proposed ILMPC is similar in structure to conventional MPC and uses a general state-space model. Thus, it can change a prediction horizon and a control horizon and various techniques applicable to MPC can be applied without further modification. Moreover, it has a single optimization step. In the proposed ILMPC, output error converges to non-zero values because the weighting factor for time index is independently tuned. If we set the weighting factor for the time index to be zero, the output error converges to zero. We provide convergence analysis for both cases of zero convergence and non-zero convergence.

The rest of the present paper is organized as follows. In Section 1, a prediction model for the proposed ILMPC is derived. Section 2 proposes the main algorithms of both unconstrained and constrained ILMPC using the prediction model. In Section 3, we prove both cases where the error goes to zero and goes to non-zero. Section 4 shows examples to illustrate the performance of the disturbance rejection and the effect of the weighting factor for time index.
1. Preliminary

We consider the following linear discrete time-invariant system.
\[
\begin{align*}
\dot{x}_t &= A_t x_t + B_t u_t(t) \\
y_t &= C_t x_t
\end{align*}
\]
Here \(x_t\) is the state vector; \(u_t\) is the input vector; \(y_t\) is the output vector; \(k\) is the batch index; \(t\) is the time index. For offset-free control, we employ delta input formulations which use the control increment \((\delta u_t(t) = u_t(t) - u_t(t-1))\) instead of the control signal \(u_t(t)\). The following delta input formulation can be obtained using \(u_t(t) = u_t(t-1) + \delta u_t(t)\).

\[
\begin{align*}
\dot{x}_t &= \tilde{A} x_t + \tilde{B} u_t(t) \\
y_t &= \tilde{C} x_t
\end{align*}
\]

Here \(\Delta\) means the increment operator with respect to the batch index, that is, \(\Delta x_t = x_t(t) - x_t(t-1)\).

The prediction model for ILMPC can be represented using the double delta formulation (4) as follows
\[
\begin{align*}
\hat{y}_t(t+1) &= y^{k-1}_t(t+1) + G \Delta \dot{u}_t(t) + F \Delta \dot{x}_t(t|t)
\end{align*}
\]

where
\[
\begin{align*}
G &= \begin{bmatrix} CB & 0 & \cdots & 0 \\ CAB & CB & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ CA^{p-1}B & CA^{p-2}B & \cdots & CA^{p-m}B \end{bmatrix} \\
F &= \begin{bmatrix} CA \\ CA^2 \\ \vdots \\ CA^p \end{bmatrix} \\
\dot{y}_t(t+1) &= \begin{bmatrix} \dot{y}_t(t+1) \\ \vdots \\ \dot{y}_t(t+1) \end{bmatrix}^T \\
\Delta \dot{u}_t(t) &= \begin{bmatrix} \Delta \dot{u}_t(t) \\ \vdots \\ \Delta \dot{u}_t(t+m-1) \end{bmatrix}^T
\end{align*}
\]

Here \(p\) is the prediction horizon and \(m\) is the control horizon. We use the state estimate \(\Delta x_t(t)\) instead of \(\Delta x_t(t)\) because we assume that all the states are not measurable where \(\hat{\cdot}\) means an estimated value. Thus, \(\Delta x_t(t)\) denotes state estimates of \(\Delta x_t(t)\) based on the information available at time \(t\) of the \(t\)-th batch. Similarly, \(\hat{y}_t(t+1)\) denote output estimates of \(y_t(t+1)\) based on \(y_t(t)\). In the ILMPC algorithm, both horizons should not exceed remaining time points. Therefore, the concept of shrinking horizons (Joseph and Hanratty, 1993) is used. Both horizons are updated as following Eq. (8).

\[
p = \begin{cases} p_0, & \text{if } p_0 \leq N - t \\ N - t, & \text{otherwise} \end{cases}
\]

\[
m = \begin{cases} m_0, & \text{if } m_0 \leq N - t \\ N - t, & \text{otherwise} \end{cases}
\]

Here, \(p_0\) is the initial prediction horizon and \(m_0\) is the initial control horizon.

2. Iterative Learning Model Predictive Controller

We use the following objective function to design the ILMPC controller.
\[
\begin{align*}
\min_{\Delta u_k(t)} & \frac{1}{2} \left[ \|e_2(t+1)\|_Q^2 + \|e_2(t+1)\|_R^2 + \|\Delta u_k(t)\|_W^2 \right] \\
\end{align*}
\]

where \(e_2(t+1) = r(t+1) - \hat{y}_2(t+1)\)

\[
r(t+1) = \begin{bmatrix} r(t+1) \\ \vdots \\ \vdots \\ r(t+1) \end{bmatrix}^T \\
\]

The following analytical solution can be obtained using Eqs. (9) and (11).
Many control applications need to ensure safety and smooth trajectory (5), (9), and (11).

\[ \Delta u_k(t) = I^*_k \Delta u^*_k(t) = -I^*_k \mathcal{H}^{-1} f \]  

where

\[ \mathcal{H} = I^*_k G^* Q I_k + I^*_k R I_k + S \]

\[ f = I^*_k G^* Q (F \hat{X}_k(t) | t) - G I_k \Delta u_k(t-1) - e_k^\#(t+1)) + I^*_k R (I_k u^*_k(t) - I_k u_k(t-1)) \]

The optimization problem can be solved by the appropriate QP solver. The first input of the optimal solution is

\[ \Delta u_k(t) \]

subject to

\[ M \Delta u^*_k(t) \leq b_k(t) \]

where

\[ M = \begin{bmatrix} -I & I & -I_k & -I_k & -GL_k \\ I & -I_k & I_k & GL_k \end{bmatrix} \]

\[ \begin{bmatrix} -u^{m}_{\text{min}} + u^{m}_{k-1}(t) \\ u^{m}_{\text{max}} - u^{m}_{k-1}(t) \\ -\delta u^{m}_{\text{min}} + I_k u^{m}_{k-1}(t) - I_k u_k(t-1) \\ \delta u^{m}_{\text{max}} - I_k u^{m}_{k-1}(t) + I_k u_k(t-1) \\ -\Delta u^{m}_{\text{min}} - I_k \Delta u_{k-1}(t-1) \\ \Delta u^{m}_{\text{max}} + I_k \Delta u_{k-1}(t-1) \\ -y^{p}_{\text{min}} + y^{p}_{k-1}(t+1) - GL_k \Delta u_{k-1}(t-1) + F \hat{X}_k(t | t) \\ y^{p}_{\text{max}} - y^{p}_{k-1}(t+1) + GL_k \Delta u_{k-1}(t-1) - F \hat{X}_k(t | t) \end{bmatrix} \]

The optimization problem can be solved by the appropriate QP solver. The first input of the optimal solution is implemented on the plant. The formulation of the proposed ILMPC is similar to the conventional MPC formulation. Thus, various techniques applicable to MPC, such as the disturbance model, time-varying model and advanced state estimation theory, can be applied without modifying the structure of the controller.

### 3. Convergence Analysis

#### 3.1 Convergence analysis for an input trajectory

First, we prove that \( \Delta u_k(t) \) converges to zero for all \( t \) as \( k \to \infty \) under the following assumptions.

1. There exists a feasible input trajectory such that \( e_k = 0 \).
2. All constraints (16) are satisfied when an input trajectory is converged.
3. A system has the same initial condition for all batches; the same input trajectory and the same state trajectory lead to the same output trajectory.
4. \( Q, R \) and \( S \) are symmetric positive definite.

**Theorem 1** Consider the assumptions and the QP problem. Then, \( \Delta u_k(t) \to 0 \) for all \( t \) as \( k \to \infty \).

**Proof** We consider the objective function and the minimizer of the optimization problem at time \( t \) of the \( k \)-th batch.

\[ \Phi_k(t) = \frac{1}{2} \left[ ||e_k^p(t+1)||_Q^2 + ||\delta u_k^m(t)||_R^2 + ||\Delta u_k^m(t)||_S^2 \right] \]

\[ J_k(t) = \min_{\Delta u_k(t)} \Phi_k(t) \geq 0 \]

subject to Eq. (16).

An optimal cost \( J_k(t) \) of an objective function is always less than or equal to a feasible cost \( \Phi_k(t) \), i.e., \( J_k(t) \leq \Phi_k(t) \). Let \( (e_k^p(t+1), u_k^m(t)) \) be the optimal solution for the \( k \)-th batch. The optimal solution of the \( k \)-th batch until time \( t \) can be used for the \((k+1)\)-th batch, then the optimal cost \( J_{k+1}(t) \) of the \((k+1)\)-th batch becomes the feasible cost \( \Phi_{k+1}(t) \) of the \((k+1)\)-th batch. Thus, \( e_{k+1}^p(t+1) = e_k^p(t+1), u_{k+1}^m(t) = u_k^m(t), \delta u_{k+1}^m = \delta u_k^m \) and \( \Delta u_{k+1}^m = \Delta u_k^m \).

We have the following inequality.

\[ J_{k+1}(t) \leq \frac{1}{2} \left[ ||e_{k+1}^p(t+1)||_Q^2 + ||\delta u_k^m(t)||_R^2 + ||\Delta u_k^m(t)||_S^2 \right] \]

By adding and subtracting the same term, we have Eq. (24) which yields Eq. (25).

\[ J_{k+1}(t) \leq \frac{1}{2} \left[ ||e_k^p(t+1)||_Q^2 + ||\delta u_k^m(t)||_R^2 + ||\Delta u_k^m(t)||_S^2 \right] \]

\[ = J_k(t) - \frac{1}{2} \||\Delta u_k^m(t)||_S^2 \]

\[ 0 \leq J_{k+1}(t) + \frac{1}{2} \sum_{j=1}^{k} \||\Delta u_j^m(t)||_S^2 \leq J_1(t) < \infty \]

Thus, \( \Delta u_k^m(t) \to 0 \) for all \( t \) as \( k \to \infty \).

#### 3.2 Convergence analysis for an output error

In this section, we show that the output error \( e_k(t) \) converges to a fixed value \( e^* \) (0) using the result of Section 3.1. If \( k \to \infty \), all constraints are satisfied by the assumptions. Thus, the unconstrained solution (13) and the constrained
solution (17) and (18) are equal if \( k \to \infty \). The purpose of this proof is to know the converged error for all time \( (1 \to N) \), not prediction time horizon \( (\infty \to t+p) \); therefore, we use the unconstrained solution and set \( t = 0 \) and \( m = p = N \). In this case, \( f \) in Eq. (15) is simplified because \( \Delta x_i(0|0) = 0 \) (the same initial condition for all batches), \( \Delta u_i(-1) = 0 \) and \( u_i(1) = 0 \). Furthermore, \( S \) in Eq. (14) can be zero because \( \Delta u_a = 0 \) and \( S \) does not affect the converged value. It affects the convergence rate. For the above reasons, \( \mathcal{H} \) in Eq. (14) and \( f \) in Eq. (15) are simplified as Eq. (26).

\[
\mathcal{H} = I_{n} G^T Q G Q + I_{p} R_{c} \\
f = -I_{n} G^T Q e_k + I_{p} R_{c} u_{k-1} 
\tag{26}
\]

**Theorem 2** Consider the proposed ILMPC controller, \( e_k \to 0 \) as \( k \to \infty \) if \( R = 0 \).

**Proof** 2 The unconstrained solution with \( t = 0 \) and \( m = p = N \) is as follows

\[
\Delta u_k = \mathcal{H}^{-1} I_{n} G^T Q e_k 
\tag{27}
\]

By Theorem 1, if \( k \to \infty \), it gives Eq. (28).

\[
\Delta u_k = 0 = \mathcal{H}^{-1} I_{n} G^T Q e_k 
\tag{28}
\]

This implies that \( e_k \to 0 \).

**Theorem 3** Consider the proposed ILMPC controller, \( e_k \to e^* \) as \( k \to \infty \).

**Proof** 3 The unconstrained solution with \( t = 0 \) and \( m = p = N \) is as follows

\[
\Delta u_k = \mathcal{H}^{-1} I_{n} G^T Q e_k + I_{p} R_{c} u_{k-1} 
\tag{29}
\]

The above equation can be expressed as follows

\[
u_k = (I - \mathcal{H}^{-1} I_{n} R_{c} ) u_{k-1} + \mathcal{H}^{-1} I_{n} G^T Q e_k 
\tag{30}
\]

To simplify, we define Eq. (30) as follows

\[
u_k = H_i u_{k-1} + H_c e_k 
\tag{31}
\]

If \( k \to \infty \),

\[
u_k = H_i u_{k-1} + H_c e_k = H_i u_{k-1} + H_c (r - y_k) 
\tag{32}
\]

The state-space model (3) can be expressed as the lifted vector form: \( y_k \) = \( G_p u_s = G_p I_i u_s \) where \( G_p \) is the plant matrix. Substituting the lifted vector form into Eq. (32) and rearranging, we have Eq. (33).

\[
u_k = (I - H_i + H_c G_p I_i)^{-1} H_c r 
\tag{33}
\]

We can obtain the final result by substituting Eq. (33) into \( e_k \) as follows

\[
e^* = r - y_k = r - G_p I_i u_{k-1} 
= \left(I - G_p I_i (I - H_i + H_c G_p I_i)^{-1} H_c\right) r
\tag{34}
\]

To make analysis of Eq. (34) simple, we consider the scalar case of the equation. Assume that terminal time \( N \) is 1 and the system has single-input single-output (SISO), then \( I = I_i = 1 \); other parameters become scalars which are written in non-bold typeface. Equation (34) is expressed as follows

\[
e^* = \left(1 - \frac{G_p H_c}{1 - H_i + H_c G_p}\right) r 
\tag{35}
\]

where

\[
H_1 = \frac{R}{G^T Q + R} \\
H_2 = \frac{G Q}{G^T Q + R} 
\tag{36}
\]

Eq. (35) is simplified as Eq. (37).

\[
e^* = \left(1 - \frac{G G Q}{G G Q + R}\right) r
\tag{37}
\]

The result indicates that a bigger \( R \) increases the size of error. If \( R = 0 \), \( e^* = 0 \); it is the same result as Theorem 2. If \( R \to \infty \), \( e^* \to r \). The reason is that if \( R \to \infty \), the input does not change from 0; thus, the output maintains zero value, i.e., \( e^* = r - y = r \).

### 4. Illustrative Example

We consider a cooling jacket temperature \( (T) \) control of a nonlinear batch reactor. A second-order exothermic reaction \( A \to B \) occurs. It is assumed that \( T_j \) is directly manipulated.

\[
\frac{dT}{dt} = -\frac{UA}{MC_p} (T - T_{i}) - \frac{\Delta H V}{MC_p} k_0 \exp\left(\frac{E}{RT}\right) C_L 
\tag{38}
\]

Here \( T \) and \( C_L \) are the state variables, \( T \) is the output variable, and \( T_i \) is the input variable. The following parameters were used for the plant

\[
\frac{UA}{MC_p} = 0.09 \quad \text{[L/min]} \\
\frac{\Delta H V}{MC_p} = -1.64 \quad \text{[K·L/mol]} \\
k_0 = 2.53 \times 10^{19} \quad \text{[L/mol·min]} \\
\frac{E}{R} = 13,500 \quad \text{[K]} \\
T(0) = 25 \quad \text{[°C]} \\
C_L(0) = 0.9 \quad \text{[mol/L]} 
\tag{39}
\]

We obtained the following linear discrete-time model using the least squares method with a step input with an initial value of 25°C and a size of 1°C; the sampling interval of 1 min. We assumed that the system was second-order.

\[
x(t+1) = \begin{bmatrix} 0.9153 & -0.0416 \\ 0.0313 & 0 \end{bmatrix} x(t) + \begin{bmatrix} 1 \\ 0 \end{bmatrix} u(t) \\
y(t) = \begin{bmatrix} 0.0436 & 1.3600 \end{bmatrix} x(t) 
\tag{40}
\]
Fig. 1  Disturbance for the 7th batch

Fig. 2  Result of the proposed ILMPC with $R = 0.01I$ and $S = 0.05I$

Fig. 3  Disturbance rejection performance of the proposed ILMPC with $R = 0.01I$ and $S = 0.05I$

Fig. 4  Convergence performance with $R = 0.01I$ and $S = 0.05I$ under the disturbance at the 7th batch

Fig. 5  Results of the proposed ILMPC with respect to different sizes of $R$ ($S = 0.01I$)
horizon are 80 and 10, respectively. The weighting factors Q was fixed as the identity matrix for all simulations. For the first simulation which aimed to show the effectiveness of the disturbance rejection, we used $R = 0.01I$ and $S = 0.05I$. Figure 2 shows the results of the 1st and the 2nd batches. The output of the 1st batch does not track the reference trajectory because of the model uncertainty; the output of the 2nd batch converges to the reference trajectory. The unknown disturbance enters the system at the 7th batch as shown in Figure 3. The disturbance is, however, rejected by the real-time feedback controller. The effect of the disturbance remains at the 8th batch because the ILMPC learns from the information of the previous batch. Therefore, the controller learns to reject the disturbance of the 7th batch. Because there is no disturbance at the 8th batch, the output rapidly converges to the reference trajectory again as shown in Figure 4. In Section 3.2, we mentioned that the error cannot go to zero if the weighting factor for $\delta u^m(t)$ is not zero. The example is provided to show the effectiveness for disturbance rejection and iterative learning. Furthermore, the simulation shows that the weighting factor for $\delta u^m(t)$ should be able to be tuned independently for obtaining a practical input trajectory.

**Conclusions**

In the present paper, we have proposed the iterative learning model predictive control technique for real-time disturbance rejection of batch processes. The proposed algorithm can independently tune the weighting factor for the rate of input change with respect to the time index. We prove that the output error cannot go to zero if the weighting factor for $\delta u^m(t)$ is not zero. The example is provided to show the effectiveness for disturbance rejection and iterative learning.

**Acknowledgement**

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). This research was supported by ‘Development of sensor-based virtual plant engineering technology for the support of plant O&M,’ funded by the Ministry of Trade, Industry & Energy (10048341).

**Literature Cited**


5683–5689 (2015)


