# COMBINED REAL-TIME AND ITERATIVE LEARNING CONTROL TECHNIQUE WITH DECOUPLED DISTURBANCE REJECTION FOR BATCH PROCESSES

**In Sik Chin** [*,1] **S. Joe Qin** [*,2] **Kwang S. Lee** [**,3]
**Moonki Cho** [**]

[*] *Department of Chemical Engineering, University of Texas at Austin, Austin, TX 78712, USA*
[**] *Department of Chemical Engineering, Sogang University, Sinsoo-dong 1, Mapo-gu, Seoul 121-742, Korea*

Abstract: A novel stochastic control framework for batch and repetitive processes is proposed. The framework provides a pertinent means to incorporate real-time feedback control (RFC) into iterative learning control (ILC) so that the performance of ILC is virtually decoupled from that of RFC. This is a new advancement since the currently practiced methods for combined RFC and ILC have suffered from the problem that RFC has undesirable effects on ILC such as digression from its convergent track along the run index when there occur run-independent real-time disturbances. Performance of the proposed technique has been demonstrated in two numerical processes.

Keywords: Run-to-run Control, Iterative Learning Control, Model Predictive Control, Stochastic Control.

## 1. INTRODUCTION

Iterative learning control (ILC) is a relatively new technique that has been developed to improve the tracking performance of a process that executes the same operation repeatedly. For the past two decades since the Arimoto's contribution (Arimoto *et al.*, 1984), ILC methods have been steadily improved from SISO (single-input single-output) modeless deterministic approach to MIMO (multi-input multi-output) model-based stochastic approach and the application areas have been extended from the mechatronic systems like robots and disk drivers (Arimoto *et al.*, 1984; Bien and Huh, 1989) to chemical processes (Lee *et al.*, 1994), microelectronic systems (Lee *et al.*, 2001; Qin *et al.*, 2002), biomedical problems (Good *et al.*, 2002), and so forth. It is anticipated that the application of ILC will be broadened more because of the growing importance of batch or run-to-run operation in various industrial processes by the recent trend to produce small quantity of high-valued multiple products.

Basically, ILC concerns the issue of learning from the past operations in order to attain the ultimate tracking performance under model uncertainty and run-wise repeated disturbances. In real applications, however, it is desirable to treat the disturbance within a run and thus real-time feed-

[1] This work was supported by the Pose-doctoral Fellowship Program of Korea Science & Engineering Foundation (KOSEF).
[2] Corresponding author: phone (1-512)471-4417, fax (1-512)471-7060, e-mail qin@che.utexas.edu
[3] Or: phone (82-2)705-8477, fax (82-2)3272-0319, e-mail kslee@ccs.sogang.ac.kr

back control (RFC) need to be combined with ILC.

There are different methods to combine RFC with ILC. A widely practiced method is to add a feedback control block to ILC such that

$$
\begin{aligned}
u_k(t) = & \; u_{k-1}(t) + H_1(t)e_{k-1}(1:N) \\
& + H_2(t)e_k(1:t)
\end{aligned}
\tag{1}
$$

where $(i:j)$ means data from $t = i$ to $j$; $k$ represents the run index; $H_1$ and $H_2$ represent the gains for ILC and FBC, respectively. In fact, RFC-ILC can be realized without $H_1$ since $u_{k-1}(t)$ already contains the information of $e_{k-1}(t)$. Nevertheless, (1) represents the most general form of RFC-ILC.

A trouble with existing combined RFC-ILC techniques is that they lack the capability of distinguishing the run-independent real-time disturbance from the run-wise persisting disturbance. Both RFC and ILC try to reject the disturbances, in real-time for RFC and run-wise for ILC, respectively. When there occurs a large run-independent disturbance in the $k^{\text{th}}$ batch, $u_k(t)$ may change excessively to reject the disturbance, and deteriorate the control performance of the following runs since $u_k(t)$ acts as a feedforward input signal for the next run. To the authors' knowledge, it seems that there has been no attempt to decouple ILC from RFC such that ILC deals with only the disturbance with strong run-wise correlation while RFC handles the run-independent disturbance.

The purpose of the research is to develop a novel framework for combined ILC and RFC, that can separately handle the run-independent real-time disturbance and the run-wise correlated disturbance. For this, a quite general form of disturbance model is first assumed in a stochastic framework by decomposing the disturbance into three parts: the run-wise persisting part, the run-independent part, and the measurement noise. Each of them is separately handled by either ILC or RBC with the aid of the Kalman filters. To complete this, not only the state but also the input is split into two parts: one for ILC and the other for RFC. Through this approach, the resulting controller is able to appropriately discriminate the real-time disturbance from the run-wise persisting disturbance and prevents the effects of the real-time disturbance from being carried over to the future runs. To realize the above concept, we propose a two-stage technique where RFC and ILC are executed in turn during and after a batch run. As a prototypical algorithm, we revised QILC (Quadratic criterion-based ILC) (Lee *et al.*, 2000) and BMPC (Batch Model-based Predictive Control) (Lee and Lee, 1997) and combine them into a two-stage algorithm.

## 2. DISTURBANCE PROPAGATION IN EXISTING TECHNIQUES

### *2.1 Process Modelling*

We consider a linear discrete-time batch process with $u(t)$ and $y(t)$ as input and output variables, respectively, defined over a finite interval with $N$ sampling steps. Although the process undergoes dynamics within a batch run, it can be represented as a linear algebraic system which relates the input sequence vector to the output sequence vector over the underlying discrete-time domain.

$$
\mathbf{y} = \mathbf{G}\mathbf{u} - \mathbf{d} \tag{2}
$$

where

$$
\mathbf{y} = \left[y^T(1)\ y^T(2) \cdots y^T(N)\right]^T \tag{3}
$$

and likewise for $\mathbf{u}$ and $\mathbf{d}$. In the above, $d(t)$ represents the effects of all possible uncertainties including the disturbance, model error, and bias term. For stochastic ILC design, it is sensible to decompose $\mathbf{d}$ into two terms: a run-wise correlated part and a run-independent part.

$$
\mathbf{d} = \mathbf{w} + \mathbf{v}. \tag{4}
$$

If we represent the run-wise correlated part as an integrated white noise process along the run index, then $\mathbf{d}_k$ can be expressed as follows:

$$
\begin{aligned}
\mathbf{w}_k &= \mathbf{w}_{k-1} + \Delta\mathbf{w}_k \\
\mathbf{d}_k &= \mathbf{w}_k + \mathbf{v}_k
\end{aligned}
\tag{5}
$$

where both $\{\Delta\mathbf{w}_k\}$ and $\{\mathbf{v}_k\}$ represent the zero-mean white noise processes along $k$ with covariance matrices $\mathbf{R}_{\Delta w}$ and $\mathbf{R}_v$, respectively.

Let $\mathbf{e}_k = \mathbf{y}_d - \mathbf{e}_k$ and $\bar{\mathbf{e}}_k = \mathbf{e}_k - \mathbf{v}_k$ where $\mathbf{y}_d$ is the desired output trajectory. Then the following inter-run transition model of tracking error can be derived from (2) and (5):

$$
\begin{aligned}
\bar{\mathbf{e}}_k &= \bar{\mathbf{e}}_{k-1} - \mathbf{G}\Delta\mathbf{u}_k + \Delta\mathbf{w}_k \\
\mathbf{e}_k &= \bar{\mathbf{e}}_k + \mathbf{v}_k.
\end{aligned}
\tag{6}
$$

### *2.2 Pure ILC*

The pure ILC algorithm can be written as

$$
\Delta\mathbf{u}_k = \mathbf{H}_1\bar{\mathbf{e}}_{k-1} \tag{7}
$$

In practice, $\bar{\mathbf{e}}_{k-1}$ is not measured, hence should be replaced by an estimate. (7) represents the idea of ILC and we rely on it for analysis purpose.

Substituting (7) into (6) gives

$$\mathbf{e}_k = [\mathbf{I} - \mathbf{G}\mathbf{H}_1]\bar{\mathbf{e}}_{k-1} + \mathbf{v}_k + \Delta\mathbf{w}_k \quad (8)$$
$$\bar{\mathbf{e}}_k = [\mathbf{I} - \mathbf{G}\mathbf{H}_1]\bar{\mathbf{e}}_{k-1} + \Delta\mathbf{w}_k \quad (9)$$

$\mathbf{v}_k$ and $\Delta\mathbf{w}_k$ appear in $\mathbf{e}_k$ without any attenuation. It is a natural result because $\mathbf{v}_k$ and $\Delta\mathbf{w}_k$ are newly entered disturbance at $k$ while $\Delta\mathbf{u}_k$ is calculated based on the previous run information. It can be seen that $\bar{\mathbf{e}}_k$ and, as a consequence, $\Delta\mathbf{u}_{k+1}$ are not affected by $\mathbf{v}_k$. This implies that ILC based on (6) can keep its integrity rejecting the effect of the real-time disturbance.

When $\mathbf{v}_k = \Delta\mathbf{w}_k = 0$ and $\|\mathbf{I}-\mathbf{G}\mathbf{H}_1\| < 1$, $\mathbf{e}_k \to 0$ as $k \to \infty$.

### *2.3 ILC combined with Real-time Feedback*

An ILC algorithm combined with RFC (RFC-ILC) can be expressed as

$$\Delta\mathbf{u}_k = \mathbf{H}_2\mathbf{e}_k \quad (10)$$

To reject the real-time disturbance, $\mathbf{e}_k$ instead of $\bar{\mathbf{e}}_k$ is fed back. For causality, $\mathbf{H}_2$ has a low-triangular structure. Again, the real algorithm may be more complicated than the above. (10) retains the key features of an RFC-ILC algorithm.

Substituting (10) into (6) results in

$$\mathbf{e}_k = [\mathbf{I} + \mathbf{G}\mathbf{H}_2]^{-1}(\bar{\mathbf{e}}_{k-1} + \mathbf{v}_k + \Delta\mathbf{w}_k) \quad (11)$$
$$\bar{\mathbf{e}}_k = [\mathbf{I} + \mathbf{G}\mathbf{H}_2]^{-1}(\bar{\mathbf{e}}_{k-1} - \mathbf{G}\mathbf{H}_2\mathbf{v}_k + \Delta\mathbf{w}_k) \quad (12)$$

It can seen that the effects of $\mathbf{v}_k$ and $\Delta\mathbf{w}_k$ are attenuated in $\mathbf{e}_k$ by the real-time control action. The larger $\mathbf{G}\mathbf{H}_2$ is, the more the disturbance is rejected. However, a large $\mathbf{G}\mathbf{H}_2$ makes $\bar{\mathbf{e}}_k$ be strongly affected by $\mathbf{v}_k$, which not only gives an harmful effect on the ILC track but also deteriorates the performance of $\mathbf{e}_{k+1}$.

## 3. PROPOSED BATCH CONTROL TECHNIQUE

One of the representative RFC-ILC techniques for batch chemical processes, called BMPC (Lee and Lee, 1997), is based on the updating rule (10), and has the problem discussed in the previous section. On the other hand, a pure ILC technique, called QILC (Lee *et al.*, 2000), is based on (7) and can keep the genuine learning track not being affected by the real-time disturbance. Bearing the above in mind, we propose a new RFC-ILC formulation where RFC is separated from ILC so that the effect of the real-time disturbance is blocked from transferring to the learning track.

The proposed technique is designed to perform two-stage calculation: ILC after a run, say it $k-1^{\text{th}}$ run, and RFC calculation during the $k^{\text{th}}$ run on the basis of the learning input. Detailed ILC and RFC algorithms are constructed by modifying existing QILC and BMPC, respectively.

### *3.1 Process Modeling*

Let us decompose the disturbance into three terms: $\mathbf{w}_k$, $\mathbf{v}_k$, $\mathbf{n}_k$ which refer to the parts that will be rejected by ILC and by RFC, and the measurement noise, respectively.

$$\mathbf{d}_k = \mathbf{w}_k + \mathbf{v}_k + \mathbf{n}_k \quad (13)$$
$$\mathbf{w}_k = \mathbf{w}_{k-1} + \Delta\mathbf{w}_k.$$

Also we decompose the a control action $\mathbf{u}_k$ into $\bar{\mathbf{u}}_k$ and $\hat{\mathbf{u}}_k$, each of which is responsible for $\Delta\mathbf{w}_k$ and $\mathbf{v}_k$, respectively. With these variables, the process model can be expressed as

$$\begin{aligned}\mathbf{y}_k &= \mathbf{G}\mathbf{u}_k - \mathbf{d}_k = \mathbf{G}(\bar{\mathbf{u}}_k + \hat{\mathbf{u}}_k) - (\mathbf{w}_k + \mathbf{v}_k + \mathbf{n}_k)\\ &= \underbrace{\underbrace{\mathbf{G}\bar{\mathbf{u}}_k - \mathbf{w}_k}_{\bar{\mathbf{y}}_k} + \mathbf{G}\hat{\mathbf{u}}_k - \mathbf{v}_k}_{\hat{\mathbf{y}}_k} - \mathbf{n}_k\\ &= \bar{\mathbf{y}}_k + \mathbf{G}\hat{\mathbf{u}}_k - \mathbf{v}_k - \mathbf{n}_k = \hat{\mathbf{y}}_k - \mathbf{n}_k \end{aligned} \quad (14)$$

$\bar{\mathbf{u}}_k$ and $\hat{\mathbf{u}}_k$ will be used by ILC and RFC to cancel $\mathbf{w}_k$ and $\mathbf{v}_k$, respectively, while steering $\mathbf{y}_k$ to follow $\mathbf{y}_d$. If perfect disturbance rejection would be made, $\mathbf{y}_k \to \mathbf{y}_d - \mathbf{n}_k$ in the end.

Similarly to (6), the following model equation can be derived from (14):

$$\begin{aligned}\bar{\mathbf{e}}_k &= \bar{\mathbf{e}}_{k-1} - \mathbf{G}\Delta\bar{\mathbf{u}}_k + \Delta\mathbf{w}_k\\ \hat{\mathbf{e}}_k &= \bar{\mathbf{e}}_k - \mathbf{G}\hat{\mathbf{u}}_k + \mathbf{v}_k\\ \mathbf{e}_k &= \hat{\mathbf{e}}_k + \mathbf{n}_k.\end{aligned} \quad (15)$$

In the above, $\bar{\mathbf{e}}_k = \mathbf{y}_d - \bar{\mathbf{y}}_k$ and $\hat{\mathbf{e}}_k = \mathbf{y}_d - \hat{\mathbf{y}}_k$, respectively.

### *3.2 Revised QILC*

ILC calculates $\Delta\bar{\mathbf{u}}_k$ instead of $\Delta\mathbf{u}_k$. After the $k-1^{\text{th}}$ run, $\mathbf{e}_{k-1}$, $\bar{\mathbf{u}}_{k-1}$, and $\mathbf{u}_{k-1}$ are available. Then $\Delta\bar{\mathbf{u}}_k$ is determined such that

$$\min_{\Delta\bar{\mathbf{u}}_k} \frac{1}{2}\left\{\|\bar{\mathbf{e}}^T_{k|k-1}\|^2_{\mathbf{Q}} + \|\Delta\bar{\mathbf{u}}_k\|^2_{\mathbf{R}}\right\} \quad (16)$$

$$\text{subject to } \bar{\mathbf{e}}_{k|k-1} = \bar{\mathbf{e}}_{k-1|k-1} - \mathbf{G}\Delta\bar{\mathbf{u}}_k \quad (17)$$
$$\bar{\mathbf{e}}_{k-1|k-1} = \bar{\mathbf{e}}_{k-1|k-2} + \mathbf{K}(\mathbf{e}_{k-1} - (\bar{\mathbf{e}}_{k-1|k-2} - \mathbf{G}\hat{\mathbf{u}}_{k-1}))$$

where $\mathbf{K}$ is the Kalman gain which depends on $\mathbf{R}_{\Delta\mathbf{w}_k}$ and $\mathbf{R}_{\mathbf{v}}$. Inequality constraints on $\bar{\mathbf{e}}_{k|k-1}$ and $\Delta\bar{\mathbf{u}}_k$ can be incorporated together. The above calculation is repeated after the each run.

### 3.3 Real-time Predictive Control

Let us define the state at $t+1$, which will be regulated by RFC, as

$$\begin{aligned}\hat{\mathbf{e}}_k(t+1) = \hat{\mathbf{e}}_k \ \text{ with } \ \Delta\bar{u}_k(t+1) = \cdots = 0,\\ \hat{u}_k(t+1) = \cdots = 0,\ v_k(t) = \cdots = 0\\ = \bar{\mathbf{e}}_{k-1} - G(0)(\Delta\bar{u}_k(0) + \hat{u}_k(0)) - \cdots\\ -G(t)(\Delta\bar{u}_k(t) + \hat{u}_k(t))\\ + \left[\, v_k^T(0) \ \cdots \ v_k^T(t) \ 0 \ \cdots \right]^T + \Delta\mathbf{w}_k. \quad (18)\end{aligned}$$

The relationship in the above can be derived from (15) and $G(i)$ represents the $i^{\text{th}}$ block column in $\mathbf{G}$. If we write (18) for $\hat{\mathbf{e}}_k(t)$ and take the difference from $\hat{\mathbf{e}}_k(t)$ while assuming the dynamics of the real-time disturbance as

$$v(t) = \alpha v(t-1) + m(t), \quad (19)$$

the state space equation in time is constructed:

$$\begin{aligned}\begin{bmatrix}\hat{\mathbf{e}}_k(t+1)\\ v(t+1)\end{bmatrix} &= \begin{bmatrix} I & H(t)\\ 0 & \alpha I\end{bmatrix}\begin{bmatrix}\hat{\mathbf{e}}_k(t)\\ v(t)\end{bmatrix} \quad (20)\\ &- \begin{bmatrix} G(t)\\ 0\end{bmatrix}(\Delta\bar{u}_k(t) + \hat{u}_k(t)) + \begin{bmatrix}0\\ I\end{bmatrix} m(t)\\ e_k(t) &= \left[\, H^T(t) \ 0 \,\right]\begin{bmatrix}\hat{\mathbf{e}}_k(t)\\ v(t)\end{bmatrix} + n(t) \quad (21)\\ \text{with } \ \begin{bmatrix}\hat{\mathbf{e}}_k(0)\\ v(0)\end{bmatrix} &= \begin{bmatrix}\bar{\mathbf{e}}_{k-1} + \Delta\mathbf{w}_k\\ v_0\end{bmatrix} \quad (22)\end{aligned}$$

where $H(t)$ is a zero matrix except $I$ at the $t^{\text{th}}$ block column such that

$$H(t) = [\overbrace{0 \ \cdots \ 0}^{t-1 \text{ cols}} \ I \ 0 \ \cdots \ 0]^T. \quad (23)$$

For simplicity, let's define

$$\underline{\Delta} u_k(t) = \Delta\bar{u}_k(t) + \hat{u}_k(t) = u_k(t) - \bar{u}_{k-1}(t) \quad (24)$$

The prediction equation can be readily derived from (20). Let $\left[\hat{\mathbf{e}}_k^T(t+m|t) \ v^T(t+m|t)\right]^T$ be the prediction of the state made at time $t$ when there are $m$ future control moves. Then we have

$$\begin{aligned}\begin{bmatrix}\hat{\mathbf{e}}_k(t+m|t)\\ v(t+m|t)\end{bmatrix} &= \begin{bmatrix} I & \sum_{j=0}^{m-1} H(t+j)\\ 0 & \alpha I\end{bmatrix}\begin{bmatrix}\hat{\mathbf{e}}_k(t|t)\\ v(t|t)\end{bmatrix}\\ &- \begin{bmatrix}\mathbf{G}^m(t)\\ 0\end{bmatrix}\underline{\Delta}\mathbf{u}_k^m(t) \quad (25)\end{aligned}$$

where

$$\begin{aligned}\mathbf{G}^m(t) &= \left[\, G(t), \ \cdots, \ G(t+m-1) \,\right],\\ \underline{\Delta}\mathbf{u}_k^m(t) &= \begin{bmatrix}\underline{\Delta} u_k(t)\\ \vdots\\ \underline{\Delta} u_k(t+m-1)\end{bmatrix} \quad (26)\end{aligned}$$

and $\left[\hat{\mathbf{e}}_k^T(t|t) \ v^T(t|t)\right]^T$ is the state estimate by the Kalman filter applied to (20)-(22).

$\underline{\Delta}\mathbf{u}_k(t)$ is calculated to minimize

$$J = \frac{1}{2}\left\{\|\hat{\mathbf{e}}_k(t+m|t)\|_\Gamma^2 + \|\underline{\Delta}\mathbf{u}_k^m(t)\|_\Lambda^2\right\} \quad (27)$$

Inequality constraints can be imposed on the output and input variables.

Note that that calculation of $\underline{\Delta} u_k(t)$ is equivalent to calculation of $\hat{u}_k(t)$ since $\bar{u}_{k-1}(t)$ is known. After the batch run, $\hat{\mathbf{u}}_k$ is available and $\Delta\bar{\mathbf{u}}_{k+1}$ can be computed by QILC.

ILC in the proposed technique is based on the updating rule in (7). Hence the learning is basically unaffected by the run-independent disturbance even under the active feedback action by the predictive control.

### 3.4 Tuning Guideline

It is thought that tuning through the noise covariance matrices is more transparent than through the quadratic cost weighting matrices in the proposed controller. In fact, it is well known that the covariance matrices and the weighting matrices have symmetric effects on the controller performance (Lee *et al.*, 2000). In the constrained case, however, the input penalty in the quadratic cost loses its meaning when the input is stuck on the constraint boundary, while the Kalman gain can still function as an effective tuning knob. In this respect, it is suggested to fix $\mathbf{Q}$ and $\Gamma$ as scaling matrices and set $\mathbf{R}$ and $\Lambda$ as small positive definite matrices only for regularization, and to use $\mathbf{R}_{\Delta w}$, $\mathbf{R}_v$ (determined by $\mathbf{R}_m$ according to (19), and $\mathbf{R}_n$ as the tuning factors. Their effects on the respective controllers are rather obvious from the nature of the Kalman filter. If $\mathbf{R}_{\Delta w}$ is given to be large in relation to $\mathbf{R}_v + \mathbf{R}_n$ (note that $\mathbf{v}_k + \mathbf{n}_k$ in (13) is equivalent to $\mathbf{v}_k$ in (4)), the run-independent disturbance is weakly filtered and has a strong influence on the ILC performance. Therefore such a choice should be made when the run-independent disturbance is not large. By the similar reasoning, the behavior of RFC is determined by the ratio $\mathbf{R}_{\Delta w} + \mathbf{R}_v$ to $\mathbf{R}_n$. When the ratio is large, real-time control is enhanced.

## 4. NUMERICAL ILLUSTRATIONS

The performance of the proposed algorithm is examined for two numerical processes, a linear single-input single-output (SISO) batch process and a semi-batch reactor system with series-parallel reactions (Chin *et al.*, 2000).

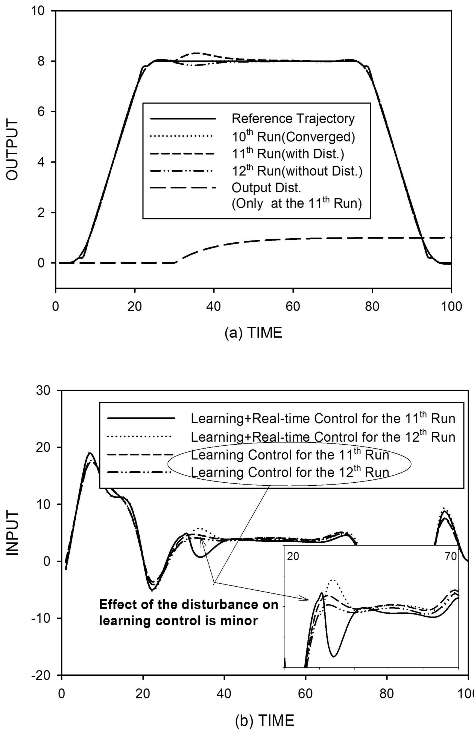


Fig. 1. (a) Controlled variables (b) Manipulated variables for the 11$^{\text{th}}$ and 12$^{\text{th}}$ runs under a run-independent disturbance.

### 4.1 Linear SISO System

The plant and nominal model are the sampled-data systems (sampling interval=1) of the following transfer functions, respectively:

$$G^p = \frac{2.5}{300s^2 + 35s + 1}, \; G^m = \frac{1.5}{270s^2 + 33s + 1} \quad (28)$$

It is assumed that a run-independent disturbance, a step response of a low pass filter, enters from $t = 31$ only at the 11$^{\text{th}}$ run.

*4.1.1. Results and Discussion* Figure 1 shows the performance of the proposed control technique. It can be seen that $\bar{u}_{12}(t)$ is only slightly influenced by the run-independent disturbance although the disturbance is aggressively rejected by the real-time predictive control yielding largely changing $u_{11}(t)$. The consequence is that the learning process can be continued with only a minor interrupt. Such a performance cannot be achieved by the existing RFC-ILC techniques.

### 4.2 Semi-Batch Reactor

The jacketed semi-batch reactor model in (Chin *et al.*, 2000), where the following reaction takes place

$$A + B \xrightarrow{k_1} C$$

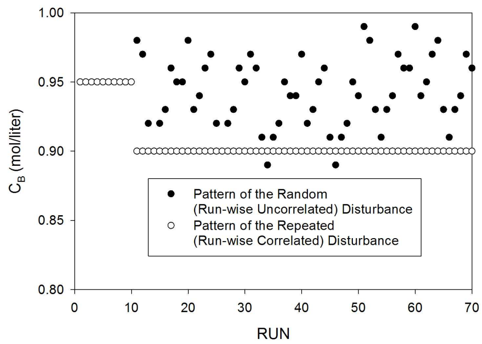


Fig. 2. Two different disturbance scenarios.

$$B + C \xrightarrow{k_2} D \quad (29)$$

is revisited in this example. $A$ is charged initially and the heat-up is followed until $B$ starts to be fed at $t = 31$ min. The reaction commences at this point and continues until the batch terminal time of $t_f = 100$ min. During this period, $A$ is sampled at every 10 min for concentration measurement. The desired product is $C$ and the main objective is to maintain the final yield of $C$ at 42 mol. We considered two manipulated variables: jacket temperature $T_j(t)$ and flow rate of $B$, $Q_B(t)$ where the following constraints are imposed:

$$20^{\circ}\text{C} \le T_j(t) \le 45^{\circ}\text{C} \quad (30)$$
$$0.5 \text{ (liter/min)} \le Q_B(t) \le 1.5 \text{ (liter/min)}$$

The sample time for control was chosen to be 1 min. In this example, two different disturbance patterns in $C_B$ (concentration of feed $B$) are assumed as shown in Figure 2. In the first case, $C_B$ changes randomly around 0.95 (mol/l) from the $11^{st}$ run. In the second case,$C_B$ is decreased from 0.95 (mol/l) to 0.9 (mol/l) at the $11^{th}$ run and kept at 0.9 (mol/l) thereafter.

*4.2.1. Results and Discussion* Figure 3 shows a result for the first disturbance scenario (run-independent disturbance). One can see that $\bar{u}_k(t)$ is almost uninfluenced by the disturbance and remains on the already-converged input trajectories. In Figure 4, a result for the repeated disturbance is given. It can be observed that $\bar{u}_k(t)$ changes and converges to new profiles that can perfect reject the repeated disturbance. Although not shown here due to limited space, the performance of quality (final yield of $C$) control as well temperature tracking control were found to be quite satisfactory for both disturbance scenarios.

## 5. CONCLUSIONS

We proposed a new learning control methodology to handle two different types of disturbances, run-wise persisting and uncorrelated disturbances

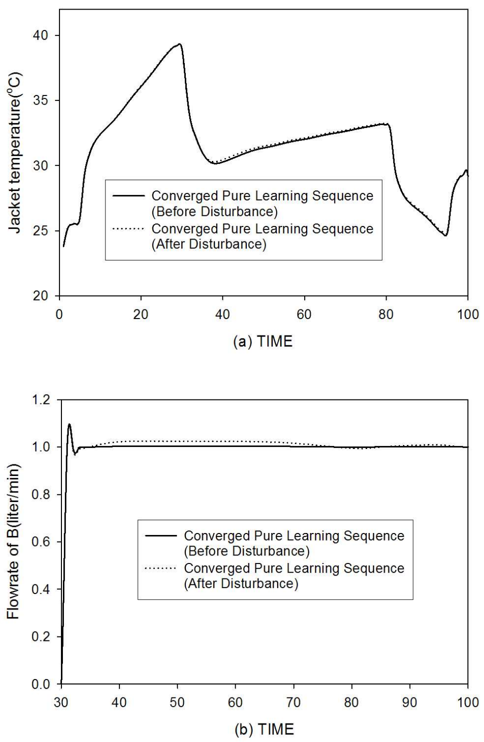


Fig. 3. (a) Jacket temperature (b) feed flowrate of $B$ under the run-independent disturbance.

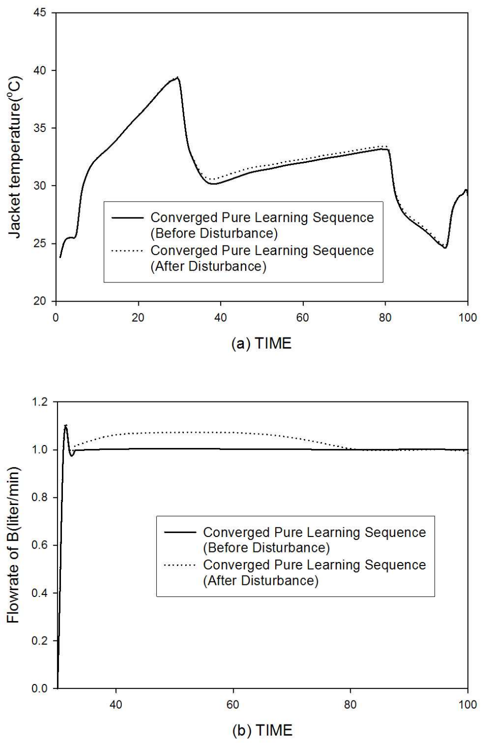


Fig. 4. (a) Jacket temperature (b) feed flowrate of $B$ under the run-wise repeated disturbance.

separately with a simple tuning guideline. The present BMPC or other combined iterative learning control (ILC) and real-time feedback control (RFC) methods share a problem that an excessive input movement by a large real-time disturbance is transferred to the next run as a feedforward input signal, which leads to deterioration of the learning performance. To solve this problem, we have devised a two-stage algorithm, RFC during a batch run to fight against run-wise uncorrelated disturbance and then ILC after the batch run for input update only by the run-wise persisting disturbance. The proposed control algorithm is based on the earlier study on BMPC and QILC for the inheritance of their advantages.

Numerical studies reveal that the proposed technique works as anticipated overcoming the problem of existing batch control methods.

To the authors' knowledge, the present paper is the first achievement that correctly deals with the disturbance rejection problem in batch process control. The two-stage technique based on QILC and BMPC has been given as a prototypical technique to realize the idea.

## REFERENCES


Arimoto, S., S. Kawamura and F. Miyazaki (1984). Bettering operation of robots by learning. *J. Robotic Syst.* **1(1)**, 123.

Bien, Z. and K. M. Huh (1989). Higher-order iterative learning control algorithm. *IEE Proc. Part D* **136(3)**, 105–112.

Chin, I. S., K. S. Lee and J. H. Lee (2000). A technique for integrated quality control, profile control and constraint handling for batch processes. *Ind. & Eng. Chem.* **39**, 693–705.

Good, R., J. Hahn, T. Edison and S. J. Qin (2002). A run-to-run control approach to drug dosage adjustment for long-term drug therapy. *Submitted to Control Engineering Practice.*

Lee, K. S. and J. H. Lee (1997). Model predictive control for nonlinear batch processes with asymptotically perfect tracking. *Com. & Chem. Eng.* **21**, 873–879.

Lee, K. S., J. H. Lee and W. C. Kim (2000). Model-based iterative learning control with a quadratic criterion for time-varying linear systems. *Automatica* **36**, 641–657.

Lee, K. S., J. Lee, I. S. Chin, J. Choi and J. H. Lee (2001). Control of wafer temperature uniformity in rapid thermal processing using an optimal iterative learning control technique. *Ind. Eng. Chem. Res.* **40**, 1661–1672.

Lee, K. S., S. H. Bang and K. S. Chang (1994). Feedback-assisted iterative learning control based on an inverse process model. *Journal of Process Control* **4**, 77.

Qin, S. J., G. Scheid and T. Riley (2002). Run-to-run optimization, monitoring, and control on a rapid thermal processor. *Submitted to JVST-B.*