Comparative Studies of Iterative Learning Control Schemes for A Batch Chemical Process *

Y. Chen, J.-X. Xu, Tong Heng Lee and S. Yamamoto †

Department of Electrical Engineering,

National University of Singapore,

10 Kent Ridge Crescent, Singapore 119260

Speaker: Mr. Chen Yangquan

`E-mail & URL : elecyq@nus.sg; http://www.ee.nus.sg/~yangquan/project.html`

*This work was supported in part by NUS under RP-3950628.

†Visiting Professor. On leave from Kogakuin University, Japan. E-mail: `yamamoto@ee.nus.sg`

Overview

1 Motivations

Could the `BATCH` property of *a batch process* be utilized for a better control performance?

What is iterative learning control (ILC)

learning is a **bridge**
between **knowledge** *and* **experience**

- **knowledge**: modeling, environment, and related uncertainties
- **experience**: repetitive operations, previous { control efforts, resulting errors}

OBJECTIVE of ILC: to utilize the system repetitions as the *experience* to improve the system control performance even under incomplete *knowledge* of the system.

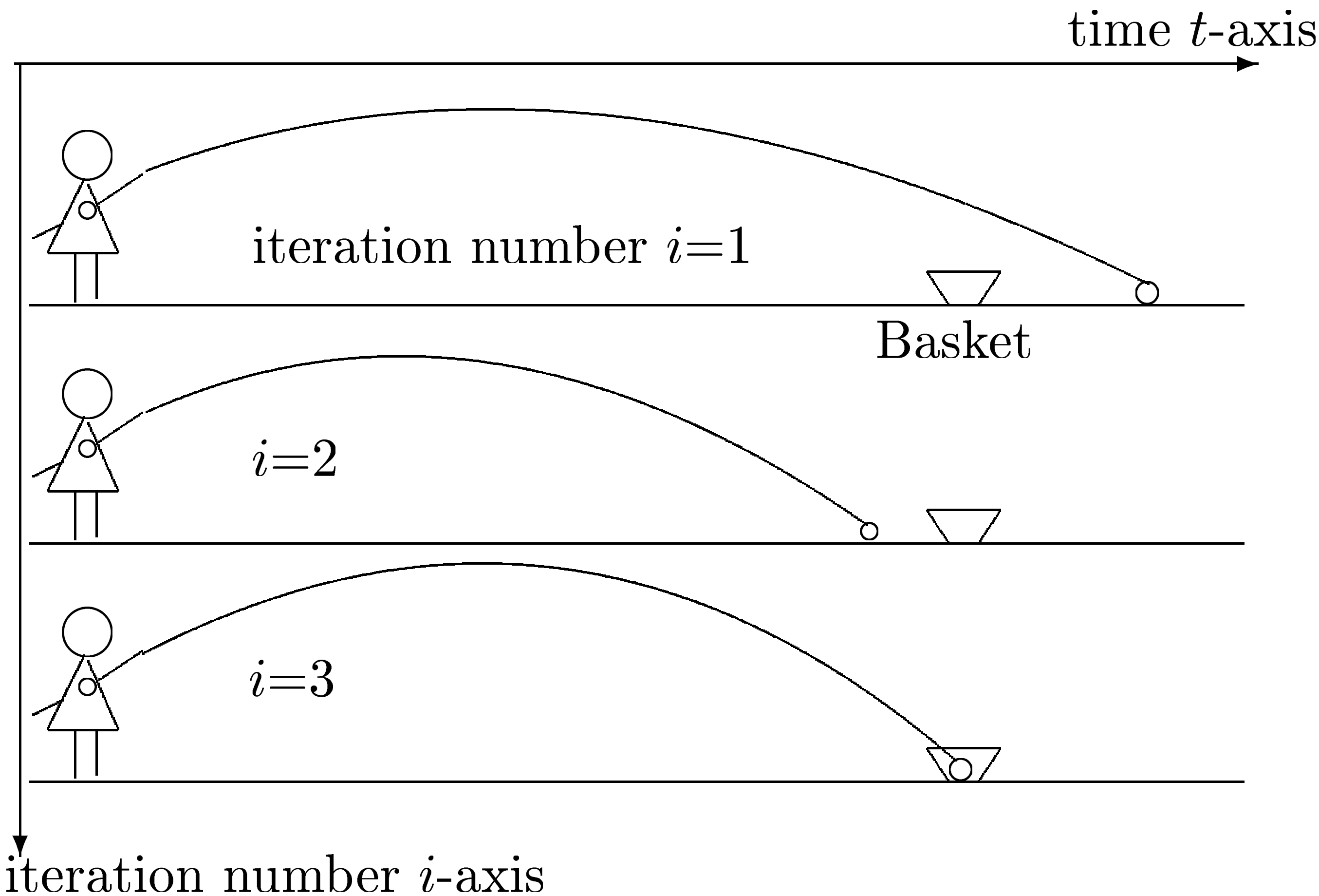


Figure 1: Illustration of *Iterative Learning*

Postulates(Arimoto* et al. *1984):

P1. Every trial (pass, cycle, batch, iteration, repetition) ends in a fixed time of duration $T > 0$.

P2. A desired output $y_d(t)$ is given *a priori* over $[0, T]$.

P3. Repetition of the initial setting. $x_k(0) = x^0$, for $k = 1, 2, \cdots$.

P4. Invariance of the system dynamics.

P5. $y_k(t)$ measurable. Tracking error $e_k(t) = y_d(t) - y_k(t)$ can be utilized to form $u_{k+1}(t)$.

P6. Invertability. For a given $y_d(t)$, $\exists$ a unique $u_d(t)$ that drives the system to produce the $y_d(t)$.

Simple D-type ILC Scheme
(Arimoto* et al. *1984):

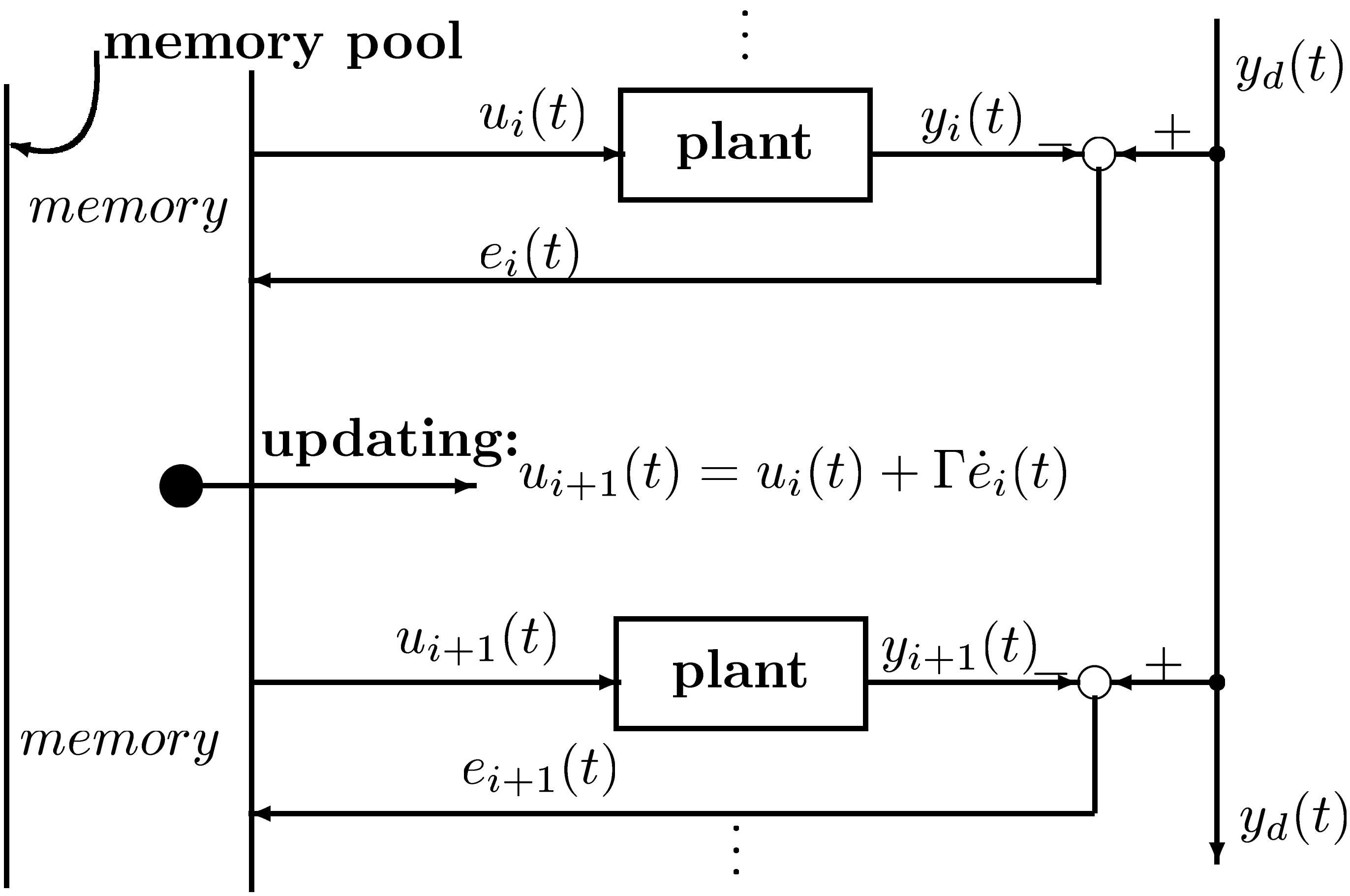


Figure 2: Block-Diagram of *Iterative Learning Control*

Control Task: to find a **recursive** control law

$$\begin{aligned} u_{k+1}(t) &= \mathbf{F}(u_k(t), e_k(t), u_{k-1}(t), e_{k-1}(t), \cdots, \\ & \quad u_{k-N+1}(t), e_{k-N+1}(t), \quad e_{k+1}(t)) \quad (1) \end{aligned}$$

and a function norm $\|\cdot\|$ such that $\|e_k(t)\|$ vanishes as k tends to infinity.

- High-order Scheme;
- D-type or P-type;
- $\cdots$

Problems to consider:

- convergence
- robustness
- $\cdots$

Stability (or convergence):

For an ILC updating law (1), to guarantee that for a certain function norm of *tracking error* $\|e(\cdot)\|$

$$\|e_{k+1}\| \leq \|e_k\|, \text{for } k = 1, 2, \cdots \tag{2}$$

or, in a stronger sense,

$$\|e_{k+1}\| \leq \theta\|e_k\|, \text{for } k = 1, 2, \cdots \tag{3}$$

with a constant $\theta \in (0, 1)$.

Norms used:

$$\|f\| = \max_{1\leq i\leq n} \mid f_i \mid,$$

$$\|G\| = \max_{1\leq i\leq m} (\sum_{j=1}^{n} \mid g_{i,j} \mid),$$

$$\|h(t)\|_\lambda = \sup_{t\in[0,N]} e^{-\lambda t} \|h(t)\|,$$

where $f = [f_1, \cdots, f_n]^T$ is a vector, $G = [g_{i,j}]_{m\times n}$ is a matrix and $h(t)\,(t \in [0, N])$ is a real function.

Robustness: relaxed **P1** - **P6**.

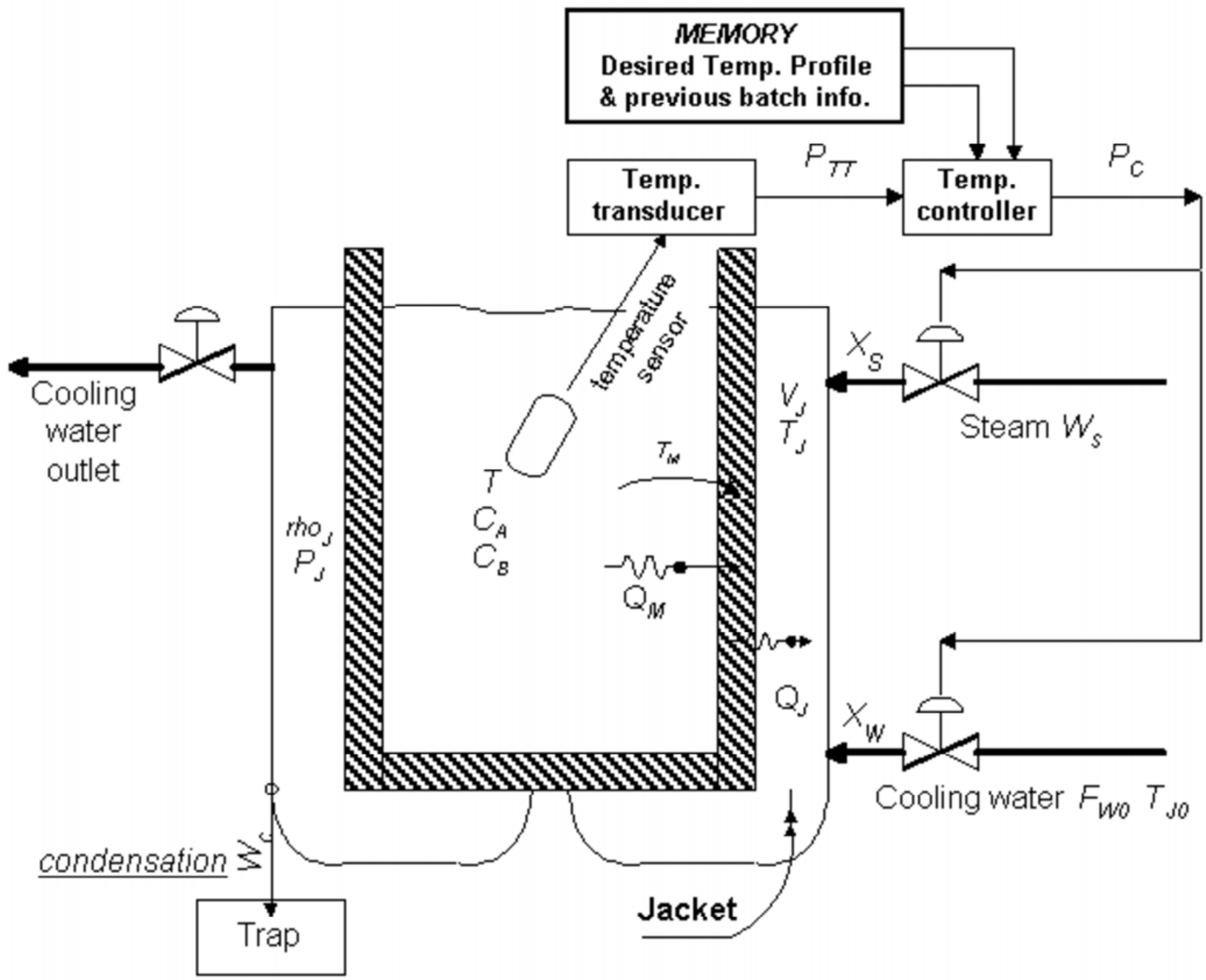


Figure 3: A Chemical Batch Reactor

2 Batch Reactor Model

Block diagram:

Model:

The reaction equations are given as follows:

$$\left\{\begin{array}{lcl} \frac{\mathrm{d}c}{\mathrm{d}t} & = & -a_1 k_d c + Q' \\ \frac{\mathrm{d}x_M}{\mathrm{d}t} & = & -a_2 k_p \sqrt{\frac{2\eta k_d}{k_t}}\sqrt{c}\, x_M \\ \frac{\mathrm{d}p}{\mathrm{d}t} & = & a_3 \eta k_d c \\ \frac{\mathrm{d}T}{\mathrm{d}t} & = & H_r \frac{\mathrm{d}x_M}{\mathrm{d}t} + Q \end{array}\right. \tag{4}$$

where c , x_M and p are the concentrations of initiator, monomer and polymer respectively; T is the temperature inside the reactor (°K). Q' is assumed to be 0 which means that no additional initiator is added during the reaction, i.e., all reactants have been filled in the reactor at the beginning. Reaction rate constants $k_i, i \in \{d, p, t\}$ are functions of T where d, p, t represent the phases of beginning, growing and stopping.

$$k_i = k_{i0} \exp(-\frac{E_i}{RT}), \quad i = d, p, t. \tag{5}$$

Challenges:

- track a given trajectory (pattern) in a finite time (batch duration) interval.

- in a point-wise manner

Iterative Learning Control,

- utilize the system's repetition to compensate or reject uncertainties and disturbances and hence able to track the prescribed trajectory in a finite interval.
- the control efforts of the current batch incorporate the control efforts and tracking errors of the previous batch.

3 Learning Control Schemes

- D-type ILC
- High-order ILC
- P-type plus feedback controller
- P-type CITE: current iteration tracking error

D-type Iterative Learning Scheme

$$Q_{i+1} = Q_i + K_d \dot{e}_i(t), \quad \dot{e}_i(t) \triangleq \dot{T}_d - \dot{T}_i \qquad (6)$$

where K_d is the learning gain which is to be properly chosen such that $e_i \to 0$ as $i \to \infty$. The convergence condition is that

$$\|1 - CBK_d\| < 1 \qquad (7)$$

where B and C are input distribution matrix and output matrix respectively. In (4), the system input is Q and output is T, hence $CB = 1$. K_d should be selected to satisfy the condition $\mid 1 - K_d \mid < 1$.

High-order Iterative Learning Scheme

WHY HIGH-ORDER?

The First-order ILC

Consider Suguru Arimoto's original ILC updating law

$$u_{k+1}(t) = u_k(t) + \Gamma \dot{e}_k(t) = \Gamma \sum_{j=0}^{k} \dot{e}_j(t)$$

along the ILC iteration number k direction, which is in an integral (I) controller form.

The Third-order ILC

Considering the PID form

$$u_{k+1}(t) = k_P \dot{e}_k(t) + k_I \sum_{j=0}^{k} \dot{e}_j(t) + k_D(\dot{e}_k(t) - \dot{e}_{k-1}(t))$$

will result in the following form of the ILC updating law

$$u_{k+1}(t) = u_k(t) + \Gamma \dot{e}_k(t) + \Gamma_1 \dot{e}_{k-1}(t) + \Gamma_2 \dot{e}_{k-2}(t).$$

where $\Gamma = k_P + k_I + k_D$, $\Gamma_1 = -k_P - 2k_D$ and $\Gamma_2 = k_D$. This is a high-order iterative learning controller.

WE CAN SAY:

High-order ILC can be better than the first-order ILC, i.e., better ILC convergence performance can be achieved by High-order ILC.

In general, an N-th order D-type ILC updating law is

$$Q_{i+1} = Q_i + \sum_{j=1}^{N} K_{d_j} \dot{e}_{i-j+1}(t) \tag{8}$$

where the learning gains should satisfy that the roots of (9) are inside the unit circle.

$$(1 - CBK_{d_1})z^{-1} - \sum_{j=2}^{N} CBK_{d_j} z^{-j} = 0 \tag{9}$$

where z is one step shifting operator. A sufficient condition is given by

$$\mid 1 - CBK_{d_1} \mid + \sum_{j=2}^{N} \mid CBK_{d_j} \mid < 1. \tag{10}$$

P-type Iterative Learning Feedback Control

$$\left\{\begin{array}{rcl} Q_i(t) & = & Q_i^{ff}(t) + Q_i^{fb}(t) \\ Q_i^{fb}(t) & = & K_p e_i(t) \\ Q_i^{ff}(t) & = & Q_{i-1}(t) + K_{pl} e_{i-1}(t) \\ & = & Q_i^{ff}(t) + (K_p + K_{pl}) e_{i-1}(t) \end{array}\right. . \quad (11)$$

blank page

P-type ILC with CITE

Consider the PI-controller in the ILC iteration direction as follows:

$$Q_i(t) = k'_I \sum_{j=0}^{i} e_j(t) + k'_P e_i(t). \qquad (12)$$

Writing (12) in an iterative form, we have

$$\begin{aligned} Q_i(t) &= Q_{i-1}(t) + k'_I e_i(t) + k'_P(e_i(t) - e_{i-1}(t)) \\ &= Q_{i-1}(t) + K_p e_i(t) + K_{pl} e_{i-1}(t) \qquad (13) \end{aligned}$$

where $K_p = k'_I + k'_P, K_{pl} = -k'_P$. Updating law (13) is called as the *ILC with Current Iteration Tracking Error (CITE)*.

4 Simulation Results

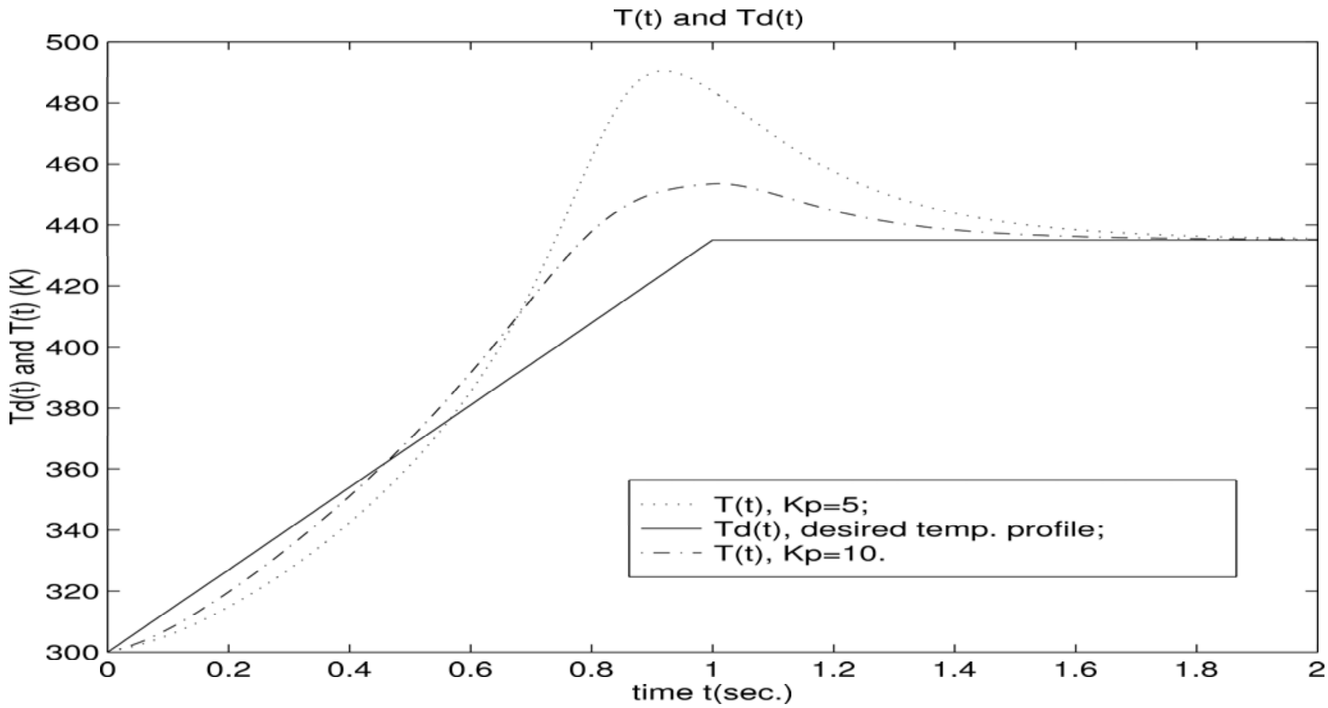
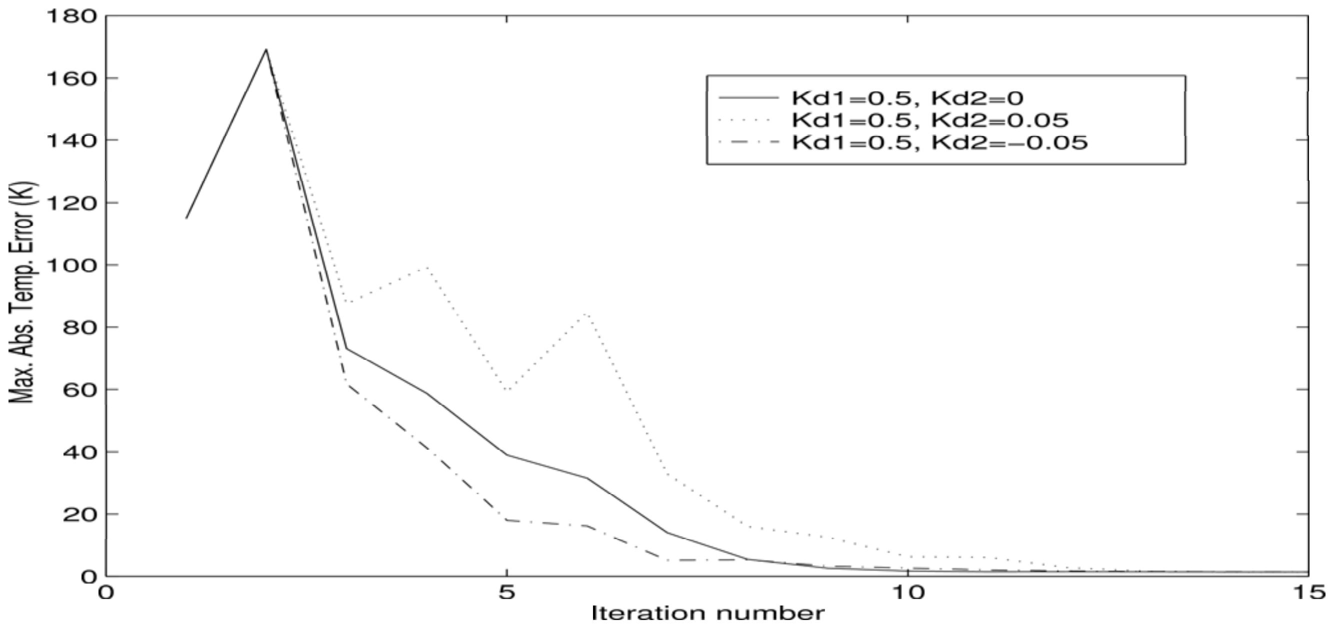
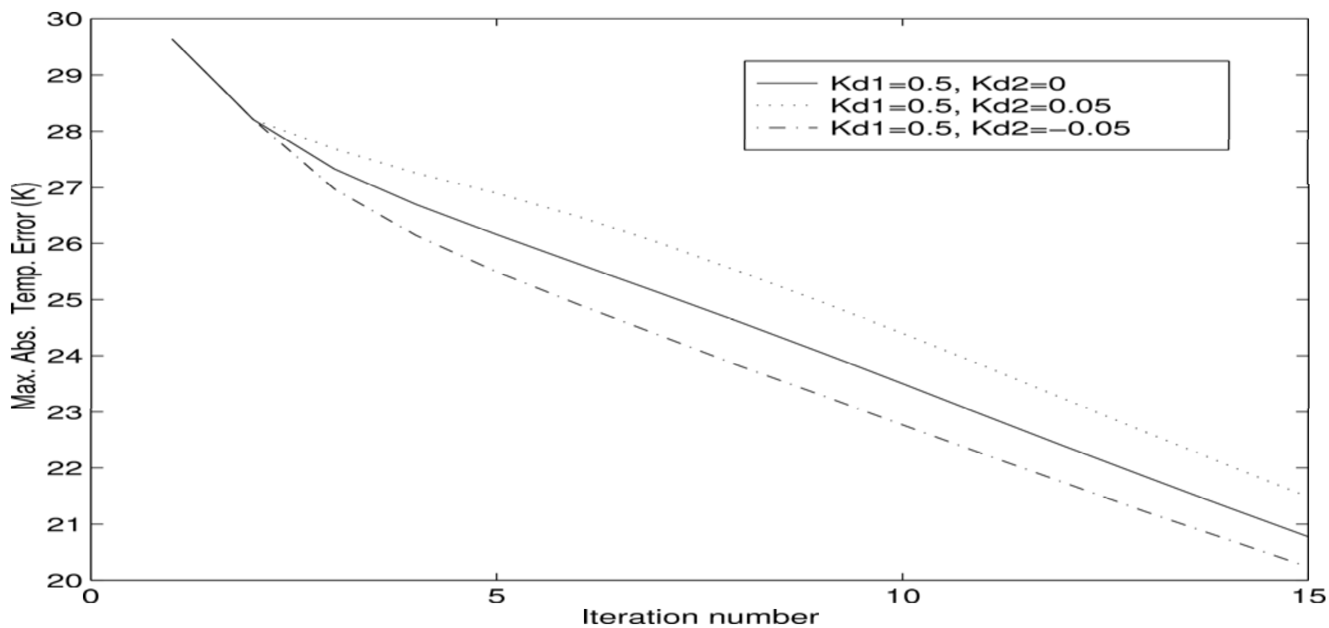


Figure 4: Responses for simple P feedback controllers



(a) $K_p = 0.0$ (D-type ILC only) (D-I scheme)



(b) $K_p(t) = 20$ (PD-I scheme)

Figure 5: Convergence comparisons for D-type ILC schemes

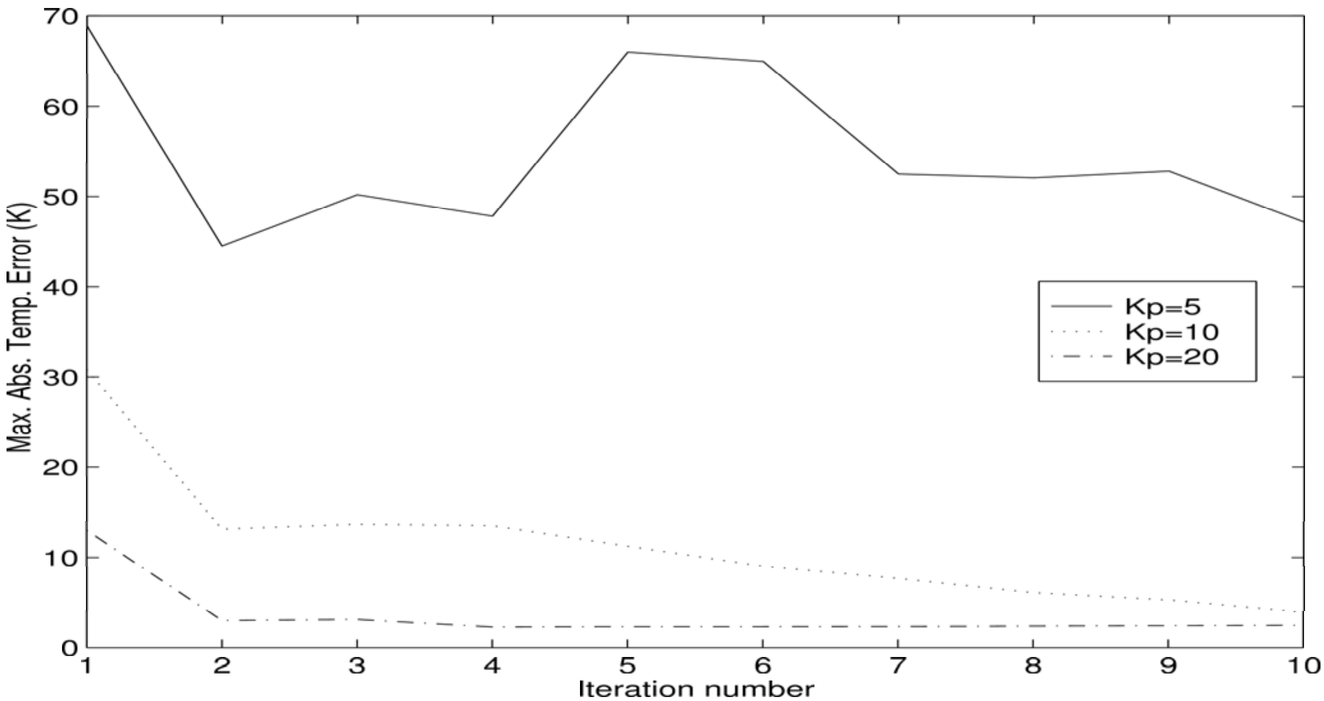


Figure 6: Convergence comparisons for P-I type (ILC+CITE) schemes

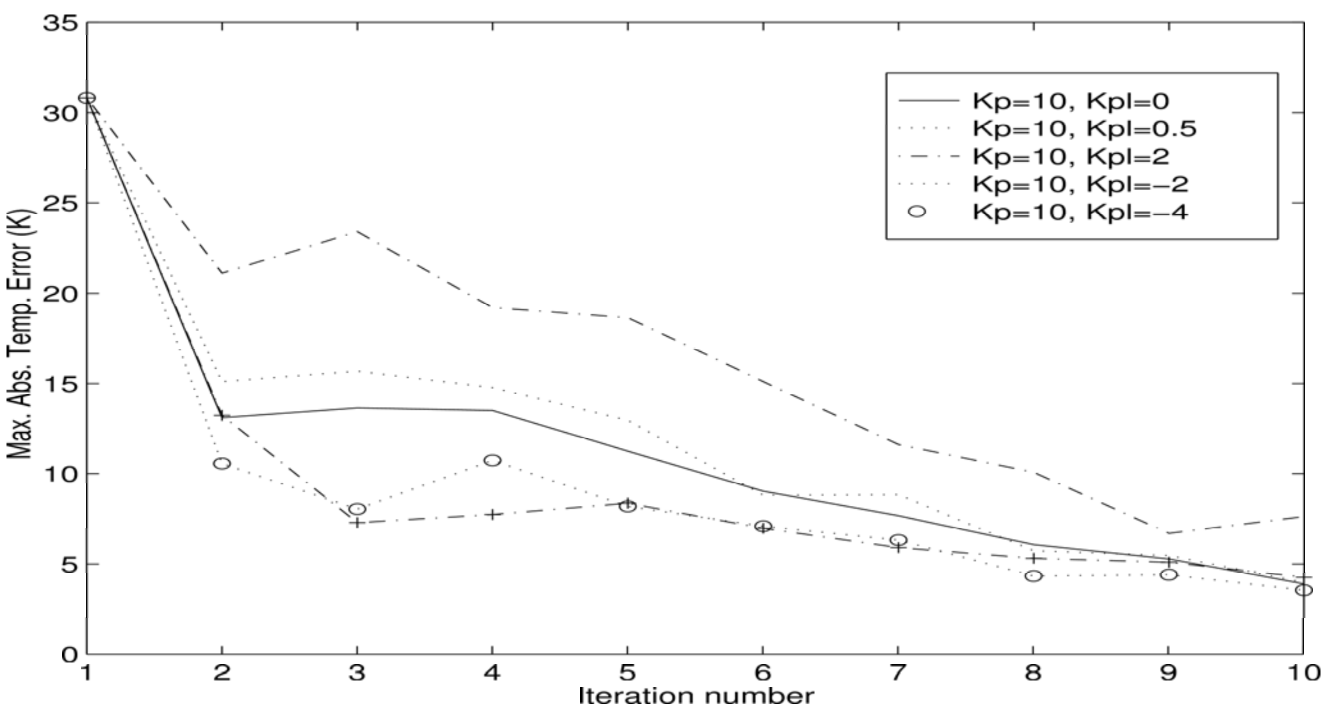


Figure 7: Convergence comparisons for P-PI type (ILC+CITE) schemes

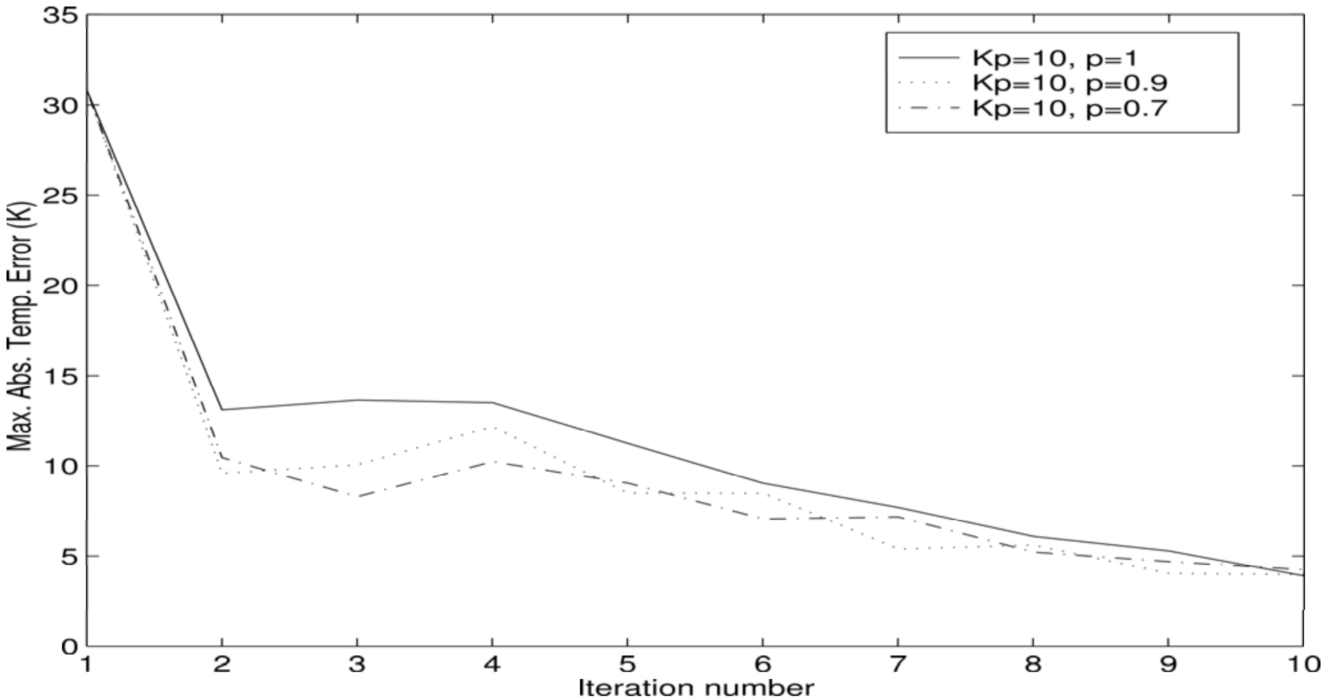


Figure 8: Convergence comparisons for P-type scheme

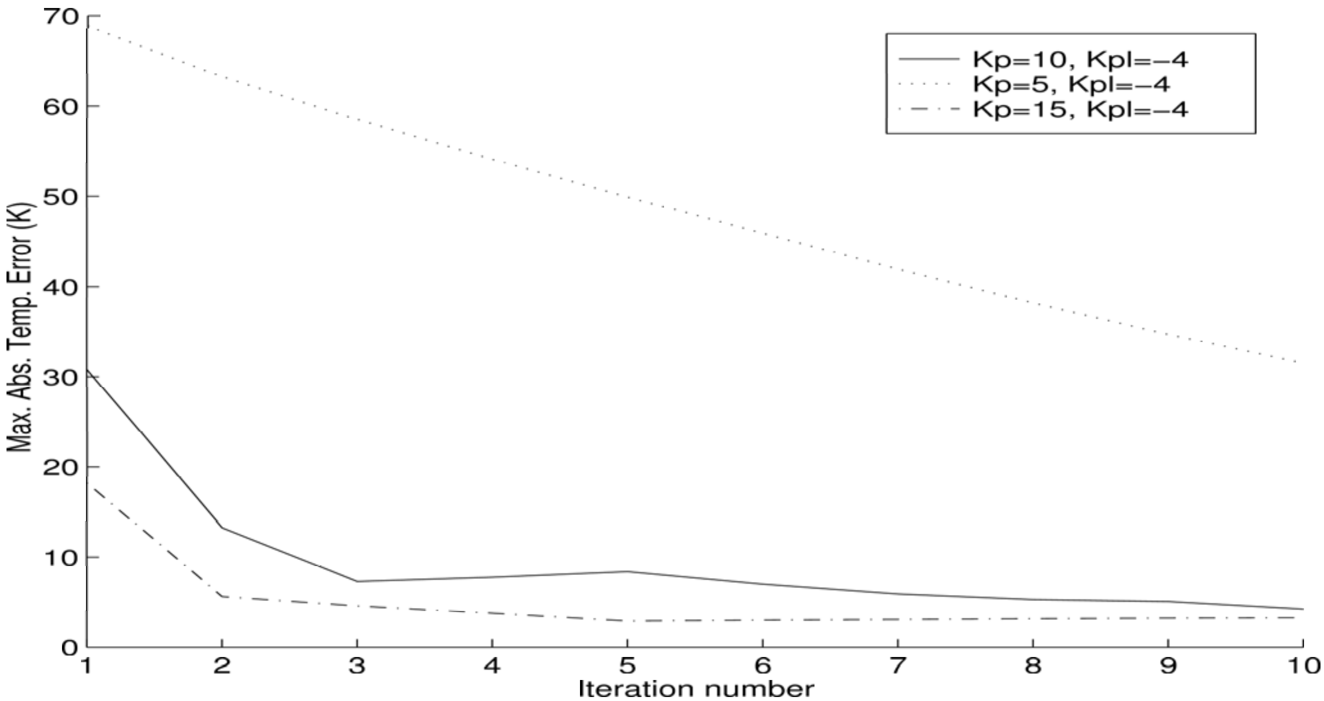


Figure 9: Convergence comparisons for P-PI type (ILC+CITE) schemes

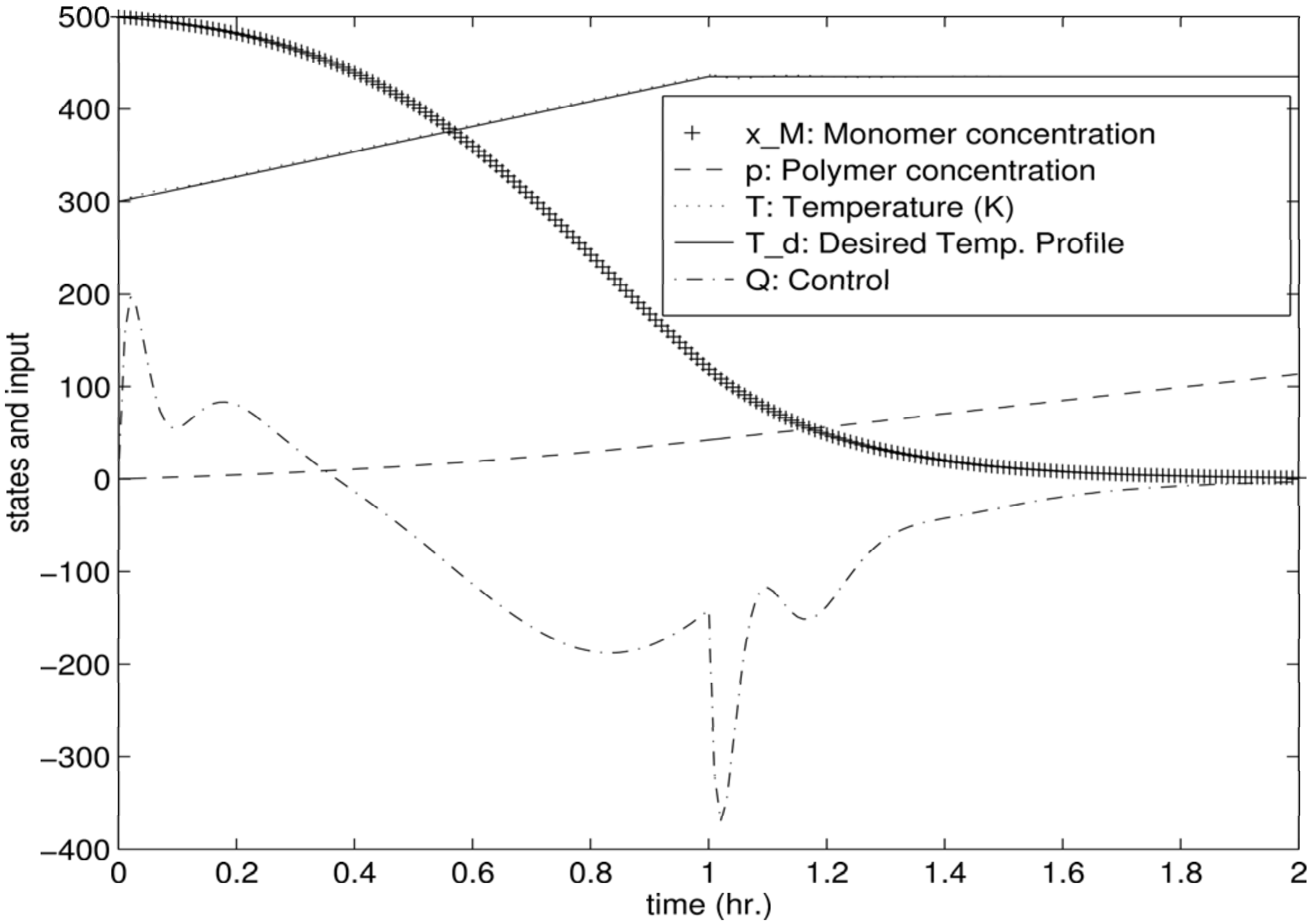


Figure 10: Converged system states and input (at the 10th ILC iteration)

5 Observations

- Suitably designed feedback controller is helpful to ILC convergence;
- Higher-order ILC can be better than conventionally used first-order one;
- P-type is effective.

Among the proposed schemes, the CITE P-type scheme is the most preferable due to its simplicity and effectiveness.

6 Concluding Remarks

- Iterative Learning Control is shown to be applicable to Batch Process Control.
- Simulations performed for illustration.

Future work:

- validation using a pilot plant.
- exploring various learning gain-tuning methods.
- convergence analysis under practical limits.
- $\cdots$

Thank you!

Comments Please ...

Don't forget to visit our **ILC-related web page:**

http://www.ee.nus.sg/~yangquan/project.html