ITERATIVE LEARNING CONTROL WITH INPUT SHIFT

C. Welz, B. Srinivasan and D. Bonvin

Laboratoire d'Automatique, École Polytechnique Fédérale de Lausanne
CH-1015 Lausanne, Switzerland

Abstract: Iterative learning control (ILC) is a technique to realize system inversion in a run-to-run manner. Though most of the techniques presented in the literature consider zero tracking error between the desired and achieved outputs, perfect inversion is often not feasible and in many cases not even desirable. Approximate inversion with good convergence and robustness properties (at the cost of a nonzero tracking error) has been proposed by using a forgetting factor on the input of the previous run. In this paper, approximate inversion is achieved by shifting the input of the previous run backwards in time. In addition, anticipatory ILC and current cycle feedback are used. The advantage of input shift over the use of a forgetting factor is that, when the reference trajectory is constant and the system stable, the tracking error decreases with run time. The proposed scheme is illustrated in simulation on a batch distillation system.

Keywords: Batch processes, Iterative learning control, Run-to-run control, Batch distillation.

1. INTRODUCTION

Iterative learning control (ILC) has been developed in order to improve the tracking performance of repetitive processes (Arimoto *et al.*, 1984). This is accomplished by utilizing the tracking error of the previous run to update the input of the current run. ILC has been successfully applied in robotics (Arimoto *et al.*, 1984; Kuc *et al.*, 1991; Wang, 2000) and in batch chemical processing (Lee *et al.*, 1996; Choi *et al.*, 1996; Lee *et al.*, 1999; Xu *et al.*, 2001).

The various ILC methods that have been reported in the literature can be classified into two main categories: (i) schemes with zero tracking error (Kuc *et al.*, 1991; Xu *et al.*, 1995), and (ii) those with nonzero tracking error (Arimoto *et al.*, 1990; Chien and Liu, 1994). System inversion is realized in the former, while the inversion is only approximated in the latter. Though the nonzero error techniques have the disadvantage of a residual error, they typically show better convergence and robustness properties (Arimoto *et al.*, 1990).

The standard technique for approximate inversion is to have a forgetting factor in the input update (Arimoto *et al.*, 1990). This causes the tracking error to be nonzero over the entire interval. However, the main difficulty with the feasibility of system inversion arises during the first part of the trajectory due to unmatched or varying initial conditions (Lee and Bien, 1991; Heinzinger *et al.*, 1992). After a certain catch-up time, trajectory following is relatively easy. Thus, the idea is to allow a nonzero tracking error early in the run and have the error decrease progressively with the run time t. This can be achieved with an *input* shift. The idea of a shift comes from anticipatory ILC techniques that use a shift in the *error* of the previous run (Park *et al.*, 1998; Wang, 2000).

The first ILC approaches used only the error from the previous run and thus could only handle repetitive disturbances. The addition of current cycle feedback has been proposed to handle non-repetitive disturbances (Xu *et al.*, 1995). The general ILC scheme proposed in this paper will include the following aspects: (i) input shift, (ii) error shift, and (iii) current cycle feedback.

The paper is organized as follows. Section 2 reviews ILC schemes and studies convergence conditions and the residual tracking error. In Section 3, a novel ILC scheme that shifts both the error and the feedforward trajectory of the previous cycle is proposed. A batch distillation system is presented as an application in Section 4, and conclusions are drawn in Section 5.

2. GENERAL ILC FORMULATION

2.1 Problem Formulation

In this section, a general ILC law is formulated in order to classify the common ILC schemes. For simplicity of notation, the following single input, single output system is considered in operator notation:

$$y_k = Gu_k + \bar{y} \tag{1}$$

where u_k and y_k represent the input and output trajectories in the run k, G is the operator representing the system, and $\bar{y}$ the response to initial conditions. Note that the input trajectory u_k and the output trajectory y_k are defined for the run time $t \in [0, t_f]$, where t_f is the final time of the run.

The idea of ILC is to improve trajectory tracking for repetitive processes by utilizing the previous cycle tracking error. The ILC update law for the input u_{k+1} in the next run is given as

$$u_{k+1} = Au_k + Be_k, \qquad e_k = y_{ref} - y_k \tag{2}$$

where y_{ref} is the reference trajectory to be tracked and A and B are operators applied to the previous cycle input and tracking error trajectories, respectively.

2.2 Convergence

An ILC law is convergent if the following limits exist:

$$\lim_{k\to\infty} y_k = y_\infty \quad \text{and} \quad \lim_{k\to\infty} u_k = u_\infty \tag{3}$$

Since the converged output trajectory y_∞ is not necessarily equal to the reference trajectory y_{ref}, the residual error can be nonzero.

One way of ensuring convergence is for the relation between u_k and u_{k+1} to be a contraction mapping in some appropriate norm $|\cdot|$:

$$|u_{k+1}| \leq \rho |u_k|, \qquad 0 \leq \rho < 1 \tag{4}$$

Using (1) in (2) gives:

$$u_{k+1} = (A - BG)\, u_k + B(y_{ref} - \bar{y}) \tag{5}$$

Note that $(y_{ref} - \bar{y})$ does not represent an error term as in (2), but the difference between two exogeneous signals, the reference trajectory and the response to initial conditions. Also, contrary to (2) where e_k depends on u_k, $(y_{ref} - \bar{y})$ is independent of u_k. Since the convergence of the algorithm depends on the homogeneous part of (5), the condition for convergence is:

$$|A - BG| < 1 \tag{6}$$

2.3 Residual Error

If the input trajectory converges to $u_k = u_{k+1} = u_\infty$, (5) gives:

$$u_\infty = (I - A + BG)^{-1} B(y_{ref} - \bar{y}) \tag{7}$$

The residual tracking error is then given by:

$$\begin{aligned} e_\infty &= y_{ref} - (Gu_\infty + \bar{y}) \\ &= \left(I - G\,(I - A + BG)^{-1} B\right)(y_{ref} - \bar{y}) \end{aligned} \tag{8}$$

Multiplying both sides with B and rearranging, this equation can be rewritten as:

$$Be_\infty = (I - A)\,(I - A + BG)^{-1} B(y_{ref} - \bar{y}) \tag{9}$$

Note that the error term Be_∞ is zero when $A = I$ and nonzero otherwise. When B is a derivative operator and $A = I$, $\dot{e}_\infty$ will be zero, but not e_∞. In other words, integral action along the run index k is needed for zero error. Zero error implies that system inversion has been achieved iteratively. However, methods with nonzero error might show better convergence properties.

2.4 Current Cycle Feedback

In conventional ILC schemes, only the previous cycle tracking error is used to adjust the input in the current cycle. To reject perturbations during a run, the error occuring during the current cycle need to be used as well. From the point of view of convergence and error analysis, these modified schemes simply correspond to different choices of the operators A and B in (2) as will be shown next. Let

A	B	ILC type	Literature
I	kD	derivative	(Arimoto *et al.*, 1984; Heinzinger *et al.*, 1992)
I	k	proportional	(Saab, 1994)
I	$k+\bar{k}D$	proportional-derivative	(Kuc *et al.*, 1991)
I	$k\Delta$	anticipatory	(Park *et al.*, 1998; Wang, 2000)
$(I+CG)^{-1}(\bar{A}+CG)$	$(I+CG)^{-1}(\bar{B}+(I-\bar{A})C)$	current cycle feedback	(Xu *et al.*, 1995; Jang *et al.*, 1995)
I	$(G^TQG+R)^{-1}G^TQ$	quadratic criterion based	(Amann *et al.*, 1996; Lee *et al.*, 2000)
βI	k	forgetting factor	(Arimoto *et al.*, 1990; Chien and Liu, 1994)
$(I+CG)^{-1}(\Delta+CG)$	$(I+CG)^{-1}(k\Delta+(I-\Delta)C)$	input shift, anticipatory, current cycle feedback	present paper

Table 1. ILC schemes with selected references (k, $\bar{k}$: proportional controllers, D: differentiation operator, Δ: shift operator, Q, R: weighting matrices, $0 \leq \beta < 1$: forgetting factor).

$$u_{k+1} = u_{k+1}^{ff} + u_{k+1}^{fb} \tag{10}$$

$$u_{k+1}^{ff} = \bar{A}u_k^{ff} + \bar{B}e_k, \quad u_{k+1}^{fb} = Ce_{k+1} \tag{11}$$

where the superscripts $(.)^{ff}$ and $(.)^{fb}$ represent the feedforward and feedback parts of the input, and $\bar{A}$, $\bar{B}$ and C are operators. Expressions (10) and (11) can be combined with (1)-(2) to give:

$$\begin{aligned} u_{k+1} = &(I+CG)^{-1}(\bar{A}(I+CG)-\bar{B}G)u_k + \\ &(I+CG)^{-1}(\bar{B}+(I-\bar{A})C)(y_{ref}-\bar{y}) \end{aligned} \tag{12}$$

which is of the form (5) with $A = (I+CG)^{-1}(\bar{A}+CG)$ and $B = (I+CG)^{-1}(\bar{B}+(I-\bar{A})C)$. Thus, the convergence condition and residual error can be analyzed using (6) and (8), respectively. Note that A and B are now functions of G due to the current cycle feedback.

2.5 Overview of ILC schemes

The various ILC schemes reported in the literature can be viewed as special cases of the general scheme presented in (5). They correspond to different choices of the operators A and B, as this was done in (12) for the current cycle feedback. Table 1 gives an overview of some ILC schemes and corresponding references.

The ILC approach first described by (Arimoto *et al.*, 1984) uses the error derivative in the ILC law, i.e. $A = I$, $B = kD$, where k is a proportional controller and D the differentiation operator. Since numerical differentiation leads to noise amplification, proportional ILC was proposed, i.e. $B = k$. However, the convergence of proportional ILC was shown for a more restrictive class than for the derivative counterpart (Saab, 1994). Also, proportional derivative controllers were proposed, $B = k+\bar{k}D$ (Kuc *et al.*, 1991). Another possibility that has been explored is a shift in the error trajectory $B = k\Delta$, where Δ is the shift operator (Wang, 2000). Since a time shift in discrete time corresponds to a differentiation, the anticipatory ILC scheme combines the robustness of derivative ILC with the low noise amplification of proportional ILC.

As discussed earlier, schemes with current cycle feedback can be cast in the above framework as well. In particular, the schemes proposed in (Xu *et al.*, 1995) and (Jang *et al.*, 1995) correspond to the choice $\bar{A} = I$, and arbitrary $\bar{B}$ and C. Note that the choice $\bar{A} = I$ results in $A = I$. Current cycle feedback is also used in ILC based on a quadratic performance criterion (Amann *et al.*, 1996).

A simple way of having a nonzero tracking error, and thus enforcing only approximate inversion, is to use a forgetting factor (Arimoto *et al.*, 1990), $A = \beta I$ with $\beta < 1$. Such a scheme has been shown to provide more robustness.

The idea of this paper is to use a non-identity operator for A so as to accept approximate inversion but improve convergence and robustness. At the same time, it is desirable to have the residual error, which could be large at the beginning of the run, get small towards the end of the run. Such a situation can be created by using $A = \Delta$ as will be shown in the next section.

3. ANTICIPATORY ILC WITH INPUT SHIFT AND CURRENT CYCLE FEEDBACK

Anticipatory ILC applies a shift to the error from the previous cycle. In the following, the use of an additional time shift of the feedforward input in order to increase robustness will be investigated. This idea was first used in (Welz *et al.*, 2004), with the same shift for the error and the input. The goal of this paper is to extend this idea and use different shifts for the error and the input.

The proposed ILC approach has three components: (i) shift of the feedforward part of the input, (ii) shift of the previous cycle error, and (iii) current cycle feedback. The iterative update law can be written as:

$$u_{k+1}(t) = u_{k+1}^{ff}(t) + k^{fb}e_{k+1}(t) \tag{13}$$

where k^{fb} is the proportional controller used in the current cycle feedback.

The feedforward part of the current input consists of shifted versions of the feedforward part of the previous input and the previous cycle tracking error:

$$u_{k+1}^{ff}(t) = u_k^{ff}(t+\delta_u) + k^{ff} e_k(t+\delta_e) \quad (14)$$

where k^{ff} is the proportional feedforward controller, δ_u the time shift of the feedforward trajectory and δ_e the time shift of the previous run error trajectory. This defines the signals $u_k^{ff}(t)$ and $e_k^{ff}(t)$ up to the times $(t_f - \delta_u)$ and $(t_f - \delta_e)$, respectively. The remaining parts of these signals are kept constant at the values $u_k^{ff}(t_f - \delta_u)$ and $e_k(t_f - \delta_e)$. In operator notation, the introduced shifts are expressed with the shift operators Δ_u and Δ_e:

$$u_{k+1}^{ff} = \Delta_u u_k^{ff} + k^{ff} \Delta_e e_k \quad (15)$$

Thus, $\bar{A} = \Delta_u$, $\bar{B} = k^{ff}\Delta_e$ and $C = k^{fb}$, and from (12), $A = (I + k^{fb}G)^{-1}(\Delta_u + k^{fb}G)$ and $B = (I + k^{fb}G)^{-1}(k^{ff}\Delta_e + (I - \Delta_u)k^{fb})$. Hence, the convergence condition follows from (6):

$$\left|(I + k^{fb}G)^{-1}\left(\Delta_u(I + k^{fb}G) - k^{ff}\Delta_e G\right)\right| < 1 \quad (16)$$

and the error term $(I - k^{fb}G)Be_\infty$ from (9):

$$\begin{aligned}(k^{ff}\Delta_e + (I - \Delta_u)k^{fb})e_\infty = (I - \Delta_u) \\ \left((I - \Delta_u)(I + k^{fb}G) + k^{ff}\Delta_e G\right)^{-1} \\ (k^{ff}\Delta_e + (I - \Delta_u)k^{fb})(y_{ref} - \bar{y}) \quad (17)\end{aligned}$$

It is seen that the residual error is zero if $\Delta_u = I$ (no shift of the feedforward input). However, the main interest for using an input shift, instead of a scalar $\beta < 1$, is the possibility to make $e_\infty(t_f)$ close to zero. In other words, the error can be shaped within the run.

The error term has the form $Se_\infty = (I - \Delta_u)T(y_{ref} - \bar{y})$, where S and T are appropriate operators. The term $(I - \Delta_u)$ corresponds to a differentiation since it approximates the derivative using forward difference. If the operator T is stable, then the trajectory $T(y_{ref} - \bar{y})$ will approach a constant value with increasing run time t. Therefore, its differentiation gives a zero error. However, since the batch time t_f is finite, it cannot be guaranteed that the error will indeed be zero at t_f.

Such an error shaping is particularly interesting when the initial conditions do not correspond to the desired reference trajectory. Without direct transmission from input to output in System (1), the initial error cannot be reduced by proportional ILC. Instead, the initial error add up from run to run, and convergence cannot be guaranteed. Direct transmission can be obtained by derivative ILC (Heinzinger *et al.*, 1992), so that the initial error is reduced from run to run. For anticipatory ILC, Chen and Wen (1999) have proposed initial state learning to compensate the effect of differences in initial conditions. On the other hand, Saab (1994) has shown convergence with the use of a forgetting factor. With the input shift presented in this paper, an initial error can be tolerated since time is provided for the system to catch up with the desired trajectory.

4. APPLICATION TO A BATCH DISTILLATION SYSTEM

4.1 Problem formulation

A binary batch distillation system is utilized to illustrate in simulation the developments of the previous sections. This example has been described in Welz *et al.* (2004), but the main features are recalled here for completeness.

Under typical assumptions for simple distillation models, a column with p equilibrium stages is considered. Writing molar balance equations for the holdup in the reboiler and for the liquid on the various stages and in the condenser, the following model of order $(p+2)$ is obtained:

$$\dot{M}_1 = -(1-r)V \quad (18)$$

$$\dot{x}_1 = \frac{V}{M_1}\left(x_1 - y_1 + r x_2\right) \quad (19)$$

$$\dot{x}_i = \frac{V}{M_i}\left(y_{i-1} - y_i + r\left(x_{i+1} - x_i\right)\right) \quad (20)$$

$$\dot{x}_c = \frac{V}{M_c}\left(y_p - x_c\right) \quad (21)$$

with $i = 2, \ldots, p$, x_i the molar liquid fraction, y_i the molar vapor fraction, M_i the molar holdup on Stage i, V the boilup rate and D the distillate flowrate. Stage 1 refers to the reboiler and Stage p to the top of the column. M_c is the holdup in the condenser. The internal reflux ratio $r = \frac{V-D}{V}$, is considered as the manipulated variable. The vapor-liquid equilibrium relationship is:

$$y_i = \frac{\alpha x_i}{1 + (\alpha - 1)x_i}, \quad i = 1, \cdots, p \quad (22)$$

where α is the relative volatility. The model parameters and the initial conditions are given in Table 2.
The composition of the accumulated distillate, x_d, which is measured with the sampling time $T_s = 0.1\ h$, is given by:

$$x_d(t) = \frac{\sum_{i=1}^{p} x_i(t)M_i(t) - x_i(0)M_i(0)}{M_1(t) - M_1(0)} \quad (23)$$

p	10		$x_{d,des}$	0.9	kmol/kmol
α	1.6		$M_1(0)$	100	kmol
M_i	0.2	kmol	$x_1(0)$	0.5	kmol/kmol
M_c	2	kmol	$x_i(0)$	0.5	kmol/kmol
V	15	kmol/h	$x_c(0)$	0.5	kmol/kmol

Table 2. Model parameters and initial conditions, $i = 2, \cdots, p$

A batch is divided into 2 intervals of operation:

(1) Start-up phase with full reflux, $r = 1$, $t = [0, t_s]$, $t_s = 1.415\ h$.
(2) Distillation phase, $r \in (0, 1)$, $t = [t_s, t_f]$, $t_f = 10\ h$.

The objective is to track a reference trajectory for the distillate purity $x_d(t)$ in Interval 2. This trajectory ends up at the desired distillate purity at final time, $x_{d,des}$. Meeting the purity constraint can then be realized by closely tracking this reference. The reference trajectory is chosen to be linear with $x_{d,ref}(t_s) = 0.925\ kmol/kmol$ and $x_{d,ref}(t_f) = 0.9\ kmol/kmol$. The accumulated distillate purity $x_d(t)$ is assumed to be measurable once some distillate has been collected in Interval 2. The initial feedforward trajectory for iterative learning schemes is also linear with $r(t_s) = 0.898$ and $r(t_f) = 0.877$.

In order to obtain a realistic test scenario, the following uncertainty is considered:

- Perturbation: Boilup rate equally distributed in the range $V = [13\ 17]\ kmol/h$, changed every 0.5 h.
- Measurement noises: Product composition x_d with 5% multiplicative gaussian noise.

The values of the squared tracking error $\sum_{t=t_s}^{t_f} e(t)^2$ and the final tracking error $e(t_f)$ are averaged over 20 realizations of the perturbation and measurement noise. Also, the variance $v_{e(t_f)}$ of the residual error is calculated from 20 realizations.

4.2 Implementation of trajectory tracking schemes

The tracking error cannot be reduced to zero because of a nonzero initial tracking error ($x_d(t_s) \neq x_{d,ref}(t_s)$) arising from uncertainties in the start-up phase. As a consequence, the methods assuming zero initial tracking error do not converge in this example. Instead, a nonzero tracking error has to be allowed in order to guarantee convergence. This can be accomplished by applying a forgetting factor or a time shift of the feedforward input.

4.2.1. Iterative learning control without on-line feedback (ILC): Three ILC schemes without on-line feedback are considered:

- (ILC $\beta = 0.999$): A forgetting factor β is applied to the feedforward input trajectory in (11) with $\bar{A} = \beta$ and $\bar{B} = k^{ff}$.
- (ILC $\delta_u = 1\ h, \delta_e = 1\ h$): The same large shift is applied to the feedforward input and the error in (14).
- (ILC $\delta_u = 0.25\ h, \delta_e = 0\ h$): A smaller shift of the feedforward input trajectory is imposed and the error trajectory is not shifted in (14).

The run-to-run gain $k^{ff} = 0.1\ kmol/kmol$ was determined as a compromise between robustness and performance. ILC with forgetting factor converges after 30 runs, while the methods with time shift of the trajectories converge after 25 runs (Figure 1). The residual error is slightly smaller with the latter methods, especially when the time shift of the feedforward trajectory is reduced to $\delta_u = 0.25\ h$ (Table 3).

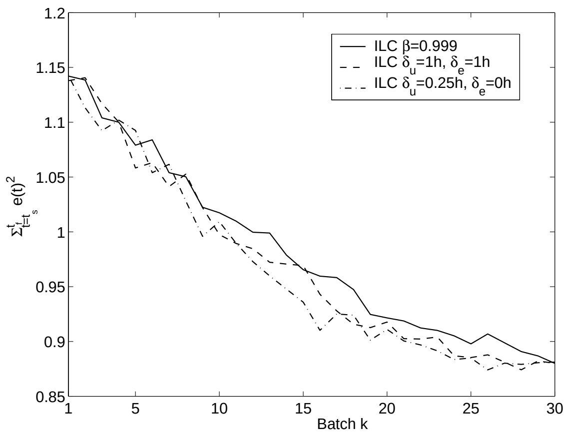

Fig. 1. Evolution of the squared tracking error for ILC methods without current cycle feedback.

Strategy	$\sum_{t=t_s}^{t_f} e(t)^2$ $[kmol^2/kmol^2]$	$e(t_f)$ $[kmol/kmol]$	$v_e(t_f)$ $[kmol^2/kmol^2]$
ILC $\beta = 0.999$	0.874	+0.0065	$4.1 \cdot 10^{-05}$
ILC $\delta_u = 1h$, $\delta_e = 1h$	0.883	-0.0022	$3.5 \cdot 10^{-05}$
ILC $\delta_u = 0.25h$, $\delta_e = 0h$	0.874	-0.0005	$3.0 \cdot 10^{-05}$
FB	0.869	-0.0069	$1.5 \cdot 10^{-05}$
FB+ILC $\beta = 0.999$	0.865	-0.0053	$3.0 \cdot 10^{-05}$
FB+ILC $\delta_u = 1h, \delta_e = 1h$	0.865	-0.0048	$2.7 \cdot 10^{-05}$
FB+ILC $\delta_u = 0.25h, \delta_e = 0h$	0.864	-0.0036	$2.0 \cdot 10^{-05}$

Table 3. Comparison of the various tracking schemes after the 30^{th} run in terms of squared tracking error $\sum_{t=t_s}^{t_f} e(t)^2$, tracking error at final time $e(t_f)$ and its variance $v_e(t_f)$.

4.2.2. ILC with current cycle feedback (FB+ILC): The on-line feedback utilized is the PI controller:

$$r(t) = r^{ff}(t) - k_p \left(e(t) + k_i \int_{t_s}^{t} e(\tau) d\tau \right) \qquad (24)$$

where r^{ff} is the feedforward term. The parameters $k_p = 8\ kmol/kmol$ and $k_i = 0.02\ 1/h$ were tuned manually. The error does not reduce to zero since integral action is not sufficient to drive it to zero within the finite time of the batch. Also, increasing the gains for faster error reduction causes instability.

Three ILC schemes are compared as in Section 4.2.1, but now with the additional current cycle feedback. Though the squared tracking errors are similar for all on-line schemes, the final error can be reduced using the time shift $\delta_u = 0.25\ h$ (Table 3). The residual error with current cycle feedback and $\delta u = 0.25$ is larger than that without feedback, but its variance is smaller, because within-run perturbations can be compensated.

5. CONCLUSION

This paper has provided an overview of ILC schemes with general conditions for convergence and an analysis of the residual tracking error. A new scheme that consists of anticipatory ILC with a shift for both the feedforward part of the input and the error trajectory and current cycle feedback has been presented. By allowing a nonzero residual error, this scheme provides good robustness properties. In contrast to ILC with a forgetting factor, which also allows a nonzero residual error, the proposed scheme has been shown to reduce the error as a function of run time for a constant input trajectory. This property is useful for batch processes, where often the initial conditions are uncertain and a constraint has to be met at final time. The proposed scheme has been applied to a simulated batch distillation system in the presence of uncertainties, where tracking of a distillate purity reference is utilized to meet the terminal constraint on distillate purity. It has been shown that shifting the input trajectory results in a smaller residual error than using a forgetting factor.

REFERENCES

Amann, N., D.H. Owens and E. Rogers (1996). Iterative learning control for discrete-time systems with exponential rate of convergence. *IEE Proc.-Control Theory Appl.* **143**(2), 217–224.

Arimoto, S., S. Kawamura and F. Miyazaki (1984). Bettering operation of robots by learning. *J. Robotic Syst.* **1**(2), 123–140.

Arimoto, S., T. Naniwa and H. Suzuki (1990). Robustness of P-type learning control with a forgetting factor for robotic motion. In: *Proceedings of the 29th Conference on Decision and Control*. Honolulu, Hawaii. pp. 2640–2645.

Chen, Y. and C. Wen (1999). *Iterative Learning Control*. Lecture Notes in Control and Information Sciences. Springer, London.

Chien, C.J. and J.S. Liu (1994). A P-type iterative learning controller for robust output tracking of nonlinear time-varying systems. In: *Proceedings of the American Control Conference*. Baltimore.

Choi, J.W., H.G. Choi, K.S. Lee and W.H. Lee (1996). Control of ethanol concentration in a fed-batch cultivation of Acinetobacter calcoaceticus RAG-1 using a feedback-assisted iterative learning algorithm. *Journal of Biotechnology* **49**, 29–43.

Heinzinger, G., D. Fenwick, B. Paden and F. Miyazaki (1992). Stability of learning control with disturbances and uncertain initial conditions. *IEEE Trans. Autom. Contr.* **37**, 110–114.

Jang, T.J., C.H. Choi and H.S. Ahn (1995). Iterative learning control in feedback systems. *Automatica* **31**(2), 243–248.

Kuc, T.Y., K. Nam and J.S. Lee (1991). An iterative learning control of robot manipulators. *IEEE Trans. Rob. Autom.* **7**(6), 835–842.

Lee, J.H., K.S. Lee and W.C. Kim (2000). Model-based iterative learning control with a quadratic criterion for time-varying linear systems. *Automatica* **36**(5), 641–657.

Lee, K.H. and Z. Bien (1991). Initial condition problem of learning control. *IEE Proceedings-D* **138**(6), 525–528.

Lee, K.S., I.S. Chin and H.J. Lee (1999). Model predictive control technique combined with iterative learning for batch processes. *AIChE J.* **45**(10), 2175–2187.

Lee, K.S., S.H. Bang, S.Yi, J.S Son and S.C. Yoon (1996). Iterative learning control of heat-up phase for a batch polymerization reactor. *J.Proces.Contr.* **6**, 255–262.

Park, K.H., Z. Bien and D.H. Hwang (1998). Design of an iterative learning controller for a class of linear dynamic systems with time delay. *IEE Proc.-Control Theory Appl.* **145**(6), 507–512.

Saab, S.S. (1994). On the P-type learning control. *IEEE Trans. Autom. Contr.* **39**, 2298–2302.

Wang, D. (2000). On D-type and P-type ILC designs and anticipatory approach. *Int. J. Control* **73**(10), 890–901.

Welz, C., B. Srinivasan and D. Bonvin (2004). Combined on-line and run-to-run optimization of batch processes with terminal constraints. In: *ADCHEM'03*. Hong-Kong.

Xu, J.X., Q. Hu, T.H. Lee and S. Yamamoto (2001). Iterative learning control with Smith time delay compensator for batch processes. *J.Proces.Contr.* **11**, 321–328.

Xu, J.X., X.W. Wang and L.T. Heng (1995). Analysis of continuous iterative learning control systems using current cycle feedback. In: *Proceedings of the American Control Conference*. Seattle, Washington. pp. 4221–4225.