University of Alberta

Performance Monitoring of Iterative Learning Control and Development of Generalized Predictive Control for Batch Processes

by

Ehsan Farasat

A thesis submitted to the Faculty of Graduate Studies and Research
in partial fulfillment of the requirements for the degree of

Master of Science
in
Process Control

Department of Chemical and Materials Engineering

Spring 2012
Edmonton, Alberta

To my parents

Abstract

Unlike continuous processes, a batch process has a certain period of operation time, and there are a number of batches in a typical operation. Hence variables in a batch process have dynamics in two dimensions, along time and across batches. Besides, batch processes involve large transient phases covering a wide range of operating envelopes, which cause challenges in both modeling and control.

To meet the control objectives of batch processes, set-point tracking and disturbance rejection, iterative learning control (ILC) has been widely attempted. This thesis is concerned with the optimal design and performance assessment of ILC based on the minimum variance benchmark.

When performance of ILC is unsatisfactory, alternative control strategies should be considered. Generalized predictive control (GPC) is a popular control strategy for continuous processes. Developing a two-dimensional GPC structure for batch processes is another focus of this research. Finally, ILC and suggested GPC are compared through simulation studies.

Acknowledgements

First of all, I would like to utter my gratitude to my supervisor, Prof. Biao Huang, for his continuous support throughout my program. His fruitful advices have always helped me through different stages of work. He has remarkably been thoughtful of me and encouraged me during my research.

Next I thank my friends in CPC group at University of Alberta that I had the pleasure of working with or helped me by different means. Shima Khatibisepehr, Iraj Hassanzadeh, Aditya Tulsyan, Marzieh Keshavarz, Yuri Shardt, and William Weng are among these people.

I would like to acknowledge the Department of Chemical and Materials Engineering at University of Alberta for providing research opportunities. Financial support of the Natural Sciences and Engineering Research Council of Canada (NSERC) is also acknowledged.

Finally, I would like to express my sincere appreciation and love to my parents, whose support have always been with me in my life. Moreover, I thank my dear friends in Canada: Milad, Keivan, Ghazaleh, Kamran, Vahid, Farshad, Mojtaba, Mitra and many others. Their friendship means the world to me.

Table of Contents

List of Tables

List of Figures

Chapter 1

Introduction

1.1 Objective of the thesis

Control and monitoring of batch processes have received remarkable attention [28], as industrial batch applications have increased over the past years. Literature has focused on developing control strategies to improve the performance in terms of set-point tracking and disturbance rejection. Although many strategies have been developed for continuous processes, most of them can hardly be directly applied to batch processes due to feasibility problem. The main features of a batch process can be classified as follow:

- The process lasts for a certain time duration, called a batch. Hence batch data are two-dimensional, which implies that the dynamics of the variables across batches as well as along time must be taken into account for modeling purposes.

- The process involves large transient phases covering a wide range of operating envelopes.

A successful control strategy of a batch process needs to address the set-point tracking and disturbance rejection problems. To meet these requirements, iterative learning control (ILC) is one of the most recommended strategies. Several structures have been proposed for design and implementation of ILC. To ensure the efficiency of the design, various methods have been investigated to assess the performance of the control. This thesis elaborates upon the optimal design and performance assessment procedures for the ILC algorithm with minimum variance control (MVC) as the benchmark.

Some of the existing control strategies for continuous processes can be modified and applied to batch processes. Several strategies from the family of predictive controllers have already been extended for control of batch processes. The basis of predictive controllers is to anticipate future output and minimize its error - difference from the set-point. From this family, generalized predictive control (GPC) utilizes the optimal predictor to anticipate the future output, which results in rejecting the stochastic disturbance as well as tracking the set-point. To derive the optimal predictor, knowledge of process models is required. However, since batch processes possess non-linearity and two-dimensional dynamics of variables, process modeling is considerably challenging. This thesis targets two-dimensional modeling and extension of GPC for batch processes.

Two strategies, ILC and GPC, are introduced for control of batch processes. The optimal design and performance assessment of ILC based on the minimum variance benchmark are elaborated, followed by two-dimensional modeling and extension of GPC. Simulation studies for control of a batch process using both strategies are carried out, and suggestions for the implementation of each strategy are made.

1.2 Technical background

1.2.1 Batch process monitoring

Among the techniques introduced for modeling and monitoring of batch processes, multivariate statistical process control (MSPC) techniques have obtained popularity due to their abilities to handle the dynamics of batch processes in three dimensions, i.e. variable , time and batch [56]. MSPC techniques include several algorithms, with different degrees of effectiveness, such as parallel factor analysis, multi-way principle component analysis (MPCA) and multi-way partial least squares (MPLS). Smilde (1992) [75] thoroughly discusses the advantages of parallel factor analysis, but a more recent contribution by Westerhuis (1999) [81] demonstrates the superior performance of the multi-way techniques.

Originally, Nomikos and MacGregor (1994) [62] used MPCA and MPLS to determine whether a batch operates well or not. Then expanding upon the applications, the multi-way techniques were implemented on-line to estimate product quality [45]. Currently, the multi-way techniques are considered for various monitoring purposes of batch processes like quality control and batch estimation [52, 82].

Considering all the variables, batch data are three-dimensional. To execute the multi-way techniques, we need to convert the data into two dimensions by unfolding them along one dimension. Conventionally, batch-wise unfolding was used for off-line batch monitoring [42, 81]. The literature covers a variety of techniques to unfold batch data [18, 23].

Alternative methods for modeling of batch processes have also been suggested. Ramaker (2005) [67] discusses the use of local and evolving models for batch processes. Local models are estimated from the data at only one sampling time across the batches, while evolving models are estimated from the measurements obtained from the beginning to the current sampling time. Rannar (1998) [68] elaborates Hierarchical PCA which uses evolving models with different weights given to the current measurements. Another method is called Moving Window PCA (MWPCA), in which the models are estimated from the current and some past data [48]. To avoid dealing with a large number of models, Batch Dynamic PCA (BDPCA) was introduced which estimates a single model based on PCA theory [8]. More theoretical and practical aspects of batch modeling and monitoring can be found in [6, 7, 9, 31, 43, 55, 63].

1.2.2 Conventional identification procedure

Estimation of process models is needed for design and performance monitoring of many control strategies. This section addresses the procedure for estimation of linear time-invariant continuous processes. The goal is to estimate a model from data which can describe the dynamics of the input and the output, as shown in Figure 1.1.

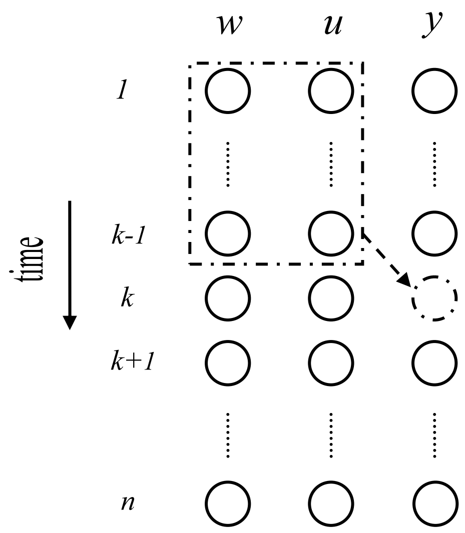


Figure 1.1: Variable relations for a SISO continuous process

Here w, u, and y denote a white noise, an input and an output, respectively. A model can be structured as:

$$y_k = T(q^{-1}, \theta)u_k + N(q^{-1}, \theta)w_k \tag{1.1}$$

where T, N are plant and disturbance models, respectively. These models consist of numerators and denominators which are functions of back-shift operator and model parameters.

Since the actual models are unknown, the identification procedure is followed to estimate the models. The estimated models are then used to derive an one-step ahead predictor, which predicts the next step output from available data. One-step ahead predictors are structured as:

$$\hat{y}_{(k|k-1)} = L_1(q^{-1}, \theta) y_k + L_2(q^{-1}, \theta) u_k \tag{1.2}$$

where L_1, L_2 are predictor models.

Optimal predictor

An optimal predictor anticipates the future output based on minimization of the prediction error variance. To derive the optimal predictor in terms of the estimated plant and disturbance models, one method is to minimize the prediction error variance with respect to L_1, L_2. The prediction error can be rewritten as:

$$\begin{aligned}\varepsilon(k,\theta) &= y_k - \hat{y}_{(k|k-1)} \\ &= T(q^{-1},\theta)u_k + N(q^{-1},\theta)w_k - L_1(q^{-1},\theta)y_k - L_2(q^{-1},\theta)u_k \\ &= [T(q^{-1},\theta) - L_2(q^{-1},\theta)]u_k + [N(q^{-1},\theta) - I]w_k - L_1(q^{-1},\theta)y_k + w_k\end{aligned} \tag{1.3}$$

From Equation 1.1, white noise can be rewritten as:

$$\begin{aligned}y_k &= T(q^{-1},\theta)u_k + N(q^{-1},\theta)w_k \\ \rightarrow w_k &= N^{-1}(q^{-1},\theta)[y_k - T(q^{-1},\theta)u_k]\end{aligned} \tag{1.4}$$

Updating Equation 1.3 yields:

$$\begin{aligned}\varepsilon(k,\theta) =&[N^{-1}(q^{-1},\theta)T(q^{-1},\theta) - L_2(q^{-1},\theta)]u_k \\ &+ [1 - N^{-1}(q^{-1},\theta) - L_1(q^{-1},\theta)]y_k + w_k\end{aligned} \tag{1.5}$$

Let us define two variables:

$$\psi_u(q^{-1},\theta) = N^{-1}(q^{-1},\theta)T(q^{-1},\theta) - L_2(q^{-1},\theta) \tag{1.6}$$

$$\psi_y(q^{-1},\theta) = 1 - N^{-1}(q^{-1},\theta) - L_1(q^{-1},\theta) \tag{1.7}$$

Substituting the new variables in Equation 1.5, we have:

$$\varepsilon(k,\theta) = \psi_u(q^{-1},\theta)u_k + \psi_y(q^{-1},\theta)y_k + w_k \tag{1.8}$$

Remark: If a transfer function has at least one time delay between input and output, its value will become zero by replacing the back-shift operator with zero. Hence $T(0, \theta) = L_1(0, \theta) = L_2(0, \theta) = 0$.

The disturbance model can be normalized so that its leading term in both numerator and denominator is 1 so that $N^{-1}(0, \theta) = 1$. Hence we can conclude:

$$\psi_u(0, \theta) = 0 \tag{1.9}$$

$$\psi_y(0, \theta) = 0 \tag{1.10}$$

In other words, the first two terms in Equation 1.8 have time delay, and are not correlated with the third term. Hence the variance of the prediction error can be computed as:

$$var(\varepsilon(k, \theta)) = var(x) + var(w_k) \geq var(w_k) \tag{1.11}$$

where

$$x = \psi_u(q^{-1}, \theta)u_k + \psi_y(q^{-1}, \theta)y_k \tag{1.12}$$

Equation 1.11 indicates that the minimum prediction error variance is reached when x=0. Equating x with zero, the optimal predictor is derived:

$$\hat{y}(k|k-1) = L_1^{opt}(q^{-1}, \theta)y_k + L_2^{opt}(q^{-1}, \theta)u_k \tag{1.13}$$

where

$$L_1^{opt}(q^{-1}, \theta) = 1 - N^{-1}(q^{-1}, \theta) \tag{1.14}$$

$$L_2^{opt}(q^{-1}, \theta) = N^{-1}(q^{-1}, \theta)T(q^{-1}, \theta) \tag{1.15}$$

Prediction error method

Prediction error method (PEM) is an algorithm to estimate plant and disturbance models. Substituting the optimal predictor in Equation 1.5, the prediction error can be written as:

$$\varepsilon(k, \theta) = y_k - [1 - \hat{N}^{-1}(q^{-1}, \theta)]y_k - [\hat{N}^{-1}(q^{-1}, \theta)\hat{T}(q^{-1}, \theta)]u_k \tag{1.16}$$

In Equation 1.16, $\hat{T}$ and $\hat{N}$ are two general linear models with unknown parameters that represent estimated models. Note that the estimated models are used to build the optimal predictor, Equation 1.13. A scalar objective function is defined based on minimization of the prediction error variance:

$$\min_{\theta} \frac{1}{N} tr\{\sum \varepsilon(k,\theta)\varepsilon^T(k,\theta)\} \tag{1.17}$$

where N denotes the number of data points available for prediction. Solving the objective function with respect to unknown parameters, $\hat{T}$ and $\hat{N}$ are found.

Model validation tests

Estimated models need to be tested to ensure there is no evidence to reject validity of the estimation. Recalling Equation 1.8 and substituting the optimal predictor, we have:

$$\begin{aligned} \varepsilon(k,\theta) =& \psi_u(q^{-1},\theta)u_k + \psi_y(q^{-1},\theta)y_k + w_k \\ \psi_u(q^{-1},\theta) &= N^{-1}(q^{-1},\theta)T(q^{-1},\theta) - \hat{N}^{-1}(q^{-1},\theta)\hat{T}(q^{-1},\theta) \\ \psi_y(q^{-1},\theta) &= 1 - N^{-1}(q^{-1},\theta) - 1 + \hat{N}^{-1}(q^{-1},\theta) \end{aligned} \tag{1.18}$$

Two tests are introduced:

1. Auto-correlation test:

   From Equation 1.18, the first two terms of the prediction error will become zero if estimated models are equal to the actual ones. Therefore, the prediction error must be a white noise.

   In practice, a signal is considered white noise if more than 99% of its autocorrelation points fall within the range of $[-\frac{3}{\sqrt{N}}, \frac{3}{\sqrt{N}}]$ [76]. Based on definition, autocorrelation is a normalized autocovariance which can be calculated as shown below:

   $$\rho_\varepsilon(\tau) = \frac{r_\varepsilon(\tau)}{r_\varepsilon(0)} \tag{1.19}$$

   where $r_\varepsilon(\tau)$ represents the prediction error autocovariance with τ lags. To conclude, if more than 99% $\rho_\varepsilon(\tau)$ satisfy $|\rho_\varepsilon(\tau)| < \frac{3}{\sqrt{N}}$, the prediction error is considered white noise and estimated models pass the test.

2. Cross-correlation test:

   From Equation 1.18, if the plant model estimation is good, $\psi_u(q^{-1},\theta)$ will become zero and there will be no cross-correlations between the prediction error and the input. Based on definition, cross-correlation is a normalized covariance which can be computed as shown below:

   $$\rho_{\varepsilon u}(\tau) = \frac{r_{\varepsilon u}(\tau)}{\sqrt{r_\varepsilon(0) r_u(0)}} \tag{1.20}$$

where $r_{\varepsilon u}(\tau)$ represents covariance between the prediction error and the input with τ lags. If more than 99% $\rho_{\varepsilon u}(\tau)$ satisfy $|\rho_{\varepsilon u}(\tau)| < \frac{3}{\sqrt{N}}$, the estimated plant model passes the test and is validated.

Note that the cross-correlation test validates only the plant model estimation, while the auto-correlation test validates both the plant and disturbance models.

If the estimated models are validated, there is no evidence to reject the estimation and the models are accepted. Otherwise, different model structures are chosen and the procedure is repeated.

Input design

Process data need to be suitable for identification. To collect suitable identification data, an input sequence is designed and given to the system in an experiment. In order to cover all process frequencies, the input is required to have a bandwidth at least equal to $\frac{1}{\tau}$ [51, 76], where τ is the time constant of the equivalent first order model obtained after a step test. Hence the input should have a maximum frequency of $\frac{k}{\tau}$, where k is a recommended integer equal to 2 or 3, in order to ensure the coverage of all desired frequencies.

In MATLAB, actual frequency is normalized to a number between 0 and 1, called the normalized frequency. The maximum normalized frequency is equivalent to the maximum actual frequency for discrete signals, which equals half the sampling frequency, i.e. $\frac{\pi}{T_s}$. Note that T_s represents the sampling time, and as a rule of thumb, can be set equal to (0.1-0.2) time constant [76].

Actual frequencies can be normalized based on the normalization standard. Therefore, the input sequence for the experiment should have a maximum normalized frequency of:

$$\omega_{max} = \frac{\frac{k}{\tau}}{\frac{\pi}{T_s}} = \frac{kT_s}{\tau\pi} \tag{1.21}$$

After setting the frequency range, the type of signal is selected, and the desired input is generated. Finally, identification data are collected after running the experiment all the way through.

1.2.3 Performance assessment of control loops

To ensure the operation of the process, the performance of control loops needs to be monitored in real-time. The performance is indicated in terms of current results compared to a benchmark. Several benchmarks have already been developed, among which the minimum variance benchmark is remarkably popular. This benchmark, first proposed by Harris (1989) [26], is established based on the minimum variance control theory introduced by Astrom (1970) [3]. Astrom developed linear time-invariant minimum variance control, which was followed by notable progress in the fields of predictive control [11, 21, 73] and adaptive control [24, 25, 44]. This theory was then extended by Harris (1989) [26] to build a benchmark, against which the performance of a process can be assessed. He proved that some terms of the process output are never affected by the feedback controller. This contribution has led to significant progress in the area of performance assessment [13, 14, 27, 30, 32, 35, 41, 53, 77, 78]. Minimum variance benchmark has also been extended for time-varying systems [33, 49, 50]. Huang and Shah (1999) [34] have discussed the theoretical and practical aspects of performance assessment of control loops.

Although simple, the minimum variance solution may not be achievable for practical controllers. The need for a more practical solution leads to the development of other benchmarks. Ko and Edgar (1998) [40] introduced the PID control benchmark. In addition, the linear quadratic Guassian (LQG) regulator benchmark, based on the process model, was proposed by Huang and Shah (1999) [34]. Some model-based benchmarks have also been suggested for performance assessment of MPC strategies [20, 74].

Since this thesis in part deals with performance assessment of iterative learning control loops based on minimum variance control law, conventional minimum variance performance assessment is introduced in details. Consider the closed-loop control of a continuous process described in Figure 1.2 and Equation 1.22.

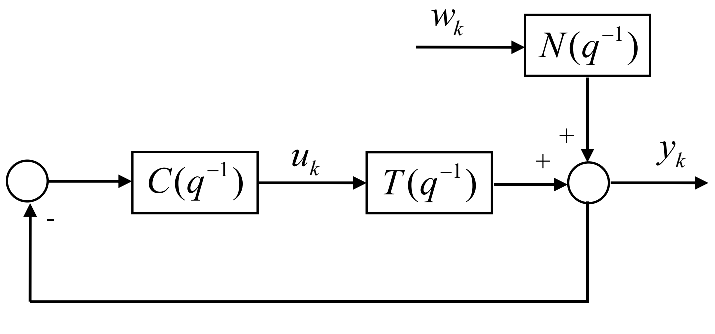


Figure 1.2: Closed-loop control of a continuous process

$$y_k = T(q^{-1})u_k + N(q^{-1})w_k \tag{1.22}$$

Note that $T(q^{-1}), N(q^{-1})$ and $C(q^{-1})$ represent the plant, disturbance and controller models. In addition, w_k and y_k denote the white noise and the output of the process.

Minimum variance control

The purpose of the conventional minimum variance control law is merely to reject the disturbance. Therefore, no set-point is assigned, and the regulatory control problem is studied, as shown in Figure 1.2. Hence control action can be described as:

$$u_k = -C(q^{-1})y_k \tag{1.23}$$

Since the output is dependent on disturbance, Diophantine identity [26] is used to split the disturbance model:

$$N(q^{-1}) = F(q^{-1}) + R(q^{-1})q^{-d} \tag{1.24}$$

where d denotes the plant delay. Transforming the disturbance model into the impulse response format, F and R are obtained. Substituting Equation 1.24 into Equation 1.22, we find:

$$y_k = \tilde{T}(q^{-1})q^{-d}u_k + [F(q^{-1}) + R(q^{-1})q^{-d}]w_k \tag{1.25}$$
$$y_k = \tilde{T}(q^{-1})u_{k-d} + F(q^{-1})w_k + R(q^{-1})w_{k-d} \tag{1.26}$$

where $\tilde{T}$ represents the delay-free part of the plant model. Note that F includes the first d terms of the disturbance impulse response model. Hence there are no correlations between $F(q^{-1})w_k$ and other terms in Equation 1.26. Consequently, the variance of the output can be written as:

$$var(y_k) = var(X) + var(F(q^{-1})w_k) \tag{1.27}$$

where

$$X = \tilde{T}(q^{-1})u_{k-d} + R(q^{-q})w_{k-d} \tag{1.28}$$

Since $F(q^{-1})w_k$ is not dependent on the controller, the optimal output is:

$$y_k^{opt} = F(q^{-1})w_k \tag{1.29}$$

To achieve the optimal solution, the following must hold:

$$X = 0 \tag{1.30}$$
$$\rightarrow \tilde{T}(q^{-1})u_{k-d}^{opt} = -R(q^{-q})w_{k-d} \tag{1.31}$$
$$\rightarrow u_{k-d}^{opt} = -\tilde{T}^{-1}(q^{-1})R(q^{-q})w_{k-d} \tag{1.32}$$

Substituting the inverse of Equation 1.29, the optimal input is simplified:

$$u_k^{opt} = -\tilde{T}^{-1}(q^{-1})R(q^{-q})F^{-1}(q^{-1})y_k^{opt} \tag{1.33}$$

From Equation 1.23, in order to set the optimal input, the following controller must be implemented:

$$C^{opt}(q^{-1}) = \tilde{T}^{-1}(q^{-1})R(q^{-1})F^{-1}(q^{-1}) \tag{1.34}$$

It is noted that to have a proper optimal controller, the plant model needs to be minimum phase, i.e. has no unstable zeros.

Conventional performance assessment

Closed-loop modeling of the process yields:

$$y_k = \frac{N(q^{-1})}{1 + T(q^{-1})C(q^{-1})} w_k \tag{1.35}$$

Substituting the Diophatine identity into Equation 1.35, we find:

$$\begin{aligned} y_k &= \frac{F(q^{-1}) + R(q^{-1})q^{-d}}{1 + T(q^{-1})C(q^{-1})} w_k \\ &= [F(q^{-1}) + \frac{R(q^{-1})q^{-d} - F(q^{-1})T(q^{-1})C(q^{-1})}{1 + T(q^{-1})C(q^{-1})}] w_k \\ &= [F(q^{-1}) + q^{-d}\frac{R(q^{-1}) - F(q^{-1})\tilde{T}(q^{-1})C(q^{-1})}{1 + T(q^{-1})C(q^{-1})}] w_k \end{aligned} \tag{1.36}$$

Let us define a new variable:

$$L(q^{-1}) \triangleq \frac{R(q^{-1}) - F(q^{-1})\tilde{T}(q^{-1})C(q^{-1})}{1 + T(q^{-1})C(q^{-1})} \tag{1.37}$$

Updating Equation 1.36 gives:

$$y_k = [F(q^{-1}) + q^{-d}L(q^{-1})]w_k \tag{1.38}$$

To assess the performance of a control loop, routine operating data are collected for the identification of F and L. From Equation 1.29, the optimal output depends only on F. Therefore, the following index is used to determine the performance of the control loop:

$$\eta = \frac{\|F\|_2^2}{\|F + q^{-d}L\|_2^2} \tag{1.39}$$

which reflects the ratio between the optimal and current performance.

1.2.4 Iterative learning control

Iterative learning control (ILC) strategy was first introduced by Arimoto and Kawamura (1984) [2] for control of robot systems. This strategy, owing to its specific structure, has proved useful for set-point tracking and disturbance rejection purposes for batch processes [19, 38, 46, 83].

Design of ILC strategy has received remarkable attention in literature [57, 64, 65, 84]. Since reliable models are required for the design, various model uncertainties of batch processes, such as parameter uncertainties or across-batch stochastic uncertainty, have been studied [1, 15, 59, 60, 61]. Another challenge in ILC design arises when the plant model is non-minimum phase, i.e. it has at least one non-stable zero. Roh (1996) [72] and Jeong (2002) [37] elaborated thoroughly on this problem. To improve the control performance, combining the iterative learning algorithm with some existing control strategies has been suggested. For instance, Lee (1999) [47] and Wang (2008) [80] demonstrate the implementation of iterative learning model predictive control. The combination of ILC and partial least squares models is another example [17]. For more on design and implementation of ILC, readers are referred to [79, 58].

ILC algorithm

In batch systems, the process lasts for a certain time period, with each period called a batch. After one batch is complete, another batch continues. The idea of ILC is to use previous batches to control the current batch. Figure 1.3 illustrates a well-known structure of ILC.

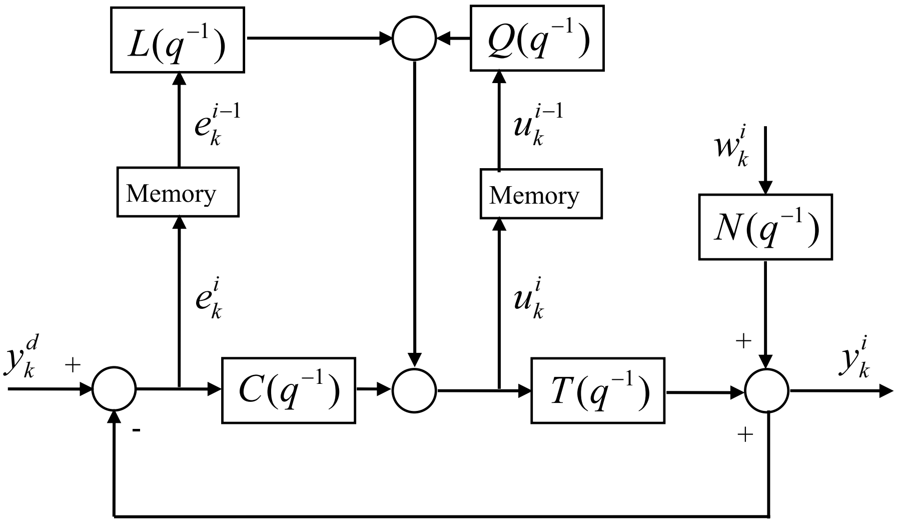


Figure 1.3: Example of an ILC structure

Note that $T(q^{-1}), N(q^{-1})$ and $C(q^{-1})$ are plant, disturbance and feedback controller models, and $L(q^{-1})$ and $Q(q^{-1})$ are iterative controllers. i, k and q^{-1} denote batch index, time index and time back-shift operator, respectively. As clear in Figure 1.3, this ILC structure consists of three loops. The loop that incorporates the plant model is the main one. The other two loops are specifically designed to cooperate in setting the input.

The mathematical equations of this structure are:

$$y_k^i = T(q^{-1})u_k^i + N(q^{-1})w_k^i \tag{1.40}$$

$$u_k^i = C(q^{-1})(y_k^d - y_k^i) + L(q^{-1})e_k^{i-1} + Q(q^{-1})u_k^{i-1} \tag{1.41}$$

In conventional closed-loop control, the feedback controller takes action solely based on real-time output error, while in ILC the use of previous batches is considered as well. At each batch, memorized input and error signals of the previous batch are filtered by $L(q^{-1})$ and $Q(q^{-1})$, then added to feedback controller's action. Simultaneously, the current input and error signals are memorized for the next batch. If a stable algorithm is designed, repeating the procedure will result in shrinking the error over batches. After a certain number of batches, the error variance remains constant, which means that convergence has been reached. However, an unstable algorithm could yield unstable output after a few batches.

1.2.5 Generalized predictive control

Predictive control strategies are based on minimization of future output error—the difference between predicted future output and set-point. In 1979, engineers at Shell Oil Company presented their predictive controller, called dynamic matrix control (DMC) [12]. DMC minimizes the least squares of output error, predicted from the step response model of linear processes. This strategy avoids taking drastic actions by penalizing the input increments over time. DMC is mostly implemented in petrochemical industries with multivariable processes [36, 66].

The next member of the predictive controllers family is model algorithmic control (MAC) [70]. Like DMC, this strategy is based on step response models. But, the objective of MAC is to minimize the predicted output error with respect to the input rather than the input increments [5]. To implement predictive controllers for a wider range of plants, predictive functional control (PFC) has been introduced, which is capable of handling non-linear and unstable linear internal models [29]. However, due to its specific structure, it can only use state space models [4, 69]. PFC is well-known for its ability to deal with quick tracking control problems [54]. Extended prediction self-adaptive control and extended horizon adaptive control are other members of this family [22, 39, 71, 85, 86]. Comprehensive reviews of all the predictive controllers can be found in literature [29, 36, 66].

In 1987, Clarke introduced generalized predictive control (GPC) [11]. This strategy can be used to reject the disturbance, as well as to track the set-point, since it implements the optimal predictor to anticipate the future output. The design procedure for this algorithm has been well-established in literature [10, 16].

Owing to a focus of this thesis on GPC, the basics of this strategy are explained here in more detail. First step is to derive the optimal predictor. Then an objective function is defined to minimize the predicted future error with respect to input increments over time.

Consider a linear process described by:

$$y_k = Tu_k + Nw_k \tag{1.42}$$

where T and N are functions of time back-shift operator. At time k, a-step ahead predictor is used to anticipate the output at time k+a:

$$\hat{y}_{(k+a|k)} = L_1 y_{k+a} + L_2 u_{k+a} \tag{1.43}$$

where L_1 and L_2 are predictor models. Splitting the disturbance model into two parts, we find:

$$N = F_a + R_a q^{-a} \tag{1.44}$$

where F_a includes the first a terms of the disturbance impulse response model. Updating Equation 1.42 with Equation 1.44 gives:

$$y_k = Tu_k + F_a w_k + R_a q^{-a} w_k \tag{1.45}$$

From Equation 1.42, we have:

$$w_k = N^{-1}[y_k - Tu_k] \tag{1.46}$$

Substituting Equation 1.46 into Equation 1.45 yields:

$$\begin{aligned} y_k &= Tu_k + F_a w_k + R_a q^{-a} N^{-1}[y_k - Tu_k] \\ &= R_a q^{-a} N^{-1} y_k + [T - R_a q^{-a} N^{-1} T] u_k + F_a w_k \\ &= R_a q^{-a} N^{-1} y_k + F_a N^{-1} T u_k + F_a w_k \end{aligned} \tag{1.47}$$

Having the output expression, the prediction error can be derived:

$$\begin{aligned} \varepsilon_{k+a} &= y_{k+a} - \hat{y}_{(k+a|k)} && (1.48) \\ &= R_a q^{-a} N^{-1} y_{k+a} + F_a N^{-1} T u_{k+a} + F_a w_{k+a} - [L_1 y_{k+a} + L_2 u_{k+a}] && (1.49) \\ &= [R_a q^{-a} N^{-1} - L_1] y_{k+a} + [F_a N^{-1} T - L_2] u_{k+a} + F_a w_{k+a} && (1.50) \end{aligned}$$

As L_1 operates on past data, it is delayed by at least a steps. Hence $[Rq^{-a}N^{-1} - L_1]$ is delayed by a steps, which makes the first term of Equation 1.50 uncorrelated with $F_a w_{k+a}$. The second term of Equation 1.50 is also uncorrelated with $F_a w_{k+a}$, since an open-loop control strategy is being implemented. Therefore, the variance of the prediction error can be written as:

$$var(\varepsilon_{k+a}) = var([R_a q^{-a}N^{-1} - L_1]y_{k+a} + [F_a N^{-1}T - L_2]u_{k+a}) + var(F_a w_{k+a}) \tag{1.51}$$

The optimal predictor can be found by minimizing the variance of the prediction error. Since the last term of Equation 1.51 is independent of the predictor, the optimal predictor is derived by equating the first term of Equation 1.51 to zero:

$$\rightarrow L_1^{opt} = R_a q^{-a} N^{-1} \tag{1.52}$$

$$\rightarrow L_2^{opt} = F_a N^{-1} T \tag{1.53}$$

A number of future outputs are chosen for the prediction purpose—called prediction horizon. The minimization of future error is executed with respect to input increments of selected steps—called control horizon. After defining prediction and control horizons, the GPC objective function can be defined:

$$\min_{\Delta u_k} J = \min_{\Delta u_k}[\sum_{i=1}^{n}(y_{k+i}^d - \hat{y}_{(k+i|k)})^2 + \lambda(\Delta u_k)^2] \tag{1.54}$$

where y_d, n and λ are the set-point, the length of the prediction horizon and a weighting variable, respectively. Note that for the sake of simplicity, control horizon is composed of a single input. To optimize the objective function, we need to write the optimal predictor in terms of the input increment. Let us define a new variable:

$$\hat{y}^*_{(k+a|k)} = L_1 y_{k+a} + L_2 u_{k+a}, u_{k+i} = u_{k-1}, i \geqslant 0 \tag{1.55}$$

Equation 1.55 represents the constant part of the predicted output. In other words, assuming the input increment of zero, the predicted output can be computed from Equation 1.55. To compensate for the input increment in prediction, another term is introduced:

$$\hat{y}(k+a|k) = \hat{y}^*(k+a|k) + h_a \Delta u_k \tag{1.56}$$

where h_a is an impulse response coefficient of L_2, computed from:

$$L_2 = h_0 + h_1 q^{-1} + h_2 q^{-2} + \cdots \tag{1.57}$$

Note that L_2 is itself dependent on a. Let us define vectors to represent data over prediction horizon:

$$Y^d = [y^d_{k+1}, \cdots, y^d_{k+n}]^T \tag{1.58}$$

$$\hat{Y} = [\hat{y}_{k+1}, \cdots, \hat{y}_{k+n}]^T \tag{1.59}$$

$$\hat{Y}^* = [\hat{y}^*_{k+1}, \cdots, \hat{y}^*_{k+n}]^T \tag{1.60}$$

Rewriting the objective function in quadratic form gives:

$$\min_{\Delta u_k} J = \min_{\Delta u_k} [(Y^d - \hat{Y}^* - H\Delta u_k)^T Q (Y^d - \hat{Y}^* - H\Delta u_k) + (\Delta u_k)^T R (\Delta u_k)] \tag{1.61}$$

where Q and R are weighting matrices. Note that H is a matrix composed of impulse response coefficients of L_2, as shown below:

$$H = \begin{bmatrix} h_1 \\ h_2 \\ \vdots \\ h_n \end{bmatrix}_{n*1} \tag{1.62}$$

Taking the derivative of Equation 1.61 with respect to the input increment, the optimal solution is found:

$$\Delta u_k = (H^T Q H + R)^{-1} H^T Q^T (Y^d - \hat{Y}^*) \tag{1.63}$$

This solution is used to compute the input at time k. Once the next output is collected, the procedure is repeated to calculate the next input.

1.3 Thesis outline

The main focus of this thesis is on performance monitoring of ILC and development of ILC and GPC strategies for control of batch processes. This is elaborated through five chapters, with the initial chapter providing an overview of the objectives, the research scope, and technical background.

Chapter 2 addresses the performance monitoring procedure for ILC strategy, based on the minimum variance benchmark for linear time varying (LTV) batch processes. First, a detailed discussion on the noncommutativity property of LTV models is presented. Then the minimum variance benchmark and the performance monitoring procedure for ILC strategy of LTV processes are elaborated. Finally, simulation results are illustrated to verify the proposed algorithm.

The minimum variance benchmark obtained in chapter 2 is designed to estimate the optimal solution in terms of the best stochastic performance of ILC strategies that yield the best deterministic performance. Chapter 3 illustrates that the optimal solution, based on the minimum variance control law, is composed of a balance between the deterministic and stochastic performance of the strategy. For all the ILC strategies with equal deterministic performance, an optimal solution in terms of the stochastic performance needs to be calculated. Chapter 3 proposes a method to estimate the optimal solutions from routine operating data. Finally, performance assessment of the control strategy based on the optimal solutions is described, followed by an illustration of simulation studies.

Chapter 4 deals with the extension and implementation of GPC for control of batch processes. Since the dynamics of batch variables are in two dimensions, along time and across batches, a method is proposed to estimate two-dimensional linear dynamics models. An optimal predictor is derived to anticipate the future output. This predictor is then used to develop a two-dimensional structure for GPC strategy. To verify the feasibility of the proposed method, simulation studies are carried out. It should be noted that the implementation of both GPC and ILC strategies is illustrated for all case studies, and comments on applications are made. To wrap up, chapter 5 gives concluding remarks and suggests the potential future work.

References

[1] H.-S. Ahn, K.L. Moore, and Y. Chen. Iterative Learning Control, Robustness and Monotonic Convergence for Interval Systems. Spring-Verlag, 2007.

[2] S. Arimoto, S. Kawamura, and F. Miyazaki. Iterative learning control for robot systems. Tokyo, Japan, 1984. Proceedings of IECON.

[3] K.J. Astrom. Introduction to Stochastic Control Theory. Academic Press, New York, 1970.

[4] S.AA. El Ata-Doss, P. Fiani, and J. Richalet. Handling input and state constraints in predictive functional control. pages 985-990. Proceedings of the IEEE Conference on Decision and Control, 1991.

[5] E. F. Camacho and C. Bordons. Model Predictive Control. Springer, 2007.

[6] J. Camacho and J. Pico. Online monitoring of batch processes using multi-phase principal component analysis. J. of Process Control, 16:1021-1035, 2006.

[7] J. Chen and H.H. Chen. On-line batch process monitoring using mhmt-based mpca. Chemical Engineering Science, 61:3223-3239, 2006.

[8] J. Chen and K. Liu. On-line batch process monitoring using dynamic pca and dynamic pls models. Chemical Engineering Science, 57:63-75, 2002.

[9] S.W. Choi, J. Morris, and In-B. Lee. Dynamic model-based batch process monitoring. Chemical Engineering Science, 63:622-636, 2008.

[10] D.W. Clarke. Generalized predictive control: A robust self-tuning algorithm. pages 990-995. Proceedings of the American Control Conference, 1987.

[11] D.W. Clarke, C. Mohtadi, and P.S. Tuffs. Generalized predictive control: parts 1 and 2. Automatica, 23(2):137-160, 1987.

[12] C.R. Cutler and B. Ramaker. Dynamic matrix control - a computer control algorithm. San Francisco, 1980. Proceedings of the Joint Automatic Control Conference.

[13] L. Desborough and T. Harris. Performance assessment measure for univariate feedback control. Can. J. Chem. Eng., 70:1186-1197, 1992.

[14] L. Desborough and T.J. Harris. Performance assessment measures for univariate feed-forward/feedback control. The Canadian J. of Chemical Engineering, 71:605-616, 1993.

[15] T.Y. Doh, J.H. Moon, K.B. Jin, and M.J. Chung. Robust iterative learning control with current feedback for uncertain linear systems. Internat. J. of Systems Sci., 30:39-47, 1999.

[16] J. Duan and M. Grimble. Design of long-range predictive control algorithms for industrial applications. IEE Colloquium on Industrial Applications of Model Based Predictive Control, Nov. 1991.

[17] J. Flores-Cerrillo and J.F. MacGregor. Iterative learning control for final batch product quality using partial least squares models. Ind. Chem. Res., 44:9146-9155, 2005.

[18] J. Flores-Cerrillo and J.F. MacGregor. Latent variable mpc for trajectory tracking in batch processes. J. of Process Control, 15:651-663, 2005.

[19] P. Fu and J.P. Barford. Simulation of an iterative learning control system for fed-batch cell culture processes. 10:53-62, 1992.

[20] J. Gao, R.S. Patwardhan, K. Akamatsu, Y. Hashimoto, G. Emoto, and S. L. Shah. Performance evaluation of two industrial mpc controllers. Control Engineering Practice, 11:1371-1387, 2003.

[21] C.E. Garcia, D.M. Prett, and M. Morari. Model predictive control: theory and practice-a survey. Automatica, 25(3):335-348, 1989.

[22] M.P. Golden, L.K. Liu, and B.E. Ydstie. Multivariable control using extended prediction horizons. pages 1550-1551. Proceedings of the IEEE Conference on Decision and Control, 1986.

[23] M. Golshan, J.F. MacGregor, M. Bruwer, and P. Mhaskar. Latent variable model predictive control (lv-mpc) for trajectory tracking in batch processes. J. of Process Control, 20:538-550, 2010.

[24] G.C. Goodwin, P.J. Ramadge, and P.E. Caines. Discrete-time stochastic adaptive control. SIAM J. Control, 19:826-853, 1981.

[25] G.C. Goodwin and K.S. Sin. Adaptive Filtering Prediction and Control. Prentice Hall, Englewood Cliffs, NJ, 1984.

[26] T. Harris. Assessment of control loop performance. The Canadian J. of Chemical Engineering, 67:856-861, 1989.

[27] T.J. Harris, F. Boudreau, and J.F. MacGregor. Performance assessment of multivariable feedback controllers. Automatica, 32(11):1505-1518, 1996.

[28] W.M. Hawkins and T.G. Fisher. Batch Control Systems Design, Application, and Implementation, 2nd Ed. ISA The Instrumentation, Systems, and Automation Society, 2006.

[29] K.S. Holkar and L.M. Waghmare. An overview of model predictive control. International Journal of Control and Automation, 3(4):47-64, 2010.

[30] A. Horch and A.J. Isaksson. A modified index for control performance assessment. J. of Process Control, 9(6):475-483, 1999.

[31] K. Hu and J. Yuan. Batch process monitoring with tensor factorization. J. of Process Control, 19:288-296, 2009.

[32] B. Huang. Performance assessment of processes with abrupt changes of disturbances. The Canadian J. of Chemical Engineering, 77(5):1044-1054, 1999.

[33] B. Huang. Minimum variance control and performance assessment of time-variant processes. J. of Process Control, 12:707-719, 2002.

[34] B. Huang and S.L. Shah. Performance Assessment of Control Loops: Theory and Applications. Springer Verlag, London, 1999.

[35] B. Huang, S.L. Shah, and K.Y. Kwok. Good, bad or optimal? performance assessment of mimo processes. Automatica, 33:1175-1183, 1997.

[36] J.C. Hung. Practical techniques for industrial control. IEEE, pages 5-10, 1996.

[37] G.-M. Jeong and C.-H. Choi. Iterative learning control for linear discrete time nonminimum phase systems. Automatica, 38:287-291, 2002.

[38] Z. Jie, J. Nguyan, J. Morris, and X. Zhihua. Batch to batch iterative learning control of a fed-batch fermentation process using linearized models. J. of Process Control, 17:745-750, 2008.

[39] R.M.C. De Keyser and A. Van Cauwenberghe. Towards robust adaptive control with extended predictive control. pages 1548-1549. Proceedings of the IEEE Conference on Decision and Control, 1986.

[40] B. Ko and T.F. Edgar. Assessment of achievable pi control performance for linear processes with dead time. Proceedings of American Control Conference, 1998.

[41] B.S. Ko and T.F. Edgar. Pid control performance assessment: the single-loop case. AIChE Journal, 50:1211-1218, 2004.

[42] T. Kourti. Multivariate dynamic data modeling for analysis and statistical process control of batch processes. Journal of Chemometrics, 17:93-109, 2003.

[43] W. Ku, R.H. Storer, and C. Georgakis. Disturbance detection and isolation by dynamic principal component analysis. Chemometrics and Intelligent Laboratory Systems, 30:179-196, 1995.

[44] T.L. Lai and C.Z. Wei. Extended least squares and their applications to adaptive control and prediction in linear systems. IEEE Trans. Autom. Control, 31:898-906, 1986.

[45] S. Lakshminarayanan, R.D. Gudi, S.L.Shah, and K. Nandakumar. Monitoring batch processes using multivariate statistical tools: Extensions and practical issues.

[46] K.S. Lee, S.H. Bang, S. Yi, J.S. Son, and Yoon. Iterative learning control of heat-up phase for a batch polymerization reactor. J. of Process Control, 6(4):255-262, 1996.

[47] K.S. Lee, I.S. Chin, and H.J. Lee. Model predictive control technique combined with iterative learning for batch processes. AIChE Journal, 45:2175-2187, 1999.

[48] B. Lennox, G. Montague, H. Hiden, G. Kornfeld, and P. Goulding. Process monitoring of an industrial fed-batch fermentation. Biotechnology and Bioengineering, 74:125., 2001.

[49] Z. Li and R.J. Evans. Minimum variance control of linear time-varying systems. Automatica, 33:1531-1537, 1997.

[50] Z. Li and R.J. Evans. Generalised minimum variance control of linear time-varying systems. IEE Pvoc-Control Theory Appl., 149(1):111-116, 2002.

[51] L. Ljung. System Identification: Theory for the User, 2nd ed. Prentice Hall, 1999.

[52] D. Louwerse and A. Smilde. Multivariate statistical process control of batch processes based on three-way models. Chemical Engineering Science, 55:1225-1235, 2000.

[53] C.B. Lynch and G.A. Dumont. Control loop performance monitoring. IEEE Trans. Control Sys. Tech., 4(2):185-192, 1996.

[54] A.I. Maalouf. Improving the robustness of a parallel robot using predictive functional control (pfc) tools. pages 6468-6473. Proceedings of the IEEE Conference on Decision and Control, 2006.

[55] O. Marjanovic, B. Lennox, D. Sandoz, K. Smith, and M. Crofts. Real-time monitoring of an industrial batch process. Computers and Chemical Engineering, 30:1476-1481, 2006.

[56] E.B. Martin and A.J. Morris. An overview of multivariate statistical process control in continuous and batch process performance monitoring. Transaction Instrument Measurement and Control, 18(1):51-60, 1996.

[57] J.H. Moon, T.Y. Doh, and M.J. Chung. A robust approach to iterative learning control design for uncertain systems. Automatica, 34:1001-1004, 1998.

[58] K.L. Moore. Iterative Learning Control for Deterministic Systems. Spring-Verlag, London, January 1993.

[59] Z.K. Nagy and R.D. Braatz. Worst-case and distributional robustness analysis of finite-time control trajectories for nonlinear distributed parameter systems. IEEE Trans. on Control Systems Technology, 11:694-704, 2003.

[60] Z.K. Nagy and R.D. Braatz. Open-loop and closed-loop robust optimal control of batch processes using distributional and worst-case analysis. J. of Process Control, 14:411-422, 2004.

[61] Z.K. Nagy and R.D. Braatz. Distributional uncertainty analysis using power series and polynomial chaos expansions. J. of Process Control, 17:229-240, 2007.

[62] P. Nomikos and J.F. MacGregor. Monitoring batch processes using multiway principal component analysis. AIChE Journal, 40(8):1361-1375, 1994.

[63] P. Nomikos and J.F. MacGregor. Multivariate spc charts for monitoring batch processes. Technometrics, 37(1):41-59, 1995.

[64] K.H. Park, Z. Bien, and D.H. Hwang. Design of an ilc for linear systems with time-delay and initial state error. pages 147-164, Netherlands, 1998. Kluwer Academic.

[65] T.H. Lee Q. Hu, J.-X. Xu . Iterative learning control design for smith predictor. Systems and control letters, 44:201-210, 2001.

[66] S.J. Qin and T.A. Badgwell. A survey of industrial model predictive control technology. Control Engineering Practice, 11:733-764, 2003.

[67] H. Ramaker, E. VanSprang, J. Westerhuis, and A. Smilde. Detection properties of global, local and time evolving models for batch process monitoring. J. of Process Control, 15(7):799-805, 2005.

[68] S. Rannar and J. MacGregor. Adaptive batch monitoring using hierarchical pca. Chemometrics and Intelligent Laboratory Systems, 41:73-81, 1998.

[69] J. Richalet, J. L. Estival, and Ph. Fiani. Industrial applications of predictive functional control to metallurgical industries. IEEE, pages 934-942, 1995.

[70] J. Richalet, A. Rault, J.L.Testud, and J. Papon. Model predictive heuristic control: Applications to industrial processes. Automatica, 14:413-428, 1978.

[71] P.D. Roberts. A Brief Overview of Model Predictive Control. The Institution of Electrical Engineers, London, UK, 1997.

[72] C.L. Roh, M.N. Lee, and M.J. Chung. Ilc for non-minimum phase system. Internat. J. of Systems Sci., 27:419-424, 1996.

[73] D.E. Seborg, S.L. Shah, and T.F. Edgar. Adaptive control strategies for process control: a survey. AIChE, 32:881-913, 1986.

[74] S. L. Shah, R. Patwardhan, and B. Huang. Multivariate controller performance analysis: Methods, applications and challenges. Proceedings of the Chemical Process Control Conference, 2001.

[75] A.K. Smilde. Three way analysis problems and prospects. Chemometrics and Intelligent Laboratory Systems, 15:143-157, 1992.

[76] T. Soderstorm and P. Stoica. System Identification. Prentice Hall, 1989.

[77] N. Stanfelj and T.E. MacGregor. Monitoring and diagnosing process control performance: the single-loop case. Ind. Eng. Chem. Res., 32:301-314, 1993.

[78] N.F. Thornhill, M. Oettinger, and P. Fedenczuk. Refinery-wide control loop performance assessment. J. of Process Control, 9(2):109-124, 1999.

[79] M.H.A. Verwoerd. Iterative learning control : a critical review. PhD thesis, Enschede, January 2005.

[80] Youqing Wang, Donghua Zhou, and Furong Gao. Iterative learning model predictive control for multi-phase batch processes. J. of Process Control, 18:543-557, 2008.

[81] A.J. Westerhuis, T. Kourti, and J.F. MacGregor. Comparing alternative approaches in batch process data analysis. Journal of Chemometrics, 13:397-413, 1999.

[82] B. Wise, N. Gallagher, S. Butler, D. White Jr., and G. Barna. A comparison of principal component analysis, multiway principal component analysis, trilinear decomposition and parallel factor analysis for fault detection in a semiconductor etch process. Journal of Chemometrics, 13:379-396, 1999.

[83] J.-X. Xu, Q. Hu, T.H. Lee, and S. Yamamoto. Iterative learning control with smith time delay compensator for batch processes. J. of Process Control, 11:321-328, 2001.

[84] J.X. Xu, S.K. Panda, and T.H. Lee. Real-time iterative learning control. Springer, 2009.

[85] B. Erik Ydstie and L.K. Liu. Single and multi-variable control with extended prediction horizons. pages 1303-1308. Proceedings of the ACC, 1984.

[86] B.E. Ydstie. Extended horizon adaptive control. Proceedings of the IFAC, 1984.

Chapter 2

Performance monitoring of iterative learning control for linear time-varying processes

2.1 Introduction

Iterative learning control (ILC) is a strategy recommended for batch processes in order to improve the control performance in terms of set-point tracking and disturbance rejection. This strategy was first introduced by Arimoto and Kawamura (1984) [2] for robot systems. The idea is to use the data collected from previous trials to control the current trial. In the case of batch processes, each operation has a certain time duration, called a batch. To implement ILC, an algorithm is defined to use previous batches for the control of the current batch [4, 12, 16]. Design of an ILC algorithm requires reliable models [18, 23, 24, 27]. But models have several kinds of uncertainties in practice, which make the design procedure more challenging [1, 6, 20, 21, 22]. To understand more about different characteristics of ILC, readers are referred to the reviews by Verwoerd (2005) [26] and Moore (1993) [19].

While operating on-line, performance of a process needs to be monitored to ensure satisfaction of the implemented control strategy. Many methods have been proposed in literature to monitor the performance of batch processes for different control strategies [25, 8, 17]. For the implementation of ILC, Kong and Chen (2009) [3] have recently suggested a method to monitor the performance based on the minimum variance benchmark. Although claimed applicable for time-varying batch processes, this method fails to consider the noncommutativity property of time-varying models by doing point-wise multiplication. This chapter proposes an extension to the existing method in order to correctly estimate the benchmark and monitor the performance of ILC loops when the batch process has time-varying property.

The remainder of this chapter is structured as follows: The noncommutativity property of time-varying models and multiplication rules are explained in Section 2. Section 3 introduces the ILC set-up and its algorithm as well as the minimum variance benchmark estimation and performance monitoring of time-varying batch processes. Simulation studies are provided in Section 4, followed by concluding remarks in Section 5.

2.2 Linear time-varying models

In this chapter, linear time-varying (LTV) models are assumed to maintain a constant structure, but parameters change over time. Suppose a process is represented by an LTV model with d samples of delay. Depending on the location of the delay, two structures are possible, as shown in Figure 2.2.

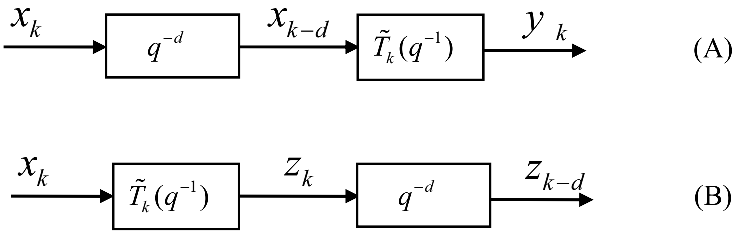


Figure 2.1: A comparison between two LTV structures

The delay-free plant model is represented by $\tilde{T}_k(q^{-1})$, and x_k denotes the input to the process. In structure (A), the delayed input (x_{k-d}) enters the plant model with parameters at time k. But in structure (B), the input at time k enters the plant model with parameters at the same time, and the output is delayed. Therefore, the outputs of (A) and (B) will not be identical. However, if the parameters of the plant model are shifted forward by d time steps in structure (B), the same output will be expected since the same model parameters are executed upon the input. Putting this in mathematical terms, we have:

$$q^{-d}\tilde{T}_k(q^{-1}) = \tilde{T}_{k-d}(q^{-1})q^{-d} \tag{2.1}$$

Equation 2.1 indicates a unique property of LTV models, called noncommutativity [14]. Based on this property, the back-shift operator alters the order of the parameters with respect to time. Assume that two LTV polynomials are given as:

$$A_k(q^{-1}) = a_{0,k} + a_{1,k}q^{-1} + \cdots + a_{n,k}q^{-n} \tag{2.2}$$

$$B_k(q^{-1}) = b_{0,k} + b_{1,k}q^{-1} + \cdots + b_{m,k}q^{-m} \tag{2.3}$$

Two methods for the multiplication of these polynomials can be used [13]:

[1] Normal multiplication, which considers the noncommutativity property by changing the order of the parameters with respect to time after the back-shift operator is moved:

$$A_k(q^{-1})B_k(q^{-1}) = \sum_{i=0}^{n}\sum_{j=0}^{m} a_{i,k}q^{-i}b_{j,k}q^{-j} = \sum_{i=0}^{n}\sum_{j=0}^{m} a_{i,k}b_{j,k-i}q^{-(i+j)} \tag{2.4}$$

$$B_k(q^{-1})A_k(q^{-1}) = \sum_{i=0}^{m}\sum_{j=0}^{n} b_{i,k}q^{-i}a_{j,k}q^{-j} = \sum_{i=0}^{m}\sum_{j=0}^{n} b_{i,k}a_{j,k-i}q^{-(i+j)} \tag{2.5}$$

Changing the location of the polynomials during the multiplication procedure affects the output. Therefore, normal multiplication is noncommutative:

$$A_k(q^{-1})B_k(q^{-1}) \neq B_k(q^{-1})A_k(q^{-1}) \tag{2.6}$$

[2] Point-wise multiplication, which disregards the noncommutativity property by maintaining the order of the parameters with respect to time after the back-shift operator is moved:

$$A_k(q^{-1})B_k(q^{-1}) = \sum_{i=0}^{n}\sum_{j=0}^{m} a_{i,k}q^{-i}b_{j,k}q^{-j} = \sum_{i=0}^{n}\sum_{j=0}^{m} a_{i,k}b_{j,k}q^{-(i+j)} \tag{2.7}$$

$$B_k(q^{-1})A_k(q^{-1}) = \sum_{i=0}^{m}\sum_{j=0}^{n} b_{i,k}q^{-i}a_{j,k}q^{-j} = \sum_{i=0}^{m}\sum_{j=0}^{n} b_{i,k}a_{j,k}q^{-(i+j)} \tag{2.8}$$

Therefore the point-wise multiplication is commutative:

$$A_k(q^{-1})B_k(q^{-1}) = B_k(q^{-1})A_k(q^{-1}) \tag{2.9}$$

Although both these methods have been used for the multiplication of LTV models, normal multiplication is the appropriate one because it provides accurate solutions by considering the noncommutativity property. Extending normal multiplication for LTV models, we find:

$$G_k = B_k^{-1}(q^{-1})A_k(q^{-1}) \tag{2.10}$$

$$H_k = D_k^{-1}(q^{-1})C_k(q^{-1}) \tag{2.11}$$

$$G_k * H_k = B_k^{-1}(q^{-1})A_k(q^{-1})D_k^{-1}(q^{-1})C_k(q^{-1}) \tag{2.12}$$

This chapter will use normal multiplication for design and implementation of LTV models. To demonstrate the differences of this method from the point-wise multiplication, simulation studies will also be carried out.

2.3 Control performance monitoring of ILC

2.3.1 ILC set-up

Figure 2.2 illustrates a recommended structure for ILC [3]. Note that $T_k(q^{-1})$, $N_k(q^{-1})$, $C_k(q^{-1})$, and $L_k(q^{-1})$ are linear time-varying models representing the plant, disturbance, inner controller, and outer controller, respectively. i, k, and q^{-1} denote batch index, time index, and time back-shift operator.

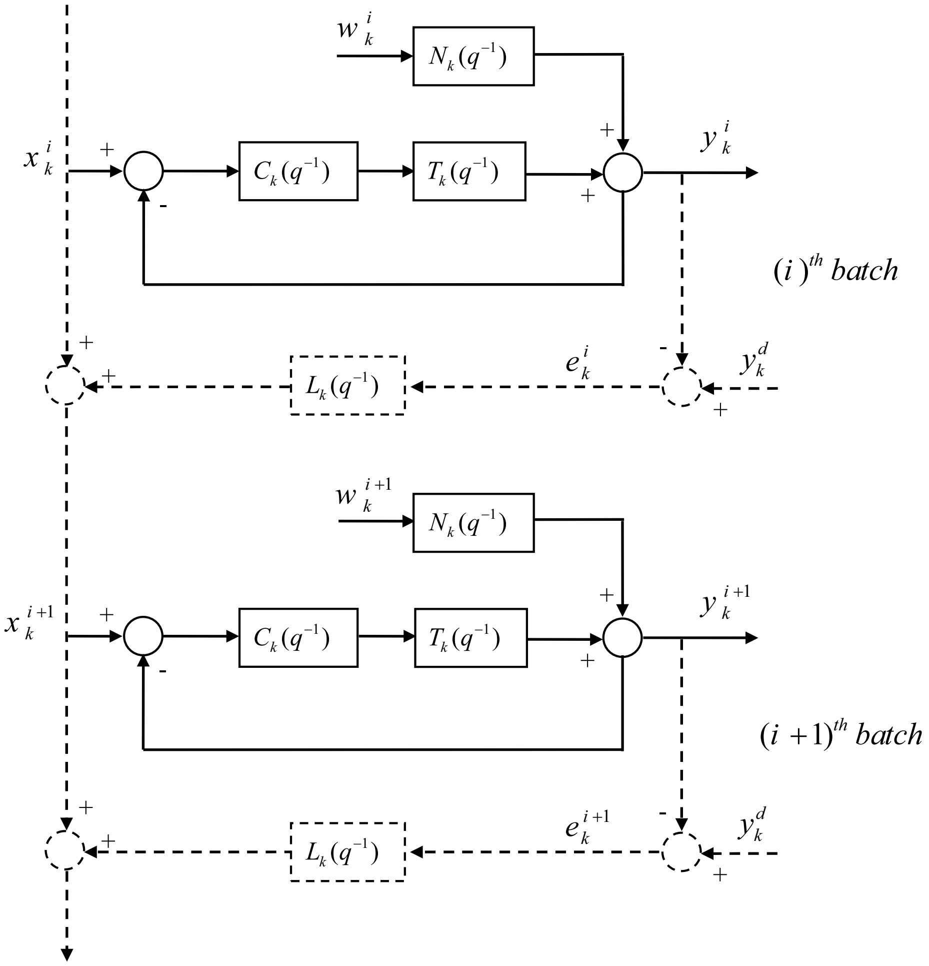


Figure 2.2: A recommended ILC structure

This structure can be used for the control of different batches with a constant batch duration. Essentially, two loops determine the structure: the inner loop, drawn with solid lines, and the outer loop, drawn with dashed lines. The inner loop operates in the same way as the conventional closed-loop feedback control. At each batch, a reference (x_k^i) is set to the inner controller. When finishing a batch, the output is compared with the set-point (y_k^d) and the error

(e_k^i) is computed in the outer loop, where the outer controller filters the error before adding it to the current reference for the next batch.

The design procedure for ILC includes deriving the inner and the outer controllers in order to achieve desirable performance in terms of set-point tracking and disturbance rejection. The inner controller operates on-line in the inner loop, while the outer controller operates off-line in the outer loop. The purpose of the off-line operation is to set the reference signal such that the inner loop tracks the set-point. The term "off-line" is in the sense that the outer control is executed after the previous batch has been completed. Since the reference signal is set by the outer controller, the inner controller can focus on disturbance rejection.

2.3.2 Minimum variance control law

The first step in performance monitoring of a process is to establish a benchmark. This thesis adopts the minimum variance benchmark. This benchmark is based on the minimum variance control law, according to which minimization of error variance yields optimal output. To derive the error expression for the ILC algorithm, the following process model is considered:

$$y_k^i = T_k u_k^i + N_k w_k^i \tag{2.13}$$

where u_k^i is the input to the plant, set by the inner controller. Note that for the sake of simplicity, the back-shift operator has been dropped. However, the subscript k in the plant and disturbance models reflects the time-varying property of the process models.

Considering the update rule of ILC, as illustrated in Figure 2.2, closed-loop model of the process is as follows:

$$y_k^i = (1 + T_k C_k)^{-1} T_k C_k x_k^i + (1 + T_k C_k)^{-1} N_k w_k^i \tag{2.14}$$

where

$$x_k^i = x_k^{i-1} + L_k (y_k^d - y_k^{i-1}) \tag{2.15}$$

Since all the variables are two-dimensional, another back-shift operator is required to shift the data across batches. Let us define z^{-1} as the batch back-shift operator:

$$z^{-1} x_k^i = x_k^{i-1} \tag{2.16}$$

Rewriting the ILC update rule gives:

$$x_k^i = (1 - z^{-1})^{-1} L_k y_k^d - (1 - z^{-1})^{-1} z^{-1} L_k y_k^i \tag{2.17}$$

Substituting the reference signal in Equation 2.14 with Equation 2.17 yields:

$$\begin{aligned} y_k^i &= (1+T_kC_k)^{-1}N_kw_k^i \\ &\quad + (1+T_kC_k)^{-1}T_kC_k[(1-z^{-1})^{-1}L_ky_k^d - (1-z^{-1})^{-1}z^{-1}L_ky_k^i] \end{aligned} \tag{2.18}$$

which can be further simplified as:

$$\begin{aligned} &[1+(1-z^{-1})^{-1}(1+T_kC_k)^{-1}z^{-1}T_kC_kL_k]y_k^i \\ &= (1+T_kC_k)^{-1}N_kw_k^i + (1-z^{-1})^{-1}(1+T_kC_k)^{-1}T_kC_kL_ky_k^d \end{aligned} \tag{2.19}$$

Multiplying both sides of Equation 2.19 by $(1-z^{-1})(1+T_kC_k)$ gives:

$$\begin{aligned} &[(1-z^{-1})(1+T_kC_k)+z^{-1}T_kC_kL_k]y_k^i \\ &= (1-z^{-1})N_kw_k^i + T_kC_kL_ky_k^d \end{aligned} \tag{2.20}$$

Hence the output can be written as a function of the set-point and the white noise:

$$\begin{aligned} y_k^i &= [(1-z^{-1})(1+T_kC_k)+z^{-1}T_kC_kL_k]^{-1}(1-z^{-1})N_kw_k^i \\ &\quad + [(1-z^{-1})(1+T_kC_k)+z^{-1}T_kC_kL_k]^{-1}T_kC_kL_ky_k^d \end{aligned} \tag{2.21}$$

Note that in the derivation of Equation 2.21, we have followed the rule of normal multiplication without altering sequence of terms in the multiplications. Once the output expression is found, the error can be derived as:

$$e_k^i = y_k^d - y_k^i \tag{2.22}$$

$$= [(1-z^{-1})(1+T_kC_k)+z^{-1}T_kC_kL_k]^{-1}(1-z^{-1})(1+T_kC_k-T_kC_kL_k)y_k^d \tag{2.23}$$

$$- [(1-z^{-1})(1+T_kC_k)+z^{-1}T_kC_kL_k]^{-1}(1-z^{-1})N_kw_k^i \tag{2.24}$$

The error is composed of two parts: the deterministic part which is derived from the set-point and the stochastic part which is derived from the white noise. Therefore, the error term can be split, as shown below:

$$e_k^i = e_k^{i,det} + e_k^{i,sto} \tag{2.25}$$

where

$$\begin{aligned} e_k^{i,det} =& [(1-z^{-1})(1+T_kC_k)+z^{-1}T_kC_kL_k]^{-1}(1-z^{-1}) \\ & (1+T_kC_k-T_kC_kL_k)y_k^d \end{aligned} \tag{2.26}$$

$$e_k^{i,sto} = [(1-z^{-1})(1+T_kC_k)+z^{-1}T_kC_kL_k]^{-1}(1-z^{-1})N_kw_k^i \tag{2.27}$$

Error variance minimization is considered in order to achieve the optimal set-point tracking and disturbance rejection. The performance of set-point tracking can be determined through the deterministic error. Similarly, disturbance rejection can be determined by the stochastic error. Therefore, two cost functions are defined:

$$
\begin{aligned}
J_{det} =& E[(e_k^{i,det})^2] && (2.28)\\
=& E[([(1-z^{-1})(1+T_kC_k)+z^{-1}T_kC_kL_k]^{-1} \\
& (1-z^{-1})(1+T_kC_k-T_kC_kL_k)y_k^d)^2] && (2.29)\\
J_{sto} =& E[(e_k^{i,sto})^2] && (2.30)\\
=& E[([(1-z^{-1})(1+T_kC_k)+z^{-1}T_kC_kL_k]^{-1}(1-z^{-1})N_kw_k^i)^2] && (2.31)
\end{aligned}
$$

By minimizing the deterministic cost function, the optimal outer controller is derived:

$$L_k^{opt} = (T_kC_k)^{-1}[1+T_kC_k] = C_k^{-1}T_k^{-1}[1+T_kC_k] \tag{2.32}$$

From Equation 2.29, the optimal outer controller equates the deterministic cost function to zero. Hence after one operation of the outer loop, the deterministic error becomes zero, and perfect tracking of the set-point is reached.

Note that owing to the delays in the plant model, the optimal outer controller has non-causal operators, which require future data for computation purposes. Since this controller executes off-line, the error signal is known at all time steps. Hence the implementation of non-causal operators is feasible. More details on non-causal iterative learning control can be found in literature [5, 26].

Substituting the optimal controller in Equation 2.31, the stochastic cost function is simplified:

$$J_{sto} = E[((1-z^{-1})[1+T_kC_k]^{-1}N_kw_k^i)^2] \tag{2.33}$$

From the Diophantine identity [9], we can split the disturbance model:

$$N_k = F_k + R_kq^{-d} \tag{2.34}$$

where d is the time delay of the process, and F_k includes the first d terms of the disturbance model in impulse response form. Substituting Equation 2.34 into Equation 2.33 yields:

$$
\begin{aligned}
J_{sto} &= E[((1-z^{-1})[1+T_kC_k]^{-1}[F_k+R_kq^{-d}]w_k^i)^2] && (2.35)\\
&= E[((1-z^{-1})F_kw_k^i+[1+T_kC_k]^{-1}(1-z^{-1})[R_kq^{-d}-T_kC_kF_k]w_k^i)^2] && (2.36)\\
&= E[((1-z^{-1})F_kw_k^i+[1+T_kC_k]^{-1}(1-z^{-1})q^{-d}[R_{k+d}-\tilde{T}_kC_kF_k]w_k^i)^2] && (2.37)
\end{aligned}
$$

Note that $\tilde{T}_k$ represents the delay-free plant model and can be written as:

$$\tilde{T}_k = q^dT_k \qquad (2.38)$$

The second term of Equation 2.37 has a d-step time delay, while the first one has finite impulse response form containing white noise terms $w_{k-d+1}^i, \ldots, w_k^i$. Therefore, the two terms are uncorrelated. Since the first term is not dependent on any controllers, it is called the controller independent term. To minimize the cost function, the optimal inner controller can be set to equate the second term of Equation 2.37 with zero. The following equation gives the optimal inner controller:

$$C_k^{opt} = \tilde{T}_k^{-1}R_{k+d}F_k^{-1} \qquad (2.39)$$

From Equation 2.37, the optimal value of the stochastic cost function can be found by applying the optimal inner controller:

$$
\begin{aligned}
J_{sto}^{opt} &= E[((1-z^{-1})F_kw_k^i)^2] && (2.40)\\
&= E[(F_ka_k^i)^2] && (2.41)
\end{aligned}
$$

where a_k^i is a white noise whose variance is double of that of w_k^i, and defined as:

$$a_k^i = w_k^i - w_k^{i-1} \qquad (2.42)$$

2.3.3 Performance monitoring

Minimum variance benchmark

In general, to monitor the performance of a control strategy, an acceptable operating range for the output needs to be defined. For ILC strategy, we define an acceptable range for the output error of each batch operation, and check if the current error falls within that range. Basically, the expected error is calculated for each batch, and upper and lower control limits are introduced.

The expected error of each batch can be written as:

$$E[e_k^i] = E[e_k^{i,det}] + E[e_k^{i,sto}] \tag{2.43}$$

where

$$\begin{aligned} E[e_k^{i,det}] =& E[[(1-z^{-1})(1+T_kC_k)+z^{-1}T_kC_kL_k]^{-1} \\ & (1-z^{-1})(1+T_kC_k-T_kC_kL_k)y_k^d] \end{aligned} \tag{2.44}$$

$$E[e_k^{i,sto}] =E[[(1-z^{-1})(1+T_kC_k)+z^{-1}T_kC_kL_k]^{-1}(1-z^{-1})(1-z^{-1})N_kw_k^i] = 0 \tag{2.45}$$

An ILC is considered to have good performance if the difference between the error term of each batch and the expected error is less than three times of the standard deviation of the optimal error [3]. To put it in mathematical terms, the following upper and lower control limits are defined:

$$UCL = E[e_k^i] + 3\sigma_k \tag{2.46}$$

$$LCL = E[e_k^i] - 3\sigma_k \tag{2.47}$$

where UCL and LCL are the upper and lower control limits, and σ_k is the standard deviation of the optimal error computed with parameters at time k. Since the optimal deterministic error is zero, the optimal error is composed of only the stochastic error. Hence the variance of the optimal error is equal to the optimal value of the stochastic cost function, which can be calculated as [10]:

$$\sigma_k^2 = \sigma_a^2 \int_{-\pi}^{\pi} \phi_e(\omega) d\omega \tag{2.48}$$

where σ_a^2 is the variance of a_k^i and

$$\phi_e(\omega) = \frac{1}{2\pi} \mid F_k(e^{-j\omega}) \mid^2 \tag{2.49}$$

This leads to:

$$\sigma_k^2 = \sigma_a^2 \int_{-\pi}^{\pi} \frac{1}{2\pi} \mid F_k(e^{-j\omega}) \mid^2 d\omega = \|F_k\|_2^2 \sigma_a^2 \tag{2.50}$$

It has been proved that the square of the 2-norm of a transfer function in impulse response form is equal to the summation of the squares of its impulse response coefficients [7]. In other words:

$$\|F_k\|_2^2 = \sum_j f_{j,k}^2 \tag{2.51}$$

where $f_{j,k}$ is an impulse response coefficient of F_k at time k, corresponding to q^{-j}.

By defining the upper and lower control limits, the control benchmark is established [3]. The error for each batch is expected to remain within the boundaries of UCL and LCL. If the error exceeds the limits of this range, the performance is deviated from the benchmark, and tuning of the control strategy can be considered.

Time-varying model identification

The computation of control limits requires knowledge of the process model and F_k, and in ILC case, identification of these two models from routine operating data is feasible. Simplifying the closed-loop model, originally presented in Equation 2.14, yields:

$$y_k^i = T_k^{cl} x_k^i + N_k^{cl} w_k^i \tag{2.52}$$

where T_k^{cl} and N_k^{cl} are the deterministic and stochastic closed-loop models, equal to:

$$T_k^{cl} = 1 - [1 + T_k C_k]^{-1} \tag{2.53}$$

$$N_k^{cl} = [1 + T_k C_k]^{-1} N_k \tag{2.54}$$

After collecting data, identification is performed with regard to Equation 2.52. For this purpose, Kalman filter algorithm is applied to recursively estimate the time-varying closed-loop models [15]. To explain the procedure, assume that θ_k and $\hat{\theta}_k$ represent the actual and estimated parameters of linear time-varying models. The following is the general equation for the recursive identification algorithm:

$$\hat{\theta}_k = \hat{\theta}_{k-1} + K_k (y_k^i - \hat{y}_k^i) \tag{2.55}$$

where $\hat{y}_k^i$ is the predicted output. The gain, K_k, determines the effects of the prediction error $y_k^i - \hat{y}_k^i$ on the estimated parameters and has the following form:

$$K_k = Q_k \psi_k \tag{2.56}$$

where Q_k and ψ_k are the weighting variable and the gradient of the predicted output $\hat{y}_k^i$ with respect to θ, respectively. For models with linear-regression form, the output can be modelled as:

$$y_k^i = \psi_k^T \theta_k + e_k^i \tag{2.57}$$

Note that e_k^i represents the noise source. The predicted output is also given by:

$$\hat{y}_k^i = \psi_k^T \hat{\theta}_{k-1} \tag{2.58}$$

Assume that the actual parameters can be described by a random walk, as shown below:

$$\theta_k = \theta_{k-1} + w_k \tag{2.59}$$

In Equation 2.59, w_k is a white noise with the covariance matrix of R_1. From Equation 2.56, Q_k is needed for the computation of the gain. Based on the Kalman filter algorithm [15], Q_k can be calculated from:

$$Q_k = \frac{P_{k-1}}{R_2 + \psi_k^T P_{k-1} \psi_k} \tag{2.60}$$

where R_2 is the covariance of e_k^i in Equation 2.57 and

$$P_k = P_{k-1} + R_1 - \frac{P_{k-1}\psi_k \psi_k^T P_{k-1}}{R_2 + \psi_k^T P_{k-1} \psi_k} \tag{2.61}$$

At each sample time, the Kalman gain is computed from Equation 2.56. Then models parameters can be estimated from Equation 2.55.

Next step is to recover the plant model and F_k from the estimated T_k^{cl} and N_k^{cl}. When doing this, care has to be taken to the noncommutativity problem by following the rules of normal multiplication. To explore the problem, consider the following two cases:

- Case 1: estimating F_k

  Substitute the Diophantine identity, Equation 2.34, into the stochastic closed-loop model, Equation 2.54:

$$\begin{aligned}
N_k^{cl} &= [1 + T_k C_k]^{-1}[F_k + R_k q^{-d}] \\
&= [1 + T_k C_k]^{-1}[F_k + T_k C_k F_k + R_k q^{-d} - T_k C_k F_k] \\
&= F_k + [1 + T_k C_k]^{-1}[R_k q^{-d} - T_k C_k F_k] \\
&= F_k + [1 + T_k C_k]^{-1}[R_k q^{-d} - q^{-d} \tilde{T}_k C_k F_k] \\
&= F_k + [1 + T_k C_k]^{-1} q^{-d}[R_{k+d} - \tilde{T}_k C_k F_k] \qquad (2.62)
\end{aligned}$$

  The second term of Equation 2.62 possesses d samples of delay. Hence the stochastic closed-loop model can be rearranged to give:

$$N_k^{cl} = F_k + \acute{R}_k q^{-d} \tag{2.63}$$

From the Diophantine identity, F_k is composed of the first d terms of the disturbance model in impulse response form. Similarly, the first d terms of the stochastic closed-loop model in impulse response form compose F_k. Hence the estimated stochastic closed-loop model needs to be transferred to an LTV moving average (MA) model. Then the first d terms are chosen to compose F_k. To verify the necessity of using normal multiplication in estimating the MA model, consider the following structure for F_k:

$$F_k = f_{0,k} + f_{1,k}q^{-1} + \cdots + f_{d-1,k}q^{d-1} \tag{2.64}$$

As an example, consider a stochastic closed-loop model N_k^{cl} is identified as an LTV AR model, which can be expanded as follows [11]:

$$[1 - \lambda_k q^{-1}]^{-1} = f_{0,k} + f_{1,k}q^{-1} + \cdots + f_{d-1,k}q^{d-1} + R_k q^{-d} \tag{2.65}$$

Simplifying Equation 2.65 yields:

$$\begin{aligned} 1 =& [1 - \lambda_k q^{-1}][f_{0,k} + f_{1,k}q^{-1} + \cdots + f_{d-1,k}q^{d-1} + R_k q^{-d}] & (2.66)\\ =& f_{0,k} + [f_{1,k} - \lambda_k f_{0,k-1}]q^{-1} + [f_{2,k} - \lambda_k f_{1,k-1}]q^{-2} + \cdots \\ &+ [f_{d-1,k} - \lambda_k f_{d-2,k-1}]q^{-d+1} + [R_k - \lambda_k f_{d-1,k-1} - \lambda_k q^{-1} R_k]q^{-d} & (2.67) \end{aligned}$$

Equating coefficients of the both hand sides of Equation 2.67 gives:

$$\begin{aligned} f_{0,k} &= 1 \\ f_{1,k} &= \lambda_k \\ f_{2,k} &= \lambda_k \lambda_{k-1} \\ &\vdots \\ f_{d-1,k} &= \lambda_k \lambda_{k-1} \cdots \lambda_{k-d+2} \end{aligned}$$

Repeating the procedure by doing point-wise multiplication would yield:

$$F_k = 1 + \lambda_k q^{-1} + \lambda_k \lambda_k q^{-2} + \cdots + \lambda_k \lambda_k \cdots \lambda_k q^{-d+1} \tag{2.68}$$

which is incorrect. Hence following the rules of normal multiplication is necessary when transferring the estimated stochastic closed-loop model into a moving average model.

- Case 2: estimating T_k

  Recall the deterministic closed-loop model, Equation 2.53:

$$T_k^{cl} = 1 - [1 + T_k C_k]^{-1} \tag{2.69}$$

  As an example, suppose the process possesses a first order plant model, and a proportional controller is implemented, as shown below:

$$T_k = (1 - \Lambda q^{-1})^{-1} \gamma_k q^{-d} \tag{2.70}$$
$$C_k = \kappa_k \tag{2.71}$$

  Substituting Equations 2.70 and 2.71 into Equation 2.69 yields:

$$\begin{aligned} T_k^{cl} &= 1 - [1 + (1 - \Lambda q^{-1})^{-1} \gamma_k q^{-d} \kappa_k]^{-1} \\ &= 1 - [(1 - \Lambda q^{-1})^{-1}(1 - \Lambda q^{-1} + \gamma_k q^{-d} \kappa_k)]^{-1} \\ &= 1 - (1 - \Lambda q^{-1} + \gamma_k \kappa_{k-d} q^{-d})^{-1}(1 - \Lambda q^{-1}) \\ &= (1 - \Lambda q^{-1} + \gamma_k \kappa_{k-d} q^{-d})^{-1}((1 - \Lambda q^{-1} + \gamma_k \kappa_{k-d} q^{-d}) - (1 - \Lambda q^{-1})) \\ &= (1 - \Lambda q^{-1} + \gamma_k \kappa_{k-d} q^{-d})^{-1} \gamma_k \kappa_{k-d} q^{-d} \end{aligned} \tag{2.72}$$

  Performing identification, the estimated deterministic closed-loop model should have the following form:

$$\hat{T}_k^{cl} = (1 - \Lambda q^{-1} + b_k q^{-d})^{-1} b_k q^{-d} \tag{2.73}$$

  Let us assume that perfect model identification results are obtained. From Equation 2.72, we know that the estimated parameter of b_k is equal to $\gamma_k \kappa_{k-d}$. Based on the inverse of Equation 2.69, the plant model can be recovered from:

$$\hat{T}_k = ([1 - \hat{T}_k^{cl}]^{-1} - 1) C_k^{-1} \tag{2.74}$$

  Substituting Equation 2.73 into Equation 2.74 gives:

$$\begin{aligned} \hat{T}_k &= ([1 - (1 - \Lambda q^{-1} + b_k q^{-d})^{-1} b_k q^{-d}]^{-1} - 1) \kappa_k^{-1} \\ &= ([(1 - \Lambda q^{-1} + b_k q^{-d})^{-1}(1 - \Lambda q^{-1} + b_k q^{-d} - b_k q^{-d})]^{-1} - 1) \kappa_k^{-1} \\ &= [(1 - \Lambda q^{-1})^{-1}(1 - \Lambda q^{-1} + b_k q^{-d}) - 1] \kappa_k^{-1} \\ &= [(1 - \Lambda q^{-1})^{-1} b_k q^{-d}] \kappa_k^{-1} \\ &= (1 - \Lambda q^{-1})^{-1} b_k \kappa_{k-d}^{-1} q^{-d} \\ &= (1 - \Lambda q^{-1})^{-1} \gamma_k q^{-d} \end{aligned}$$

which is equal to the actual plant model. For the sake of comparison, we also derive the plant model by using point-wise multiplication. Substituting Equation 2.73 into Equation 2.74 and simplifying the result, we have:

$$\begin{aligned}\hat{T}_k &= [(1-\Lambda q^{-1})^{-1} b_k q^{-d}]\kappa_k^{-1} \\ &= (1-\Lambda q^{-1})^{-1} b_k \kappa_k^{-1} q^{-d}\end{aligned}$$

Replacing b_k with its actual value, $\gamma_k \kappa_{k-d}$, yields:

$$\hat{T}_k = (1-\Lambda q^{-1})^{-1} \gamma_k \kappa_{k-d} \kappa_k^{-1} q^{-d}$$

which is not equal to the actual plant model. Although the estimation of the deterministic closed-loop model is assumed to be sufficiently good, point-wise multiplication for recovering the plant model brings inaccuracy into the estimation.

As shown in the above two cases, using point-wise multiplication can be problematic particularly when the time-varying parameters vary sufficiently fast. Therefore, it is necessary to use normal multiplication for recovering desired models from the estimated closed-loop models if the batch process does have time-varying property.

2.4 Simulation studies

Consider the following process and disturbance models [11]:

$$T_k = (1-0.67q^{-1})^{-1} \gamma_k q^{-4} \tag{2.75}$$
$$N_k = (1-\lambda_k q^{-1})^{-1} \tag{2.76}$$

where γ_k and λ_k are time-varying parameters. This process is expected to follow the set-point given in Figure 2.4.

Two ILC strategies are designed for control of this process. Both strategies implement the optimal inner and outer controllers, but the first uses normal multiplication for design purposes, while the second uses point-wise multiplication. After collecting routine batch operating data, models parameters are estimated using the time varying model identification algorithm, presented in the previous section.

Three case studies with different rates of parameter change over time are considered:

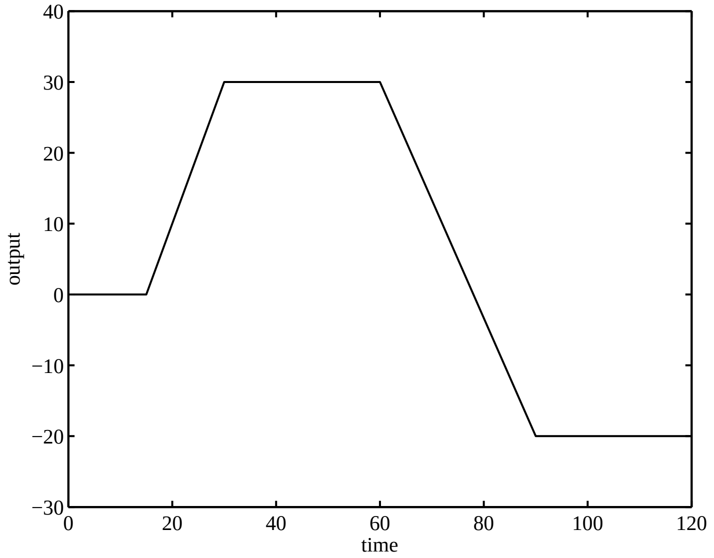


Figure 2.3: Set-point for the simulation studies

- Case 1) Quick change of time-varying parameters is considered:

$$\gamma_k = 0.33 + 0.3sin(0.2k + 0.1) \tag{2.77}$$

$$\lambda_k = 0.9|sin(0.2k + 0.1)| \tag{2.78}$$

  Figures 2.4 and 2.5 illustrate the simulation results for performance monitoring of both strategies. The second strategy needs more batches to achieve perfect tracking of the set-point. In terms of the performance, the first strategy yields satisfactory outputs for all batches. But the second one yields unsatisfactory outputs for the first few batches, which is due to disregarding the noncommutativity property of LTV models.

- Case 2) Slower change of time-varying parameters is considered:

$$\gamma_k = 0.33 + 0.3sin(0.05k + 0.1) \tag{2.79}$$

$$\lambda_k = 0.9|sin(0.05k + 0.1)| \tag{2.80}$$

  Simulation results of performance monitoring are given in Figures 2.6 and 2.7. It is clear that a few batches exceed the benchmark limits after the implementation of the second strategy. However, compared to the previous case, there is less difference between the outputs of the two strategies.

- Case 3) All the parameters remain constant over time:

  $$\gamma_k = 0.5 \tag{2.81}$$
  $$\lambda_k = 0.6 \tag{2.82}$$

  Comparing the simulation results of both strategies, given in Figure 2.8, no difference between the outputs is identified. Since the models are time-invariant, both normal and point-wise multiplication methods yield the same results.

According to the simulations, the slower the model parameters change, i.e. the less models are dependent on time, the less difference is made between the results of normal and point-wise multiplication. If the parameters remain constant over time, there will be no differences between the results of the multiplication methods, which is expected since the models are time-invariant.

2.5 Conclusions

This chapter addressed performance monitoring of ILC loops for linear time-varying (LTV) batch processes. After a literature review, the basic noncommutativity property of LTV models was explained. Two multiplication methods were introduced: normal multiplication which considers time varying property, and point-wise multiplication which disregards it. Following the rules of normal multiplication, the minimum variance benchmark for the ILC algorithm of time-varying processes was established, and the procedure for monitoring the performance based on the proposed benchmark was explained. To verify the necessity of using normal multiplication, two control strategies were designed using normal and point-wise multiplications respectively, and simulation results illustrated the difference. The faster model parameters change over time, the less accurate results are obtained using point-wise multiplication. However, regardless of the rate of parameter change, the use of normal multiplication is recommended for design and implementation of ILC strategy for LTV batch processes.

References

[1] H.-S. Ahn, K.L. Moore, and Y. Chen. Iterative Learning Control, Robustness and Monotonic Convergence for Interval Systems. Spring-Verlag, 2007.

[2] S. Arimoto, S. Kawamura, and F. Miyazaki. Iterative learning control for robot systems. Tokyo, Japan, 1984. Proceedings of IECON.

[3] J. Chen and C.K. Kong. Performance assessment for iterative learning control of batch units. J. of Process Control, 19:1043-1053, 2009.

[4] Y. Chen, J.X. Xu, T.H. Lee, and S. Yamamoto. Comparative studies of iterative learning control schemes for a batch chemical process. Singapore, 1997. IEEE Int. Symposium on Control Theory and Applications.

[5] B.G. Dijkstra. Iterative learning control, with applications to a wafer stage. PhD thesis, Delft University of Technology, 2003.

[6] T.Y. Doh, J.H. Moon, K.B. Jin, and M.J. Chung. Robust iterative learning control with current feedback for uncertain linear systems. Internat. J. of Systems Sci., 30:39-47, 1999.

[7] J.C. Doyle, B.A. Francis, and A.R. Tannenbaum. Feedback control theory. Macmillan publishing company, 1992.

[8] M. Guo, L. Xie, and S q. Wang. Model-based multiscale performance monitoring for batch process. pages 357-360. Proceedings of the IEEE Int. Conf. Neural Networks and Signal Processing, 2003.

[9] T. Harris. Assessment of control loop performance. The Canadian J. of Chemical Engineering, 67:856-861, 1989.

[10] T. Soderstorm and P. Stoica. System Identification. Prentice Hall, 1989.

[11] B. Huang. Minimum variance control and performance assessment of time-variant processes. J. of Process Control, 12:707-719, 2002.

[12] J.H. Lee and K.S. Lee. Iterative learning control applied to batch processes: An overview. Control Engineering Practice, 15:1306-1318, 2007.

[13] Z. Li and R.J. Evans. Minimum variance control of linear time-varying systems. Automatica, 33:1531-1537, 1997.

[14] Z. Li, R.J. Evans, and B. Wittenmark. Minimum variance prediction for linear time-varying systems. Automatica, 33(4):607-618, 1997.

[15] L. Ljung. System Identification: Theory for the User, 2nd ed. Prentice Hall, 1999.

[16] P. Lucibello. Learning control of linear systems. Chicago, US, 1992. Proceedings of ACC.

[17] L.A. McPherson. Model-based performance monitoring of batch processes. PhD thesis, Newcastle University, 2008.

[18] J.H. Moon, T.Y. Doh, and M.J. Chung. A robust approach to iterative learning control design for uncertain systems. Automatica, 34:1001-1004, 1998.

[19] K.L. Moore. Iterative Learning Control for Deterministic Systems. Spring-Verlag, London, January 1993.

[20] Z.K. Nagy and R.D. Braatz. Worst-case and distributional robustness analysis of finite-time control trajectories for nonlinear distributed parameter systems. IEEE Trans. on Control Systems Technology, 11:694-704, 2003.

[21] Z.K. Nagy and R.D. Braatz. Open-loop and closed-loop robust optimal control of batch processes using distributional and worst-case analysis. J. of Process Control, 14:411-422, 2004.

[22] Z.K. Nagy and R.D. Braatz. Distributional uncertainty analysis using power series and polynomial chaos expansions. J. of Process Control, 17:229-240, 2007.

[23] K.H. Park, Z. Bien, and D.H. Hwang. Design of an ilc for linear systems with time-delay and initial state error. pages 147-164, Netherlands, 1998. Kluwer Academic.

[24] T.H. Lee Q. Hu, J.-X. Xu . Iterative learning control design for smith predictor. Systems and control letters, 44:201-210, 2001.

[25] C. Undey, S. Ertunc, and A.C. Nar. Online batch/fed-batch process performance monitoring, quality prediction, and variable-contribution analysis for diagnosis. Ind. Eng. Chem., 42:4645-4658, 2003.

[26] M.H.A. Verwoerd. Iterative learning control : a critical review. PhD thesis, Enschede, January 2005.

[27] J.X. Xu, S.K. Panda, and T.H. Lee. Real-time iterative learning control. Springer, 2009.

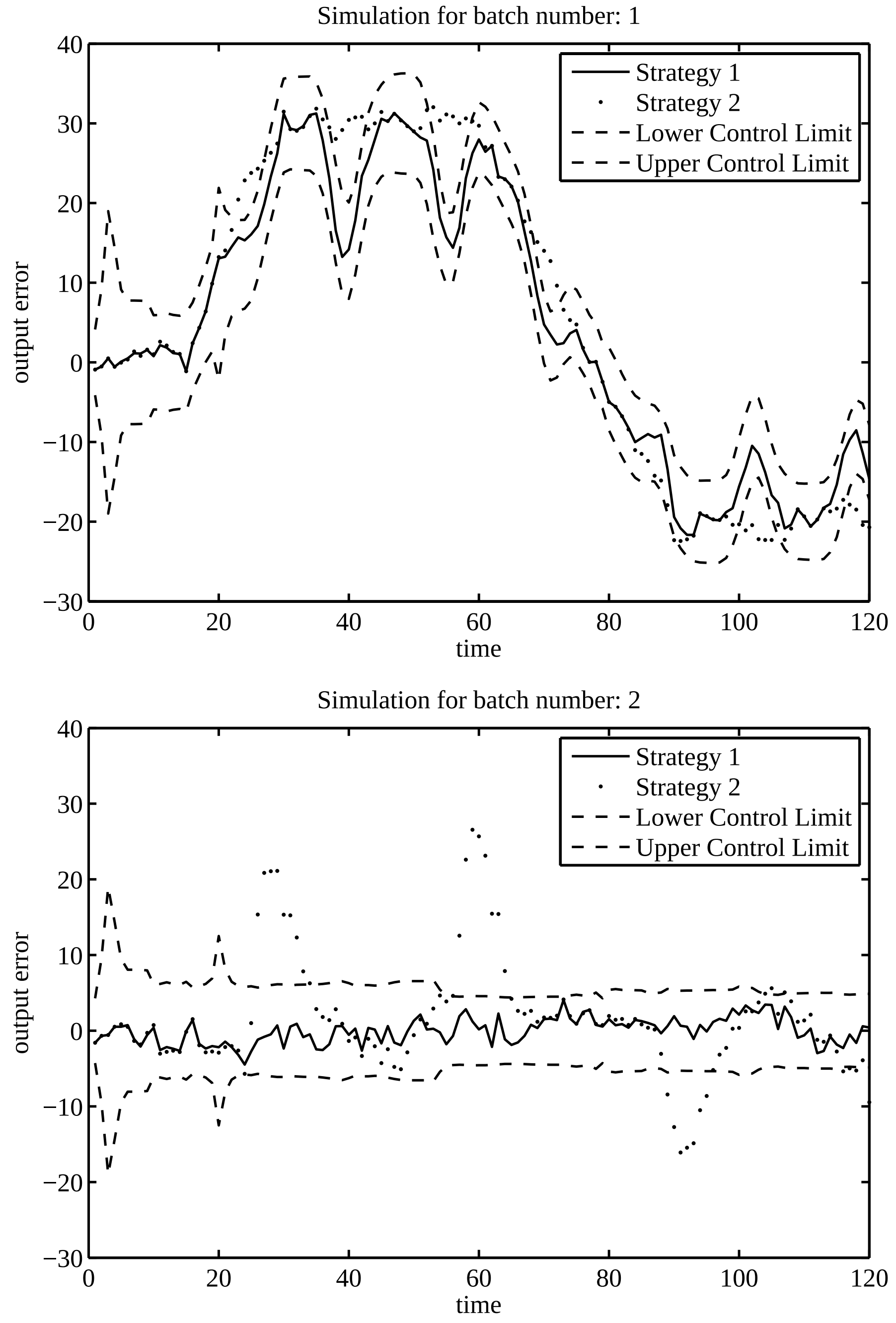
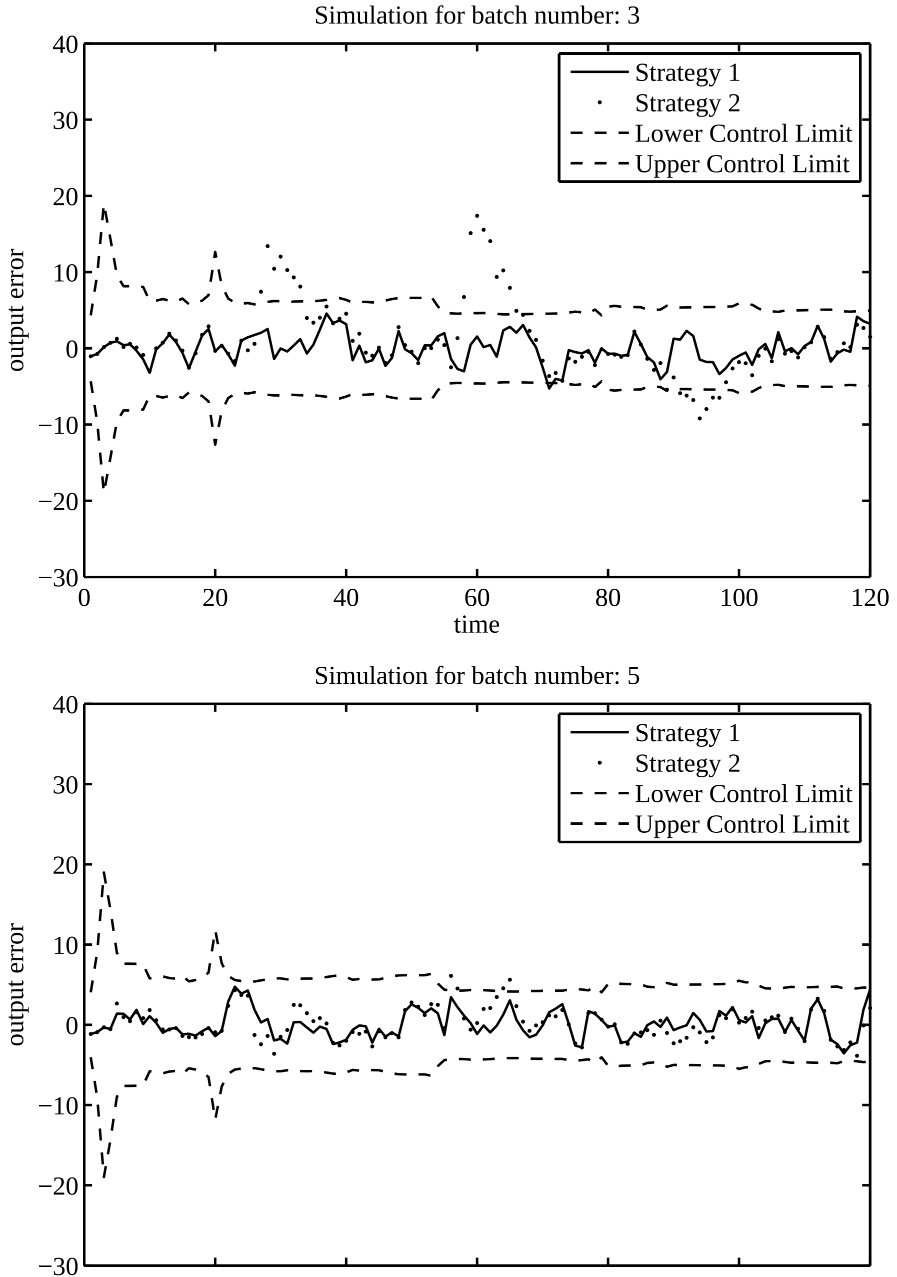


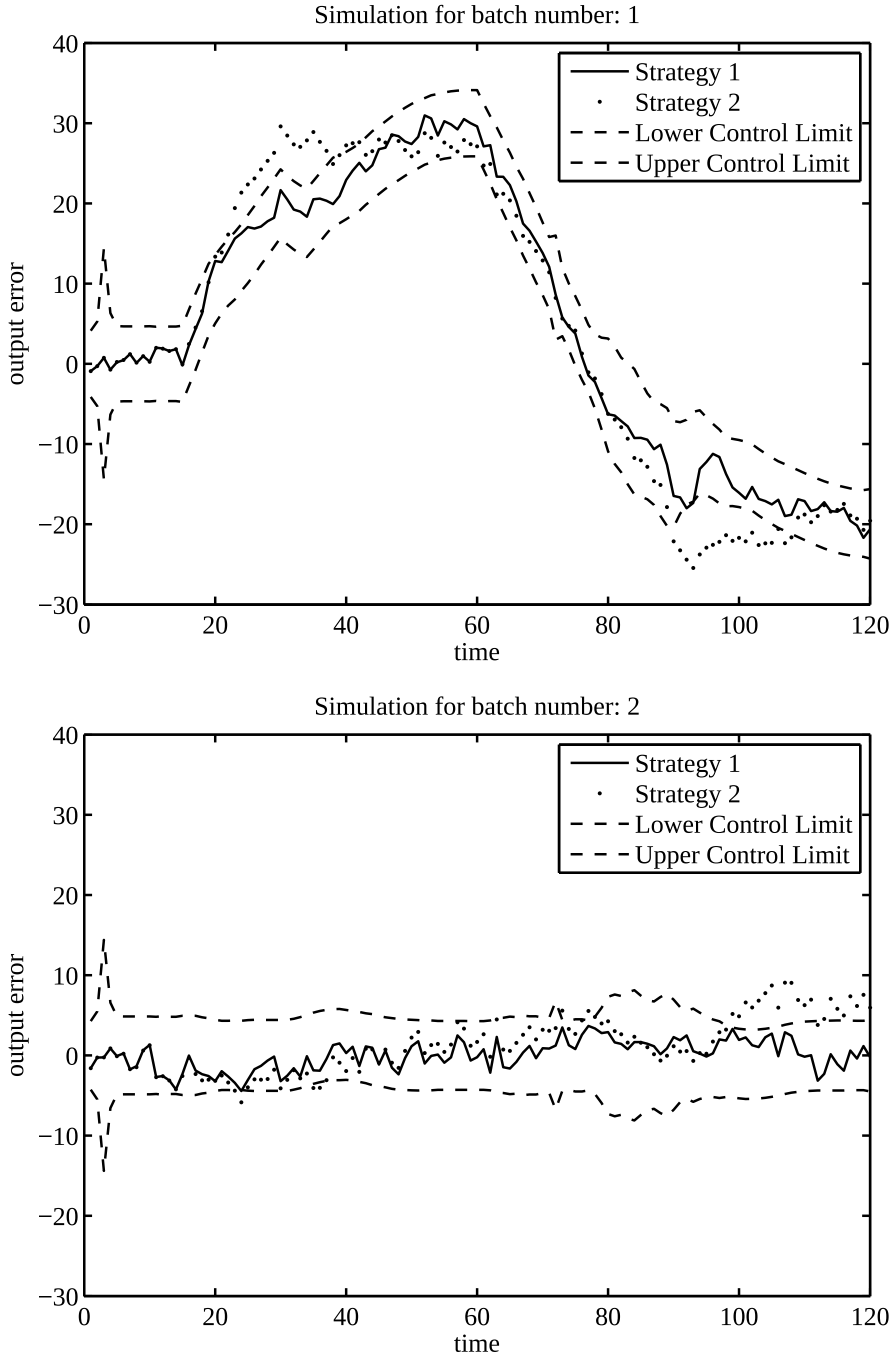
Figure 2.4: Performance monitoring simulation results for case 1



Simulation for batch number: 3

Simulation for batch number: 5

Figure 2.5: Performance monitoring simulation results for case 1



Simulation for batch number: 1

Simulation for batch number: 2

Figure 2.6: Performance monitoring simulation results for case 2

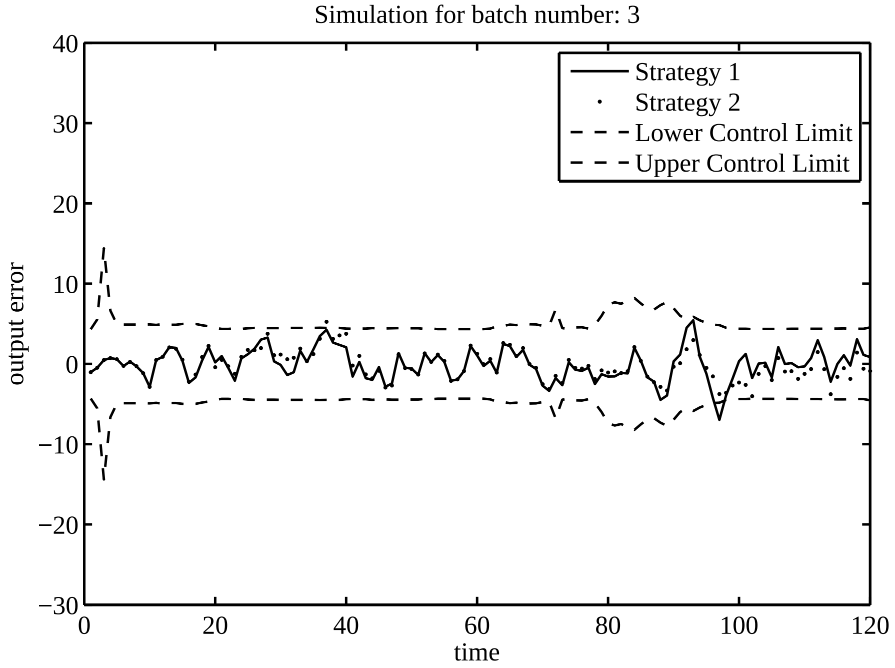


Figure 2.7: Performance monitoring simulation results for case 2

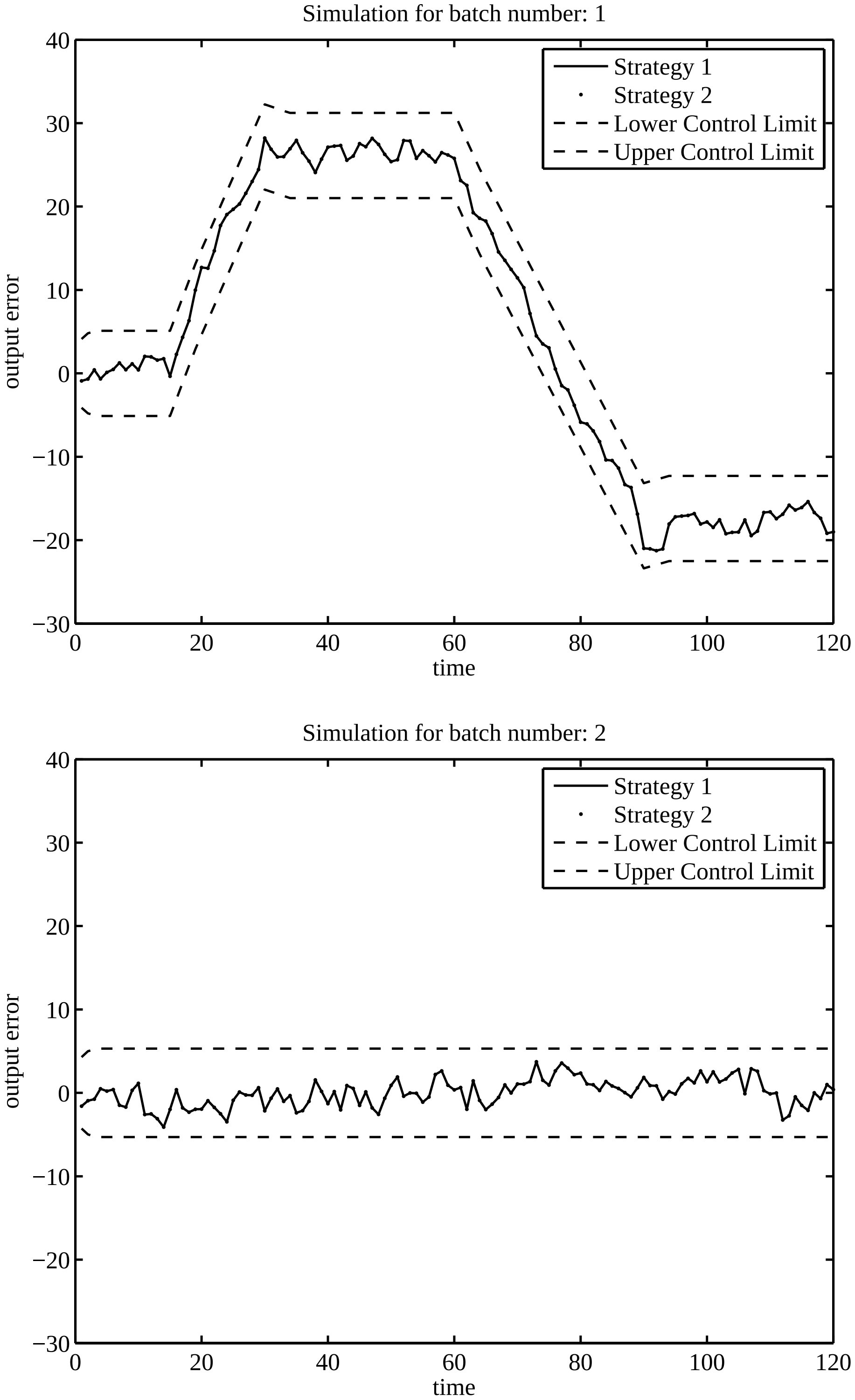


Figure 2.8: Performance monitoring simulation results for case 3

Chapter 3

Deterministic versus stochastic performance assessment of iterative learning control loops

3.1 Introduction

Due to increasing industrial applications of batch processes, there is a growing interest in investigating different control and monitoring methods [17] in order to improve control performance in terms of set-point tracking and disturbance rejection. Batch processes inherit large transient phases covering a wide range of operating envelopes. This implies that both the tracking and disturbance rejection problems must be addressed in any control design of batch processes.

To achieve the mentioned objectives, iterative learning control (ILC) has been widely attempted [11, 24, 27, 45]. This method, first introduced by Arimoto and Kawamura (1984) [2] for robot systems, considers the use of information from the previous batch to control the current one. At each batch, the input and error signals are kept for the use of the next batch. Design of ILC needs reliable models [32, 37, 38, 46], but in practice models have several kinds of uncertainties including parameter uncertainties or across-batch stochastic uncertainty, which have been discussed in literature [1, 9, 34, 35, 36]. To improve the performance of batch process control, combination of iterative learning algorithm with the model predictive control strategy has been proposed [28, 44]. For certain aspects of ILC such as design and convergence, readers are referred to Verwoerd (2005) [43] and Moore (1993) [33].

Since the global competition in chemical industry is increasing rapidly, performance assessment of control loops is receiving more attention as a means to ensure high efficiency of process operations. To assess the performance of a control loop, a benchmark is required as a reference. Minimum variance benchmark has gained popularity after it was first proposed by Harris (1989) [15], based on the minimum variance control introduced by Astrom (1970) [3].

Astrom developed linear time-invariant minimum variance control, which was followed by remarkable progress in the fields of predictive control [5, 12, 40] and adaptive control [13, 14, 26]. Harris (1989) [15] suggested the use of minimum variance control as the benchmark. It was shown that there exists a feedback controller independent term in the process output that represents the minimum variance output and can be estimated from routine operating data. Since then, great progress in the area of performance assessment has been achieved [6, 7, 16, 18, 20, 23, 25, 31, 41, 42]. The reason behind the popularity of this method is its simplicity and non-intrusiveness. It provides the best possible control, with respect to the output error variance, without relying on the process model or complicated computation process. This benchmark has also been extended to time-varying processes [21, 29, 30]. Theoretical and practical aspects of this subject have been well-covered by Huang and Shah (1999) [22].

Performance assessment of continuous processes deals with disturbance rejection problem since the process often operates at a constant set-point. But for batch processes set-point tracking performance must also be taken into account. Qin (1998) [39] has categorized two different performance assessment problems. Chen and Kong (2009) [4] proposed a method to assess the performance of an ILC algorithm, that consists of two levels of controllers, by deriving the optimal control law for each controller. However, this method assumes that each of the two-level ILC controllers affects either the stochastic or deterministic control performance but without considering interaction.

This chapter elaborates the ILC controllers effects on both the stochastic and deterministic performance of batch control. For stochastic performance, the goal is to reject the disturbance, while for deterministic performance faster convergence is the goal. It is shown that there is a trade-off between convergence and disturbance rejection. A method is proposed to estimate the trade-off curve, which is then used as the benchmark to assess the performance of ILC controllers.

The remainder of this chapter is structured as follows: Section 2 contains ILC set-up and a detailed explanation of its algorithm. In Section 3, the optimal design of ILC controllers and estimation of the trade-off curve are discussed. Section 4 addresses performance assessment of the ILC controllers. Simulation studies are conducted in Section 5, followed by concluding remarks in Section 6.

3.2 Preliminaries

Figure 3.1 illustrates an ILC algorithm that is applied to a process which operates over different batches with a constant batch period. Note that i, k, and q^{-1} represent batch index, time index, and time back-shift operator, respectively. As shown in Figure 3.1, the ILC algorithm consists of two loops. The inner loop, drawn with solid lines, operates similarly to a conventional control loop and includes the inner controller (C), plant model (T) and disturbance model (N). At each batch, the reference (x_k^i) is set by the outer loop, drawn with dashed lines. Disturbance is considered as another input to the process which can be modeled by a colored noise derived from a white noise (w_k^i). After a batch is over, the output trajectory is compared with the set-point trajectory (y_k^d) and the error (e_k^i) is computed in the outer loop. The outer controller (L) filters the error trajectory which will then be added to the current reference to set the new reference for the next batch.

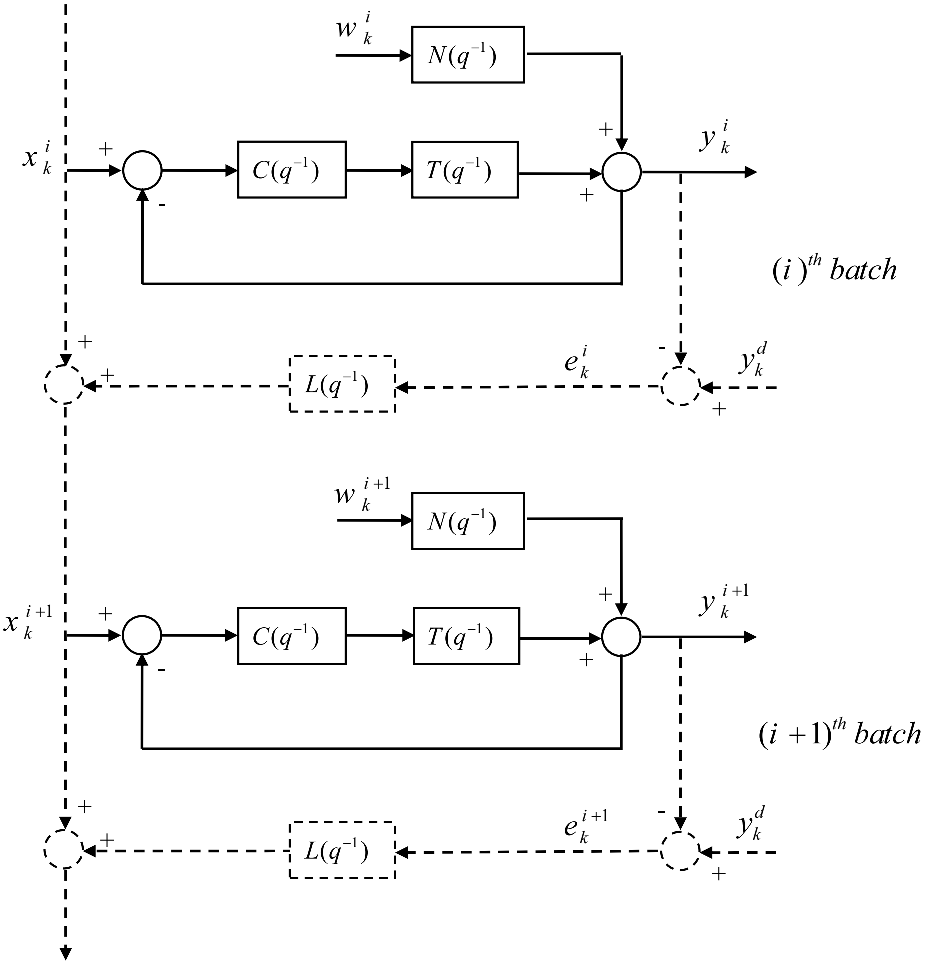


Figure 3.1: A recommended ILC structure

The outer loop sets the reference for each batch to reduce the error in the next batch. Under a good control policy, the process can track the set-point after a few batches. This is called convergence. For instance, in Figure 3.2 the third batch tracks the desired output, and its output error is caused only by stochastic disturbance. Hence the designed control converges after three batches.

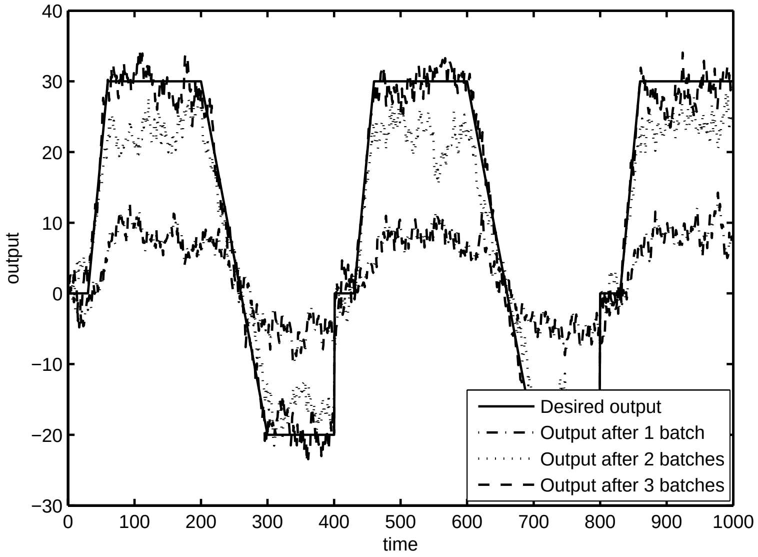


Figure 3.2: ILC convergence

Equation 3.1 gives the relation between the inputs and output in a discrete-time transfer function format. Note that for the sake of simplicity, the back-shift operator has been dropped.

$$y_k^i = [1 + TC]^{-1} TC x_k^i + [1 + TC]^{-1} N w_k^i \tag{3.1}$$

The outer loop holds an update rule, as follows:

$$x_k^i = x_k^{i-1} + L e_k^{i-1} \tag{3.2}$$

Let us define z^{-1} as the batch back-shift operator, as shown below:

$$z^{-1} x_k^i = x_k^{i-1} \tag{3.3}$$

Hence Equation 3.2 can be rearranged to give:

$$x_k^i = (z - 1)^{-1} L e_k^i \tag{3.4}$$

3.3 Optimal ILC design

This section elaborates upon the optimal design procedure for the ILC controllers based on the minimum variance control law. Error terms are derived and effects from both controllers on the error terms are discussed. To achieve desired output with respect to set-point tracking and disturbance rejection, control algorithms are developed.

3.3.1 General derivations

From the Diophantine identity [15], we can write the closed-loop disturbance model in an impulse response form:

$$[1 + TC]^{-1}N = F + Rq^{-d} \tag{3.5}$$

where F includes the first d terms of the impulse response and R includes the remainder. Substituting the closed-loop disturbance model with the Diophantine identity, Equation 3.1 can be rewritten as:

$$y_k^i = [1 + TC]^{-1}TCx_k^i + [F + Rq^{-d}]w_k^i \tag{3.6}$$

The reference signal is set according to the update rule, presented in Equation 3.4. Substituting the update rule into Equation 3.6 yields:

$$y_k^i = [1 + TC]^{-1}TC(z - 1)^{-1}Le_k^i + (F + Rq^{-d})w_k^i \tag{3.7}$$

Equation 3.7 can be rearranged to give:

$$(z - 1)y_k^i = [1 + TC]^{-1}TCLe_k^i + (z - 1)(F + Rq^{-d})w_k^i \tag{3.8}$$

For the sake of simplicity, let us define a new variable:

$$G \triangleq [1 + TC]^{-1}TCL \tag{3.9}$$

Substituting G into Equation 3.8 yields:

$$(z - 1)y_k^i = Ge_k^i + (z - 1)(F + Rq^{-d})w_k^i \tag{3.10}$$

which can be further simplified as:

$$y_k^{i+1} - y_k^i = Ge_k^i + (F + Rq^{-d})(w_k^{i+1} - w_k^i) \tag{3.11}$$

By adding and subtracting the set-point to the left side of Equation 3.11, the error can be written as a function of the previous batch error:

$$y_k^{i+1} - y_k^i + y_k^d - y_k^d = Ge_k^i + (F + Rq^{-d})(w_k^{i+1} - w_k^i) \tag{3.12}$$

$$\rightarrow e_k^i - e_k^{i+1} = Ge_k^i + (F + Rq^{-d})(w_k^{i+1} - w_k^i) \tag{3.13}$$

$$\rightarrow e_k^{i+1} = (1 - G)e_k^i + (F + Rq^{-d})(w_k^i - w_k^{i+1}) \tag{3.14}$$

Equation 3.14 gives the relation between the error sequences of different batches. In order to write the error as a function of the set-point and white noises, the error of the first batch is needed. For the first batch, it is common to use the set-point directly as the reference signal [33]. Let us rewrite Equation 3.6 for the first batch:

$$y_k^1 = [1 + TC]^{-1} TC y_k^d + (F + Rq^{-d}) w_k^1 \tag{3.15}$$

Subsequently, the error is found to be:

$$\begin{aligned} e_k^1 = y_k^d - y_k^1 &= y_k^d - [1 + TC]^{-1} TC y_k^d - [F + Rq^{-d}] w_k^1 \\ &= [1 + TC]^{-1} y_k^d - (F + Rq^{-d}) w_k^1 \end{aligned} \tag{3.16}$$

Having calculated the error expression of the first batch and according to Equation 3.14, all the error terms can be derived, as shown below:

$$e_k^2 = (1 - G)[1 + TC]^{-1} y_k^d + G(F + Rq^{-d}) w_k^1 - (F + Rq^{-d}) w_k^2 \tag{3.17}$$

$$\vdots$$

$$\begin{aligned} e_k^n = &(1 - G)^{n-1}[1 + TC]^{-1} y_k^d + (1 - G)^{n-2} G(F + Rq^{-d}) w_k^1 + \cdots \\ &+ (1 - G)G(F + Rq^{-d}) w_k^{n-2} + G(F + Rq^{-d}) w_k^{n-1} - (F + Rq^{-d}) w_k^n \end{aligned} \tag{3.18}$$

Since all the terms are derived from the set-point or white noises, the error of each batch can be split into deterministic and stochastic parts:

$$e_k^n = e_k^{n,det} + e_k^{n,sto} \tag{3.19}$$

where

$$e_k^{n,det} = (1 - G)^{n-1}[1 + TC]^{-1} y_k^d \tag{3.20}$$

$$\begin{aligned} e_k^{n,sto} = &(1 - G)^{n-2} G(F + Rq^{-d}) w_k^1 + \cdots + (1 - G)G(F + Rq^{-d}) w_k^{n-2} \\ &+ G(F + Rq^{-d}) w_k^{n-1} - (F + Rq^{-d}) w_k^n \end{aligned} \tag{3.21}$$

Consider two special cases:

[1] G=0

The outer controller is inactive, and the outer loop is therefore disconnected. Hence the reference signal remains constant across batches, and convergence cannot be achieved. From Equation 3.18, the error can be computed as follows:

$$e_k^n = [1 + TC]^{-1} y_k^d - (F + Rq^{-d}) w_k^n \tag{3.22}$$

[2] G=1

From Equation 3.20, the deterministic error becomes zero after the operation of the outer loop, which implies that convergence will be achieved after one batch. From Equation 3.18, the error is found to be:

$$e_k^n = (F + Rq^{-d})(w_k^{n-1} - w_k^n), n > 1 \tag{3.23}$$

3.3.2 Rate of convergence

Based on the definition, convergence of the ILC algorithm is achieved when the output error is caused only by stochastic disturbance, i.e. the deterministic error becomes zero. Rate of convergence is defined as the number of batches required to achieve convergence, which can be used as a measure for the deterministic performance of the algorithm. The higher the rate of convergence, i.e. the fewer batches required for convergence, the better the deterministic performance.

Let us define a new variable:

$$\bar{y}_k^d \triangleq [1 + TC]^{-1} y_k^d \tag{3.24}$$

which is the deterministic error of the first batch. Updating the deterministic error, Equation 3.20, gives:

$$e_k^{n,det} = (1 - G)^{n-1} \bar{y}_k^d \tag{3.25}$$

To achieve convergence, we need to equate Equation 3.25 with zero:

$$e_k^{n,det} = (1 - G)^{n-1} \bar{y}_k^d = 0 \tag{3.26}$$

Based on the expansion of Parseval's theorem [10], we propose the following method to determine the convergence.

Expansion of Parseval's theorem: Let *u(t)* and *y(t)* be the input and output of a system with the following relation:

$$y(t) = Hu(t) \tag{3.27}$$

where H is a discrete time transfer function. The following inequality holds between the 2-norms of the input and output [10]:

$$\|y\|_2 \leq \|H\|_\infty \|u\|_2 \tag{3.28}$$

Applying this theorem to Equation 3.25 yields:

$$\|e_k^{n,det}\|_2 \leq \gamma_n \|\bar{y}_k^d\|_2 \tag{3.29}$$

where

$$\gamma_n = \|(1-G)^{n-1}\|_\infty \tag{3.30}$$

The algorithm is called converged after n batches, if the 2-norm of the deterministic error of the n^{th} batch decreases to less than $100\varepsilon\%$ of the 2-norm of the first batch deterministic error, where $0 < \varepsilon << 1$. Therefore, the rate of convergence is equal to the smallest number of n that satisfies the following:

$$\gamma_n \le \varepsilon \tag{3.31}$$

Convergence condition

In some cases, the ILC algorithm may never converge. When designing the algorithm, we need to consider the convergence property.

From the submultiplicative property of the ∞-norm [10], we can expand Equation 3.30:

$$\gamma_n = \|(1-G)^{n-1}\|_\infty = \|(1-G)\|_\infty^{n-1} \tag{3.32}$$

If the following holds:

$$\|(1-G)\|_\infty < 1 \tag{3.33}$$

Then Equation 3.31 will certainly hold for a finite value of n. Otherwise, γ_n increases (or never decreases) after each batch and Equation 3.31 will not hold. Therefore, Equation 3.33 represents a sufficient condition of a convergent algorithm.

3.3.3 The inner controller

Recall the error expressions:

$$\begin{aligned}
e_k^{n,det} =& (1-G)^{n-1}[1+TC]^{-1}y_k^d \\
e_k^{n,sto} =& (1-G)^{n-2}G(F+Rq^{-d})w_k^1 + \cdots + (1-G)G(F+Rq^{-d})w_k^{n-2} \\
&+ G(F+Rq^{-d})w_k^{n-1} - (F+Rq^{-d})w_k^n
\end{aligned}$$

where G, F, and R are related in the following equations:

$$\begin{aligned}
G &= [1+TC]^{-1}TCL \\
F + Rq^{-d} &= [1+TC]^{-1}N
\end{aligned}$$

As shown in the first chapter, F is the feedback controller independent term. The outer controller affects only G, while the inner controller affects both G and R. Since the outer controller is capable of executing both causal and non-causal operators [8, 43], all the possible values of G can be reached by changing the outer controller. The inner controller can focus on changing R which is in the stochastic error expression. Therefore, the optimal inner controller is derived by minimizing the stochastic error of the inner control loop. The derivation of the minimum variance control law is the same as the conventional minimum variance control law, as presented in the first chapter. The result is given by:

$$C^{opt} = \tilde{T}^{-1}\acute{R}F^{-1} \tag{3.34}$$

where $\tilde{T}$ is the delay-free plant model and $\acute{R}$ is obtained from the following Diophantine identity:

$$N = F + \acute{R}q^{-d} \tag{3.35}$$

3.3.4 The outer controller

The outer controller affects both the deterministic and stochastic error terms. Among all the ILC algorithms with the same deterministic performance (the same rate of convergence), the one with the highest stochastic performance (the minimum variance of the stochastic error) is considered optimal. Therefore, a different optimal solution exists at each convergence rate.

To find the optimal solutions, the variance of the stochastic error needs to be minimized with respect to the outer controller while the rate of convergence is set at the desired value. Note that the existing inner controller (not necessarily the minimum variance inner controller) is considered in the stochastic error term. The optimal solution for an ILC algorithm that converges after n batches meets two conditions:

[1] The rate of convergence is n. From Equation 3.31, the following holds:

$$\gamma_n = \|(1-G)^{n-1}\|_\infty \leq \varepsilon \tag{3.36}$$

[2] The variance of the stochastic error, Equation 3.21, is minimized. Originated from different white noises, all the terms of the stochastic error are uncorrelated. Therefore, the variance of the stochastic error is equal to:

$$\begin{aligned} var(e_k^{n,sto}) =& var((1-G)^{n-2}G(F+Rq^{-d})w_k^1) \\ &+ \cdots + var((1-G)G(F+Rq^{-d})w_k^{n-2}) \\ &+ var(G(F+Rq^{-d})w_k^{n-1}) + var((F+Rq^{-d})w_k^n) \end{aligned} \tag{3.37}$$

To calculate the variance of each term, the following Lemma is used:

Lemma1: Given an expression $y(t) = Hw(t)$ where $w(t)$ is a white noise with variance of σ_w^2, the variance of $y(t)$ can be computed as follows [19]:

$$var(y(t)) = \int_{-\pi}^{\pi} \phi_y(\omega) d\omega \tag{3.38}$$

where

$$\phi_y(\omega) = \frac{1}{2\pi} \mid H(e^{-j\omega}) \mid^2 \sigma_w^2 \tag{3.39}$$

This leads to:

$$var(y(t)) = \int_{-\pi}^{\pi} \frac{1}{2\pi} \mid H(e^{-j\omega}) \mid^2 d\omega \sigma_w^2 = \|H\|_2^2 \sigma_w^2 \tag{3.40}$$

According to Lemma1, the variance of the stochastic error is found to be:

$$\begin{aligned} var(e_k^{n,sto}) = (&\|(1-G)^{n-2}G(F+Rq^{-d})\|_2^2 + \cdots \\ &+ \|(1-G)G(F+Rq^{-d})\|_2^2 + \|G(F+Rq^{-d})\|_2^2 + \|(F+Rq^{-d})\|_2^2)\sigma_w^2 \end{aligned} \tag{3.41}$$

Hence the following objective function is defined to minimize the variance of the stochastic error:

$$J(G) = \min_G [var(e_k^{n,sto})] \tag{3.42}$$

$$\begin{aligned} &= \min_G [\|(1-G)^{n-2}G(F+Rq^{-d})\|_2^2 + \cdots + \|(1-G)G(F+Rq^{-d})\|_2^2 \\ &\quad + \|G(F+Rq^{-d})\|_2^2 + \|(F+Rq^{-d})\|_2^2] \end{aligned} \tag{3.43}$$

Optimization algorithm

To find the optimal solutions, we combine the two conditions into one objective function, as shown below:

$$G^{opt} = arg \min_G \, [var(e_k^{n,sto}) + \lambda|\gamma_n - \varepsilon|] \tag{3.44}$$

where λ is a weighting variable. As an example, consider a first order structure for G^{opt}:

$$G^{opt} = \frac{b_0 + b_1 q^{-1}}{1 + f_1 q^{-1}} \tag{3.45}$$

Optimization is performed with respect to three parameters of G^{opt}. For this example, the procedure is as follows:

[1] Assign an integer to n.

[2] A small initial value is set for λ.

[3] A numerical method, like Newton-Raphson, is used to do the minimization.

[4] If $\gamma_n \leq \varepsilon$ (e.g. $\varepsilon = 0.05$) is satisfied, the solution is found. If not, continue to step 4.

[5] A larger weighting variable is chosen, and step 2 is repeated.

Following the algorithm for different values of n, all the optimal solutions for the considered G^{opt} can be found. Plotting the stochastic error variance of all the solutions versus the rate of convergence, we obtain a trade-off curve which illustrates the trade-off between the deterministic and stochastic performance of the ILC algorithm.

For each optimal solution, the optimal outer controller can be computed from the inverse of the definition of G, as shown below:

$$L^{opt} = (TC)^{-1}[1 + TC]G^{opt} \tag{3.46}$$

3.4 Performance assessment

After collecting routine operating data from ILC controlled batch process, estimation of the process and disturbance models can be conducted. Let us rewrite the closed loop model of the inner loop, originally presented in Equation 3.1:

$$y_k^i = (1 - S)x_k^i + SNw_k^i \tag{3.47}$$

where

$$S = [1 + TC]^{-1} \tag{3.48}$$

Performing identification using output data y_k^i and the batch set-point reference data x_k^i, S and N are estimated. Since the existing inner controller is available, the plant model can also be estimated from S. F, R, and G are accordingly determined based on definitions, presented in Equations 3.5 and 3.9.

3.4.1 The inner controller

As mentioned in section 3.3.3, the inner controller minimizes the stochastic error of the inner loop, and the optimal inner controller is equivalent with the optimal feedback controller with respect to the conventional minimum variance benchmark. Hence performance index of the inner controller is defined the same as the conventional method, as presented in the first chapter:

$$\eta_C = \frac{\|F\|_2^2}{\|F + Rq^{-d}\|_2^2} \tag{3.49}$$

which reflects the ratio between the optimal and existing performance.

3.4.2 The outer controller

Following the procedure for the optimal design of the outer controller for different convergence rates (n), a trade-off curve is obtained, as illustrated in Figure 3.3.

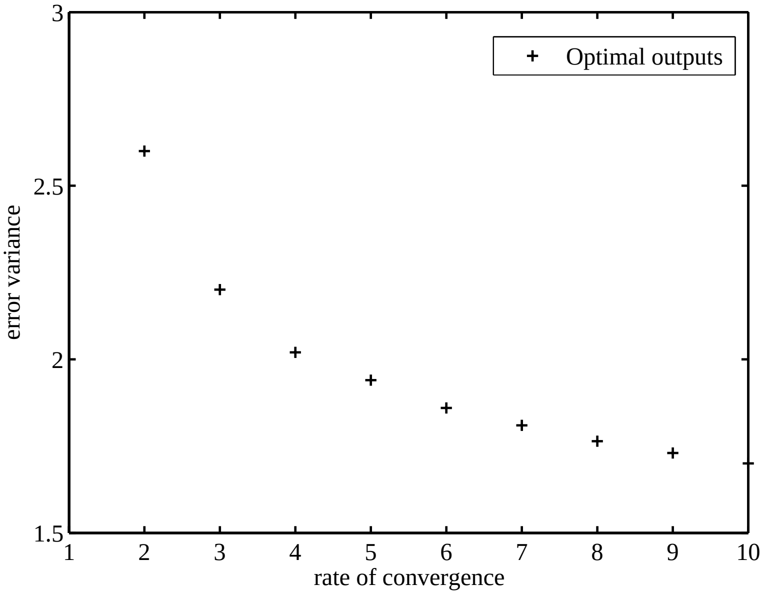


Figure 3.3: An example of the trade-off curve

Having estimated F, R, and G, the rate of convergence and the stochastic error variance for the current ILC can be obtained from Equations 3.31 and 3.41, respectively. If the current algorithm converges after n batches, the optimal solution with the rate of convergence equal to n is selected from the curve for performance assessment. The comparison of the variance of the current stochastic error with the variance of the optimal stochastic error for the same rate of convergence can offer a measure to determine the performance of the outer controller. Hence we define an index equal to the variance of the stochastic error under the optimal outer control loop divided by the variance of the current stochastic error, as shown below:

$$\eta_L = \frac{\|(1-G^{opt})^{n-2}G^{opt}(F+Rq^{-d})\|_2^2 + \cdots + \|(F+Rq^{-d})\|_2^2}{\|(1-G)^{n-2}G(F+Rq^{-d})\|_2^2 + \cdots + \|(F+Rq^{-d})\|_2^2} \tag{3.50}$$

The numerator in Equation 3.50 represents the solution under the optimal outer control loop. Note that since performance assessment of the outer controller is the objective, the existing inner controller (which is not necessarily optimal) is used for the calculation of the solution. Hence the difference in the error terms of the current and optimal solutions is caused only by the outer controller.

3.5 Simulation studies

A batch process is given as below:

$$\begin{aligned} T &= \frac{0.9q^{-2} - 0.6q^{-3}}{1 - 0.7q^{-1}} \\ N &= \frac{1 - 0.2q^{-1}}{1 - 0.9q^{-1}} \end{aligned}$$

This process is expected to follow the set-point trajectory given in Figure 3.4. Three cases with different sets of ILC controllers are studied:

- *Case 1*

  The optimal inner controller is implemented. The outer controller is chosen to achieve the fastest convergence, as shown below:

  $$G = 1 \to L = (TC)^{-1}[1 + TC]$$

  After collecting data, G, F, and R are estimated:

  $$\begin{aligned} \hat{F} &= 1 + 0.7015q^{-1} \\ \hat{R} &\simeq 0 \\ \hat{G} &= 1.004 - 0.007q^{-1} \end{aligned}$$

  To obtain the trade-off curve, the optimization procedure is applied. Table 3.1 shows the parameters of the optimal G and the minimum

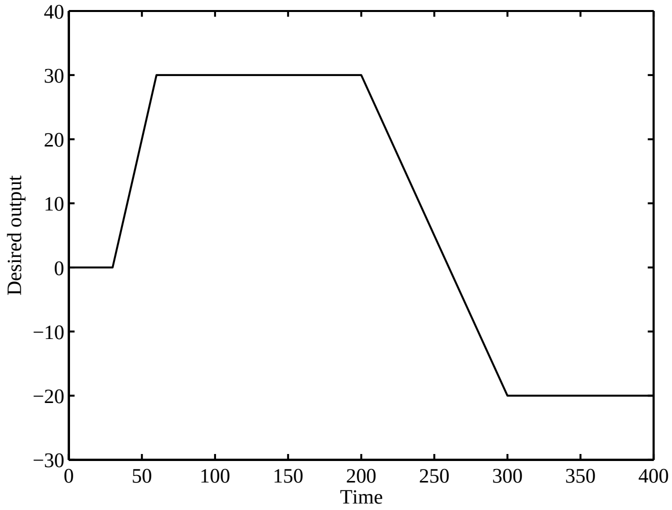


Figure 3.4: Desired output for simulation

variance solutions (optimal stochastic error variance) under the optimal outer loop at different rates of convergence. Figure 3.5 also illustrates the trade-off curve and the current performance of the designed algorithm.

Table 3.1: Optimal controller and minimum variance values for Case 1

Conv.	2	3	4	5	6	7	8	9
b_0	0.789	0.54	0.404	0.322	0.267	0.228	0.201	0.192
b_1	0.039	0.058	0.046	0.037	0.028	0.033	0.031	0.029
f_1	0.143	0.253	0.281	0.293	0.293	0.340	0.362	0.376
min. var.	2.404	2.001	1.839	1.754	1.702	1.667	1.641	1.622

The current ILC algorithm converges after two batches with the error variance of 2.98. The minimum variance solution under the optimal outer loop at the same rate of convergence is 2.4041. Hence the performance index for the outer controller is:

$$\eta_L = \frac{2.4041}{2.98} = 0.807$$

Based on 3.49, the performance index of the inner controller is 1, which is expected since the optimal inner controller is implemented.

- *Case 2*

  The optimal inner controller and a relatively slower outer controller ($L = 0.5$) are implemented. Estimated G, F, and R are as follow:

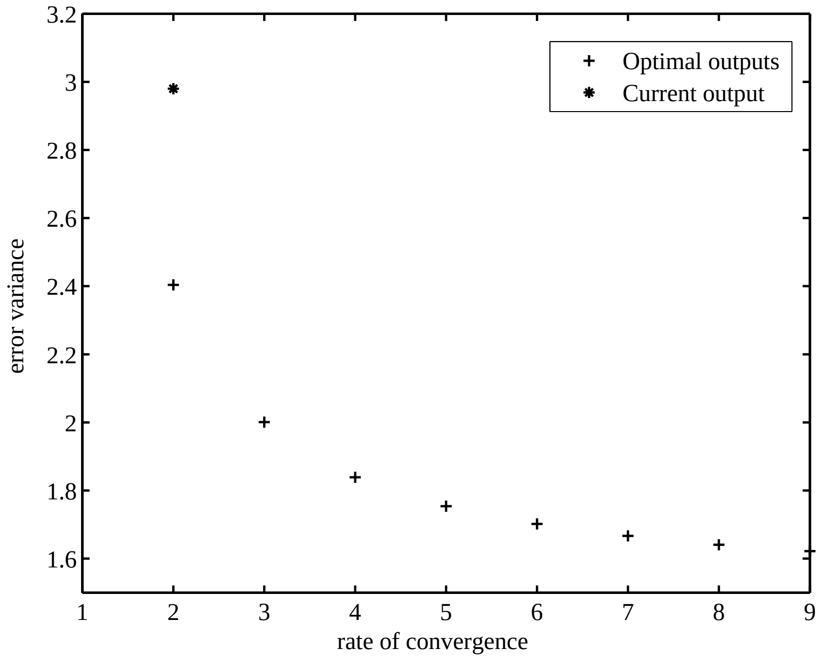


Figure 3.5: Trade-off curve for Case 1

$$\hat{F} = 1 + 0.7033q^{-1}$$
$$\hat{R} \simeq 0$$
$$\hat{G} = 0.3173 + 0.053q^{-1} + 0.022q^{-2} + 0.016q^{-3} + 0.006q^{-4}$$

Table 3.2 shows the parameters of the optimal G and the minimum variance solutions under the optimal outer loop. The trade-off curve and the current performance of the designed algorithm are also presented in Figure 3.6.

Table 3.2: Optimal controller and minimum variance values for Case 2

Conv.	2	3	4	5	6	7	8	9
b_0	0.789	0.540	0.405	0.322	0.267	0.228	0.199	0.190
b_1	0.046	0.061	0.048	0.039	0.029	0.034	0.034	0.027
f_1	0.151	0.257	0.287	0.299	0.295	0.342	0.370	0.368
min. var.	2.408	2.005	1.843	1.758	1.705	1.670	1.645	1.620

The inner controller is the same as that in the previous case. Hence its performance index remains to be 1. This can be verified by calculating

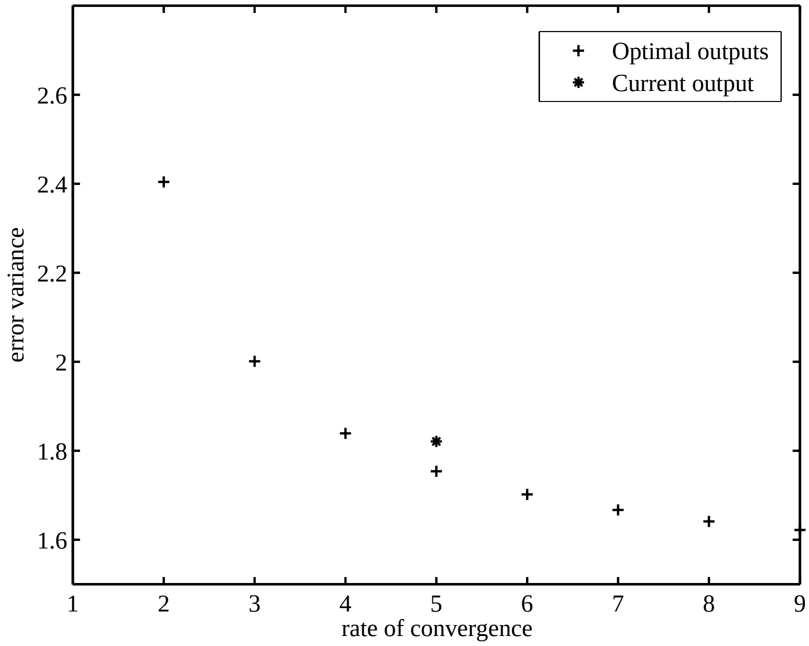


Figure 3.6: Trade-off curve for Case 2

3.49. The current ILC converges after 5 batches with the error variance of 1.821, while the minimum variance solution under the optimal outer loop for the same rate of convergence is 1.758. Hence the performance index of the outer controller is:

$$\eta_L = \frac{1.758}{1.821} = 0.962$$

- *Case 3*

  Two non-optimal inner and outer controllers are implemented, as shown below:

$$C = \frac{0.5}{1 - 0.9q^{-1}}$$
$$L = 0.5$$

After obtaining data, G, F, and R are estimated:

$$\hat{F} = 1 + 0.6983q^{-1}$$
$$\hat{R} \simeq 0.1836 - 0.1681q^{-1} - 0.2499q^{-2} - 0.1663q^{-3} - 0.043q^{-4} + 0.034q^{-5}$$
$$\hat{G} = 0.2241 + 0.205q^{-1} + 0.0913q^{-2} - 0.0097q^{-3} - 0.0578q^{-4} - 0.0532q^{-5}$$

Table 3.3 and Figure 3.7 show the parameters of the optimal G and the minimum variance solutions under the optimal outer loop, as well as the current performance of the designed algorithm.

Table 3.3: Optimal controller and minimum variance values for Case 3

Conv.	4	5	6	7	8	9	10
b_0	0.407	0.325	0.252	0.233	0.187	0.175	0.164
b_1	-0.004	-0.003	-0.003	-0.002	-0.002	-0.002	-0.001
f_1	0.180	0.193	0.271	0.332	0.356	0.381	0.394
min. var.	2.024	1.932	1.875	1.837	1.810	1.788	1.762

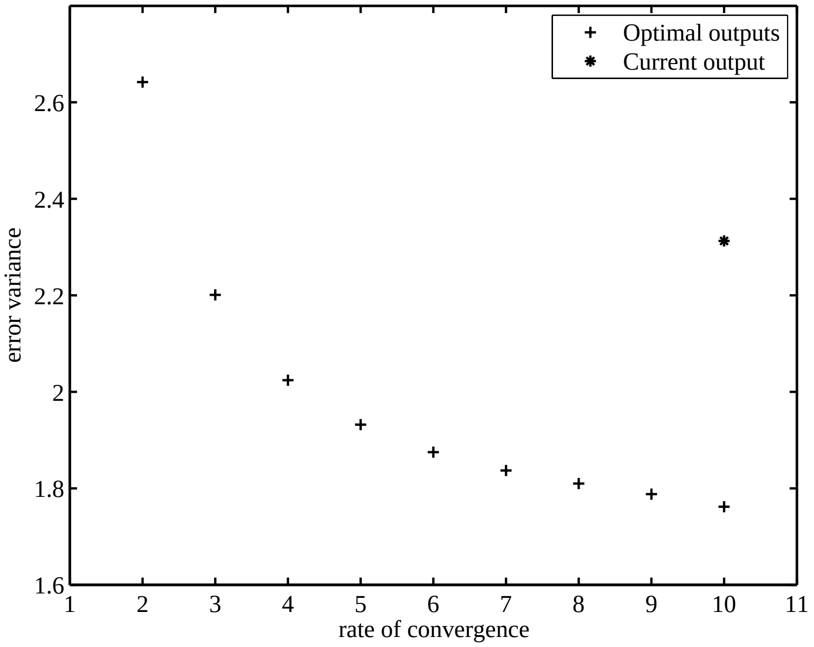


Figure 3.7: Trade-off curve for Case 3

The current performance indicates convergence after 10 batches with the error variance of 2.3129. The minimum variance solution under the optimal outer loop for the ILC algorithm with the same rate of convergence is 1.7627. Therefore, the performance index of the outer controller is:

$$\eta_L = \frac{1.7627}{2.3129} = 0.762$$

Based on 3.49, the performance index of the inner controller can be determined as:

$$\eta_C = \frac{1^2 + 0.6983^2}{1^2 + 0.6983^2 + 0.1836^2 + 0.1681^2 + \cdots} = 0.9$$

3.6 Conclusions

In this chapter certain aspects of performance assessment for the ILC algorithm including identification, the extension of minimum variance benchmark and performance assessment were discussed. First the ILC set-up and its algorithm were explained. Then the effects of both ILC controllers on the stochastic and deterministic performance were studied, followed by the optimal design procedure for each controller. The deterministic control performance of a process was determined by its rate of convergence, i.e. the number of batches required for convergence, while the stochastic control performance was determined by the stochastic error variance. Based on this consideration, it was shown that the solution of the inner loop under the optimal inner controller is equivalent to the conventional minimum variance solution. But for the outer controller, there is a trade-off between the deterministic and stochastic performance of control loops. Therefore, optimal solutions under the optimal outer controller were equivalent to the minimum variance solutions at given rates of convergence. All the optimal solutions were presented by a curve to demonstrate the mentioned trade-off. To compute the trade-off curve, error variance minimization was conducted to find the optimal solutions under the optimal outer controller for different rates of convergence. This chapter also elaborated the performance assessment of ILC controllers and verified the feasibility of the proposed methods through three simulation case studies.

References

[1] H.-S. Ahn, K.L. Moore, and Y. Chen. Iterative Learning Control, Robustness and Monotonic Convergence for Interval Systems. Spring-Verlag, 2007.

[2] S. Arimoto, S. Kawamura, and F. Miyazaki. Iterative learning control for robot systems. Tokyo, Japan, 1984. Proceedings of IECON.

[3] K.J. Astrom. Introduction to Stochastic Control Theory. Academic Press, New York, 1970.

[4] J. Chen and C.K. Kong. Performance assessment for iterative learning control of batch units. J. of Process Control, 19:1043-1053, 2009.

[5] D.W. Clarke, C. Mohtadi, and P.S. Tuffs. Generalized predictive control: parts 1 and 2. Automatica, 23(2):137-160, 1987.

[6] L. Desborough and T. Harris. Performance assessment measure for univariate feedback control. Can. J. Chem. Eng., 70:1186-1197, 1992.

[7] L. Desborough and T.J. Harris. Performance assessment measures for univariate feed-forward/feedback control. The Canadian J. of Chemical Engineering, 71:605-616, 1993.

[8] B.G. Dijkstra. Iterative learning control, with applications to a wafer stage. PhD thesis, Delft University of Technology, 2003.

[9] T.Y. Doh, J.H. Moon, K.B. Jin, and M.J. Chung. Robust iterative learning control with current feedback for uncertain linear systems. Internat. J. of Systems Sci., 30:39-47, 1999.

[10] J.C. Doyle, B.A. Francis, and A.R. Tannenbaum. Feedback control theory. Macmillan publishing company, 1992.

[11] P. Fu and J.P. Barford. Simulation of an iterative learning control system for fed-batch cell culture processes. 10:53-62, 1992.

[12] C.E. Garcia, D.M. Prett, and M. Morari. Model predictive control: theory and practice-a survey. Automatica, 25(3):335-348, 1989.

[13] G.C. Goodwin, P.J. Ramadge, and P.E. Caines. Discrete-time stochastic adaptive control. SIAM J. Control, 19:826-853, 1981.

[14] G.C. Goodwin and K.S. Sin. Adaptive Filtering Prediction and Control. Prentice Hall, Englewood Cliffs, NJ, 1984.

[15] T. Harris. Assessment of control loop performance. The Canadian J. of Chemical Engineering, 67:856-861, 1989.

[16] T.J. Harris, F. Boudreau, and J.F. MacGregor. Performance assessment of multivariable feedback controllers. Automatica, 32(11):1505-1518, 1996.

[17] W.M. Hawkins and T.G. Fisher. Batch Control Systems Design, Application, and Implementation, 2nd Ed. ISA The Instrumentation, Systems, and Automation Society, 2006.

[18] A. Horch and A.J. Isaksson. A modified index for control performance assessment. J. of Process Control, 9(6):475-483, 1999.

[19] T. Soderstorm and P. Stoica. System Identification. Prentice Hall, 1989.

[20] B. Huang. Performance assessment of processes with abrupt changes of disturbances. The Canadian J. of Chemical Engineering, 77(5):1044-1054, 1999.

[21] B. Huang. Minimum variance control and performance assessment of time-variant processes. J. of Process Control, 12:707-719, 2002.

[22] B. Huang and S.L. Shah. Performance Assessment of Control Loops: Theory and Applications. Springer Verlag, London, 1999.

[23] B. Huang, S.L. Shah, and K.Y. Kwok. Good, bad or optimal? performance assessment of mimo processes. Automatica, 33:1175-1183, 1997.

[24] Z. Jie, J. Nguyan, J. Morris, and X. Zhihua. Batch to batch iterative learning control of a fed-batch fermentation process using linearised models. J. of Process Control, 17:745-750, 2008.

[25] B.S. Ko and T.F. Edgar. Pid control performance assessment: the single loop case. AIChE Journal, 50:1211-1218, 2004.

[26] T.L. Lai and C.Z. Wei. Extended least squares and their applications to adaptive control and prediction in linear systems. IEEE Trans. Autom. Control, 31:898-906, 1986.

[27] K.S. Lee, S.H. Bang, S. Yi, J.S. Son, and Yoon. Iterative learning control of heat-up phase for a batch polymerization reactor. J. of Process Control, 6(4):255-262, 1996.

[28] K.S. Lee, I.S. Chin, and H.J. Lee. Model predictive control technique combined with iterative learning for batch processes. AIChE Journal, 45:2175-2187, 1999.

[29] Z. Li and R.J. Evans. Minimum variance control of linear time-varying systems. Automatica, 33:1531-1537, 1997.

[30] Z. Li and R.J. Evans. Generalised minimum variance control of linear time-varying systems. IEE Pvoc-Control Theory Appl., 149(1):111-116, 2002.

[31] C.B. Lynch and G.A. Dumont. Control loop performance monitoring. IEEE Trans. Control Sys. Tech., 4(2):185-192, 1996.

[32] J.H. Moon, T.Y. Doh, and M.J. Chung. A robust approach to iterative learning control design for uncertain systems. Automatica, 34:1001-1004, 1998.

[33] K.L. Moore. Iterative Learning Control for Deterministic Systems. Spring-Verlag, London, January 1993.

[34] Z.K. Nagy and R.D. Braatz. Worst-case and distributional robustness analysis of finite-time control trajectories for nonlinear distributed parameter systems. IEEE Trans. on Control Systems Technology, 11:694-704, 2003.

[35] Z.K. Nagy and R.D. Braatz. Open-loop and closed-loop robust optimal control of batch processes using distributional and worst-case analysis. J. of Process Control, 14:411-422, 2004.

[36] Z.K. Nagy and R.D. Braatz. Distributional uncertainty analysis using power series and polynomial chaos expansions. J. of Process Control, 17:229-240, 2007.

[37] K.H. Park, Z. Bien, and D.H. Hwang. Design of an ilc for linear systems with time-delay and initial state error. pages 147-164, Netherlands, 1998. Kluwer Academic.

[38] T.H. Lee Q. Hu, J.-X. Xu . Iterative learning control design for smith predictor. Systems and control letters, 44:201-210, 2001.

[39] S.J. Qin. Control performance monitoring - a review and assessment. Computers and Chemical Engineers, 23:173-186, 1998.

[40] D.E. Seborg, S.L. Shah, and T.F. Edgar. Adaptive control strategies for process control: a survey. AIChE, 32:881-913, 1986.

[41] N. Stanfelj and T.E. MacGregor. Monitoring and diagnosing process control performance: the single-loop case. Ind. Eng. Chem. Res., 32:301-314, 1993.

[42] N.F. Thornhill, M. Oettinger, and P. Fedenczuk. Refinery-wide control loop performance assessment. J. of Process Control, 9(2):109-124, 1999.

[43] M.H.A. Verwoerd. Iterative learning control : a critical review. PhD thesis, Enschede, January 2005.

[44] Youqing Wang, Donghua Zhou, and Furong Gao. Iterative learning model predictive control for multi-phase batch processes. J. of Process Control, 18:543-557, 2008.

[45] J.-X. Xu, Q. Hu, T.H. Lee, and S. Yamamoto. Iterative learning control with smith time delay compensator for batch processes. J. of Process Control, 11:321-328, 2001.

[46] J.X. Xu, S.K. Panda, and T.H. Lee. Real-time iterative learning control. Springer, 2009.

Chapter 4

Two-dimensional generalized predictive control for batch processes

4.1 Introduction

Minimization of the predicted future output error (from the set-point) is the basis for many control strategies, referred to as predictive control strategies. Dynamic matrix control (DMC), first presented by Shell oil Company engineers [4], is probably one of the simplest predictive controllers and deals with step response model of linear processes. This method minimizes the least squares of the future output error while penalizing the input increments to avoid drastic changes in control actions [7]. DMC has been applied in many multivariable petrochemical processes [8, 12].

Model algorithmic control (MAC), also called model predictive heuristic control (MPHC), is another predictive controller introduced by Richalet (1978) [14]. This strategy also uses the step response model of the process, but minimizes the output error with respect to input rather than the input increment. One disadvantage of this method is that penalizing input reduces the ability to remove offset [7]. Next member of predictive controllers family is predictive functional control (PFC), which is popular due to its ability to deal with nonlinear and unstable linear internal models [7]. Because of the specific structure, this control strategy is capable of handling quick tracking control problems [11]. It is noted that only state space models are used in this algorithm [1, 13]. Generalized predictive control (GPC), developed by Clarke (1987) [3], has the same objective of the output error minimization, but utilizes an optimal predictor to anticipate the future output. Hence this method is capable of rejecting the stochastic disturbance in addition to set-point tracking. Design and robustness analysis of GPC have been well-covered in literature [2, 5]. Extended prediction self-adaptive and extended horizon adaptive control are other members of this family [6, 9, 15, 17, 18]. Interested readers are referred to the reviews of predictive control strategies [7, 8, 12].

This work extends the structure of GPC to batch processes. As mentioned, GPC utilizes an optimal predictor to predict the future output. To obtain an optimal predictor, knowledge of the process is required. Hence batch process modeling constitutes the first part of this chapter. Since batch processes are correlated across batches as well as along time, two-dimensional models are necessary for batch process identification. As a result, two-dimensional model structure is used for the development of GPC for batch processes, followed by an elaboration on design and implementation procedures. Simulation studies are conducted to illustrate the feasibility of the proposed methods.

The remainder of this chapter is structured as follows: Section 2 elaborates upon the identification procedure for two-dimensional batch processes. A new method is proposed to estimate two-dimensional linear models. Section 3 extends the structure of GPC to batch processes and explains the design and implementation procedures. Simulation studies are carried out in Section 4, followed by concluding remarks in Section 5.

4.2 Two-dimensional identification

Since continuous processes operate solely along time dimension, the conventional identification algorithm considers the dynamics of process variables along time dimension only [10, 16]. However, batch processes operate in two dimensions, along time and across batches. Hence the dynamics of the process variables in both dimensions need to be taken into account in identification.

Figure 4.1 illustrates the dependency of the actual and predicted outputs (y and $\hat{y}$) on the inputs (u which is the deterministic input and w which is a white noise). The area shown by dot-dashed lines includes the data needed to model the actual output at batch i and time k. As shown in Figure 4.1, the actual output depends on both inputs. The area shown by dashed lines includes the data needed to predict the output at batch i and time k. The past deterministic inputs and the past actual outputs are used to predict the future output.

Stochastic responses can be originated from various disturbance sources. Since some sources can affect more than one batch, it is reasonable to assume the stochastic responses have correlations across batches. Similarly, deterministic responses can also be interacted across batches. This chapter uses data across batches for identification.

Since process data of concern are two-dimensional, two back-shift operators are required to shift along time and across batches. z^{-1} and q^{-1} are defined as batch and time back-shift operators, respectively. Equation 4.1 shows a general linear model of the process at batch i and time k.

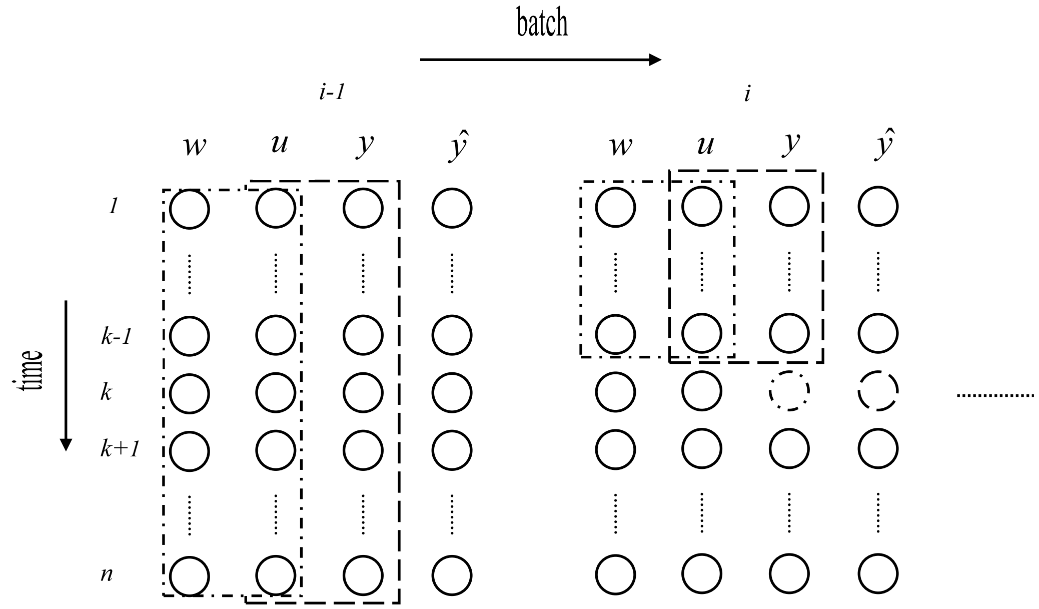


Figure 4.1: Variable relations for a SISO batch process

$$y_k^i = T(q^{-1}, z^{-1}, \theta)u_k^i + N(q^{-1}, z^{-1}, \theta)w_k^i \tag{4.1}$$

To model the stochastic response of a batch process, a two-dimensional white noise is defined. A two-dimensional white noise has no autocorrelations along time or across batches. In mathematical terms, the following equation holds for a two-dimensional white noise when $k \neq m$ and $i \neq n$.

$$E[w_k^i][w_m^n] = 0 \tag{4.2}$$

For identification purposes, output predictors are developed first. General predictors require the input and output sequences for all previous time and previous batches. Equation 4.3 presents a general structure for one-step ahead predictors.

$$\hat{y}_{(k|k-1)}^i = L_1(q^{-1}, z^{-1}, \theta)y_k^i + L_2(q^{-1}, z^{-1}, \theta)u_k^i \tag{4.3}$$

4.2.1 Optimal one-step ahead predictor

Chapter 1 has provided an overview of the optimal one-step ahead predictor for one-dimensional linear models. This section extends the mentioned procedure to two-dimensional linear models. Like the conventional identification procedure, the optimal predictor is found by minimizing the prediction error. Rewrite the definition of the prediction error for a two-dimensional process:

$$\varepsilon_k^i(\theta) = y_k^i - \hat{y}_{(k|k-1)}^i \tag{4.4}$$

Substituting Equations 4.1 and 4.3 into Equation 4.4 yields:

$$\begin{aligned}\varepsilon_k^i(\theta) =& T(q^{-1}, z^{-1}, \theta)u_k^i + N(q^{-1}, z^{-1}, \theta)w_k^i \\ &- L_1(q^{-1}, z^{-1}, \theta)y_k^i - L_2(q^{-1}, z^{-1}, \theta)u_k^i \qquad (4.5) \\ =& T(q^{-1}, z^{-1}, \theta)u_k^i + (N(q^{-1}, z^{-1}, \theta) - 1)w_k^i + w_k^i \\ &- L_1(q^{-1}, z^{-1}, \theta)y_k^i - L_2(q^{-1}, z^{-1}, \theta)u_k^i \qquad (4.6)\end{aligned}$$

From the inverse of Equation 4.1, we have:

$$w_k^i = N^{-1}(q^{-1}, z^{-1}, \theta)[y_k^i - T(q^{-1}, z^{-1}, \theta)u_k^i] \tag{4.7}$$

Replacing the white noise in Equation 4.6 with Equation 4.7 gives:

$$\begin{aligned}\varepsilon_k^i(\theta) =& T(q^{-1}, z^{-1}, \theta)u_k^i + (N(q^{-1}, z^{-1}, \theta) - 1) \\ & N^{-1}(q^{-1}, z^{-1}, \theta)[y_k^i - T(q^{-1}, z^{-1}, \theta)u_k^i] + w_k^i \\ &- L_1(q^{-1}, z^{-1}, \theta)y_k^i - L_2(q^{-1}, z^{-1}, \theta)u_k^i \qquad (4.8)\end{aligned}$$

which can be further simplified as:

$$\varepsilon_k^i(\theta) = \psi_u(q^{-1}, z^{-1}, \theta)u_k^i + \psi_y(q^{-1}, z^{-1}, \theta)y_k^i + w_k^i \tag{4.9}$$

where

$$\begin{aligned}\psi_u(q^{-1}, z^{-1}, \theta) =& N^{-1}(q^{-1}, z^{-1}, \theta)T(q^{-1}, z^{-1}, \theta) - L_2(q^{-1}, z^{-1}, \theta) \\ \psi_y(q^{-1}, z^{-1}, \theta) =& 1 - N^{-1}(q^{-1}, z^{-1}, \theta) - L_1(q^{-1}, z^{-1}, \theta) \qquad (4.10)\end{aligned}$$

Prediction is derived based on data from previous time as well as previous batches. Therefore, all the parameters of the predictor have either time delay or batch delay. In mathematical terms, the following holds:

$$L_1(q^{-1} = 0, z^{-1} = 0, \theta) = 0 \tag{4.11}$$

$$L_2(q^{-1} = 0, z^{-1} = 0, \theta) = 0 \tag{4.12}$$

The disturbance model can be normalized so that its leading term in both numerator and denominator is 1 and the following equation holds:

$$N(0, 0, \theta) = 1 \tag{4.13}$$

The plant model also has either time delay or batch delay in all terms. Hence from Equation 4.9, the following holds:

$$\psi_u(0,0,\theta) = 0 \tag{4.14}$$
$$\psi_y(0,0,\theta) = 0 \tag{4.15}$$

which implies that both ψ_u, ψ_y have either time delay or batch delay. Therefore, in the right hand side of Equation 4.9 there are no correlations between the first two terms and the white noise. Consequently, the variance of the prediction error can be derived as below:

$$var(\varepsilon_k^i(\theta)) = var(x) + var(w_k^i) \geq var(w_k^i) \tag{4.16}$$

where

$$x \triangleq \psi_u(q^{-1}, z^{-1}, \theta)u_k^i + \psi_y(q^{-1}, z^{-1}, \theta)y_k^i \tag{4.17}$$

Equation 4.16 presents the minimum variance of the prediction error. The optimal predictor is derived by equating x with zero, which results in:

$$\psi_u(q^{-1}, z^{-1}, \theta) = 0 \rightarrow N^{-1}(q^{-1}, z^{-1}, \theta)T(q^{-1}, z^{-1}, \theta) - L_2(q^{-1}, z^{-1}, \theta) = 0 \tag{4.18}$$
$$\psi_y(q^{-1}, z^{-1}, \theta) = 0 \rightarrow 1 - N^{-1}(q^{-1}, z^{-1}, \theta) - L_1(q^{-1}, z^{-1}, \theta) = 0 \tag{4.19}$$

Therefore, the optimal predictor is found to be:

$$\hat{y}^i_{(k|k-1)} = L_1^{opt}(q^{-1}, z^{-1}, \theta)y_k^i + L_2^{opt}(q^{-1}, z^{-1}, \theta)u_k^i \tag{4.20}$$

where

$$L_1^{opt}(q^{-1}, z^{-1}, \theta) = 1 - N^{-1}(q^{-1}, z^{-1}, \theta) \tag{4.21}$$
$$L_2^{opt}(q^{-1}, z^{-1}, \theta) = N^{-1}(q^{-1}, z^{-1}, \theta)T(q^{-1}, z^{-1}, \theta) \tag{4.22}$$

4.2.2 Prediction error method

Extending the prediction error method from one-dimensional process [10, 16] to two-dimensional process, the following objective function is defined:

$$\min_{\theta} \frac{1}{N} tr\{\sum_{i=1}^{n_i}\sum_{k=1}^{n_k} \varepsilon_k^i(\theta)\varepsilon_k^{i^T}(\theta)\} \tag{4.23}$$

where n_i and n_k denote the number of batches and data samples collected, respectively. N represents the total data points. The prediction error is computed from:

$$\varepsilon_k^i(\theta) = y_k^i - [1 - \hat{N}^{-1}(q^{-1}, z^{-1}, \theta)]y_k^i - [\hat{N}^{-1}(q^{-1}, z^{-1}, \theta)\hat{T}(q^{-1}, z^{-1}, \theta)]u_k^i \quad (4.24)$$

where $\hat{T}, \hat{N}$ are the plant and disturbance models to be estimated. A general linear two-dimensional model is structured as:

$$G(q^{-1}, z^{-1}, \theta) = \frac{b_{00} + b_{01}q^{-1} + b_{10}z^{-1} + b_{11}z^{-1}q^{-1} + \cdots}{f_{00} + f_{01}q^{-1} + f_{10}z^{-1} + f_{11}z^{-1}q^{-1} + \cdots} \quad (4.25)$$

Two linear two-dimensional models are considered for $\hat{T}$ and $\hat{N}$, and model orders with respect to each back-shift operator are chosen. Model parameters can then be estimated by optimizing Equation 4.23.

4.2.3 Model validation tests

In the prediction error method, selected model orders may not be suitable for the process. Hence we need to validate the estimated models. Recall the optimal predictor:

$$\hat{y}_{(k|k-1)}^i = [1 - \hat{N}^{-1}(q^{-1}, z^{-1}, \theta)]y_k^i + [\hat{N}^{-1}(q^{-1}, z^{-1}, \theta)\hat{T}(q^{-1}, z^{-1}, \theta)]u_k^i \quad (4.26)$$

Substituting Equation 4.1 into Equation 4.26 gives:

$$\begin{aligned}\hat{y}_{(k|k-1)}^i =& [1 - \hat{N}^{-1}(q^{-1}, z^{-1}, \theta)][T(q^{-1}, z^{-1}, \theta)u_k^i + N(q^{-1}, z^{-1}, \theta)w_k^i] \\ &+ [\hat{N}^{-1}(q^{-1}, z^{-1}, \theta)\hat{T}(q^{-1}, z^{-1}, \theta)]u_k^i \end{aligned} \quad (4.27)$$

Hence the prediction error can be rewritten as:

$$\begin{aligned}\varepsilon_k^i(\theta) =& y_k^i - \hat{y}_{(k|k-1)}^i \\ =& [\hat{N}^{-1}(q^{-1}, z^{-1}, \theta)(T^{-1}(q^{-1}, z^{-1}, \theta) - \hat{T}^{-1}(q^{-1}, z^{-1}, \theta))]u_k^i \\ &+ [\hat{N}^{-1}(q^{-1}, z^{-1}, \theta)N(q^{-1}, z^{-1}, \theta)]w_k^i \end{aligned} \quad (4.28)$$

Three tests can be considered:

[1] Cross-correlation test:

This test checks the plant model estimation. From Equation 4.28, if the estimated plant model is sufficiently close to the actual plant model, then the prediction error and the current batch input will not be cross-correlated. Cross-correlation between the prediction error and the input is computed from:

$$\rho_{\varepsilon^i u^i}(\tau) = \frac{r_{\varepsilon^i u^i}(\tau)}{\sqrt{r_\varepsilon^i(0) r_u^i(0)}} \tag{4.29}$$

where $r_{\varepsilon^i u^i}(\tau)$ represents the covariance between the prediction error and input sequences of i^{th} batch with τ time lags. In practice, if more than 99% $\rho_{\varepsilon^i u^i}(\tau)$ satisfy $|\rho_{\varepsilon^i u^i}(\tau)| < \frac{3}{\sqrt{N}}$, plant model estimation is acceptable and the estimated plant model is validated [16].

[2] Auto-correlation test along time:

From Equation 4.28, if the estimated models are sufficiently close to the actual models, the prediction error will be a two-dimensional white noise. Therefore, two tests are defined to ensure there are no autocorrelations in the prediction error along time or across batches. The first test is to calculate the autocovariance for each batch, as shown below:

$$\rho_{\varepsilon^i}(\tau) = \frac{r_{\varepsilon^i}(\tau)}{r_{\varepsilon^i}(0)} \tag{4.30}$$

where $r_{\varepsilon^i}(\tau)$ denotes the prediction error autocovariance for i^{th} batch with τ time lags.

If more than 99% $\rho_{\varepsilon^i}(\tau)$ satisfy $|\rho_{\varepsilon^i}(\tau)| < \frac{3}{\sqrt{N}}$, the prediction error is a white noise along time and the estimated models pass the test.

[3] Auto-correlation test across batches

To ensure that the prediction error is a white noise across batches, its autocovariance at a given time step is calculated as below:

$$\rho_{\varepsilon_k}(\tau) = \frac{r_{\varepsilon_k}(\tau)}{r_{\varepsilon_k}(0)} \tag{4.31}$$

where $r_{\varepsilon_k}(\tau)$ denotes the prediction error autocovariance for all batches at time k with τ batch lags.

If more than 99% $\rho_{\varepsilon_k}(\tau)$ satisfy $|\rho_{\varepsilon_k}(\tau)| < \frac{3}{\sqrt{N}}$, the prediction error is a white noise across batches and the estimated models pass the test.

If all the tests are passed, the estimated models are validated. Otherwise, a different model structure is chosen and the prediction error method is repeated.

4.2.4 Equivalent one-dimensional models

In terms of mathematical modeling, batch processes can be represented by one-dimensional linear models equivalent to two-dimensional models. In other words, two-dimensional models can be converted into one-dimensional models. To elaborate this conversion, first we represent two-dimensional data by a vector, as shown in Figure 4.2.

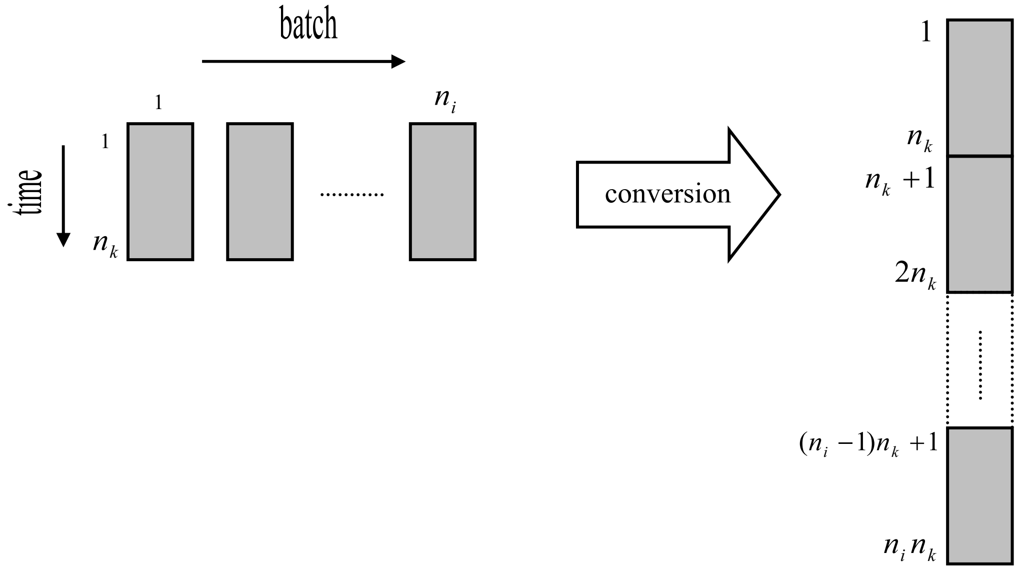


Figure 4.2: Conversion of two-dimensional data into one dimension

Note that n_k and n_i represent the total number of time and batch steps. A new dimension is defined which we call time-batch.

Definition: For two-dimensional data, two indices are used to distinguish data. When turning the data into a vector, one index along the new defined dimension is needed. A function, called 1-D converter, is defined to calculate the equivalent index:

$$\Gamma(i,k) = (i-1) * n_k + k \tag{4.32}$$

The sample data obtained at batch i and time k can be found at time-batch $\Gamma(i,k)$ after building the vector.

All the dynamics of data along time and across batches can be presented in time-batch dimension. To explain the idea further, suppose the following two-dimensional linear model holds for a batch process:

$$y_k^i = (b_{01}q^{-1} + b_{10}z^{-1})u_k^i = b_{01}u_{k-1}^i + b_{10}u_k^{i-1} \tag{4.33}$$

The time back-shift operator shifts data one time step within the batch, which is equivalent to one time-batch step. The batch back-shift operator shifts data one batch step at current time, which is equivalent to n_k time-batch steps. Hence Equation 4.33 is equivalent to:

$$Y(l) = b_{01}U(l-1) + b_{10}U(l-n_k) \tag{4.34}$$

where m, U and Y denote the time-batch index, input and output along time-batch, respectively. Index l can be computed from 1-D converter function.

Equation 4.34 presents an equivalent linear correlation between variables along time-batch. By defining p^{-1} as time-batch back-shift operator, the following model is obtained:

$$Y(l) = (b_{01}p^{-1} + b_{10}p^{-n_k})U(l) \tag{4.35}$$

which shows an equivalent one-dimensional linear model for the process.

4.2.5 Example

Consider the following process:

$$T = \frac{0.7q^{-2}}{1-0.3q^{-1}} \tag{4.36}$$

$$N = \frac{1+0.2q^{-1}}{1-0.7z^{-1}} \tag{4.37}$$

In this process, the plant model has dynamics along the time but not along the batches while the disturbance model has two-dimensional dynamics. Conventional identification procedure is applied first to identify the process by one-dimensional models. Note that this procedure can be done by any batch data. In this example, the tenth batch data were used. The estimated models are:

$$\hat{T} = \frac{0.6911q^{-2}}{1-0.301q^{-1}} \tag{4.38}$$

$$\hat{N} = \frac{1+0.2782q^{-1}}{1+0.0572q^{-1}} \tag{4.39}$$

which fail to capture the dynamics of the disturbance model across batches. Performing the residual tests, Figure 4.3 is obtained. Obviously, the autocorrelation test across batches is not passed.

Following the proposed procedure for identification of two-dimensional models, the following models are estimated:

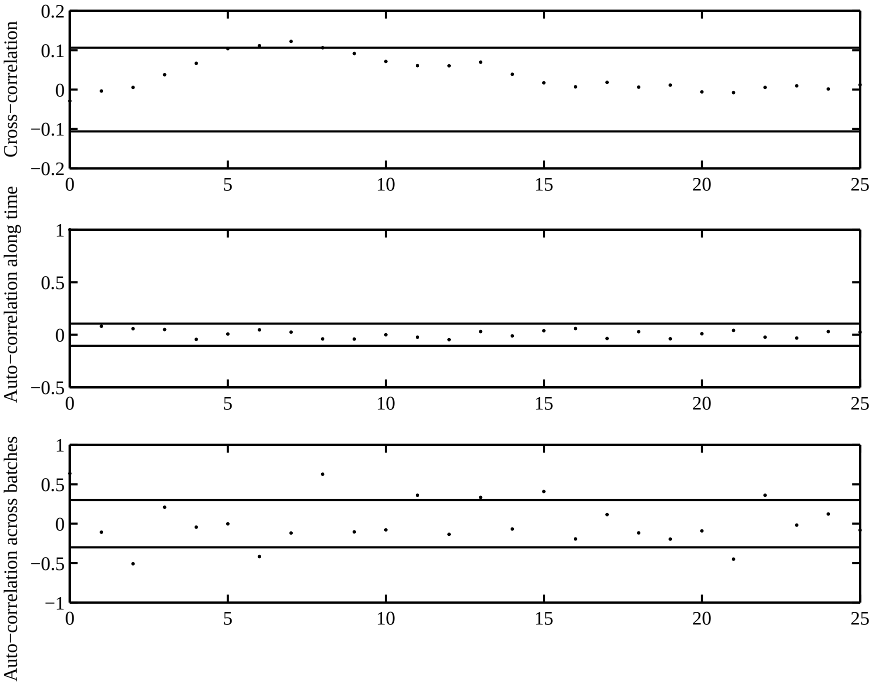


Figure 4.3: Residual tests for the conventional identification procedure for the Example

$$\hat{T} = \frac{0.6962q^{-2}}{1 - 0.3041q^{-1}} \tag{4.40}$$

$$\hat{N} = \frac{1 + 0.3018q^{-1}}{1 - 0.7048z^{-1}} \tag{4.41}$$

which are close to the actual models. Performing the residual tests, as shown in Figure 4.4, the estimated models are validated.

4.3 Generalized predictive control

This control strategy deals with minimization of the predicted output error from the set-point trajectory. For this purpose, an optimal predictor is derived based on the estimated plant and disturbance models. Since the predicted outputs depend on the input, minimizing the predicted output error can determine optimal inputs for the process.

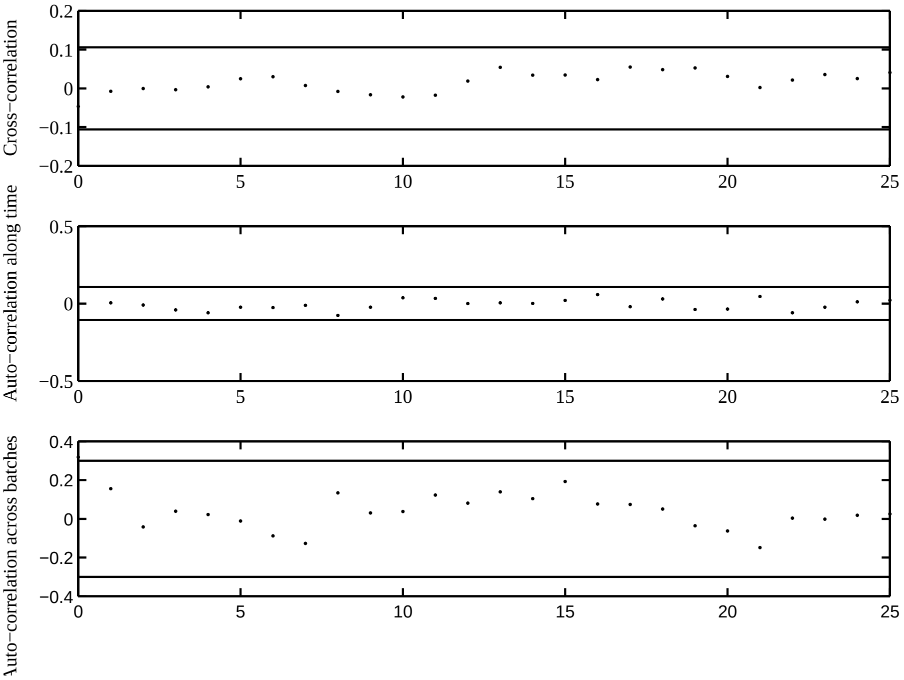


Figure 4.4: Residual tests for the two-dimensional identification procedure for the Example

4.3.1 Algorithm set-up

Since batch variables are correlated along time and across batches, the future outputs in both dimensions need to be considered for the optimization. To demonstrate this idea, Figure 4.5 is considered. Note that filled circles represent past outputs.

Assume that the process is at i^{th} batch and k^{th} time. The input is set by solving the optimization problem. Prior to setting up the objective function, some concepts need to be introduced:

[1] **Prediction horizon:** Minimization of the predicted output error is performed over some future outputs along both dimensions. The dashed rectangular area is called prediction horizon, and includes the selected outputs. This horizon consists of $m_2 * n_2 - 1$ outputs in both dimensions.

[2] **Control horizon:** Current and several future inputs are chosen to be the minimization decision parameters. The area including the selected inputs is called control horizon, shown by solid rectangle in Figure 4.5. This horizon consists of $m_1 * n_1$ inputs in both dimensions.

[3] **Available index set:** This set encompasses batch and time indices of all the available data. In mathematical terms:

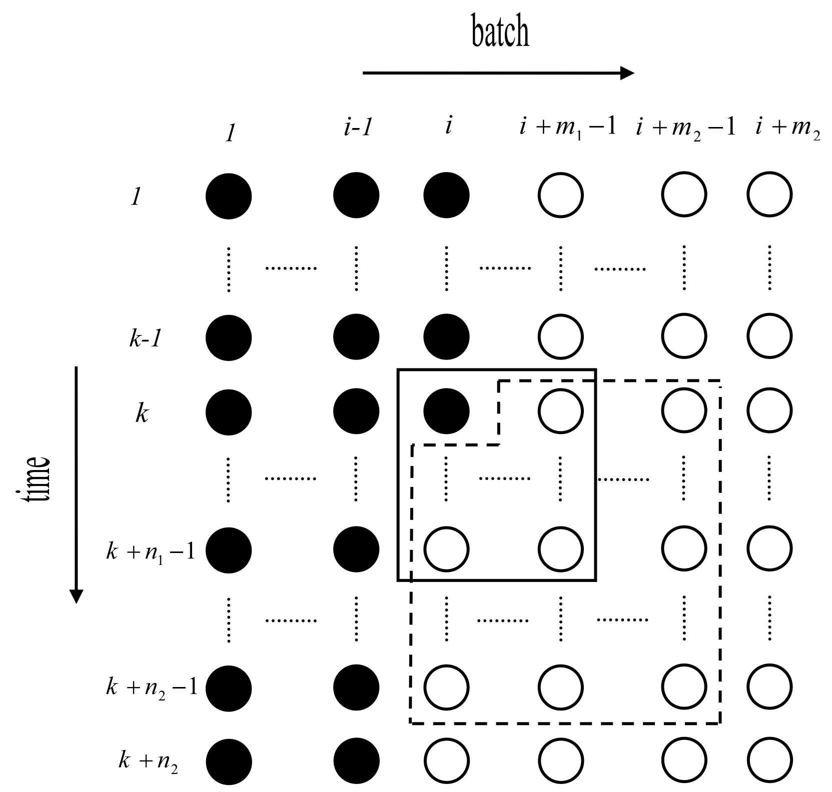


Figure 4.5: Two-dimensional GPC

$$\phi(i,k) = \{(m,n)|(m < i) \text{ or } (m = i \text{ and } n <= k)\}$$

Equivalently, we can introduce the set in time-batch dimension:

$$\Phi(l) = \{l|l <= \Gamma(i,k)\} \tag{4.42}$$

[4] **Future index set:** This set encompasses batch and time indices of the future data. In mathematical terms:

$$\acute{\phi}(i,k) = \{(m,n)|(m > i) \text{ or } (m = i \text{ and } n > k)\}$$

The equivalent set in time-batch dimension is:

$$\acute{\Phi}(l) = \{l|l > \Gamma(i,k)\} \tag{4.43}$$

[5] **Computable transfer function:** Assume G is a two-dimensional transfer function operating on y_{k+b}^{i+a}. If all the terms of G address the available

data, i.e. obtained prior to i^{th} batch and k^{th} time, then G is called computable with respect to *(a,b)*. Equation 4.44 gives a general structure for G.

$$G = \sum_{m=0}^{m_g} \sum_{n=0}^{n_g} g_{m,n} z^{-m} q^{-n} \tag{4.44}$$

Each term that satisfies Equation 4.45 is called computable with respect to *(a,b)*. If all the terms of G satisfy the equation, then G is called computable with respect to *(a,b)*.

$$\begin{aligned} &(i+a-m, k+b-n) \in \phi(i,k) \\ &\Gamma(i+a-m, k+b-n) \in \Phi(\Gamma(i,k)) \end{aligned} \tag{4.45}$$

Note that the conditions in Equation 4.45 are equivalent. The first one is used for two-dimensional models, while the second one is used for the converted one-dimensional models.

Remark: If a transfer function is computable with respect to *(a,b)*, its equivalent one-dimensional model has a delay of at least $\Gamma(a,b)$ because it does not address the $\Gamma(a,b)$ data samples that are collected between y_k^i and y_{k+b}^{i+a}.

Computable transfer functions have two important properties:

- Summation: If G and H are computable transfer functions with respect to *(a,b)*, then G+H will also be computable with respect to *(a,b)*.

  Proof:

$$G = \sum_{m=0}^{m_g} \sum_{n=0}^{n_g} g_{m,n} z^{-m} q^{-n} \tag{4.46}$$

$$H = \sum_{m=0}^{m_h} \sum_{n=0}^{n_h} h_{m,n} z^{-m} q^{-n} \tag{4.47}$$

$$G + H = \sum_{m=0}^{max(m_h,m_g)} \sum_{n=0}^{max(n_h,n_g)} g_{m,n} + h_{m,n} z^{-m} q^{-n} \tag{4.48}$$

  Since G and H are computable, all the values of m and n satisfy Equation 4.45. Therefore, from Equation 4.48, G+H is computable with respect to *(a,b)*.

- Multiplication: If G is a computable transfer function with respect to *(a,b)*, then G*R will also be computable with respect to *(a,b)*.

  Proof: Let us write the equivalent one-dimensional models for G and R:

$$G = \sum_{l=\Gamma(a,b)}^{l_g} g_l p^{-l} \tag{4.49}$$

$$R = \sum_{l=0}^{l_r} r_l p^{-l} \tag{4.50}$$

$$\tag{4.51}$$

  The equivalent one-dimensional model for G*R is:

$$G * R = \sum_{l=\Gamma(a,b)}^{l_g * l_r} gr_l p^{-l} \tag{4.52}$$

  G*R has a delay of at least $\Gamma(a,b)$, which indicates that it is computable with respect to *(a,b)*.

4.3.2 Design procedure

The basis of the predictive controllers is minimization of the output error from the set-point trajectory. Hence the following cost function is considered over the prediction horizon:

$$J(m_2, n_2) = E\{\sum_{a=0}^{m_2-1} \sum_{b=0}^{n_2-1} [y_{(k+b)}^{(i+a)} - y_{k+b}^d]^2\} \tag{4.53}$$

where m_2, n_2, and y_{k+b}^d represent the across-batch and along-time lengths of the prediction horizon and the set-point at time $(k + b)$, respectively. Note that since only data up to i^{th} batch and k^{th} time is available, expectation of the future output error is used. However, we can substitute the future output with its predicted value.

Optimal a-b-step ahead predictor

Output of a process consists of deterministic and stochastic responses. Since deterministic response is derived from the process input, it can be directly calculated. But predicting the stochastic response is more challenging because its input is noise. However, an optimal predictor can use previous outputs to estimate parts of future disturbance effects. The optimal a-b-step ahead predictor estimates the output a-step ahead across batches and b-step ahead along time.

Consider the following linear process model:

$$y_k^i = T(q^{-1}, z^{-1}, \theta)u_k^i + N(q^{-1}, z^{-1}, \theta)w_k^i \tag{4.54}$$

General a-b-step ahead predictors are structured as:

$$\hat{y}_{(k+b|k)}^{(i+a|i)} = L_1(q^{-1}, z^{-1}, \theta)y_{k+b}^{i+a} + L_2(q^{-1}, z^{-1}, \theta)u_{k+b}^{i+a} \tag{4.55}$$

Lemma: The optimal a-b-step ahead predictor is:

$$L_1^{opt}(q^{-1}, z^{-1}, \theta) = R_b^a(q^{-1}, z^{-1}, \theta)N^{-1}(q^{-1}, z^{-1}, \theta) \tag{4.56}$$

$$L_2^{opt}(q^{-1}, z^{-1}, \theta) = F_b^a(q^{-1}, z^{-1}, \theta)N^{-1}(q^{-1}, z^{-1}, \theta)T(q^{-1}, z^{-1}, \theta) \tag{4.57}$$

where F and R are computed from:

$$N(q^{-1}, z^{-1}, \theta) = F_b^a(q^{-1}, z^{-1}, \theta) + R_b^a(q^{-1}, z^{-1}, \theta) \tag{4.58}$$

with R_b^a including all the computable terms of N with respect to *(a,b)*.

Proof: Substituting Equation 4.58 into Equation 4.54 and writing for the future output yields:

$$y_{k+b}^{i+a} = Tu_{k+b}^{i+a} + F_b^a w_{k+b}^{i+a} + R_b^a w_{k+b}^{i+a} \tag{4.59}$$

Note that for the sake of simplicity, the back-shift operators have been dropped. From the process model, we find:

$$w_{k+b}^{i+a} = N^{-1}[y_{k+b}^{i+a} - Tu_{k+b}^{i+a}] \tag{4.60}$$

Let us rewrite Equation 4.59:

$$y_{k+b}^{i+a} = Tu_{k+b}^{i+a} + F_b^a w_{k+b}^{i+a} + R_b^a N^{-1}[y_{k+b}^{i+a} - Tu_{k+b}^{i+a}] \tag{4.61}$$

The optimal predictor is defined by minimizing the prediction error. Based on definition, the prediction error can be written as:

$$\varepsilon_{(k+b|k)}^{(i+a|i)} = y_{k+b}^{i+a} - \hat{y}_{(k+b|k)}^{(i+a|i)} \tag{4.62}$$

$$= Tu_{k+b}^{i+a} + F_b^a w_{k+b}^{i+a} + R_b^a N^{-1}[y_{k+b}^{i+a} - Tu_{k+b}^{i+a}] - [L_1 y_{k+b}^{i+a} + L_2 u_{k+b}^{i+a}] \tag{4.63}$$

$$= [T - R_b^a N^{-1}T - L_2]u_{k+b}^{i+a} + [R_b^a N^{-1} - L_1]y_{k+b}^{i+a} + F_b^a w_{k+b}^{i+a} \tag{4.64}$$

Let us define two new variables:

$$\psi_u = T - R_b^a N^{-1} T - L_2 \tag{4.65}$$
$$\psi_y = R_b^a N^{-1} - L_1 \tag{4.66}$$

Updating Equation 4.64 yields:

$$\varepsilon_{(k+b|k)}^{(i+a|i)} = \psi_u u_{k+b}^{i+a} + \psi_y y_{k+b}^{i+a} + F_b^a w_{k+b}^{i+a} \tag{4.67}$$

Because L_1 uses available data for prediction, it is computable with respect to *(a,b)*. In addition, R_b^a is, by definition, computable with respect to *(a,b)*. Therefore, from the multiplication property of computable transfer functions, ψ_y is computable with respect to *(a,b)*. On the other hand, F_b^a is composed of only non-computable terms with respect to *(a,b)*. Consequently, $\psi_y y_{k+b}^{i+a}$ and $F_b^a w_{k+b}^{i+a}$ are uncorrelated. Besides, there are no cross-correlations between the future input and the white noise. Therefore, the variance of the prediction error is equal to:

$$\begin{aligned} var(\varepsilon_{(k+b|k)}^{(i+a|i)}) &= var(\psi_u u_{k+b}^{i+a} + \psi_y y_{k+b}^{i+a}) + var(F_b^a w_{k+b}^{i+a}) && (4.68)\\ &\geq var(F_b^a w_{k+b}^{i+a}) && (4.69) \end{aligned}$$

To achieve the minimum variance solution for the prediction error, the following must hold:

$$\psi_u = 0 \rightarrow L_2^{opt} = T - R_b^a N^{-1} T = F_b^a N^{-1} T \tag{4.70}$$
$$\psi_y = 0 \rightarrow L_1^{opt} = R_b^a N^{-1} \tag{4.71}$$

Predictive control law

Each batch process has a set-point trajectory to follow. The objective of the predictive control law is to compute control actions such that the difference between the future output and set-point is minimized. Since the future outputs are unknown at the time of making the decision, the difference between the predicted output and set-point (predicted output error) is minimized. Nonetheless, drastic control actions may be required to achieve the minimum output error, which may be expensive or infeasible to take. Hence control actions must be considered to be within an acceptable range with regard to cost and feasibility.

The challenge is to manipulate the objective function in order to guarantee acceptable control actions. For a continuous process, this problem is solved by minimizing control increments over time. In other words, change of control actions over time is penalized, and each control action cannot be too far away from the previous ones. For a batch process, penalizing control increments over batch can also be considered. Therefore, various objective functions can be defined:

[1] This objective function combines the minimization of the predicted output error and control increments over time. For this purpose, the following cost function is defined:

$$\begin{aligned} J(m_1, n_1, m_2, n_2) &= \sum_{a=0}^{m_2-1} \sum_{b=0}^{n_2-1} [\hat{y}_{(k+b|k)}^{(i+a|i)} - y_{k+b}^{d}]^2 \\ &+ \sum_{a=0}^{m_1-1} \sum_{b=0}^{n_1-1} \lambda(a,b)[\Delta^{i+a} u_{k+b}^{i+a}]^2 \end{aligned} \tag{4.72}$$

where $\lambda(a, b)$ is a two-dimensional control-weighting sequence, and control increment is defined as:

$$\Delta^{i+a} u_{k+b}^{i+a} = u_{k+b}^{i+a} - u_{k+b-1}^{i+a} \tag{4.73}$$

To compute the cost function, all the terms should be written as functions of control increments. Let us recall the optimal a-b-step ahead predictor:

$$\hat{y}_{(k+b|k)}^{(i+a|i)} = F_b^a N^{-1} T u_{k+b}^{i+a} + R_b^a N^{-1} y_{k+b}^{i+a} \tag{4.74}$$

The input is undetermined after batch i and time *k-1*. A new variable (r_m^n) is defined equal to the input for all available data. For unavailable data, after batch i and time *k-1*, input increments over time are assumed zero, and r_m^n contains the inputs at previous time steps. In mathematical terms:

$$r_m^n = u_m^n, (n, m) \in \phi(i, k-1) \tag{4.75}$$

$$r_m^n = u_k^n, (n, m) \in \acute{\phi}(i, k-1) \tag{4.76}$$

Hence r_m^n is known for all the batches and time steps. Replacing the input in Equation 4.74 with r_m^n yields:

$$\hat{y}_{(k+b|k)}^{(i+a|i)} = H_b^a \tilde{u}_k^i(m_1, n_1) + F_b^a N^{-1} T r_{k+b}^{i+a} + R_b^a N^{-1} y_{k+b}^{i+a} \tag{4.77}$$

where

$$\tilde{u}^i_k(m_1, n_1) = \begin{bmatrix} \Delta^i u^i_k & \Delta^{i+1} u^{i+1}_k & \cdots & \Delta^{i+m_1-1} u^{i+m_1-1}_k \\ \Delta^i u^i_{k+1} & \Delta^{i+1} u^{i+1}_{k+1} & \cdots & \Delta^{i+m_1-1} u^{i+m_1-1}_{k+1} \\ \vdots & \vdots & \vdots & \vdots \\ \Delta^i u^i_{k+n_1-1} & \Delta^{i+1} u^{i+1}_{k+n_1-1} & \cdots & \Delta^{i+m_1-1} u^{i+m_1-1}_{k+n_1-1} \end{bmatrix}_{n_1*m_1} \tag{4.78}$$

H^a_b is a transfer function defined for the incremental input. Prior to calculating H^a_b, note that only the first term of Equation 4.77 depends on the input increments and the rest are constant. A new variable can be defined to represent the constant part:

$$\bar{y}^{i+a}_{k+b} \triangleq F^a_b N^{-1} T r^{i+a}_{k+b} + R^a_b N^{-1} y^{i+a}_{k+b} \tag{4.79}$$

Substituting Equation 4.79 into Equation 4.77 yields:

$$\hat{y}^{(i+a|i)}_{(k+b|k)} = H^a_b \tilde{u}^i_k(m_1, n_1) + \bar{y}^{i+a}_{k+b} \tag{4.80}$$

For the sake of simplicity, all two-dimensional matrices are converted to vectors:

$$\hat{Y} = [\hat{y}^i_{(k+1|k)}, \hat{y}^i_{(k+2|k)}, \cdots, \hat{y}^i_{(k+n_2-1|k)}, \hat{y}^{(i+1|i)}_k, \cdots, \hat{y}^{(i+m_2-1|i)}_{(k+n_2-1|k)}]^T_{(m_2n_2-1)*1} \tag{4.81}$$

$$\tilde{U} = [\Delta^i u^i_k, \Delta^i u^i_{k+1}, \cdots, \Delta^i u^i_{k+n_1-1}, \cdots, \Delta^{i+m_1-1} u^{i+m_1-1}_{k+n_1-1}]^T_{(m_1n_1)*1} \tag{4.82}$$

$$\bar{Y} = [\bar{y}^i_{k+1}, \bar{y}^i_{k+2}, \cdots, \bar{y}^i_{k+n_2-1}, \bar{y}^{i+1}_k, \cdots, \bar{y}^{i+m_2-1}_{k+n_2-1}]^T_{(m_2n_2-1)*1} \tag{4.83}$$

$$Y^d = [y^d_{k+1}, y^d_{k+2}, \cdots, y^d_{k+n_2-1}, y^d_k, \cdots, y^d_{k+n_2-1}]^T_{(m_2n_2-1)*1} \tag{4.84}$$

$\hat{Y}$ includes the future outputs within the prediction horizon, and $\tilde{U}$ includes the input increments within the control horizon. $\bar{Y}$ is a constant vector that can be computed based on available data. Extending Equation 4.80, we have:

$$\hat{Y} = H\tilde{U} + \bar{Y} \tag{4.85}$$

Equation 4.85 presents the optimal predictor over the entire prediction horizon with respect to the control horizon. To complete the structure of the predictor, definition of H is required. First, we model the future outputs one by one:

$$
\begin{aligned}
&\hat{y}^{i}_{(k+1|k)} = h^{0}_{1}(0,1)\Delta^{i}u^{i}_{k} + \bar{y}^{i}_{k+1} \\
&\hat{y}^{i}_{(k+2|k)} = h^{0}_{2}(0,2)\Delta^{i}u^{i}_{k} + h^{0}_{2}(0,1)[\Delta^{i}u^{i}_{k} + \Delta^{i}u^{i}_{k+1}] + \bar{y}^{i}_{k+2} \\
&\vdots \\
&\hat{y}^{(i+1|i)}_{k} = h^{1}_{0}(1,0)\Delta^{i}u^{i}_{k} + \bar{y}^{i+1}_{k} \\
&\hat{y}^{(i+1|i)}_{(k+1|k)} = h^{1}_{1}(1,1)\Delta^{i}u^{i}_{k} + h^{1}_{1}(1,0)[\Delta^{i}u^{i}_{k} + \Delta^{i}u^{i}_{k+1}] + \bar{y}^{i+1}_{k+1} \\
&\vdots
\end{aligned}
\tag{4.86}
$$

where $h^{a}_{b}(i,j)$ is an impulse response coefficient calculated from:

$$F^{a}_{b}N^{-1}T = \sum_{i}\sum_{j} h^{a}_{b}(i,j)z^{-i}q^{-j} \tag{4.87}$$

Generalizing the derivations of the future outputs, we have:

$$
\begin{aligned}
\hat{y}^{(i+a|i)}_{(k+b|k)} =& [h^{a}_{b}(a,0) + \cdots + h^{a}_{b}(a,b)]\Delta^{i}u^{i}_{k} \\
&+[h^{a}_{b}(a,0) + \cdots + h^{a}_{b}(a,b-1)]\Delta^{i}u^{i}_{k+1} \\
&\ \vdots \\
&+h^{a}_{b}(a,0)\Delta^{i}u^{i}_{k+b} \\
&+[h^{a}_{b}(a-1,0) + \cdots + h^{a}_{b}(a-1,b)]\Delta^{i+1}u^{i+1}_{k} \\
&+[h^{a}_{b}(a-1,0) + \cdots + h^{a}_{b}(a-1,b-1)]\Delta^{i+1}u^{i+1}_{k+1} \\
&\ \vdots \\
&+[h^{a}_{b}(a-m_2+1,0) + \cdots \\
&\quad + h^{a}_{b}(a-m_2+1,b-n_2+1)]\Delta^{i+m_2-1}u^{i+m_2-1}_{k+n_2-1} \\
&+\bar{y}^{i+a}_{k+b}
\end{aligned}
\tag{4.88}
$$

Note that the future input increments beyond the control horizon are assumed zero. In other words,

$$\Delta^{i+a}u^{i+a}_{k+b} = 0, a \geq 0, b \geq m_2 \tag{4.89}$$

Modeling all the future outputs within the prediction horizon, H can be written in terms of the impulse responses of $F^{a}_{b}N^{-1}T$.

Having introduced the optimal predictor in vector format, i.e. Equation 4.85, the cost function defined in Equation 4.72 can be updated:

$$J = (H\tilde{U} + \bar{Y} - Y^{d})^{T}Q(H\tilde{U} + \bar{Y} - Y^{d}) + \tilde{U}^{T}R\tilde{U} \tag{4.90}$$

where Q and R are weighting matrices. Accordingly, the following objective function is considered:

$$\min_{\tilde{U}} J = \min_{\tilde{U}}[(H\tilde{U} + \bar{Y} - Y^d)^T Q(H\tilde{U} + \bar{Y} - Y^d) + \tilde{U}^T R\tilde{U}] \quad (4.91)$$

Assuming no constraints on the input or the output, the minimization can be conducted by taking the derivative of the cost function with respect to $\tilde{U}$ and equating it with zero. Equation 4.92 yields the optimal solution for the input increments.

$$\tilde{U}^{opt} = (H^T QH + R)^{-1} H^T Q^T (Y^d - \bar{Y}) \quad (4.92)$$

[2] The second objective function combines the minimization of the predicted output error and control increments over batches. The following cost function is defined:

$$\begin{aligned} J(m_1, n_1, m_2, n_2) = & \sum_{a=0}^{m_2-1} \sum_{b=0}^{n_2-1} [\hat{y}^{(i+a|i)}_{(k+b|k)} - y^d_{k+b}]^2 \\ & + \sum_{a=0}^{m_1-1} \sum_{b=0}^{n_1-1} \lambda(a,b)[\Delta_{k+b} u^{i+a}_{k+b}]^2 \end{aligned} \quad (4.93)$$

where input increment is defined as:

$$\Delta_{k+b} u^{i+a}_{k+b} = u^{i+a}_{k+b} - u^{i+a-1}_{k+b} \quad (4.94)$$

Let us define a new variable, r^i_k, to contain all the available data of the input sequence with the future input increments over batch equal to zero, as shown below:

$$r^n_m = u^n_m, (n, m) \in \phi(i, k-1) \quad (4.95)$$

$$r^n_m = u^i_m, (n, m) \in \acute{\phi}(i, k-1) \quad (4.96)$$

Updating the optimal predictor gives:

$$\hat{y}^{(i+a|i)}_{(k+b|k)} = H^a_b \tilde{u}^i_k(m_1, n_1) + \bar{y}^{i+a}_{k+b} \quad (4.97)$$

where

$$\bar{y}_{k+b}^{i+a} = F_b^a N^{-1} Tr_{k+b}^{i+a} + R_b^a N^{-1} y_{k+b}^{i+a} \tag{4.98}$$

and

$$\tilde{u}_k^i(m_1, n_1) = \begin{bmatrix} \Delta_k u_k^i & \cdots & \Delta_k u_k^{i+m_1-1} \\ \Delta_{k+1} u_{k+1}^i & \cdots & \Delta_{k+1} u_{k+1}^{i+m_1-1} \\ \vdots & \vdots & \vdots \\ \Delta_{k+n_1-1} u_{k+n_1-1}^i & \cdots & \Delta_{k+n_1-1} u_{k+n_1-1}^{i+m_1-1} \end{bmatrix}_{n_1 * m_1} \tag{4.99}$$

Two-dimensional matrices are converted into vectors. All the vectors are defined the same as the previous case, apart from the input increment which is:

$$\tilde{U}^* = [\Delta_k u_k^i, \Delta_{k+1} u_{k+1}^i, \cdots, \Delta_{k+1} u_k^{i+1}, \cdots, \Delta_{k+n_1-1} u_{k+n_1-1}^{i+m_1-1}]^T_{(m_1 n_1)*1} \tag{4.100}$$

Writing the optimal predictor in terms of the vectors, Equation 4.101 is obtained.

$$\hat{Y} = H^* \tilde{U}^* + \bar{Y} \tag{4.101}$$

To build H^*, a general structure for the future outputs is given:

$$\begin{aligned} \hat{y}_{(k+b|k)}^{(i+a|i)} =& [h_b^a(0,b) + \cdots + h_b^a(a,b)] \Delta_k u_k^i \\ &+ [h_b^a(0,b-1) + \cdots + h_b^a(a,b-1)] \Delta_{k+1} u_{k+1}^i \\ &\ \vdots \\ &+ [h_b^a(0,0) + \cdots + h_b^a(a,0)] \Delta_{k+b} u_{k+b}^i \\ &+ [h_b^a(0,b) + \cdots + h_b^a(a-1,b)] \Delta_k u_k^{i+1} \\ &+ [h_b^a(0,b-1) + \cdots + h_b^a(a-1,b-1)] \Delta_{k+1} u_{k+1}^{i+1} \\ &\ \vdots \\ &+ [h_b^a(0,b-n_2+1) + \cdots + h_b^a(a-m_2+1,b-n_2+1)] \\ &\ \Delta_{k+n_2-1} u_{k+n_2-1}^{i+m_2-1} \\ &+ \bar{y}_{k+b}^{i+a} \end{aligned} \tag{4.102}$$

where $h_b^a(i,j)$ is an impulse response coefficient calculated from:

$$F_b^a N^{-1} T = \sum_i \sum_j h_b^a(i,j) z^{-i} q^{-j} \tag{4.103}$$

The future input increments beyond the control horizon are assumed zero and H^* is constructed. Then, the following cost and objective functions are defined:

$$J = (H^*\tilde{U}^* + \bar{Y} - Y^d)^T Q (H^*\tilde{U}^* + \bar{Y} - Y^d) + \tilde{U}^{*T} R \tilde{U}^* \tag{4.104}$$

$$\min_{\tilde{U}^*} J = \min_{\tilde{U}^*} [(H^*\tilde{U}^* + \bar{Y} - Y^d)^T Q (H^*\tilde{U}^* + \bar{Y} - Y^d) + \tilde{U}^{*T} R \tilde{U}^*] \tag{4.105}$$

Conducting the minimization under no constraints, the optimal input increments are achieved:

$$\tilde{U}^{*opt} = (H^{*T} Q H^* + R)^{-1} H^{*T} Q^T (Y^d - \bar{Y}) \tag{4.106}$$

This section has introduced two cost functions, Equations 4.90 and 4.104, which are equivalent in terms of predicted error minimization, but compute the input increments differently, one along time and the other along batch. However, only one cost function is needed while designing GPC strategy. Applications of both cost functions are shown in the following section. When making the decision, minimizing the cost function yields the optimal input increments at each step. For receding horizon implementation, only the first input increment is actually implemented. This procedure repeats at each time of each batch.

4.4 Simulation studies

Two case studies are considered for simulation of controlling batch processes, having GPC strategies with different objective functions and settings. In addition, ILC as a common control strategy for batch processes, introduced in previous chapters, is implemented for the sake of comparison. The optimal solutions of the ILC algorithm are computed based on the procedure mentioned in the previous chapter.

- **Case 1:** In this example, only the stochastic model is two-dimensional, and the plant model has dynamics only along time. The process is given as:

$$T(q^{-1}, z^{-1}) = \frac{0.5q^{-2}}{1 - 0.8q^{-1}} \tag{4.107}$$

$$N(q^{-1}, z^{-1}) = \frac{1 + 0.8q^{-1}}{1 - 0.7z^{-1}} \tag{4.108}$$

Figure 4.6 illustrates the set-point trajectory for the process.

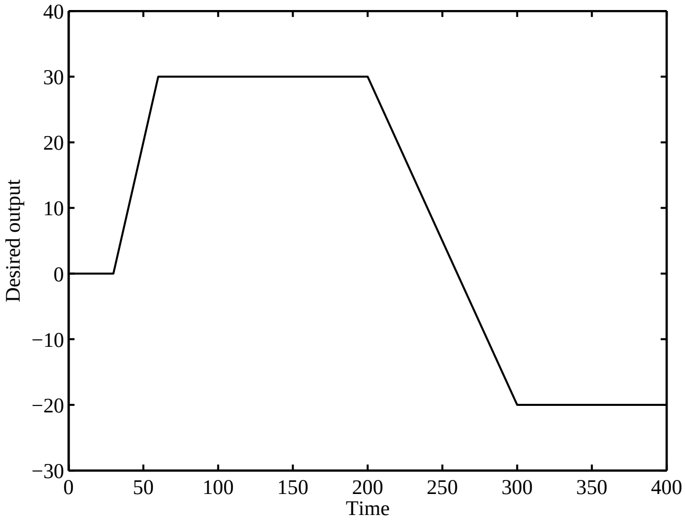


Figure 4.6: Desired output for simulation

A number of control strategies are implemented, as listed in Table 4.1. Note that objective functions GPC1 and GPC2 refer to the strategy penalizing over time and batch, respectively. Besides, (m_1, n_1) and (m_2, n_2) determine the control and prediction horizons.

Table 4.1: Control strategies structure for case 1

Name	Objective function	m_1	n_1	m_2	n_2	Q
gpc-time-1	GPC1	1	1	1	3	10R
gpc-time-2	GPC1	1	1	1	3	100R
gpc-time-3	GPC1	2	1	2	3	10R
gpc-batch-1	GPC2	1	1	1	3	10R
gpc-batch-2	GPC2	1	1	3	2	10R
ILC-1	ILC	-	-	-	-	-
ILC-2	ILC	-	-	-	-	-

For ILC strategies, the optimal solutions are computed and shown in Figure 4.7. Recalling from the previous chapter, the process is converged when the output error consists of only stochastic error. The rate of convergence is equal to the number of batches required before process achieves convergence. Table 4.2 shows the results of all the algorithms.

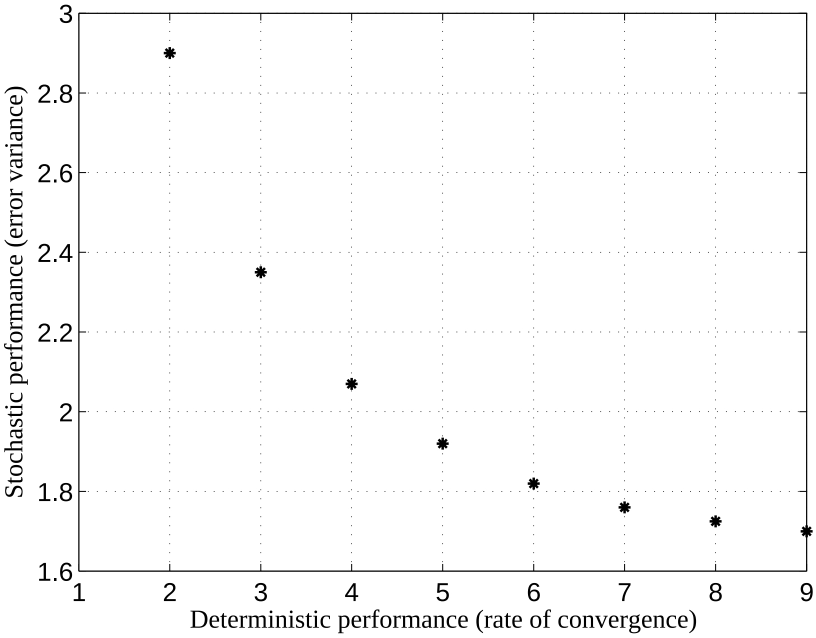


Figure 4.7: ILC optimal solutions for case 1

Table 4.2: Results of control strategies for case 1

Name	Convergence	Rate of convergence	Error variance
gpc-time-1	√	1	2.25
gpc-time-2	√	1	1.74
gpc-time-3	√	1	2.30
gpc-batch-1	√	1	2.28
gpc-batch-2	-	-	-
ILC-1	√	4	2.07
ILC-2	√	8	1.72

The first strategy, gpc-time-1, minimizes the cost function over a prediction horizon of 3 with respect to a control horizon of 1. Increasing the output error weight over the input increment weight, in the second design, results in less error variance, at the cost of more input changes. Enlarging the control and prediction horizons, on the other hand, does not affect the outputs significantly. Fourth strategy, gpc-batch-1, implements the GPC objective function which penalizes the input increment over batch. The results, however, are similar to previous strategies.

For the last GPC design, gpc-batch-2, the process cannot track the set-point and convergence will not be achieved. To find the problem, consider the structure illustrated in Figure 4.8. Solid and dashed line areas represent the control and prediction horizons. The dot line area includes the outputs that are dependent on the selected control horizon, obtained from the plant model. As shown in Figure 4.8, since the plant model has two samples of delay along time and no dynamics along batch, none of the outputs within the prediction horizon depends on the selected control horizon. Therefore, minimizing the prediction error is not considered in the objective function, and decisions are made based on minimizing the input increments only.

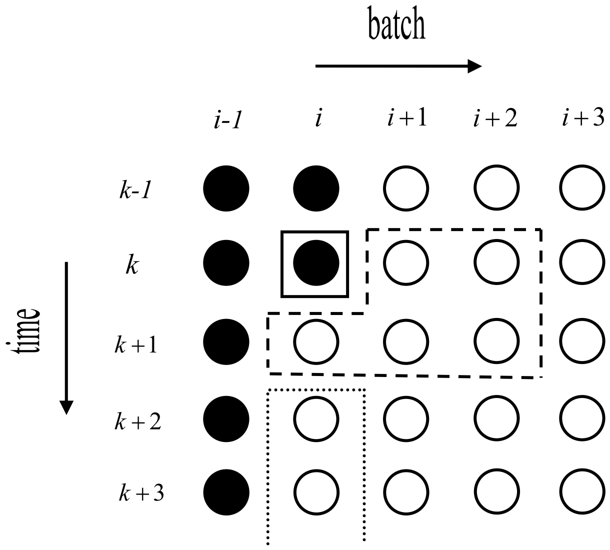


Figure 4.8: Selected horizons for the last GPC design for case 1

ILC strategies provide smaller output error variance for the process. However, the first few batches cannot track the set-point. For instance, ILC-1 provides the final output error variance of 2.07, which is less than gpc-time-1's. But at least 4 batches of the process are required to achieve convergence.

Figure 4.9 gives the results obtained from gpc-time-1 at the first batch and ILC-1 at 4^{th} batch. Error variance gained from ILC is relatively smaller. However it is only beneficial if we can afford to lose three batches. Otherwise, implementation of GPC is recommended.

- **Case 2:** We consider the two-dimensional dynamics of both the plant and disturbance models, along time and across batches. The following

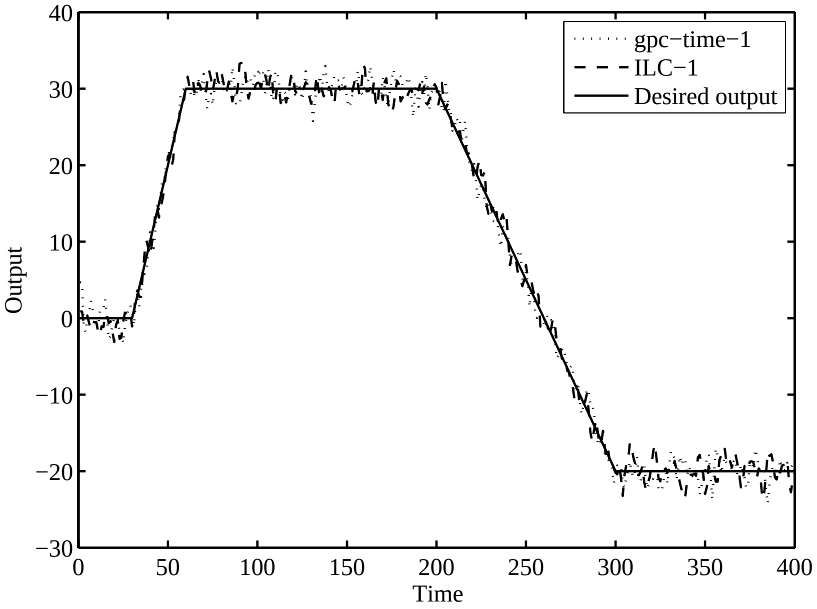


Figure 4.9: Simulation results for gpc-time-1 and ILC-1 strategies

is the process:

$$T(q^{-1}, z^{-1}) = \frac{0.9q^{-2} + 0.8z^{-1}}{1 - 0.7q^{-1}} \tag{4.109}$$

$$N(q^{-1}, z^{-1}) = \frac{1 + 0.8q^{-1}}{1 - 0.7z^{-1}} \tag{4.110}$$

The same set-point trajectory given in Figure 4.6 is desired. Table 4.3 lists the selected strategies.

Table 4.3: Control strategies structure for case 2

Name	Objective function	m_1	n_1	m_2	n_2	Q
gpc2-time-1	GPC1	1	1	1	3	10R
gpc2-time-2	GPC1	2	1	2	3	10R
gpc2-batch-1	GPC2	1	1	1	3	10R
gpc2-batch-2	GPC2	2	1	2	3	10R
ILC2-1	ILC	-	-	-	-	-
ILC2-2	ILC	-	-	-	-	-

Applying the ILC optimization algorithm, the optimal solutions are found and shown in Figure 4.10.

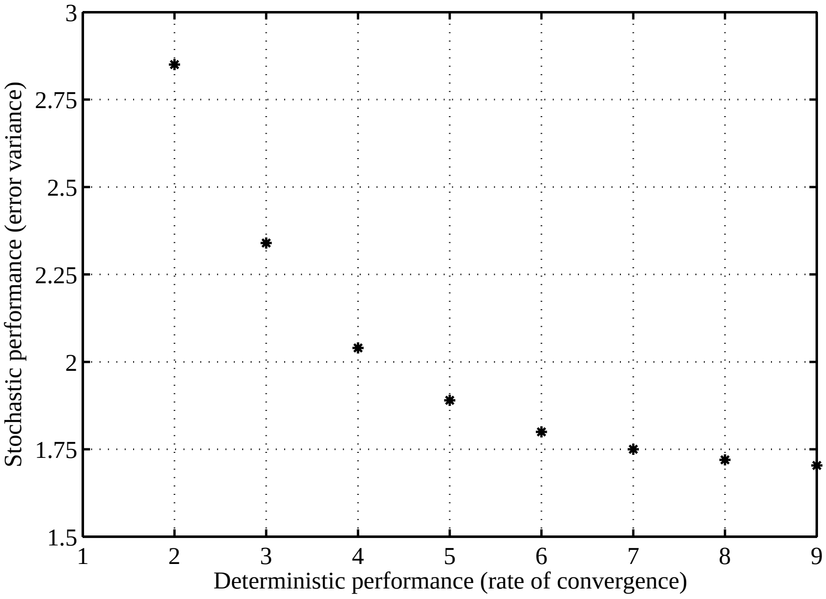


Figure 4.10: ILC optimal solutions for case 2

Implementing the GPC strategies, the results are obtained. Table 4.4 presents the results of the selected strategies.

Table 4.4: Results of control strategies for case 2

Name	Convergence	Rate of convergence	Error variance
gpc2-time-1	$\surd$	1	1.80
gpc2-time-2	-	-	-
gpc2-batch-1	$\surd$	1	1.81
gpc2-batch-2	$\surd$	2	2.15
ILC2-1	$\surd$	2	2.85
ILC2-2	$\surd$	7	1.75

As shown in Table 4.4, choosing a short prediction horizon of only three along time, in gpc2-time-1, can yield a relatively small error variance for all the batches. The second strategy, gpc2-time-2, does not even guarantee convergence. This strategy considers the minimization of the prediction error for two batches in a row. Because the input and output are strongly interacted across batches, the prediction error of the second batch has a large weight in the objective function. Hence the prediction error of the first batch cannot be minimized, and the process cannot track the set-point.

In the last GPC strategy, gpc2-batch-2, convergence is achieved after one batch operation. For the first batch, due to the initial values of the input and penalizing the input increments, set-point tracking is infeasible. But for the next batches the input is previously valued and convergence is reached. Figure 4.11 shows the outputs for the first two batches when gpc2-batch-2 is implemented.

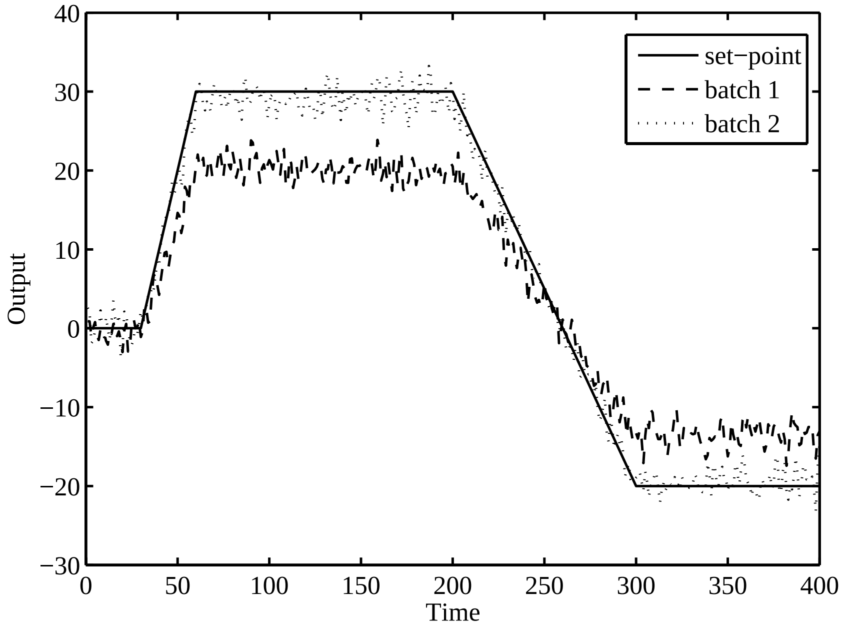


Figure 4.11: The first two batches outputs for gpc2-batch-2

Comparing the results of ILC strategies, ILC-1 converges after seven batches and has an error variance of 1.75. If we can afford to lose six batches, the outputs will have smaller error variance compared to GPC strategies. Otherwise, GPC is capable of converging for all the batches (with appropriate choices of control and prediction horizons) and yields a reasonable error variance.

4.5 Conclusions

Identification of two-dimensional batch processes was discussed in this work. Since the process operates along time and across batches, batch variables are correlated in two dimensions. This should be taken into account for identification purposes. This chapter proposed a new method for two-dimensional identification, followed by an extension of generalized predictive control for two-dimensional batch processes. The optimal two-dimensional predictor was derived and embedded inside the GPC structure. Then a new objective function was established and an algorithm to solve it was developed. All the algorithms were successfully verified through simulation studies.

Iterative learning control (ILC) strategy has been elaborated in the previous chapters. For the sake of comparison, all simulation studies were carried out by implementation of ILC as well as GPC. Based on the simulation results, ILC can be designed to perform better in terms of disturbance rejection. However, the first few batches yield undesirable results. If we can afford to lose a few batches, then ILC implementation will be beneficial. Otherwise, the use of GPC is recommended.

References

[1] S.AA. El Ata-Doss, P. Fiani, and J. Richalet. Handling input and state constraints in predictive functional control. pages 985-990. Proceedings of the IEEE Conference on Decision and Control, 1991.

[2] D.W. Clarke. Generalized predictive control: A robust self-tuning algorithm. pages 990-995. Proceedings of the American Control Conference, 1987.

[3] D.W. Clarke, C. Mohtadi, and P.S. Tuffs. Generalized predictive control: parts 1 and 2. Automatica, 23(2):137-160, 1987.

[4] C.R. Cutler and B. Ramaker. Dynamic matrix control - a computer control algorithm. San Francisco, 1980. Proceedings of the Joint Automatic Control Conference.

[5] J. Duan and M. Grimble. Design of long-range predictive control algorithms for industrial applications. IEE Colloquium on Industrial Applications of Model Based Predictive Control, Nov. 1991.

[6] M.P. Golden, L.K. Liu, and B.E. Ydstie. Multivariable control using extended prediction horizons. pages 1550-1551. Proceedings of the IEEE Conference on Decision and Control, 1986.

[7] K.S. Holkar and L.M. Waghmare. An overview of model predictive control. International Journal of Control and Automation, 3(4):47-64, 2010.

[8] J.C. Hung. Practical techniques for industrial control. IEEE, pages 5-10, 1996.

[9] R.M.C. De Keyser and A. Van Cauwenberghe. Towards robust adaptive control with extended predictive control. pages 1548-1549. Proceedings of the IEEE Conference on Decision and Control, 1986.

[10] L. Ljung. System Identification: Theory for the User, 2nd ed. Prentice Hall, 1999.

[11] A.I. Maalouf. Improving the robustness of a parallel robot using predictive functional control (pfc) tools. pages 6468-6473. Proceedings of the IEEE Conference on Decision and Control, 2006.

[12] S.J. Qin and T.A. Badgwell. A survey of industrial model predictive control technology. Control Engineering Practice, 11:733-764, 2003.

[13] J. Richalet, J. L. Estival, and Ph. Fiani. Industrial applications of predictive functional control to metallurgical industries. IEEE, pages 934-942, 1995.

[14] J. Richalet, A. Rault, J.L.Testud, and J. Papon. Model predictive heuristic control: Applications to industrial processes. Automatica, 14:413-428, 1978.

[15] P.D. Roberts. A Brief Overview of Model Predictive Control. The Institution of Electrical Engineers, London, UK, 1997.

[16] T. Soderstorm and P. Stoica. System Identification. Prentice Hall, 1989.

[17] B. Erik Ydstie and L.K. Liu. Single and multi-variable control with extended prediction horizons. pages 1303-1308. Proceedings of the ACC, 1984.

[18] B.E. Ydstie. Extended horizon adaptive control. Proceedings of the IFAC, 1984.

Chapter 5

Conclusions and future work

5.1 Concluding remarks

The objective of this thesis is to study two control strategies, iterative learning control (ILC) and generalized predictive control (GPC), for batch processes. The main contributions are classified as follows:

- Performance monitoring of ILC strategy for linear time-invariant models based on the minimum variance benchmark has been discussed in literature. This work proposes the extension of this algorithm for linear time varying models.
- Based on the minimum variance control law, there is a trade-off between the deterministic and stochastic performance of the ILC strategy, which can be illustrated in a trade-off curve. A method is proposed to compute the curve from routing operating data and assess the performance of the control strategy.
- Since batch processes operate along time and across batches, the dynamics of process variables in both dimensions should be considered for modeling purposes. This work proposes a method to estimate two-dimensional linear models.
- A two-dimensional linear optimal predictor is developed, followed by an extension of GPC for two-dimensional batch processes. Design and implementation challenges are also elaborated.

5.2 Future work

Here are some suggestions for future work:

- Deriving minimum variance benchmark for ILC strategy of LTV MIMO plants or non-minimum phase plants

- Performance assessment of ILC strategy for MIMO plants

- A more robust numerical method to calculate the trade-off curve for the minimum variance solutions

- Although the proposed identification procedure considers the dynamics of process variables in two dimensions, it still fails to cover nonlinearities of the process. Combining two-dimensional identification with multi-local linear modeling can be a potential research topic.