UKACC International Conference on Control 2012
Cardiff, UK, 3-5 September 2012

Batch-to-batch Iterative Learning Control Using Linearised Models with Adaptive Model Updating

J. Jewaratnam, J. Zhang and J. Morris
School of Chemical Engineering and Advanced Materials
Newcastle University
Newcastle Upon Tyne NE1 7RU, UK
jegalakshimi.lingeson@ncl.ac.uk; jie.zhang@ncl.ac.uk

A. Hussain
Department of Chemical Engineering
Engineering Faculty, University of Malaya
Kuala Lumpur, Malaysia

Abstract— **This paper presents batch-to-batch iterative learning control (ILC) of a fed-batch fermentation process using batch-wise linearised models identified from process operation data. The newly obtained process operation data after each batch is added to the historical data base and an updated linearised model is adaptively identified. In an effort to adapt to the current process environment, the updated model is identified from a moving window of the most recent historical batches. The new model is used to compute control policy for the next batch. The control policy at different batch stages are generally correlated as the overall control policy is obtained to maximize the amount of product at the end of a batch. To address the colinearity issue, partial least square (PLS) is used in estimating the linearised model parameters. The proposed strategy is applied to a simulated fed-batch fermentation process and the performance is evaluated. The effect of window sizes was studied. Simulation results show that the proposed approach improves the batch-to-batch ILC performance.**

Keywords-batch-to-batch control; iterative learning control; partial least square; fermentation process

I. INTRODUCTION

Fed-batch fermentation is vital in manufacturing high value added pharmaceutical and biochemical products. Fed-batch process is an evolution from batch reactor. The only difference between batch and fed-batch reactor is the feeding technique. In a batch reactor the substrate is only fed at the beginning of the process while in a fed-batch reactor it is fed over a few intervals with varying feed rates if necessary depending on the cell growth curve. The objective of using fed-batch in a fermentation process is to feed the substrate at the same rate that the organism utilizes it. That way, fed-batch fermentation increases input-yield ratio compared to the batch fermentation.

A core issue in fed-batch fermentation is the inability to sustain end-product quality specification due to the presence of model plant mismatches and unknown disturbances. Although every single batch repeats with exactly the same nominal initial parameters, somehow the end-product concentration varies due to unknown process variations. At present, the optimal control policy (feed rate) to obtain desired product quality in fed-batch fermentation is calculated off-line. The off-line calculated control policy may not be optimal when implemented to the real process due to model-plant mismatches and presence of unknown disturbances.

The repetitive nature of batch processes allows information from previous batches being used in modifying the control policy of the next batch in the framework of iterative learning control (ILC). ILC exploits every possibility to incorporate past control information into construction of present control action through memory based learning. The basic idea of ILC is to update the control trajectory for a new batch run using information from previous batch runs so that the output trajectory converges asymptotically to the desired reference trajectory [1],[2],[3]. The concept of ILC was first developed in the robotics industry to render a high precision in performing repetitive action of a given task [4], [5], [6]. In the recent years, ILC have been actively studied for application in injection moulding, batch reactor, chemical batch process, extrusion and batch distillation [7], [8], [9].

In a batch reactor, final product quality is usually controlled through controlling measurable variables such as pH, temperature and feed rate. The optimal trajectory of the measured variable is set and every batch run for the same process follows the same fixed trajectory [10]. This strategy fails when process disturbance in non-measured variables such as feedback condition, raw material properties, impurities and catalyst activities is present and affects the product quality [1]. This means a consistent input trajectory will not ensure product quality especially when unknown disturbances are present. This setback brings about the idea to update and re-optimize the input trajectory after every run to achieve desired product quality iteratively [10]. In a recent development, batch to batch ILC based on linearized perturbation model identified using multiple linear regressions (MLR) is reported [10]. In that work [10], the perturbation model is obtained using deviations of process input and output from their nominal trajectories and is updated after every batch by using the immediate previous batch as the nominal batch. This way, the unexpected process and parametric disturbances is expected to be captured and removed to render a more precise model prediction.

This paper presents an ILC strategy for a fed-batch fermentation process using linearised models identified from process operational data. The control policy updating is calculated using a model linearised around a reference batch. In order to cope with process variations and disturbances, the

978-1-4673-1558-6/12/$31.00 ©2012 IEEE

reference batch can be taken as the immediate previous batch. In such a way, the model is a batch wise linearised model and is updated after each batch. The newly obtained process operation data after each batch is added to the historical data base and an updated linearised model is re-identified. In order to overcome the colinearity among the predictor variables, this paper proposes that the linearised model can be identified using partial least square regression (PLS) [11].

In order for the updated model to capture the process behavior in the face of process variations, a new technique using a moving window of the historical batches to update batch-wise linearised models is developed in this paper. The historical batches were updated after every batch run but using only the *M* recent number of batches. In other words, after every run the "oldest" batch is forgotten and the new batch is included into the sliding "window" of historical batches.

The proposed strategy is applied to a simulated fed-batch fermentation process. The results show that enhanced control performance is obtained under the proposed approach. Model updating using PLS leads to better control performance than model updating using MLR. Different window sizes were studied and the performances were evaluated.

The paper is organized as follows. Section II presents batch-to-batch ILC with updated linearised model. Application to a simulated fed-batch fermentation process is presented in Section III. Section IV concludes this paper.

II. Batch-to-batch Iterative Learning Control with Updated Linearised Model

A. Linearised Models for Batch Processes

Consider batch processes where the batch run length (t_f) is fixed and consists of N control intervals. For simplicity in implementation, the manipulated variable, $u \in R^m$ (m=1 in this work), is kept constant within each control interval and, thus, the control policy for a batch is a vector with N elements. Product quality variables (outputs), $y \in R^n$ ($n \geq 1$), can be obtained off-line by analysing the samples taken during the batch run. The product quality and control trajectories are defined, respectively, as

$$\mathbf{Y}_k=[y_k^{\mathrm{T}}(1), y_k^{\mathrm{T}}(2),\ldots, y_k^{\mathrm{T}}(N)]^{\mathrm{T}} \qquad (1)$$

$$\mathbf{U}_k=[u_k(0), u_k(1),\ldots,u_k(N\text{-}1)]^{\mathrm{T}} \qquad (2)$$

where the subscript k denotes the batch index. The desired reference trajectories of product quality are defined as

$$\mathbf{Y}_d=[y_d^T(1), y_d^T(2),\ldots, y_d^T(N)]^T \qquad (3)$$

A batch process is typically modelled with a dynamic model, but it would be convenient to consider a static function relating the control sequence to the product quality sequences over the whole batch duration [10].

$$\mathbf{Y}_k = \mathbf{F}(\mathbf{U}_k) + \mathbf{v}_k \qquad (4)$$

where $\mathbf{F}(\cdot)$ represents the non-linear static functions between $\mathbf{U}_k(t)$ and $y_k(t)$ at different sampling times and $\mathbf{v}_k=[v_k^T(0),v_k^T(1),\ldots, v_k^T(N\text{-}1)]^T$ is a vector of measurement noises. Linearising the non-linear batch process model described by Eq(4) with respect to $\mathbf{U}_s$ around the nominal trajectories ($\mathbf{U}_s$, $\mathbf{Y}_s$), the following can be obtained.

$$\mathbf{Y}_k = \mathbf{Y}_s + \left.\frac{\partial \mathbf{F}(\mathbf{U}_k)}{\partial \mathbf{U}_k}\right|_{\mathbf{U}_s} (\mathbf{U}_k - \mathbf{U}_s) + \mathbf{w}_k + \mathbf{v}_k \qquad (5)$$

where $\mathbf{w}_k=[w_k^T(1), w_k^T(2), \ldots, w_k^T(N)]^T$ is a sequence of model errors due to the linearization (i.e., due to neglecting the higher order terms) and $\mathbf{v}_k$ represents the effects of noise and unmeasured disturbances. Define the linearised model $\mathbf{G}_s$ as

$$\mathbf{G}_s = \left.\frac{\partial \mathbf{F}(\mathbf{U}_k)}{\partial \mathbf{U}_k}\right|_{\mathbf{U}_s} \qquad (6)$$

The structure of $\mathbf{G}_s$ is restricted to the following lower-block-triangular form due to the causality.

$$\mathbf{G}_s = \begin{bmatrix} g_{10} & 0 & \cdots & 0 \\ g_{20} & g_{21} & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ g_{N0} & g_{N1} & \cdots & g_{NN-1} \end{bmatrix} \qquad (7)$$

The linearised model can be identified from historical process operation data using MLR [10]. Let $\mathbf{X}$ and $\mathbf{Y}$ be the deviations from the reference trajectories of historical data in the manipulated variables and product quality variables respectively, then $\mathbf{Y} = \mathbf{G}_s\mathbf{X}$ and the linearised model $\mathbf{G}_s$ can be obtained through MLR as

$$\mathbf{G}_s = (\mathbf{X}^{\mathrm{T}}\mathbf{X})^{-1}\mathbf{X}^{\mathrm{T}}\mathbf{Y} \qquad (8)$$

To cope with process drift, the linearised model can be re-identified after each batch run with data from the most recent batch added to the historical process data. Furthermore, the control trajectory and quality variable trajectory from the most recent batch can be used as the reference trajectories.

B. Partial Least Squares

PLS projects the $\mathbf{X}$ and $\mathbf{Y}$ matrices to a subset of latent variables, $\mathbf{t}$ and $\mathbf{u}$, respectively.

$$\mathbf{X} = \sum_{j=1}^{k} \mathbf{t}_j \mathbf{p}_j^T + \mathbf{E} \qquad (9)$$

$$\mathbf{Y} = \sum_{j=1}^{k} \mathbf{u}_j \mathbf{q}_j^T + \mathbf{F} \tag{10}$$

In the above equations, **E** and **F** are residual matrices of unfitted variations in the **X** and **Y** data respectively. If sufficient numbers of latent variables (sufficiently large *k*) are used, then both **E** and **F** can be made zero or close to zero if the modeled relationship is linear. The objective is to fit a linear relationship between **X** and **Y** by performing least square regression between each pair of corresponding **t** and **u** latent vectors while making $\|\boldsymbol{F}\|$ as small as possible.

$$\hat{\mathbf{u}}_j = \mathbf{t}_j \mathbf{b}_j \qquad j=1,2,\ldots k \tag{11}$$

where $\mathbf{b}_j$ is the coefficient from the inner linear regression between the jth latent variables $\mathbf{u}_j$ and $\mathbf{t}_j$ which is

$$\mathbf{b}_j = (\mathbf{t}_j^T \mathbf{t}_j)^{-1} \mathbf{t}_j^T \mathbf{u}_j \tag{12}$$

PLS provides a bilinear decomposition of the **X** and **Y** matrices into a number of rank-one matrices. The decomposition can be defined as the product between each pair of input scores vector, **t**, and predicted output scores vector, $\hat{\mathbf{u}}_j$, and a set of corresponding input and output loading vectors **p** and **q**.

The number of latent variables, *k*, to be retained in the model for PLS is usually determined through cross-validation [12]. The data set for building a model is partitioned into a training data set and a testing data set. PLS models with different number of principal components are developed on the training data and then tested on the testing data. The model with the smallest testing errors is then selected.

C. *Model Updating Using a Sliding Window of Historical Batches*

Let *M* be the size of a sliding window of the past batches and use the immediate previous batch, the (k-1)th batch, as the nominal batch, then the deviations of the process input and output trajectories from their nominal trajectories in the sliding window can be represented as:

$$\Delta\mathbf{X}_{k=1} = \begin{bmatrix} \mathbf{U}_{k-M} - \mathbf{U}_{k=1} \\ \mathbf{U}_{k-M+1} - \mathbf{U}_{k=1} \\ \cdots \\ \mathbf{U}_{k-1} - \mathbf{U}_{k=1} \end{bmatrix} \tag{13}$$

$$\Delta\mathbf{Y}_{k=1} = \begin{bmatrix} \mathbf{Y}_{k-M} - \mathbf{Y}_{k=1} \\ \mathbf{Y}_{k-M+1} - \mathbf{Y}_{k=1} \\ \cdots \\ \mathbf{Y}_{k-1} - \mathbf{Y}_{k=1} \end{bmatrix} \tag{14}$$

The updated model parameters can be obtained using MLR or PLS. If correlations exist among the control actions at different stages of a batch, then PLS will give robust and reliable estimation of the model parameters.

D. *Iterative Learning Control*

The batch to batch iterative learning control strategy was developed in [10] and is briefly introduced here.

The prediction of perturbation model is defined as

$$\hat{\overline{\mathbf{Y}}}_k = \hat{\mathbf{G}}_s \overline{\mathbf{U}}_k \tag{15}$$

and the absolute model prediction is defined as

$$\hat{\mathbf{Y}}_k = \mathbf{Y}_s + \hat{\overline{\mathbf{Y}}}_k = \mathbf{Y}_s + \hat{\mathbf{G}}_s \overline{\mathbf{U}}_k \tag{16}$$

After completion of the k^{th} batch run, prediction errors between off-line measured or analysed product qualities and their model predictions can be calculated as

$$\varepsilon_k = \mathbf{Y}_k - \hat{\mathbf{Y}}_k = \overline{\mathbf{Y}}_k - \hat{\overline{\mathbf{Y}}}_k \tag{17}$$

Based on the prediction errors of the k^{th} batch run, the modified prediction of perturbation model in the $(k+1)^{th}$ batch run is obtained as

$$\tilde{\overline{\mathbf{Y}}}_{k+1} = \hat{\overline{\mathbf{Y}}}_{k+1} + \varepsilon_k \tag{18}$$

The absolute modified model prediction is defined as

$$\tilde{\mathbf{Y}}_{k+1} = \hat{\mathbf{Y}}_{k+1} + \varepsilon_k = \mathbf{Y}_s + \hat{\overline{\mathbf{Y}}}_{k+1} + \varepsilon_k \tag{19}$$

The modified prediction error is defined as

$$\tilde{\varepsilon}_{k+1} = \mathbf{Y}_{k+1} - \tilde{\mathbf{Y}}_{k+1} = \overline{\mathbf{Y}}_{k+1} - \tilde{\overline{\mathbf{Y}}}_{k+1} \tag{20}$$

From the definitions in Eq(17) and Eq(18), we have

$$\tilde{\varepsilon}_{k+1} = \varepsilon_{k+1} - \varepsilon_k \tag{21}$$

We assume that the prediction error of the perturbation model is bounded by a certain small positive constant B_m such that

$$|\varepsilon_k| < B_m \tag{22}$$

The prediction error bound B_m is a measure to represent the deviation of $\hat{\overline{\mathbf{Y}}}_k$ from $\overline{\mathbf{Y}}_k$ or $\hat{\mathbf{Y}}_k$ from $\mathbf{Y}_k$. The higher the value of B_m is, the poorer the identified model is. The modified prediction error is bounded by $2B_m$ as follows

$$|\tilde{\varepsilon}_k| < |\varepsilon_k| + |\varepsilon_{k-1}| < 2B_m \tag{23}$$

The tracking errors of process and perturbation model are respectively defined as

$$\mathbf{e}_k = \mathbf{Y}_d - \mathbf{Y}_k = \overline{\mathbf{Y}}_d - \overline{\mathbf{Y}}_k \tag{24}$$

$$\hat{\mathbf{e}}_k = \mathbf{Y}_d - \hat{\mathbf{Y}}_k = \overline{\mathbf{Y}}_d - \hat{\overline{\mathbf{Y}}}_k \tag{25}$$

where $\overline{\mathbf{Y}}_d$ is the deviated desired trajectory and defined as

$$\overline{\mathbf{Y}}_d = \mathbf{Y}_d \text{-} \mathbf{Y}_s \tag{26}$$

The tracking errors of modified prediction of perturbation model is defined as

$$\tilde{\mathbf{e}}_k = \mathbf{Y}_d - \tilde{\mathbf{Y}}_k = \overline{\mathbf{Y}}_d - \tilde{\overline{\mathbf{Y}}}_k \tag{27}$$

From the definitions in Eq(17), Eq(24) and Eq(27), the following relationship among these three tracking errors can be obtained

$$\varepsilon_k = \hat{\mathbf{e}}_k - \mathbf{e}_k \tag{28}$$

$$\tilde{\mathbf{e}}_k = \hat{\mathbf{e}}_k - \boldsymbol{\varepsilon}_{k-1} \tag{29}$$

From Eq(25) and Eq(15), an iterative relationship for $\hat{\mathbf{e}}_k$ along the batch index k can be obtained as

$$\hat{\mathbf{e}}_{k+1} = \hat{\mathbf{e}}_k - \hat{\mathbf{G}}_s \Delta\overline{\mathbf{U}}_{k+1} \tag{30}$$

where $\Delta\overline{\mathbf{U}}_{k+1}$ is defined as

$$\Delta\overline{\mathbf{U}}_{k+1} = \overline{\mathbf{U}}_{k+1} - \overline{\mathbf{U}}_k \tag{31}$$

From the definition of perturbation variables, we can have

$$\Delta\overline{\mathbf{U}}_{k+1} = \overline{\mathbf{U}}_{k+1} - \overline{\mathbf{U}}_k = \mathbf{U}_{k+1} - \mathbf{U}_k \tag{32}$$

Substitute Eq(28) and Eq(30) to Eq(29), we have

$$\tilde{\mathbf{e}}_{k+1} = \hat{\mathbf{e}}_{k+1} - (\hat{\mathbf{e}}_k - \mathbf{e}_k) = \mathbf{e}_k - \hat{\mathbf{G}}_s \Delta\overline{\mathbf{U}}_{k+1} \tag{33}$$

On the other hand, Eq(28) can be rewritten as

$$\mathbf{e}_k = \hat{\mathbf{e}}_k - \varepsilon_k \tag{34}$$

From Eq(34) and Eq(30), an iterative relationship for $\mathbf{e}_k$ along the batch index k can also be obtained as

$$\mathbf{e}_{k+1} = \mathbf{e}_k - \hat{\mathbf{G}}_s \Delta\overline{\mathbf{U}}_{k+1} - \tilde{\varepsilon}_{k+1} \tag{35}$$

Given the error transition model in the form of Eq(33) and Eq(35), the objective of ILC is to design a learning algorithm to manipulate the control policy so that the product qualities follow the specific desired reference trajectories. The following quadratic objective function based on the modified prediction errors upon the completion of the k^{th} batch run is minimised to update the input trajectory for the $(k+1)^{\text{th}}$ batch run

$$J_{k+1} = \min_{\Delta\overline{\mathbf{U}}_{k+1}} \frac{1}{2}[\tilde{\mathbf{e}}_{k+1}^T \mathbf{Q} \tilde{\mathbf{e}}_{k+1} + \Delta\overline{\mathbf{U}}_{k+1}^T \mathbf{R} \Delta\overline{\mathbf{U}}_{k+1}] \tag{36}$$

where **Q** and **R** are positive definitive matrices. Note that the objective function, Eq(36), has a penalty term on the input change $\Delta\overline{\mathbf{U}}_{k+1}$ between two adjacent batch runs, the algorithm has an integral action with respect to the batch index k [10]. The weighting matrices **Q** and **R** should be selected carefully. A larger weight on the input change will lead to more conservative adjustments and slower convergence. There are also other variants of the objective function. For example, the weighting matrices **Q** and **R** may be set as **Q** = diag$\{Q(1), Q(2), \ldots, Q(N)\}$, **R** = diag$\{R(0), R(1), \ldots, R(N\text{-}1)\}$, where $Q(i)$ and $R(j)$ increase with respect to the time intervals t in proportion to its effect of the final product quality. For the sake of simplicity, **Q** and **R** are selected in this study as $\mathbf{Q}=\lambda_q \cdot \mathbf{I}_N$ and $\mathbf{R}=\lambda_r \cdot \mathbf{I}_N$.

By finding the partial derivative of the quadratic objective function Eq(36) with respect to the input change $\Delta\overline{\mathbf{U}}_{k+1}$ and through straightforward manipulation, the following ILC law can be obtained

$$\Delta\overline{\mathbf{U}}_{k+1} = \hat{\mathbf{K}} \mathbf{e}_k \tag{37}$$

where $\hat{\mathbf{K}}$ is defined as the learning rate

$$\hat{\mathbf{K}} = [\hat{\mathbf{G}}_s^T \mathbf{Q} \hat{\mathbf{G}}_s + \mathbf{R}]^{-1} \hat{\mathbf{G}}_s^T \mathbf{Q} \tag{38}$$

From Eq(32) and Eq(37), the ILC law can be written as

$$\mathbf{U}_{k+1} = \mathbf{U}_k + \hat{\mathbf{K}} \mathbf{e}_k \tag{39}$$

III. Application to a Fed-batch Fermentation Process

A. A fed-batch Fermentation Process

The process considered in this paper is a fed-batch yeast fermentation process taken from [14], where a detailed kinetic and dynamic model is presented. The kinetic model of yeast metabolism is based on the bottleneck hypothesis by [15] and a dynamic model is developed based on mass balance equations for glucose, ethanol, oxygen and biomass concentrations. In this study, a simulation programme is developed in MATLAB using the kinetic and dynamic model and is verified with the results presented in [14]. The operation objective is to produce maximum amount of biomass by adjusting the glucose feed rate subject to operation constraints.

In this case study, each batch had a finite run time of 16.5hrs. The batch duration was divided into 10 equal stages and the feed rate remains constant within each stage. An initial feed rate profile was obtained from [14]. Then, 20 historical batches were generated by adding random variations to the initial feed rate profile. The end-batch biomass concentration of the historical batches ranged between 45-60g/L. Then, MLR and PLS regression methods were used in estimating the linearised model parameters from these historical process data. Batch –to-batch iterative learning control with updated historical batches was applied to the simulation. The **Q** and **R** values were fixed at 1**I** and 0.0001**I** respectively. The **R** and **Q** weighting were decided using a trial and error method.

B. Results for Batch to Batch ILC with Updated Models

A batch-to-batch control study using linearised models from updated historical batches was conducted. The number of historical bathes used to develop a current batch process model keeps building up after every batch run. In other words, after every batch trial, the data is added into the pile of historical batches. Then all previous batches are used to identify a new process model which is used to generate a new control policy for the current batch. The cycle repeats and the process model is developed using both old and new batch data.

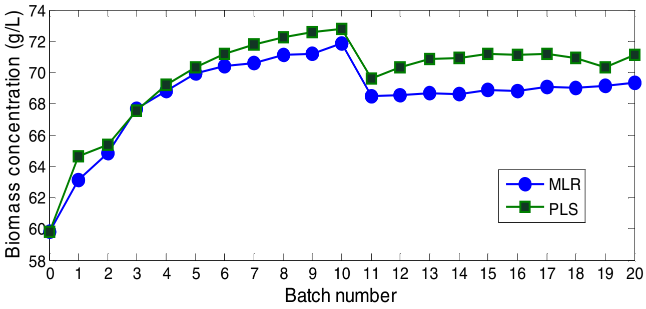

Figure 1. End of batch biomass concentration under ILC with batch-wise updated models

Figure 1 shows the performance of batch to batch ILC using updated MLR and PLS models identified from growing number of historical batches. The desired final biomass concentration was set at 74g/L. Batch 0 represents the last historical batch before implementing ILC. It is used as reference point to show process improvements due to the implementation of ILC. Batches 1 to 10 were used to test the ability of tracking desired trajectory without the presence of disturbances. From batch 11, a disturbance was introduced in that the initial substrate concentration was changed to 305g/l from its nominal value of 325g/l. Note that the initial substrate concentration is not measured and, hence, this is an unmeasured disturbance. Overall, MLR and PLS model revealed improving process operation with slight instability for both with and without disturbances. Comparing MLR and PLS models, it is evident that ILC based on the PLS model delivered higher biomass concentration for almost all the batches. This is due to the ability of PLS model to alleviate co-linearity issue. Although PLS model resulted in higher biomass concentration in the presence of disturbance, the convergence rate and stability was quite unsatisfactory. MLR model exhibited steadier performance from batch 11 to batch 20 because larger number of historical batches is favourable to obtain a better MLR model, thus better performance. The PLS model based ILC gives higher biomass but it does not always improve from batch to batch.

C. Results for Batch to Batch ILC with Updated Models and Moving window Historical Batches

Further improvement was done for PLS model using a sliding window of historical batches to develop process models. After each batch run, the new batch data is added into the window of historical batches. The oldest batch in the window is removed. The idea is to use latest information to update the model and calculate the control policy for the current batch. Three sliding window sizes of 10, 15 and 20 historical batches were studied.

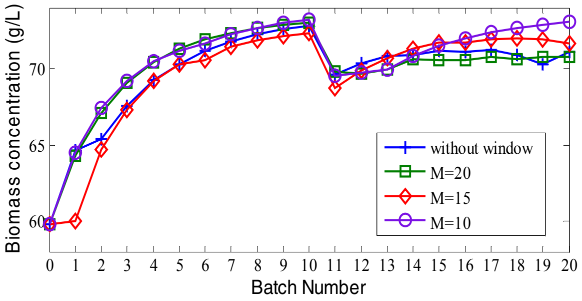

Figure 2. End of batch biomass concentration under ILC with batch-wise updated models using a sliding window of historical batches

Figure 2 shows that all three windows sizes exhibit improving results with varying stability before and after the disturbance was introduced. Performances of different window sizes (N) were compared with the one without using sliding window. For window size of 20, the convergence rate and stability were better and satisfactory when there is no disturbance. In the presence of disturbance, the results fluctuated and were not any better compared to ILC without using a sliding window. As for window size of 15, the biomass concentrations were lesser for most of the batches with no disturbances but in the presence of disturbance, the convergence rate were improving steadily from batch 11 to batch 15. From batch 16 to batch 20 slight fluctuations were noticed though the biomass concentrations were higher than the plot with no window. For window size of 10, the performance with no disturbance is as good as window size of 20. For batch 12 and 13 the biomass yield was lesser than that under ILC without using sliding window, but in the following 7 batches the performance improved steadily. The convergence rate was very satisfactory. The PLS model was able to attain final output almost as good as without disturbance within 10 batches. Amongst the three window sizes, window size of 10 gave the most stable and fastest converging performance. It is shown in the results that PLS method does not need a growing number of historical batches to develop a reliable model. An updated historical batch data with window size equal to the number of control policies used in the fed-batch fermentation process is able to generate optimal process model by using the PLS method.

IV. Conclusions

An ILC technique with model adaptation using a sliding window of historical batches is developed in this paper. PLS is used to estimate model parameter in order to address the colinearity issues. The proposed method is applied to a simulated fed-batch fermentation process. Application results show that ILC based on batch-wise updated model using a sliding window of recent historical batches improves the control performance with and without disturbance. The effect of window sizes is studied. It is shown that model updating using PLS does not need large window size in providing enhanced control performance.

Acknowledgment

The work is supported by University Malaya and Ministry of Higher Education, Malaysia, through a scholarship for the first author and the EU through the project iREMO (grant No. NMP2-SL-2009-228662).

References

[1] K. S. Lee, I. S. Chin, H. J. Lee and J. H. Lee, "Model predictive control technique combined with iterative learning for batch processes", AIChE Journal, vol.45 (10) October, 1999.

[2] W .J. Campbell, S .K. Firth, A. J. Toprac and T .F. Edgar, "A comparison of run-to-run control algorithms", Proceedings of the American Control Conference, May 8-10, 2002.

[3] Z. Xiong and J. Zhang, "A batch-to-batch iterative optimal control strategy based on recurrent neural networks models", Journal of Process Control, vol.15, pp. 11-21, 2005.

[4] M. Garden, "Learning control of actuators in control systems", US Patent 3555252, 1971.

[5] M. Uchiyama, "Formation of high-speed motion pattern of a mechanical arm by trial", Transactions of Society for Implementation and Control Engineers, vol.14, pp.706-712, 1978.

[6] S. Arimoto, S. Kawamura and F. Miyazaki, "Bettering operation of robots by learning", Journal of Robotic Systems, vol.1, pp.123-140, 1984.

[7] F. Gao, Y. Yang and C. Shao, "Robust iterative learning control with application to injection molding process", Chemical Engineering Science, vol.56, pp. 7025-7034, 2001.

[8] D. A. Bristow, M. Tharayil and A. G. Alleyne. "A survey of iterative learning control: a learning-based method for high-performance tracking control", IEEE Control Systems Magazine, pp. 96-114, June 2006.

[9] K. S. Lee and J. H. Lee, "Iterative learning control-based batch process control technique for integrated control of end product properties and transient profiles of process variables", Journal of Process Control, vol.13: pp.607-621, 2003.

[10] Z. Xiong and J. Zhang, "Product quality trajectory tracking in batch processes using iterative learning control based on time-varying perturbation models", Industrial Engineering Chemical Research, vol. 42, pp. 6802-6814, 2003.

[11] P. Geladi, and B. R. Kowalski, "Partial least squares regression: a tutorial", Analytica Chimica Acta, vol. 185, pp.1-17, 1986.

[12] Wold, S., "Cross validatory estimation of the number of components in factor and principal components models", Technometrics, vol. 20, pp. 397-404, 1978.

[13] K. S. Lee and J. H. Lee, "Model predictive control for nonlinear batch processes with asymptotically perfect tracking", Computers and Chemical Engineering, vol. 21, pp. s873-s879, 1997.

[14] U. Yuzgec, M. Turker and A. Hocalar, "On-line evolutionary optimization of an industrial fed-batch yeast fermentation process", ISA Transactions, vol. 48, pp.79-92, 2009.

[15] B. Sonnleitnert and O. Kappeli, "Growth of saccharomyces cerevisiae is controlled by its limited respiratory of capacity: formulation and verification of a hypothesis", Biotechnology and Bioengineering, vol. 28, pp. 927-937, 1986.