Proceedings of the 11th International Symposium on
Computer Applications in Biotechnology (CAB 2010),
Leuven, Belgium, July 7-9, 2010
Julio R. Banga, Philippe Bogaerts, Jan Van Impe,
Denis Dochain, Ilse Smets (Eds.)

Batch-to-Batch Iterative Learning Control of a Fed-batch Fermentation Process Using Incrementally Updated Models

J. Jewaratnam [1], J. Zhang[1], A. Hussain[2] and J. Morris[1]

[1]*School of Chemical Engineering and Advanced Material, Newcastle University, Newcastle Upon Tyne, NE1 7RU, UK*

[2]*Department of Chemical Engineering, University of Malaya, Kuala Lumpur, Malaysia*
(e-mail: Jegalakshimi.Lingeson@ncl.ac.uk; jie.zhang@ncl.ac.uk)

Abstract: Batch-to-batch iterative learning control of a fed-batch fermentation process using batchwise linearised models identified from process operation data is presented in this paper. Due to model-plant mismatches and the present of unknown disturbances, off-line calculated control policy may not be optimal when implemented to the real process. The repetitive nature of batch process allows information from the previous batches being used in modifying the control policy of the next batch in the framework of iterative learning control. In order to cope with nonlinear behaviour of batch fermentation processes, the model is linearised using the immediate previous batch as a reference batch and the model is updated from batch to batch. The control policy (feed rates) 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 of the control variable, principal component regression and partial least squares regression are used in estimating the linearised model parameters. Application results on a simulated industrial scale fed-batch fermentation process demonstrate that the proposed strategy is effective.

Keywords: Batch to batch Control, Fermentation Processes, Iterative Learning Control, Principal Component Regression, Partial Least Squares Regression.

1. INTRODUCTION

Fermentation is an important process in pharmaceutical and biochemical industry. Process optimisation is important in improving product quality and production efficiency. In fed-batch fermentation process, the glucose feed rate is usually obtained by solving an optimisation problem based on a process model such as a mechanistic model (Park and Ramirez, 1988) or an empirical model (Tian et al., 2002). The off-line calculated control policies (recipes) for batch fermentation processes may not be optimal when implemented on the processes due to model plant mismatches and/or the presence of unknown disturbances (Zhang, 2004). Utilising the repetitive nature of batch processes, fermentation operation recipes can be modified from batch to batch in order to overcome the detrimental effect of model-plant mismatches and unknown disturbances.

Recently, iterative learning control (ILC) has been used in the run-to-run control of batch processes to directly update input trajectory (Gao et al., 2001; Chin et al., 2000; Lee et al., 2000; Alvarez et al., 2009). The basic idea of ILC is to update the control trajectory for a new batch run using the information from previous batch runs so that the output trajectory converges asymptotically to the desired reference trajectory. Refinement of control signals based on ILC can significantly enhance the performance of tracking control systems. Zhang (2005) proposes a neural network based batch to batch control strategy where a linearised model is obtained from the neural network model. Xiong and Zhang (2005) present a recurrent neural network based ILC scheme for batch processes where filtered recurrent neural network prediction errors from previous batches are added to the model predictions for the current batch and optimisation is performed based on the updated predictions. Xiong and Zhang (2003) propose a batch to batch ILC strategy using linearised time variant perturbation models that are identified from process operational data. In (Xiong and Zhang, 2003), the linearised models are identified using multiple linear regression (MLR). In many batch processes, the control actions at different stages of a batch can be correlated since they are such determined to optimize the final product quality at the end of the batch. In such situations, models obtained using MLR may not be appropriate.

This paper presents an iterative learning control strategy for a 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 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 from principal component regression

(PCR) and partial least square regression (PLS) (Geladi and Kowalski, 1986).

The paper is organised as follows. Section 2 presents a batch to batch control strategy based on linearised model. Section 3 presents application results on a simulated fed-batch fermentation process. Some concluding remarks are given in Section 4.

2. ITERATIVE LEARNING CONTROL WITH BATCHWISE UPDATED MODELS

2.1 Linearised models for batch processes

Consider batch processes where the batch run length (t_f) is fixed and consists of N sampling intervals (i.e. $N=t_f/h$, with h being the sampling time). 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 and the manipulated variable, $u\in R^m$ ($m=1$ in this work), can be measured at each sampling time on-line. The product quality and control trajectories are defined, respectively, as

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

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

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

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

A batch operation 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 (Lee et al., 1999). Due to the causality, the product quality variables at time t, $y_k(t)$, is a non-linear function of all control actions up to time t, $\mathbf{U}_k(t)=[u_k(0), u_k(1),\ldots,u_k(t\text{-}1)]^T$, i.e.

$$y_k(t)=f_t(\mathbf{U}_k(t)) + v_k(t), \quad t=1, 2, ..., N, \quad y_k(0)=y_0 \quad (4)$$

where $f_t(\cdot)$ represents the non-linear function between $\mathbf{U}_k(t)$ and $y_k(t)$ and $v_k(t)$ is the measurement noise at time t. Eq(4) can be rewritten in matrix form as

$$\mathbf{Y}_k = \mathbf{F}(\mathbf{U}_k) + \mathbf{v}_k \quad (5)$$

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.

$$\mathbf{Y}_k = \mathbf{Y}_s + \left.\frac{\partial \mathbf{F}(\mathbf{U}_k)}{\partial \mathbf{U}_k}\right|_{\mathbf{U}_s} (\mathbf{U}_k - \mathbf{U}_s) + \mathbf{w}_k + \mathbf{v}_k \quad (6)$$

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 linearisation (i.e., due to negalecting the higher oder terms) and $\mathbf{v}_k$ represents the effects of noise and unmeasured disturbances. Define the linearised model $\mathbf{G}_s$ as

$$\mathbf{G}_s = \left.\frac{\partial \mathbf{F}(\mathbf{U}_k)}{\partial \mathbf{U}_k}\right|_{\mathbf{U}_s} \quad (7)$$

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

$$\mathbf{G}_s = \begin{bmatrix} g_{10} & 0 & \cdots & 0 \\ g_{20} & g_{21} & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ g_{N0} & g_{N1} & \cdots & g_{NN-1} \end{bmatrix} \quad (8)$$

The linearised model can be identified from historical process operation data using MLR (Xiong and Zhang, 2003). To cope with process drift, the linearised model can be re-identified after each batch 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.

In many batch processes, the control policies are typically determined to optimise the product quality at the end of a batch. Therefore, the control actions during different stages of a batch are usually correlated. In such cases, appropriate linearised model may not be obtained from MLR. To overcome the colinearity in the regression variables, PCR or PLS (Geladi and Kowalski, 1986) can be used to obtain the linearised models.

A brief introduction of PCR is given here. Details about PCR can be found in (Geladi and Kowalski, 1986). Consider the following linear model

$$y = x_1\theta_1 + x_2\theta_2 + \ldots + x_n\theta_n \quad (9)$$

where y is the model output, x_1 to x_n are model inputs, and θ_1 to θ_n are model parameters.

Given a set of input and output data, $\mathbf{X}$ and $\mathbf{Y}$, the model parameters can be obtained from MLR as

$$\hat{\boldsymbol{\theta}} = (\mathbf{X}^T\mathbf{X})^{-1}\mathbf{X}^T\mathbf{Y} \quad (10)$$

where $\hat{\boldsymbol{\theta}} = (\hat{\theta}_1\ \hat{\theta}_2 \cdots \hat{\theta}_n)^T$ is a vector of the estimates of model parameters.

When the model input variables are correlated, Eq(10) gives unreliable estimates since $(\mathbf{X}^T\mathbf{X})$ is close to singular. In this case the model parameters can be estimated using PCR.

The matrix $\mathbf{X}$ can be decomposed as the sum of a series of rank one matrices through principal component analysis.

$$\mathbf{X} = \mathbf{t}_1\mathbf{p}_1^T + \mathbf{t}_2\mathbf{p}_2^T + ... + \mathbf{t}_N\mathbf{p}_N^T \quad (11)$$

In the above equation, $\mathbf{t}_i$ and $\mathbf{p}_i$ are the ith score vector and loading vector respectively. The score vectors are orthogonal, likewise the loading vectors, in addition they are of unit length. The loading vector $\mathbf{p}_1$ defines the direction of the greatest variability and the score vector $\mathbf{t}_1$, also known as the first principal component, represents the projection of each column of $\mathbf{X}$ onto $\mathbf{p}_1$. The first principal component is thus that linear combination of the columns in $\mathbf{X}$ explaining the greatest amount of variability ($\mathbf{t}_1=\mathbf{X}\mathbf{p}_1$). The second principal component is that linear combination of the columns in $\mathbf{X}$ explaining the next greatest amount of variability ($\mathbf{t}_2=\mathbf{X}\mathbf{p}_2$) subject to the condition that it is orthogonal to the first principal component. Principal components are arranged in decreasing order of variability explained. Since the columns in $\mathbf{X}$ are highly correlated, the first a few principal components can explain the majority of data variability in $\mathbf{X}$.

$$\mathbf{X} = \mathbf{T}_k\mathbf{P}_k^T + \mathbf{E} = \sum_{i=1}^{k}\mathbf{t}_i\mathbf{p}_i^T + \mathbf{E} \quad (12)$$

where $\mathbf{T}_k = [\mathbf{t}_1\ \mathbf{t}_2\ ...\ \mathbf{t}_k]$, $\mathbf{P}_k = [\mathbf{p}_1\ \mathbf{p}_2\ ...\ \mathbf{p}_k]$, k represents the number of principal components to retain, and $\mathbf{E}$ is a matrix of residuals of unfitted variation.

If the first k principal components can adequately represent the original data set $\mathbf{X}$, then regression can be performed on the first k principal components. The model output is obtained as a linear combination of the first k principal components of $\mathbf{X}$ as

$$\hat{\mathbf{Y}} = \mathbf{T}_k\mathbf{w} = \mathbf{X}\mathbf{P}_k\mathbf{w} \quad (13)$$

where $\mathbf{w}$ is a vector of model parameters in terms of principal components.

The least squares estimation of $\mathbf{w}$ is:

$$\mathbf{w} = (\mathbf{T}_k^T\mathbf{T}_k)^{-1}\mathbf{T}_k^T\mathbf{Y} \ = (\mathbf{P}_k^T\mathbf{X}^T\mathbf{X}\mathbf{P}_k)^{-1}\mathbf{P}_k^T\mathbf{X}^T\mathbf{Y} \quad (14)$$

The model parameters in Eq(9) calculated through PCR is then

$$\boldsymbol{\theta} = \mathbf{P}_k\mathbf{w} \ = \mathbf{P}_k(\mathbf{P}_k^T\mathbf{X}^T\mathbf{X}\mathbf{P}_k)^{-1}\mathbf{P}_k^T\mathbf{X}^T\mathbf{Y} \quad (15)$$

The number of principal components, k, to be retained in the model is usually determined through cross-validation (Wold, 1978). The data set for building a model is partitioned into a training data set and a testing data set. PCR 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.

3.2 Batch to batch iterative learning control

The batch to batch iterative learning control strategy was developed by Xiong and Zhang (2003) and is briefly introduced here. As batch process dynamics are non-linear and the perturbation model is linearised around the nominal operation trajectories of a batch process, offsets always occur due to modelling errors and unmeasured disturbances. The perturbation model predictions of the current batch run can be corrected by adding model prediction residuals of previous batch runs.

The prediction of perturbation model is defined as

$$\hat{\overline{\mathbf{Y}}}_k = \hat{\mathbf{G}}_s\overline{\mathbf{U}}_k \quad (16)$$

and the absolute model prediction is defined as

$$\hat{\mathbf{Y}}_k = \mathbf{Y}_s + \hat{\overline{\mathbf{Y}}}_k = \mathbf{Y}_s + \hat{\mathbf{G}}_s\overline{\mathbf{U}}_k \quad (17)$$

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

$$\varepsilon_k = \mathbf{Y}_k - \hat{\mathbf{Y}}_k = \overline{\mathbf{Y}}_k - \hat{\overline{\mathbf{Y}}}_k \quad (18)$$

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

$$\tilde{\overline{\mathbf{Y}}}_{k+1} = \hat{\overline{\mathbf{Y}}}_{k+1} + \varepsilon_k \quad (19)$$

The absolute modified model prediction is defined as

$$\tilde{\mathbf{Y}}_{k+1} = \hat{\mathbf{Y}}_{k+1} + \varepsilon_k = \mathbf{Y}_s + \hat{\overline{\mathbf{Y}}}_{k+1} + \varepsilon_k \quad (20)$$

The modified prediction error is defined as

$$\tilde{\varepsilon}_{k+1} = \mathbf{Y}_{k+1} - \tilde{\mathbf{Y}}_{k+1} = \overline{\mathbf{Y}}_{k+1} - \tilde{\overline{\mathbf{Y}}}_{k+1} \quad (21)$$

From the definitions in Eq(18) and Eq(19), we have

$$\tilde{\varepsilon}_{k+1} = \varepsilon_{k+1} - \varepsilon_k \quad (22)$$

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

$$|\varepsilon_k| < B_m \quad (23)$$

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

$$|\tilde{\varepsilon}_k| < |\varepsilon_k| + |\varepsilon_{k-1}| < 2B_m \quad (24)$$

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

$$\mathbf{e}_k = \mathbf{Y}_d - \mathbf{Y}_k = \overline{\mathbf{Y}}_d - \overline{\mathbf{Y}}_k \quad (25)$$

$$\hat{\mathbf{e}}_k = \mathbf{Y}_d - \hat{\mathbf{Y}}_k = \overline{\mathbf{Y}}_d - \hat{\overline{\mathbf{Y}}}_k \quad (26)$$

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

$$\overline{\mathbf{Y}}_d = \mathbf{Y}_d - \mathbf{Y}_s \quad (27)$$

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

$$\widetilde{\mathbf{e}}_k = \mathbf{Y}_d - \widetilde{\mathbf{Y}}_k = \overline{\mathbf{Y}}_d - \overline{\widetilde{\mathbf{Y}}}_k \quad (28)$$

From the definitions in Eq(18), Eq(25) and Eq(28), the following relationship among these three tracking errors can be obtained

$$\varepsilon_k = \hat{\mathbf{e}}_k - \mathbf{e}_k \quad (29)$$

$$\widetilde{\mathbf{e}}_k = \hat{\mathbf{e}}_k - \boldsymbol{\varepsilon}_{k-1} \quad (30)$$

From Eq(26) and Eq(16), an iterative relationship for $\hat{\mathbf{e}}_k$ along the batch index k can be obtained as

$$\hat{\mathbf{e}}_{k+1} = \hat{\mathbf{e}}_k - \hat{\mathbf{G}}_s \Delta\overline{\mathbf{U}}_{k+1} \quad (31)$$

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

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

From the definition of perturbation variables, we can have

$$\Delta\overline{\mathbf{U}}_{k+1} = \overline{\mathbf{U}}_{k+1} - \overline{\mathbf{U}}_k = \mathbf{U}_{k+1} - \mathbf{U}_k \quad (33)$$

Substitute Eq(29) and Eq(31) to Eq(30), we have

$$\widetilde{\mathbf{e}}_{k+1} = \hat{\mathbf{e}}_{k+1} - (\hat{\mathbf{e}}_k - \mathbf{e}_k) = \mathbf{e}_k - \hat{\mathbf{G}}_s \Delta\overline{\mathbf{U}}_{k+1} \quad (34)$$

On the other hand, Eq(29) can be rewritten as

$$\mathbf{e}_k = \hat{\mathbf{e}}_k - \varepsilon_k \quad (35)$$

From Eq(35) and Eq(31), an iterative relationship for $\mathbf{e}_k$ along the batch index k can also be obtained as

$$\mathbf{e}_{k+1} = \mathbf{e}_k - \hat{\mathbf{G}}_s \Delta\overline{\mathbf{U}}_{k+1} - \widetilde{\varepsilon}_{k+1} \quad (36)$$

Given the error transition model in the form of Eq(34) and Eq(36), 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

$$J_{k+1} = \min_{\Delta\overline{\mathbf{U}}_{k+1}} \frac{1}{2}[\widetilde{\mathbf{e}}_{k+1}^T \mathbf{Q} \widetilde{\mathbf{e}}_{k+1} + \Delta\overline{\mathbf{U}}_{k+1}^T \mathbf{R} \Delta\overline{\mathbf{U}}_{k+1}] \quad (37)$$

where **Q** and **R** are positive definitive matrices. Note that the objective function, Eq(37), 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 (Lee et al., 2000). 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 $\mathbf{Q} = \text{diag}\{Q(1), Q(2), \ldots, Q(N)\}$, $\mathbf{R} = \text{diag}\{R(0), R(1), \ldots, R(N\text{-}1)\}$, 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(37) with respect to the input change $\Delta\overline{\mathbf{U}}_{k+1}$ and through straightforward manipulation, the following ILC law can be obtained

$$\Delta\overline{\mathbf{U}}_{k+1} = \hat{\mathbf{K}}\mathbf{e}_k \quad (38)$$

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

$$\hat{\mathbf{K}} = [\hat{\mathbf{G}}_s^T \mathbf{Q} \hat{\mathbf{G}}_s + \mathbf{R}]^{-1} \hat{\mathbf{G}}_s^T \mathbf{Q} \quad (39)$$

From Eq(33) and Eq(38), the ILC law for the control trajectory can be written as

$$\mathbf{U}_{k+1} = \mathbf{U}_k + \hat{\mathbf{K}}\mathbf{e}_k \quad (40)$$

3. APPLICATION TO A FED-BATCH FERMENTATION PROCESS

3.1 A fed-batch fermentation process

The process considered in this paper is a fed-batch yeast fermentation process taken from Yuzgec et al. (2009), where detailed kinetic and dynamic model is presented. The kinetic model of yeast metabolism is based on the bottleneck hypothesis by Sonnleitner and Kappeli (1986) and a dynamic model is developed based on mass balance equations for glucose, ethanol, oxygen and biomass concentrations (Yuzgec et al., 2009). In this study, a simulation programme is developed in MATLAB using the kinetic and dynamic model given in (Yuzgec et al., 2009) and is verified with the results presented in (Yuzgec et al., 2009). The operation objective is to produce maximum amount of biomass by adjusting the glucose feed rate subject to operation constraints.

3.2 Results

The developed process simulator was used to generate 40 historical batch runs to be used in the generation of the time variant linearised model, Gs. MLR, PCR and PLS are used in estimating the model parameters. The 40 batches were generated by adding random perturbations to a nominal feed profile.

The **R** and **Q** weighting values were decided using a trial and error method. The **Q** values were fixed at 1 while different **R** values were studied. Fig. 1 to Fig. 3 show the results of batch to batch control with different R values with MLR, PCR, and PLS models respectively. It can be seen in all three cases that **R** value of 0.00001 gives good performance. Too large **R** values lead to sluggish performance while too small R values lead to oscillation or unstable performance.

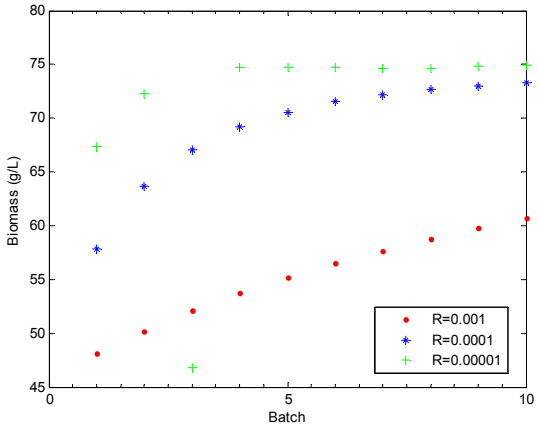

Fig. 1. Batch to batch control under MLR model

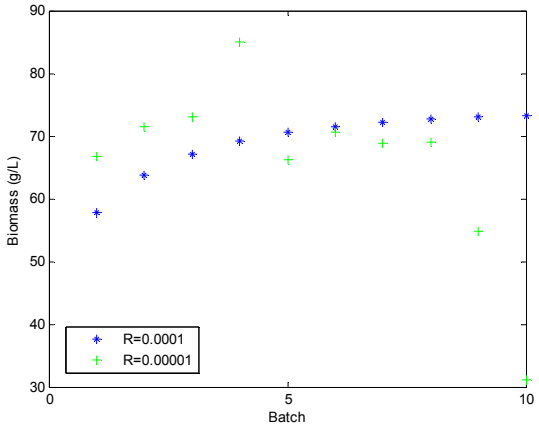

Fig. 2. Batch to batch control under PCR model

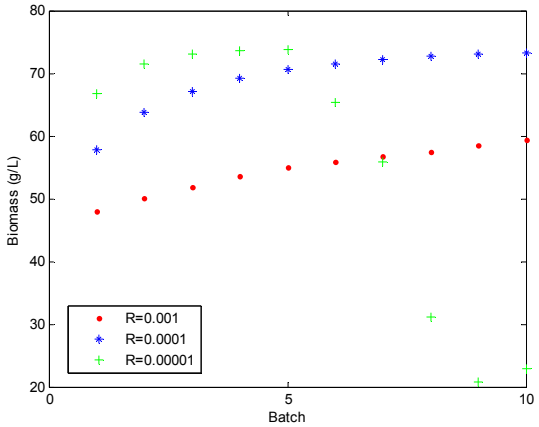

Fig. 3. Batch to batch control under PLS model

Batch to batch control based on the three types of models (MLR, PCR, and PLS) were tested and compared for the following three different cases:

Case 1: Constant Gs, Ys and Us
Case 2: Updating Gs, Ys and Us
Case 3: Updating Gs

Whereby, Gs is the time variant linearised model, Ys is the nominal biomass output and Us is the nominal substrate feed.

Each of the cases was run for 30 batches to investigate the control performance. The desired final bio-mass concentration value was set at 74g/L. A disturbance was introduced from batch 21. The initial substrate concentration, S_0, was changed to 305g/l from its nominal value of 325g/l from batch 21.

Fig. 4 shows the control performance under the MLR model for the three different cases. It can be seen from Fig. 4 that batch to batch control with fixed model and fixed reference trajectories does not give good performance and becomes unstable even without the presence of disturbances. This is due to the fact that the fixed nominal model becomes invalid when the operation trajectories shift away from the nominal trajectories. The control performance with updated model appears to be satisfactory when there is no disturbance. However, with the presence of unknown disturbance, batch to batch control based on MLR model does not give satisfactory performance even with updated model and updated reference trajectories. This could be due to that the MLR model is not appropriate due to the correlations among the control actions during different batch stages.

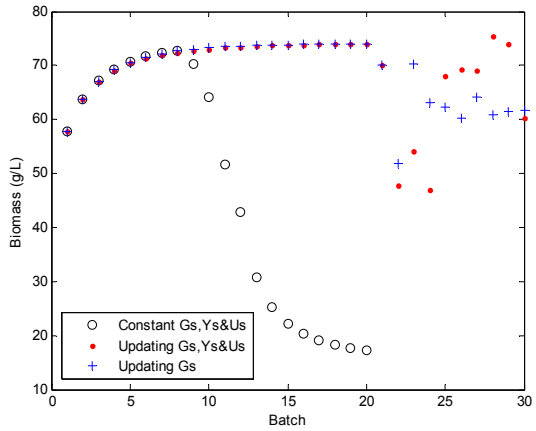

Fig. 4. Control performance under the MLR model

Fig. 5 and Fig. 6 show, respectively, the control performance under PCR model and PLS model. Once again, batch to batch control under fixed model does not give satisfactory performance even without disturbances. With updated linearised models but constant reference trajectories, batch to batch control under PCR model or PLS model gives satisfactory performance when there is no disturbance, but gives unsatisfactory performance when disturbance presents. However, with updated linearised models and updated reference trajectories, batch to batch control under PCR model or PLS model gives satisfactory performance when unknown disturbances are present. This remarkable improvement over batch to batch control under MLR model is due to that the PCR model and PLS model are more appropriate than the MLR model.

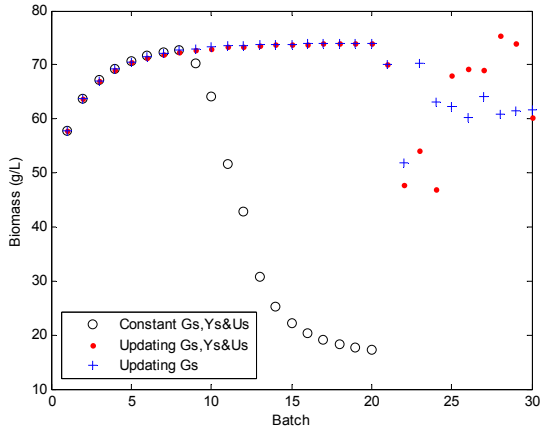

Fig. 5. Control performance under the PCR model

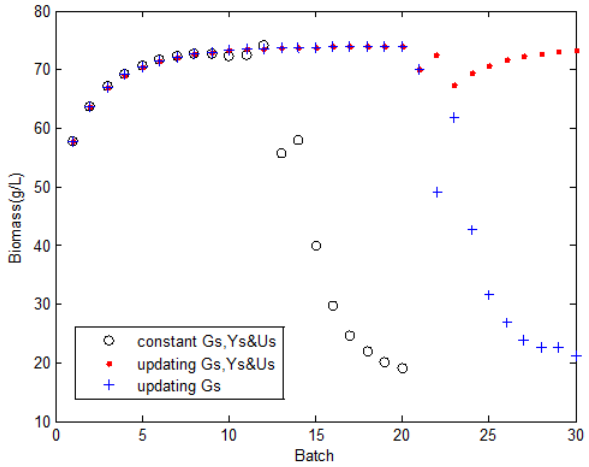

Fig. 6. Control performance under the PLS model

4. CONCLUSIONS

A batch to batch iterative learning control strategy based on incrementally updated linearised model is presented. The proposed strategy overcomes the detrimental effects of model-plant mismatches and unknown disturbances by incrementally updating the control policy using information from the previous batches. Control policy updating is calculated using a linearised model identified from process operation data. To cope with nonlinear behaviour of fermentation processes, the linearised model is updated from batch to batch and the immediate previous batch is used as the reference batch. To address the colinearity among the control actions during different batch stages, the linearised model is identified using PCR and PLS. Application results on a simulated fed-batch fermentation process demonstrate that the proposed technique is very effective.

REFERENCES

Alvarez, M. A., S. M. Stocks, S. B. Jørgensen (2009). Bioprocess modelling for learning model predictive control (L-MPC), in *Computational Intelligence Techniques for Bioprocess Modelling, Supervision and Control*, Springer-Berlin, 237-280.

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

Gao, F., Y. Yang, and C. Shao (2001). Robust iterative learning control with applications to injection molding process, *Chemical Engineering Science*, **56**, 7025-7034.

Geladi, P. and B. R. Kowalski (1986). Partial least squares regression: a tutorial, *Analytica Chimica Acta*, **185**, 1-17.

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

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

Lee, K. S. and J. H. Lee (1997). Model predictive control for nonlinear batch processes with asymptotically perfect tracking, *Computers Chem. Engng.*, **21**, s873-s879.

Park, S. and W. F. Ramirez (1988). Optimal production of secreted protein in fed-batch reactors. *AIChE J.*, **34**, 1550-1558.

Sonnleitner, B. and O.Kappeli (1986). Growth of S. cerevisiae is controlled by its limited respiratory capacity: Formulation and verification of a hypothesis. *Biotech & Bioeng*; **28**, 79-84.

Tian, Y., J. Zhang, and A. J. Morris (2002). Optimal control of a fed-batch bioreactor based upon an augmented recurrent neural network models, *Neurocomputing*, **48**(1-4), 919-936.

Wold, S. (1978). Cross validatory estimation of the number of components in factor and principal components models, Technometrics, **20**, 397-404.

Xiong, Z. and J. Zhang (2005). Batch-to-batch iterative optimisation control based on recurrent neural network models, *Journal of Process Control*, **15**(1), 11-21.

Xiong, Z. and J. Zhang (2003). Product quality trajectory tracking in batch processes using iterative learning control based on time-varying perturbation models, *Ind. Eng. Chem. Res.*, **42**(26), 6802-6814.

Yüzgeç, U., M. Türker, A. Hocalar (2009). On-line evolutionary optimization of an industrial fed-batch yeast fermentation process. *ISA Transactions*, **48**, 79-92.

Zhang, J. (2004). A reliable neural network model based optimal control strategy for a batch polymerisation reactor, *Ind. Eng. Chem. Res.*, **43**, 1030-1038.

Zhang, J. (2005) A neural network based strategy for the integrated batch-to-batch control and within batch control of batch processes, *Transactions of the Institute of Measurement and Control*, **27**(5), 391-410,.