Chemistry and Materials Research
ISSN 2224- 3224 (Print) ISSN 2225- 0956 (Online)
Vol.14 No.3, 2022

Batch-to-Batch Iterative Learning Control of a Fed-Batch Fermentation Process

Nwamaka Linda Okoli [1*] Franklin Okoro [2]
1. University of Newcastle, United Kingdom
2. CleanScript Group, Nigeria
* E-mail of the corresponding author: f.okoro@cleanscriptgroup.com

Abstract
In this work, Iterative Learning Control on a fed-batch fermentation process using linearised models has been studied. The repetitive nature of batch processes enables ILC to obtain information from a previous batch in order to improve the performance of the current batch such that the product quality converges asymptotically to the desired trajectory The basic batch to batch ILC law presents the control action of a current batch as a summation of the control action from the previous batch and the deviation of the output trajectory from the desired reference trajectory incorporation with a learning rate. In a bid to address the issue of the process non-linearity, the control policy and the output trajectory were linearised around their respective nominal trajectories. The linearised models were then identified using Multi Linear Regression (MLR), Principal Component Analysis (PCR) and Partial Least Squares (PLS). In order to curb the effects of plant-model mismatches and process variations, the linearised models were reidentified after each batch operation. This was done by selecting the immediate previous batch as the nominal batch and then adding the recently obtained process data into the historical data batch on completion of the current batch run. The weighting matrices in the objective function were carefully selected taking into consideration that they have a major influence on the robust performance of the process. In using PLS and PCR models the issue of process collinearity was effectively addressed. The proposed batch to batch ILC strategy was applied to a simulated fed-batch fermentation process for the production of secreted protein. The results of the optimal control policy were comparable to that obtained in using full mechanistic model. ILC, a simple but yet an effective optimal control strategy has demonstrated to be a viable option in complex processes such as batch processes where mechanistic models are difficult to develop.
Keywords: Iterative Learning Control, batch process, fed-batch fermentation, batch to batch ILC, control policy.
DOI: 10.7176/CMR/14-3-02
Publication date: August 31st 2022

1. Introduction

In a batch process, the fermentation process is carried out with the addition of substrate or removal of the products during the course of the process. However, oxygen, an antifoam agent, and acid or base is added to control the ph. On completion of the batch run, the products are taken out and the reactor is cleaned out for another run. Fed-batch fermentation is an extension of the batch process. It involves the injection of nutrients into the bioreactor during fermentation in which the product(s) remain in the bioreactor till the batch run ends (Yamanè and Shimizu, 1984). In a fed-batch process the medium is supplemented with nutrients that are depleted or that may be needed for the terminal stages of the culture. Following that there is no withdrawal of liquid, the volume of the reactor increases till the final batch time. The difference between fed-batch and batch operation is that in the former feed is continuously supplied. An obvious advantage of fed-batch operation is that nutrient levels are continuously varied to achieve favorable growth conditions without significant risk of culture contamination (Henson, 2006). Over the years, fed batch process has been preferred over the normal batch process in the manufacturing industry. Production of by-product is a major threat in any production process. Fed batch fermentation limits the production of bye products by controlling the growth limitation of substrate.

Due to the ever-increasing market competition and need for high value-added products, there is a growing need to develop optimal control strategies in the manufacturing industry. Lot of researchers have made efforts to come up with different optimal control strategies each with its success and limits. In order to come up with a good optimal strategy, the dynamics of the process involved have to be well understood. According to Van Impe and Bastin (1995), the design of high-performance model-based control algorithms for batch processes is limited by two major problems. Firstly, due to the complexity of these processes, their kinetics are poorly understood nonlinear functions coupled with the parameters that are generally time-varying. Secondly, there is a limitation of reliable online sensors suitable for real-time monitoring of the process variables. This calls for an adequate optimal solution. Attempts have been made in the use of mechanistic models to describe these processes. However due to their complex nature developing mechanistic models can be time consuming, costly and at the end can turn up being unrealistic. A lot of researchers have deemed it wise to resort to data-based methods to solve this growing need. Data based methods are empirical methods that develops a relationship between the measured inputs to the outputs that describe the process response to changes in inputs (Cinar et al, 2003). An important objective of batch

processes lies in improving the performance of the process from batch to batch. Due to the repetitive nature of these processes, information from the previous batches can provide an insight on how to improve subsequent batches. Most learning control strategies utilize this set of information. Iterative learning control is one of such promising methods.

Iterative Learning Control (ILC) is an optimal control strategy that tries to address the problem of transient response performance for systems that operate in a repetitive manner. (Moore, 2006). It adjusts the process feed rate based on the errors observed from past operation. (Moore, 2006). ILC overcomes the shortcomings posed by the need for a complete and accurate knowledge on the process model before a desired product quality is achieved. Utilizing the information on the control policy and error on the previous batch, it iteratively updates the input from trial to trial till the desired reference trajectory is achieved. In order to avoid plant model mismatches and unknown disturbances, the basic approach to ILC is to linearise the process around a fixed nominal control trajectory. However significant batch to batch variation could lead to the optimal control strategy becoming prone to error and failing in its principal objective of achieving the desired trajectory. To overcome this limitation, the linearised models are preferably updated after each batch operation. In doing this, the non-linear behaviour of processes such as fermentation is fully addressed.

ILC has relatively become well established in the area of optimal control. ILC is a form of an intelligent control which overcomes the shortcomings of traditional controller design. Its application is mostly found in complex systems which operate in a repetitive manner. With issues of non-linearity and complexity mostly associated with batch processes, ILC has become a good optimal control option for most batch related industries due to its ability to address these issues. In this study, the specific objectives included to: develop an ILC strategy for a fed batch bioreactor for the production of secreted protein to address model-plant mismatches and unknown plant disturbances, generate and compare the performance of different linearised models used in developing the batch to batch ILC strategy (the model parameters were generated using Multiple Linear Regression (MLR), Partial Least Square (PLS) and Principal Component Regression (PCR)), and to compare batch to batch ILC performance based on fixed and updated model parameters, control policy and product quality.

2. ILC Algorithm

Arimoto *et al*. (1984) was the first to formulate a basic ILC algorithm. This is represented in Figure 1.

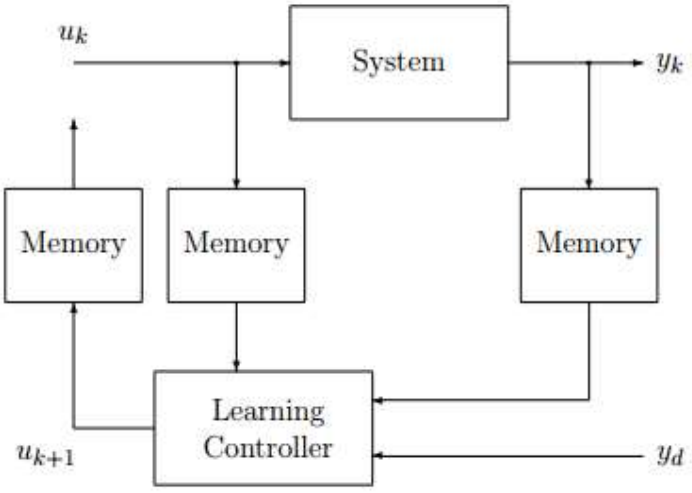


Figure 1: Standard iterative learning control Strategy (Moore, 2006)

From Figure 1, the learning controller uses the memory stored up as the control policy for the current trial $u_k(t)$ and the memory of the error, $e_k(t)$ which is the difference between the current trial's output $y_k(t)$ and the desired trajectory, $y_d(t)$ to update the next control policy for the next trial $u_{k+1}(t)$. $u_{k+1}(t)$ becomes a summation of control profile and the process tracking error from the immediate previous trial. This process is continued iteratively until y_k converges to the desired reference trajectory. Therefore, an ILC algorithm operates in such a mode that the input is made better from trial to trial till the process output reaches the desired trajectory. From Figure 1, a typical ILC algorithm can be deduced as;

$$u_{k+1}(t) = u_k(t) + Le_k(t) \quad (2.1)$$

$$e_k(t) = r(t) - y_k(t) \quad (2.2)$$

Where L is the learning rate.

Besides the general algorithm above, so many researchers have suggested different forms of basic algorithm. Some do not require prior knowledge of the plant model while some need a mathematical representation of the plant (Cai, 2009). These algorithms are summarised by Cai (2009) and is shown in Table 1.

Table 1: Basic ILC Algorithms in Equations

Algorithm	Equations
D-type ILC	$u_{k+1}(t) = u_k(t) + L\dot{e}_k(t)$
P-type ILC	$u_{k+1}(t) = u_k(t) + Le_k(t)$
PID-type ILC	$u_{k+1}(t) = u_k(t) + K_p e_k(t) + K_i \int e_k(t)dt + K_d \dot{e}_k(t)$
Higher order ILC	$u_{k+1}(t) = u_k(t) + \sum_{n=0}^{N} \beta_n(t) e_l(t)$
Phase lead ILC	$u_{k+1}(t) = u_k(t) + Le_k(t+\tau), \tau \in \mathbb{Z}$
Forgetting factors	$u_{k+1}(t) = \beta[u_k(t) + Le_k(t)]$

Nonlinear Representations of Batch Processes

Batch processes are usually non-linear. In order to address this issue, a mathematical representation of batch process non-linearity is given as reviewed by (Xiong and Zhang, 2003). Consider a batch process with a fixed batch run length (t_f) and N sampling intervals

where $N = t_f/h$

h= sampling time

The variables of the product quality which are the outputs is represented as $y \in R^n \; with \; n \geq 1$. This can be obtained off-line by analyzing the samples taken during the batch run. The manipulated variable is represented as $u \in R^m \; with \; m = 1$. This can be measured at each sampling time on-line. The product quality is defined as;

$Y_k = [{y_k}^T(1), {y_k}^T(2), \ldots . {y_k}^T(N)]^T$ (2.3)

The control trajectories can be defined as;

$U_k = [{u_k}^T(0), {u_k}^T(1), \ldots . {u_k}^T(N-1)]^T$ (2.4)

Where k=batch index

The desired reference trajectories of product quality are defined as;

$Y_d = [{y_d}^T(1), {y_d}^T(2), \ldots . {y_d}^T(N)]^T$ (2.5)

In as much as a batch operation is modelled with a dynamic model, it would be much easier to consider a static function relating the control sequence to the product quality sequences over the whole batch duration (Lee *et al*, 1999). In consideration of the casualty, a product variable $y_k(t)$, at time t, is a nonlinear function of all control actions up to time t,

$U_k(t) = [u_k(0), u_k(1), \ldots . u_k(N-1)]^T$ that is;

$y_k(t) = f_t\big(U_k(t)\big) + v_k(t), \;\; t = 1, 2, \ldots . , N; y_k(0) = y_0$ (2.6)

Where $f_t(*)$ is the nonlinear function between $U_k(t)$ and $y_k(t)$

$v_k(t)$ is the measurement noise at time t

Equation (2.6) can be rewritten in matrix form as;

$Y_k = F(U_k) + v_k$ (2.7)

Where $F(*)$ is the nonlinear function relating $U_k(t)$ to $y_k(t)$ at different sampling times.

v_k, is a vector of measurement noises represented as $[v_k(0), v_k(1), \ldots . v_k(N-1)]^T$

Linearisation of nonlinear batch process model

This is done by linearising the non-linear model around some nominal trajectories. Nominal trajectories can be chosen as the mean profile of all the past control trajectories and its corresponding output trajectories, best batch data which is reidentified after every batch run, or immediate previous batch data which is updated after every batch run (Jewaratham, 2013).

Let the nominal control trajectory be defined as;

$U_k = [[u_s(0), u_s(1), \ldots . u_s(N-1)]^T \;]$ (2.8)

The corresponding product quality trajectory can be defined as;

$Y_k = [{y_s}^T(1), {y_s}^T(2), \ldots . {y_s}^T(N)]^T$ (2.9)

Where the subscript s denotes the nominal batch and $y_s(0) = y_0$

Linearizing the nonlinear batch process model described by eqn. 2.7 with respect to U_s around the nominal trajectories (U_s, Y_s), the following equation is obtained;

$Y_d = Y_s + \frac{\partial F(U_k)}{\partial U_k}|_{U_s}(U_k - U_s) + w_k + v_k$ (2.10)

Where w_k is a matrix of model errors due to linearization and is represented as; $[w_k{}^T(1), w_k{}^T(2), \ldots\ldots w_k{}^T(N)]^T$

v_k represents the effects of noise and unmeasured disturbances.

The model parameter, G_s is represented as;

$$G_s = \frac{\partial F(U_k)}{\partial U_k}|_{U_s} \quad (2.11)$$

It should be noted that G_s is known as linear time varying because it varies with U_s which is updated from batch to batch.

The structure of G_s is restricted to the following lower-block-triangular form because of the causality

$$G_s = \begin{bmatrix} g_{10} & 0 & \cdots & 0 \\ g_{20} & g_{21} & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ g_{N0} & g_{N1} & \cdots & g_{NN-} \end{bmatrix} \quad (2.12)$$

Where $g_{ij} \in R^m$

Linear Time-Varying Perturbation Models

The perturbation variable of the control and product quality variables are defined as:

$$\bar{U}_k = U_k - U_s \quad (2.13)$$

The perturbation variable of the product quality variables is defined as;

$$\bar{Y}_k = Y_k - Y_s \quad (2.14)$$

The linearized time-varying perturbation model is then obtained from eqn. 2.10 as;

$$\bar{Y}_k = G_s\bar{U}_k + d_k \quad (2.15)$$

Where d_k, the model disturbance sequence is defined as;

$$d_k = w_k + v_k \quad (2.16)$$

This is bounded by small positive constant B_d such that

$$d_k < B_d \quad (2.17)$$

Optimal Iterative Learning Control of Batch Processes

Offsets usually occur in batch processes as a result of modelling errors and unmeasured disturbances. This is because of the process nonlinearity and linearization of perturbation model around the nominal operation trajectories (Xiong and Zhang, 2003). Corrections can be made to the predictions of the perturbation model by addition of the model prediction residuals of previous batch runs (Xiong and Zhang, 2003).

The prediction of the perturbation model is defined as;

$$\hat{\bar{Y}}_k = \hat{G}_s\bar{U}_k \quad (2.18)$$

The absolute model prediction is given as;

$$\hat{Y}_k = Y_s + \hat{\bar{Y}}_k = Y_s + \hat{G}_s\bar{U}_k \quad (2.19)$$

On completion of the kth batch run, prediction errors between off-line-measured or analysed product qualities and their model predictions can be calculated as;

$$\varepsilon_k = Y_s - \hat{Y}_k = \bar{Y}_k - \hat{\bar{Y}}_k \quad (2.20)$$

The modified prediction of the perturbation model in the (k + 1)th batch run is obtained in on the basis of the prediction errors of the kth batch run as;

$$\tilde{\bar{Y}}_{k+1} = \hat{\bar{Y}}_{k+1} + \varepsilon_k \quad (2.21)$$

Defining the absolute modified model prediction;

$$\tilde{Y}_{k+1} = \hat{Y}_{k+1} + \varepsilon_k = Y_s + \hat{\bar{Y}}_{k+1} + \varepsilon_k \quad (2.22)$$

The modified prediction error becomes;

$$\tilde{\varepsilon}_{k+1} = Y_{k+1} - \tilde{Y}_{k+1} = \bar{Y}_{k+1} - \tilde{\bar{Y}}_{k+1} \quad (2.23)$$

From equations 2.19 and 2.20;

$$\tilde{\varepsilon}_{k+1} = \varepsilon_{k+1} - \varepsilon_k \quad (2.24)$$

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

$$|\varepsilon_k| < B_m \quad (2.25)$$

B_m is a measure used to represent the deviation of $\hat{\bar{Y}}_k$ from $\bar{Y}_k$ or $\hat{Y}_k$ from Y_k. The modified prediction error is bounded by $2B_m$;

$$|\tilde{\varepsilon}_k| < |\varepsilon_k| + |\varepsilon_{k-1}| < 2B_m \quad (2.26)$$

The structure of the modified prediction based optimal ILC is shown in figure 2;

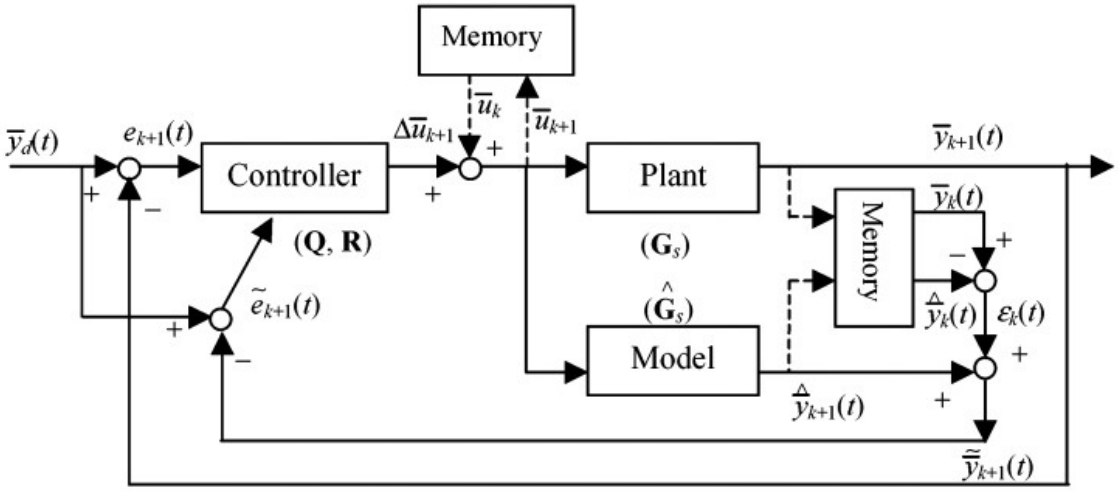


Figure 2: Modified Prediction-based optimal ILC (Xiong and Zhang, 2003)

Figure 2 gives a summary of the optimal ILC for batch-to-batch process. The control trajectory U_k is implemented at the kth batch run. On the completion of the batch run, the product qualities Y_k is obtained by off-line analysis of samples taken during the batch run. The model prediction errors e_k are then evaluated and used to correct the model predictions for the next batch (Xiong and Zhang, 2003). Based on the modified predictions $\tilde{\bar{Y}}_{k+1}$, a new control policy U_{k+1} for the next batch is calculated using the ILC law. At the next batch, this procedure is repeated until the desired trajectory is attained.

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

$$e_k = Y_d - Y_k = \bar{Y}_d - \bar{Y}_k \quad (2.27)$$

$$\hat{e}_k = Y_d - \hat{Y}_k = \bar{Y}_d - \hat{\bar{Y}}_k \quad (2.28)$$

Where $\bar{Y}_d$ is the deviated desired trajectory and defined as;

$$\bar{Y}_d = Y_d - Y_s \quad (2.29)$$

The tracking error of the modified prediction of the perturbation model is defined as;

$$\tilde{e}_k = Y_d - \tilde{Y}_k = \bar{Y}_d - \tilde{\bar{Y}}_k \quad (2.30)$$

From eqns. 2.18, 2.27, and 2.30, the following relationships among the three tracking errors can be obtained;

$$\varepsilon_k = \hat{e}_k - e_k \quad (2.31)$$

$$\tilde{e}_k = \hat{e}_k - \varepsilon_{k-1} \quad (2.32)$$

From eqns. 2.18 and 2.28, an iterative relationship for $\hat{e}_k$ along the batch index k can be obtained as;

$$\hat{e}_{k+1} = \hat{e}_k - \hat{G}_s \Delta\bar{U}_{k+1} \quad (2.33)$$

Where $\Delta\bar{U}_{k+1}$ is defined as;

$$\Delta\bar{U}_{k+1} = \bar{U}_{k+1} - \bar{U}_k \quad (2.34)$$

From eqn. 2.10

$$\Delta\bar{U}_{k+1} = \bar{U}_{k+1} - \bar{U}_k = U_{k+1} - U_k \quad (2.35)$$

Substituting of eqns. 2.29 and 2.31 into eqn. 2.32;

$$\tilde{e}_{k+1} = \hat{e}_{k+1} - (\hat{e}_k - e_k) = e_k - \hat{G}_s \Delta\bar{U}_{k+1} \quad (2.36)$$

Eqn. 2.31 can be rewritten as;

$$e_k = \hat{e}_k - \varepsilon_k \quad (2.37)$$

From eqns. 2.31 and 2.33, an iterative relationship for e_k along the batch index k can also be obtained as

$$e_{k+1} = e_k - \hat{G}_s \Delta\bar{U}_{k+1} - \tilde{\varepsilon}_{k+1} \quad (2.38)$$

Given the error transition model in the form of eqns. 2.36 and 2.38, the objective of ILC is to design a learning algorithm such that the product qualities follow the specific desired reference trajectories by manipulating the control policy. According to Lee *et al* (2007) the learning algorithm is expected to have the following property;

$$\lim_{k\to\infty} ||e_k||^2_Q = \min_U ||e||^2_Q \quad (2.39)$$

By the certainty-equivalence principle (Lee *et al.*,1997), solving the following quadratic objective function using the modified prediction errors upon completion of the kth batch run to update the input trajectory for the (k + 1)th

batch run is considered;

$$J_{k+1} = \min_{\Delta U_{k+1}} \frac{1}{2}[\tilde{e}_{k+1} + {}^{T}Q\tilde{e}_{k+1} + \Delta\bar{U}_{k+1}{}^{T}R\Delta\bar{U}_{k+1}] \quad (2.40)$$

where Q and R are positive-definitive matrices. Q is the weighting for product quality control errors while R is the weighting for control action. It can be noticed that the objective function, eqn. 2.40, has a penalty term on the input change $\Delta\bar{U}_{k+1}$ between two adjacent batch runs, and the algorithm has an integral action with respect to the batch index k. Careful consideration should be made when selecting the weighting matrices Q and R. A larger weight on the input change will lead to more conservative adjustments and slower convergence. For the sake of simplicity, Q and R are selected in this study as $Q = \lambda_q I_N$ and $R = \lambda_r I_N$. By finding the partial derivative of the quadratic objective function, equation 2.40 with respect to the input change $\Delta\bar{U}_{k+1}$ and performing straightforward manipulations, the following ILC law is obtained;

$$\Delta\bar{U}_{k+1} = \hat{K}e_k \quad (2.41)$$

Where $\hat{K}$ is known as the learning rate

$$\hat{K} = [\hat{G}_s{}^{T}Q\hat{G}_s + R]^{-1}\hat{G}_s{}^{T}Q \quad (2.42)$$

From eqns. 2.35 and 2.41, the ILC law for the control trajectory is written as

$$U_{k+1} = U_k + \hat{K}e_k \quad (2.43)$$

3. Simulation of Fed batch Reactor

The process considered in this work is the fed batch reactor for the production of secreted protein. The process' dynamic models presented here are adapted from work done by Park and Ramirez (1988) on optimal production of secreted protein. Baker's yeast was used as the host organism. The state variables in the fed batch reactor are described by the following differential equations:

$$\dot{P}_M = A(S)(P_T - P_M)\frac{q}{V}P_M \quad (3.1)$$

$$\dot{P}_T = B(S) - \frac{q}{V}P_T \quad (3.2)$$

$$\dot{X} = C(S)X - \frac{q}{v}X \quad (3.3)$$

$$\dot{S} = -YC(S)X + \frac{q}{v}(m - S) \quad (3.4)$$

$$\dot{V} = q \quad (3.5)$$

Where

$$A(S) = \phi(\mu_x) = \frac{4.75C(S)}{0.12+C(S)} \quad (3.6)$$

$$B(S) = f_p(S) = \frac{Se^{-5S}}{0.1+S} \quad (3.7)$$

$$C(S) = \mu_x(S) = \frac{21.87S}{(S+0.4)(S+62.5)} \quad (3.8)$$

The state variables are defined as the following;

$P_M =$ Amount of secreted protein on a unit culture volume basis

$P_T =$ Total amount of protein on a unit culture volume basis

$X =$ Culture cell density, g/l

$S =$ Culture glucose concentration, g/l

$V =$ Culture Volume, l

Other parameters include;

$q =$ Feed flow rate, l/h. This is the control variable which is the rate at which glucose is fed to the reactor.

$m =$ Glucose concentration of the feed stream, g/l

Y = Yield of glucose (g)/cell mass (g)

$\phi =$ Protein secretion rate

$f_p =$ Protein expression rate

$\mu_x =$ Specific growth rate of the host cell

The process initial conditions are set as;

$P_M(t_0) = 0$, $P_T(t_0) = 0$, $X(t_0) = 1.0\ g/l$, $S(t_0) = 5.0\ g/l$, $V(t_0) = 1L$, $m = 20\ g/l$, $Y = 7.3$, $t_f = 15h$

In this work, glucose is chosen as the substrate. The process input and output are the feed rate and the amount of secreted protein respectively. From equations 3.1 to 3.4 the ratio $\frac{q}{V}$ represents the dilution rate, that is, dilution of each of the state variable as the feed is added during operation. It is important that this ratio remains less than

the specific growth rate of the cells in order to avoid washout of cells from the bioreactor. As can be seen in equations 3.6, 3.7 and 3.8, the different rates are all functions of the substrate. Aiba *et al.* (1968) on their study of the kinetics of product inhibition in alcohol fermentation discovered that the specific growth rate of cells is significantly reduced by ethanol accumulation. However, in order to avoid additional complexity, the effect of this accumulation is not considered in this work since it does not change the control strategy for the optimization problem (Park and Ramirez, 1988).

The objective in this study is to maximize the secreted protein at the final batch time by manipulating the feed rate known as the control policy. Thus, the performance index is formulated as:

$$Q = P_M(t_f)V(t_f)$$

The feed rate is bounded by;

$$0\ L/h \leq q \leq 10L/h$$

For the purpose of this work, the batch time was divided into 10 equal stages with a constant feed rate at each stage after which there was only one output at the end of each batch run. The baseline control policy used for this work was gotten from that reported by Xiong and Zhang (2003) in comparison to the control policy from the mechanistic model from Park and Ramirez, (1988) and that from augmented recurrent neural network. This is shown in Figure 3.

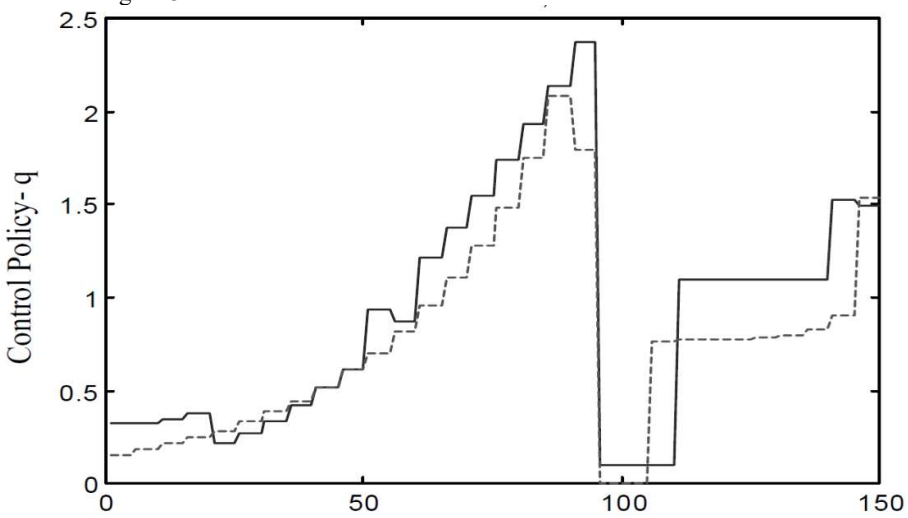


Figure 3: Fed batch reactor Control Policy. (Xiong and Zhang, 2003)

The control policy on dotted lines represents that from mechanistic model while the control policy on solid lines represents that from neural networks. It can be seen from this figure that both profiles are quite close. Therefore, the combination of these two is used as a guideline in selecting the baseline control profile in this study. This is given as;

$$q = [0.3, 0.3, 0.5, 0.8, 1.9, 2.5, 0.1, 1.2, 1.1, 1.5]$$

With the initial conditions, the simulation of the fed batch reactor was performed using MATLAB software where ODE45 function was implemented to solve the differential equations. After simulation with the baseline control policy the corresponding amount of secreted protein evaluated is 20g. Random perturbations are added to the baseline policy to generate the rest of the control policies for other batches. ILC algorithm parameters are obtained using the data from this simulation. Profiles of the state variables after simulation with the control profile is shown in figures 4 to 7.

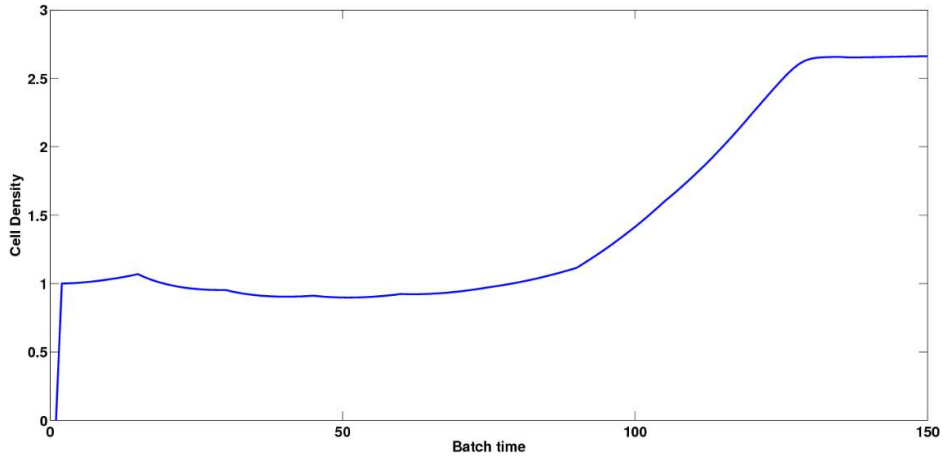


Figure 4: Cell Density

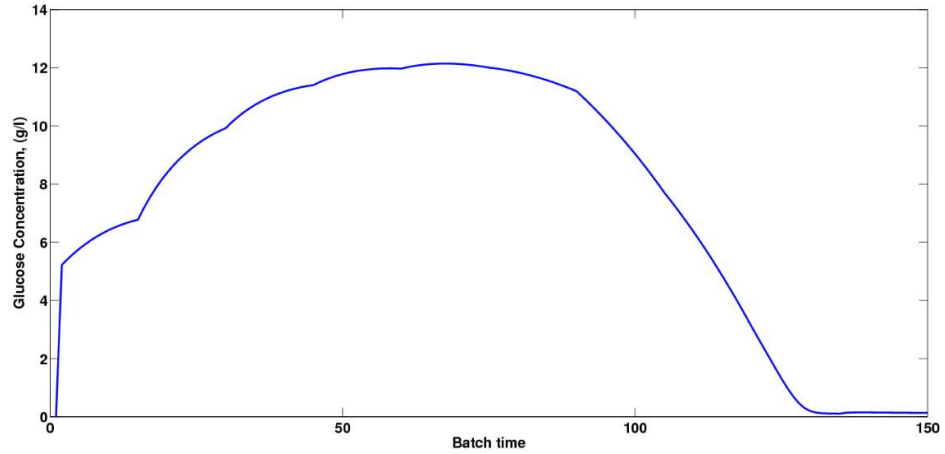


Figure 5: Glucose Concentration

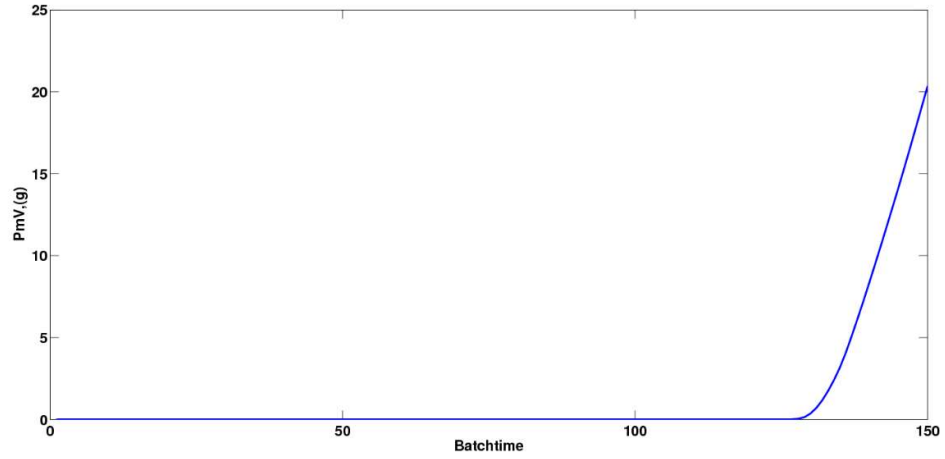


Figure 6: Amount of secreted protein, $P_M V$

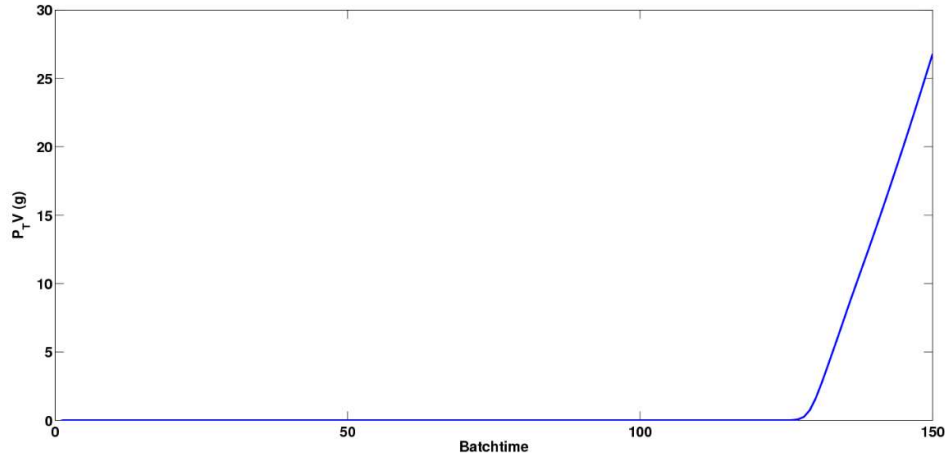


Figure 7: Total amount of protein, $P_T V$

From these figures, it is observed that at the beginning of the fermentation process, the substrate shows an increase after the initial condition of 5g/l. However, it can be noticed that increase in the substrate above its initial condition does not show any significant change in the cell growth. This is because the initial condition 5g/l is chosen such that it supports the maximum growth rate of cells. As the batch time progresses the glucose gradually gets consumed. At this stage it is being utilized for the cell growth. This stage is important because it is in the cells that the secretion and expression of protein occur. However, formation of protein does not occur at this stage due to catabolic repression. As the glucose consumption increases, the cell growth increases exponentially. At the depletion of the substrate which is at a very low concentration of glucose, the formation of secreted and total protein is formed building up to 25 g and 27g respectively while the cell growth remains stationary at about 2.7g/l till the end of batch time.

Developing mechanistic model for processes such as fed batch processes are usually difficult. However, Park and Ramirez (1988) successfully developed such models for the secretion of protein in fed batch reactors. The kinetic model for protein secretion in a fed batch reactor was given here. The initial conditions for the process were also outlined and plots of the various state variables were shown to understand the mechanism of the fermentation process. Subsequently, the plots will be compared to those obtained after the application of the ILC algorithm to the simulated model. Successful works done using this reactor model show that the set of differential equations defining the process is reliable.

4. Methodology

Linearised Model Building

In order to build the ILC models, 30 historical batch runs were selected for this work. These batch runs initially consist of the input which is the feed rate. Each batch length is divided in 10 equal intervals and at each interval the feed rate is maintained at a constant value. Thus, each batch is regarded as a control profile or policy. The baseline control policy was gotten based on that reported by Xiong and Zhang (2003) as shown in section 3. The rest of the batches were then generated by adding random perturbation to the initial control policy. The batch time, t_f as earlier mentioned is taken as $15h$ and the sampling time as $0.10h$ therefore each feed rate in a batch is introduced at the specified sample time for a duration of $15h$. The desired performance index, $P_M(t_f)V(t_f)$ is set as $27g$. This is set slightly higher than that achievable in a normal batch run in order to demonstrate the ability of ILC in achieving the maximum amount of product obtainable.

The procedure for the model building is outlined as follows;

- The data is divided into 20 sets of training data and 10 sets of testing data. The control profiles are represented in MATLAB as;

U=[u(1),u(2)……., u(30)];

- The output Y which is mass of the secreted protein is obtained at the end of each batch run and is represented as;

Y=[y(1),y(2)……., y(30)];

It is noted that a single output is expected from each batch run so the model is in form of a multiple input- single output (MISO) model.

- The nominal input and output trajectories are then defined as last batch in the historical database, both for testing and training data

- For the perturbation variables, the nonlinear function is linearized around U_k and its corresponding output set of data Y_k is generated. This is done by subtracting their respective nominal trajectories from the corresponding historical batches.
- The model parameters are then estimated using MLR, PCR and PLS. It is to be noted that the models relate the input to the output. Thus, they are of the form;

$$Y = \theta_1 x_1 + \theta_2 x_2 + \cdots + \theta_N x_N$$

Where θ is the model parameters and N, the no of stages $= 10$

Once the model parameters are evaluated, the batch-to-batch ILC is applied to the process. This is outlined as follows;

- The weighting matrix $Q\ and\ R = \lambda I$ are selected. I is a 10×10 identity matrix. The value of R is selected based on that which converges faster to the desired trajectory.
- The control action also known as the learning rate is calculated using equation 2.42
- The tracking error for the current batch is calculated using equation 2.38. The control policy for the next batch is then estimated using equation 2.43. This is followed by generating the next batch of output data.
- This previous step is repeated till the last batch.

For the updated models the model parameters, control profiles, X and output, Y are re- estimated after each batch. This is done as follows;

- For k[th] batch, the nominal trajectory is selected as the control profile and product quality of the immediate previous batch.
- As in the previous procedure, the perturbation variable is obtained by linearizing the nonlinear function around U_k and its corresponding output set of data Y_k.
- The model parameters are estimated.
- On completion of the current batch run, the new set of input, U_k and output, Y_k are added to the initial historical set of data. The updated historical batch set becomes;

$$\mathrm{C}_k{}^{u} = [\,\mathrm{C}_{k-1}{}^{uT}\ \ U_k]$$
$$\mathrm{C}_k{}^{y} = [\,\mathrm{C}_{k-1}{}^{yT}\ \ Y_k]$$

Where $\mathrm{C}_{k-1}{}^{uT}$ is the previous historical set of data for the control profile and $\mathrm{C}_{k-1}{}^{yT}$ is the corresponding historical set of data for the product quality.

- K is set as $K = K + 1$ and step 1 is returned to.

The above algorithm summarises the two cases that will be considered in this work; fixed and updating models.

Selection of Weighting Matrices

The reason behind adding the weighting matrices is to provide the best performance for the process while ensuring robustness to the system's uncertainties (Bristow, 2008). In this work Q is chosen as 1 and R is manipulated in such a way that a fast convergence is achieved and yet not compromising the stability of the system. One way of tuning the R is starting with a value and then increasing it while observing the response of the process to any sign of instability. In this work, values of $100I$ and $120I$ are selected for trial and the one finally used for this work is chosen based on the overall ILC performance. The first 15 batches are used.

The results of the selected values on ILC based on MLR, PCR and PLS are shown in figures 8, 9 and 10.

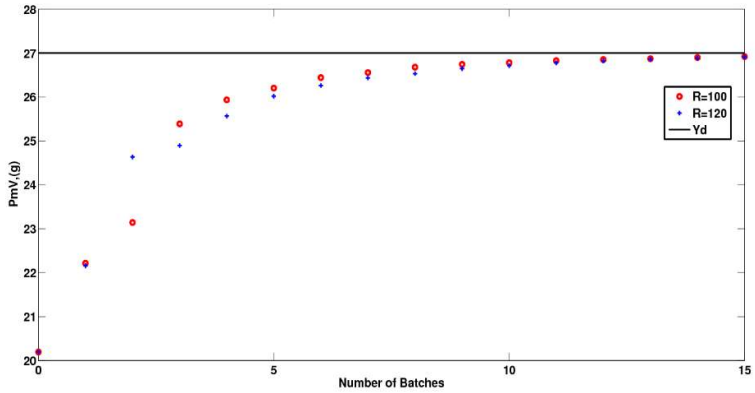


Figure 8: Different R values for ILC with MLR

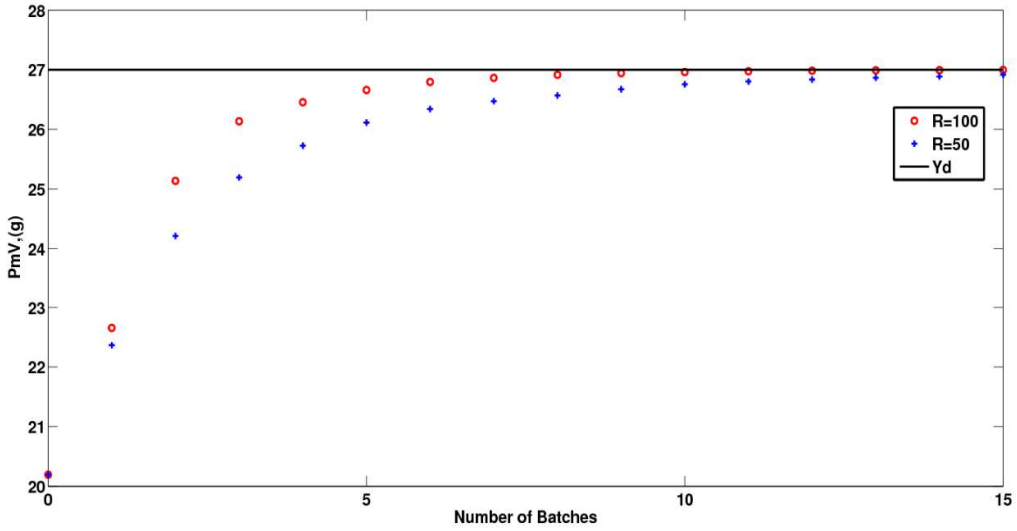


Figure 9: Different R values for ILC with PCR

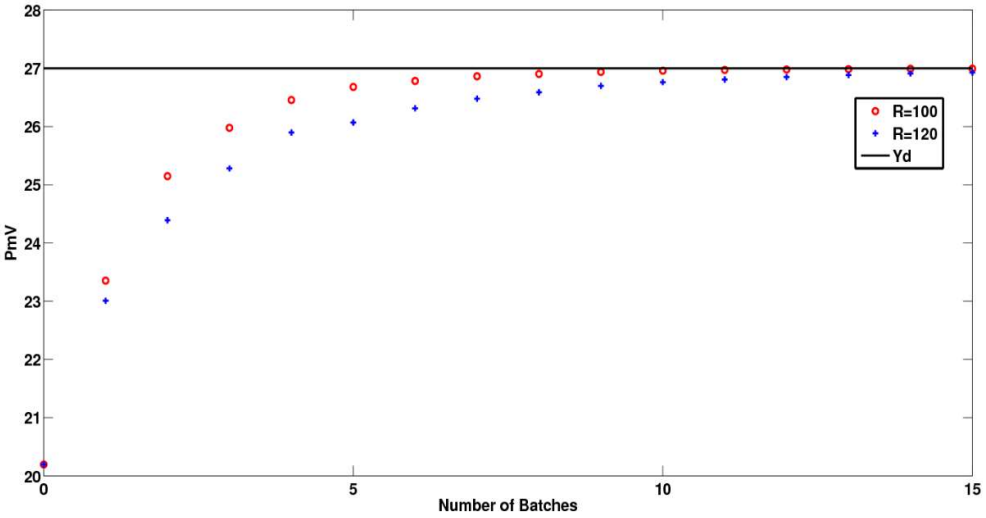


Figure 10: Different R values for ILC with PLS

In all the three figures, it can be seen that R=100I gives the fastest and most stable performance amongst the tested R values. Therefore, the weighting matrix R for this work is chosen as 100I

5. Results and Discussion

Batch to Batch ILC Performance based on Fixed and Updated MLR, PCR and PLS Model

The simulation is run for 30 batches for the regression models and a disturbance is introduced at the 18th batch to study the behaviour or performance of ILC on such condition. The glucose concentration of the feed, m is manipulated in this case. It is moved 5% above its normal operating condition and it is expected in such condition that the product quality declines. The plots of each ILC performance based on each regression method both for fixed and updated model parameters are shown in the subsequent figures.

The plot of the control policy is shown in figure 11. It can be seen that the control policy during different batch stages is highly correlated.

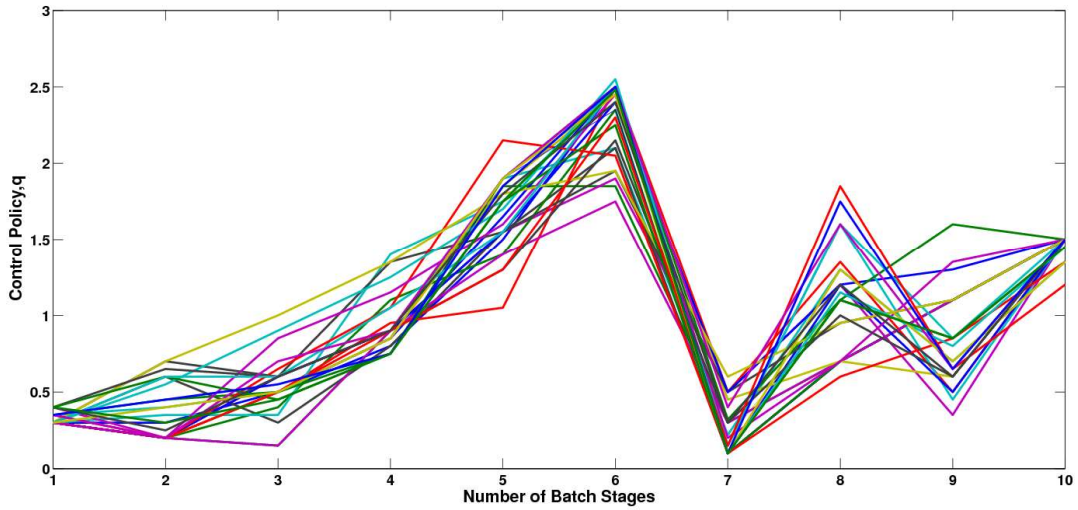


Figure 11: Plot of Historical batches of Control Policy

ILC Performance based on MLR Model

Firstly, the performance of the ILC without the addition of disturbance is considered. Figure 12 shows the plot of product quality against number of batches.

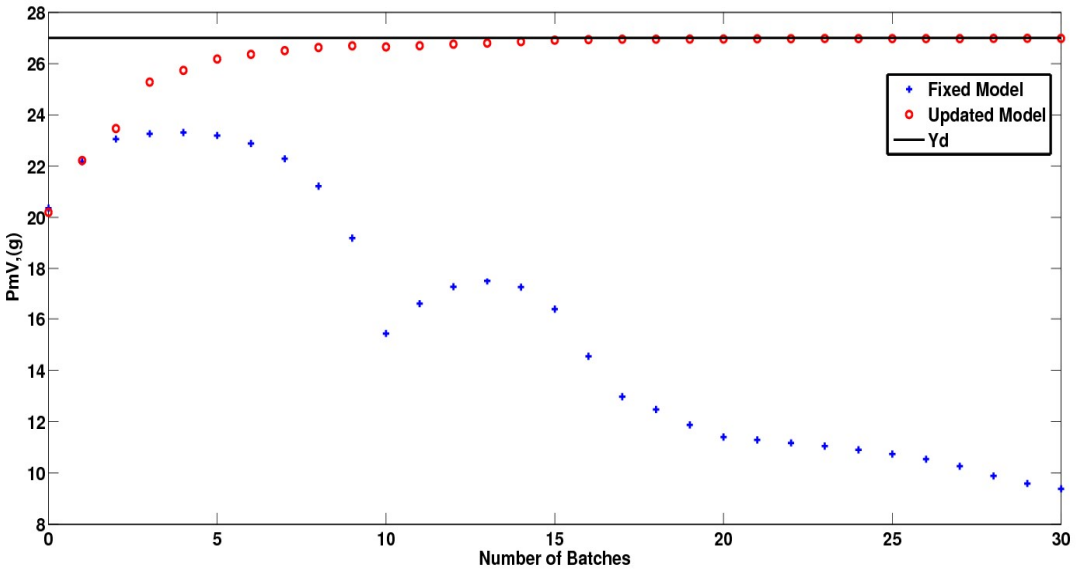


Figure 12: ILC Performance based on MLR Model without Disturbance

It can be seen from this plot that for the fixed MLR model parameters, the performance index makes an attempt to track the reference point for the first three batches. From the 4th batch, at 23g, the mass of secreted protein drastically moves downwards from batch to batch up to 10g. A decrease in the product is majorly caused by a decrease in the feed rate. A very low feed rate could lead to low cell growth rate and subsequently, to a poor product yield which is caused by an increased substrate concentration. However, with the updated model parameters, there is a significant improvement in the batch-to-batch performance of the ILC. The secreted protein moves from 20g and converges to the desired reference trajectory at about the 11th batch. In updating the model parameters after each batch run, the process is able to cope with unknown variation. Figure 13 shows the effect of disturbance on the performance of ILC. The reason behind this is to demonstrate the ability of ILC to overcome variations caused by unknown disturbances.

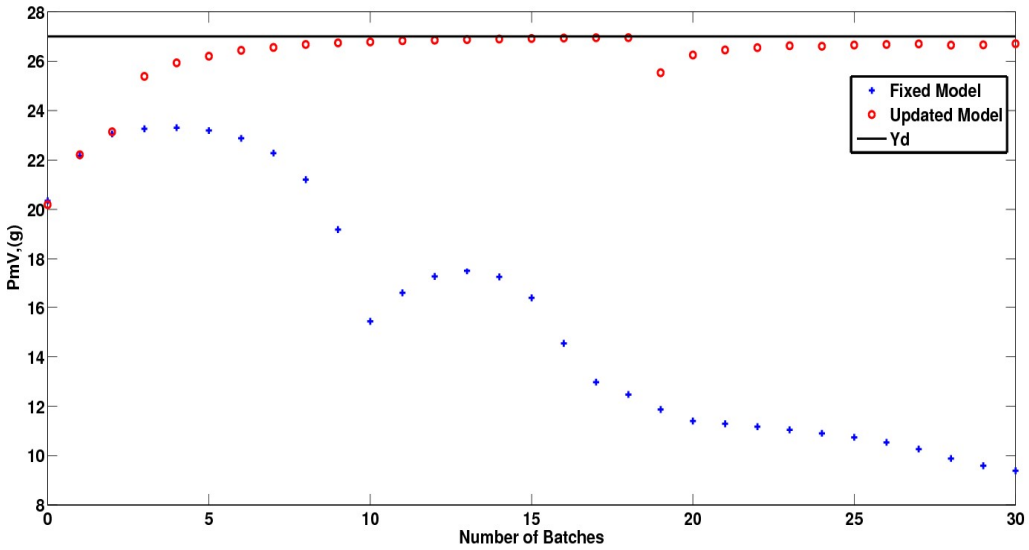


Figure 13: ILC Performance based on MLR Model with Disturbance

When the disturbance is introduced at the 18th batch, the product quickly moves away asymptotically in an opposite direction from the desired trajectory. However, from the 20th batch it approximately converges back to the desired trajectory. This is the case for the updated model but for the fixed model, the ILC performance is already deteriorated from the 4th batch. Thus, there is no better improvement in an attempt to track the desired product quality by updating the model parameters. The bad performance shown with the fixed model explains the inability of the MRL model to handle the issue of collinearity in the process. This is also reflected in the plot of the tracking error shown in figure 14. This is an alternative plot to the plots showing the evolution of product since it is the difference between the desired trajectory and the product. The tracking error fails to converge to zero for the fixed model but for the updated model the tracking error converges to the desired reference before and shortly after the introduction of disturbance.

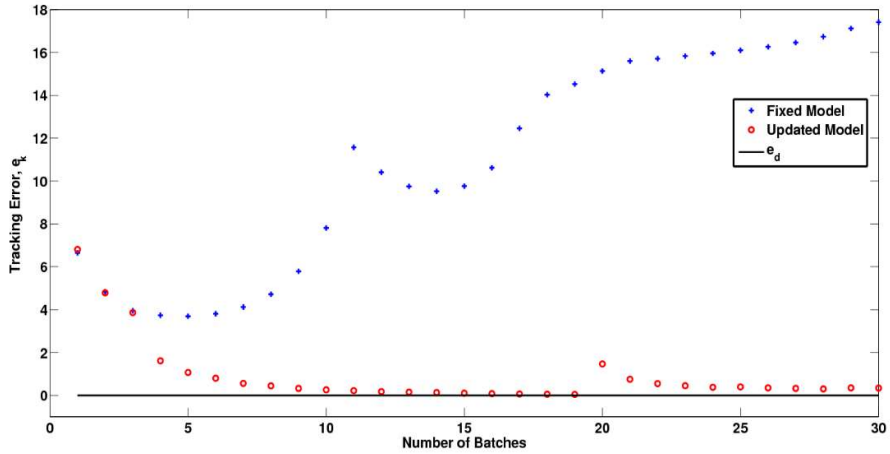


Figure 14: Evolution of Tracking Error for MLR Model

ILC Performance based on PCR Model

For PCR model, the number of principal components kept was determined through cross validation. The sum of squared error (SSE) using for both the training and testing data are evaluated and that which gives the lowest SSE the testing data is selected. 10 number of principal components were tried. Figure 15 shows the amount of variance explained in each principal component.

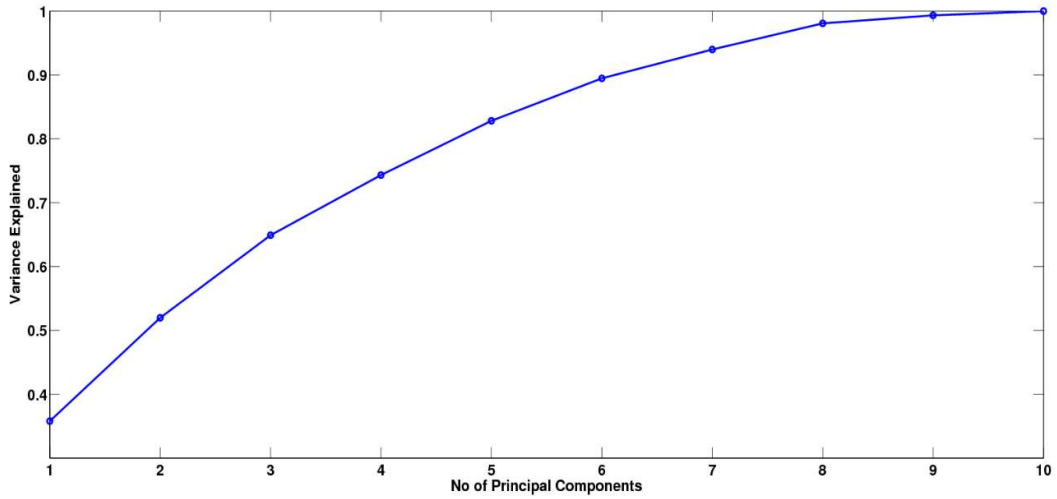


Figure 15: Variation explained by the number of principal components

It can be seen from this figure that the first four principal components can explain over 75% of the variation. Going further to determine the principal component with the minimum SSE on testing data, the SSE is plotted against the number of principal components. This is shown in figure 16.

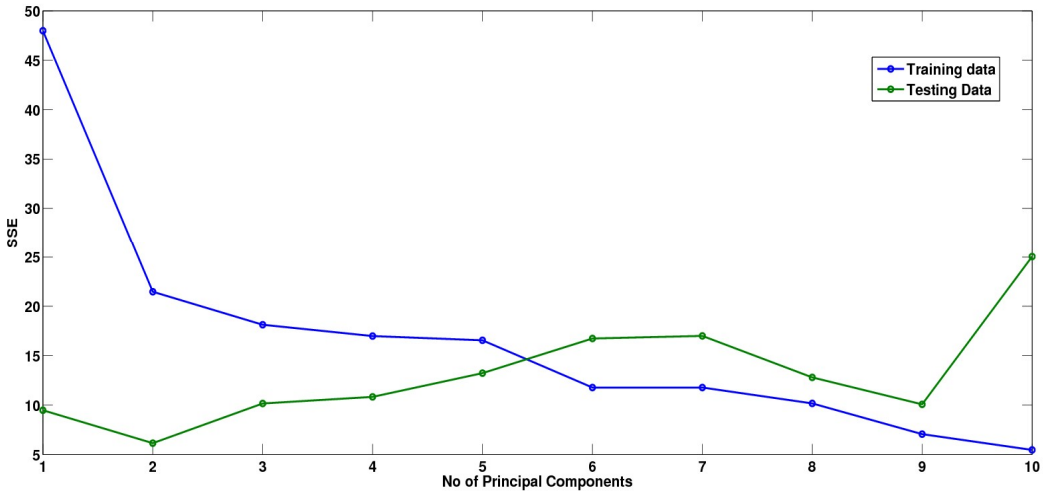


Figure 16: Sum of Squared Error of PCR model

It can be seen from this figure that that the principal component which gave the least sum of squares error on the testing data is 2 thus the number of principal components used is 2. In the case of the updated model, the number of principal components captured from batch to batch is shown in figure 17. It can be seen from this figure that the number of latent variables across each batch varies.

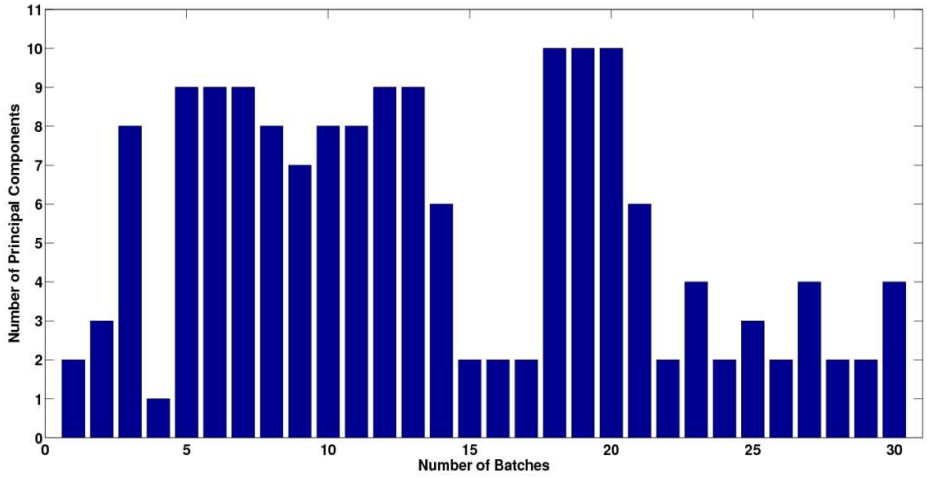


Figure 17: Numbers of Principal Components kept in the updated PCR model

Figure 18 shows the ILC performance of performance index against the number of batches without the presence of disturbance.

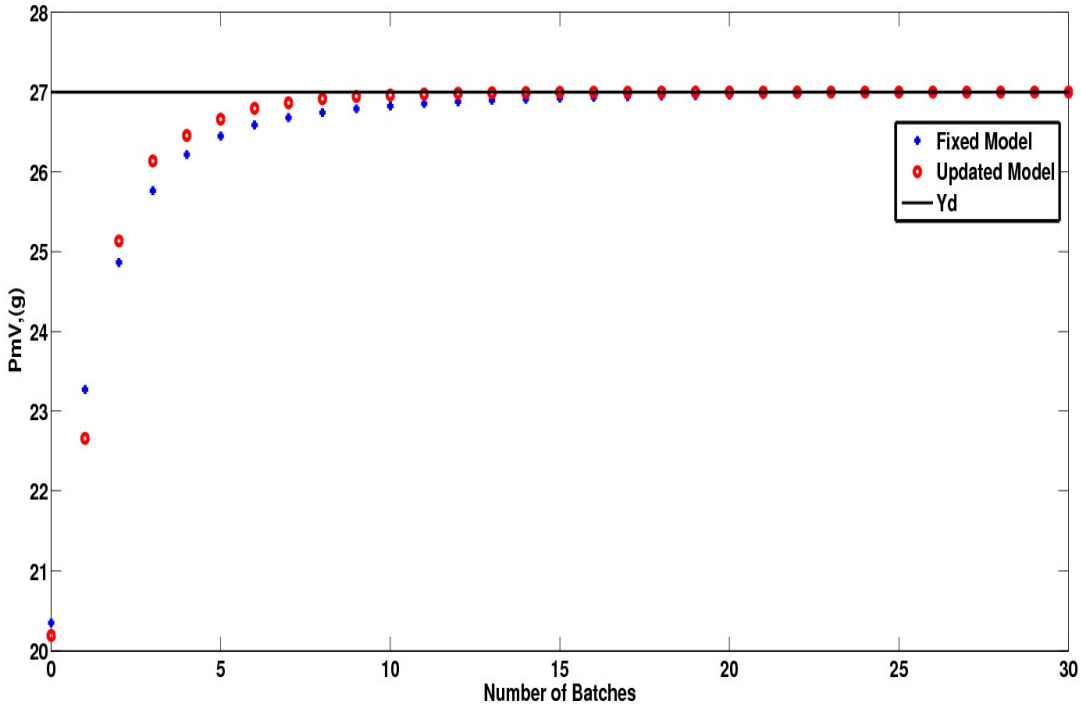


Figure 18: ILC Performance based on PCR Model without Disturbance

Unlike the ILC performance based on fixed MLR model, it can be seen in this figure that for the fixed model, the product quality was able to converge to the desired trajectory at the 12th batch. However, the updated model gave a better performance. It can be noticed that both start off at same pace for first batch but from the 2nd batch, the updated model moves faster than that of the fixed model. At the 3rd batch the product quality is at 25.8g and 26.2g for the fixed and updated model respectively. The fixed and updated models reach the desired convergence at the 11th and 8th batch respectively. The difference at the batch of convergence between these models might seem insignificant but what happens in the presence of disturbance gives a very distinct performance. Figure 19 shows the same performance but with introduction of disturbance. At the 18th batch, the disturbance is introduced. The product moves away from the desired trajectory at the 19th batch. For the updated model, the product quality increases from the 20th batch till it converges to the desired trajectory at the 25th batch. For the fixed model the

product quality fails to converge to the desired trajectory. This highlights the inability of fixed model to thrive in the presence of disturbances

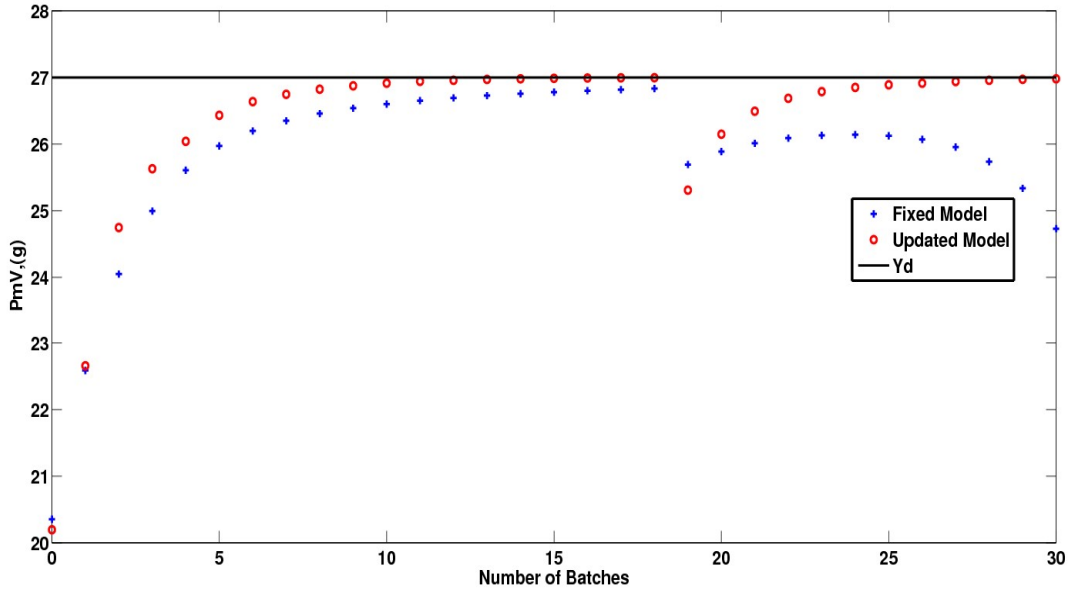


Figure 19: ILC Performance based on PCR Model with Disturbance

The tracking error plot is shown in figure 20. The tracking error from the updated model successfully converges to zero before and after disturbance as can be seen in the figure while that of the fixed model acts otherwise as it fails to converge in the presence of disturbance.

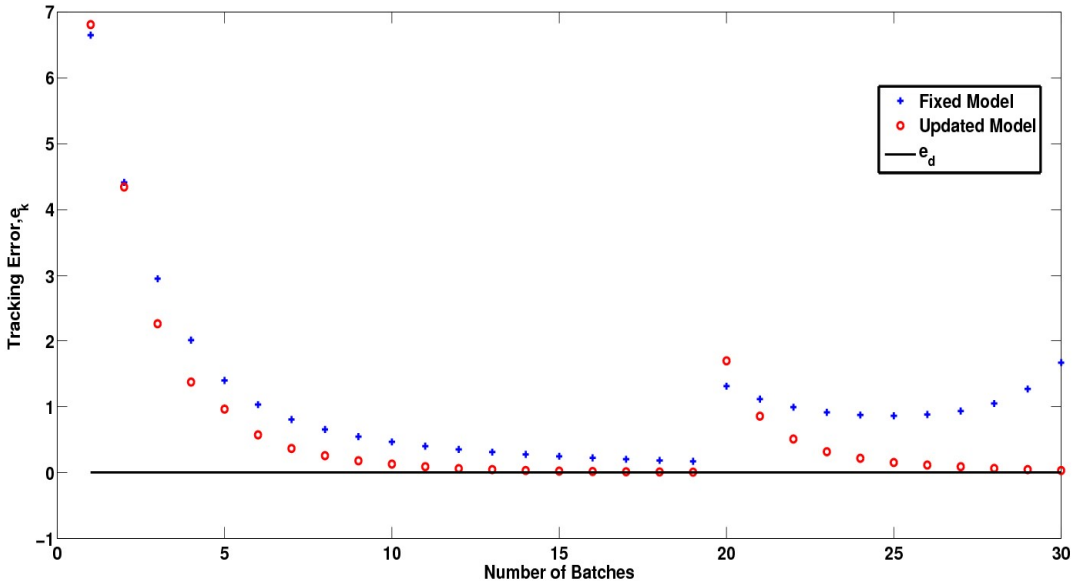


Figure 20: Evolution of Tracking Error for PCR model

ILC Performance based on PLS Model

For the fixed PLS model, the number of latent variables kept was determined through cross validation. Similarly, to PCR, 10 number of latent variables were tried. The Sum of Squared Error is plotted against the number of latent variables. This is shown in figure 21. From the figure, the one which gave the least sum of squares error on the testing data is 2 thus the number of latent variables used is 2.

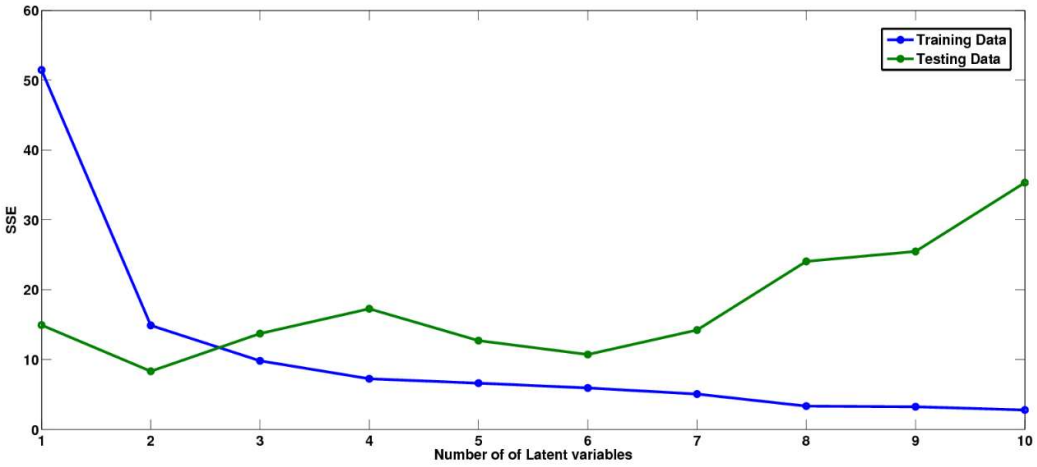


Figure 21: Sum of Squared Error of PLS model

For the updated model parameters, the number of latent variables kept was again determined through cross validation. The ones which gave the least sum of squares error on the testing data for each batch was kept. A bar plot of this is shown in figure 22. It can be seen from this figure that the number of latent variables across each batch varies.

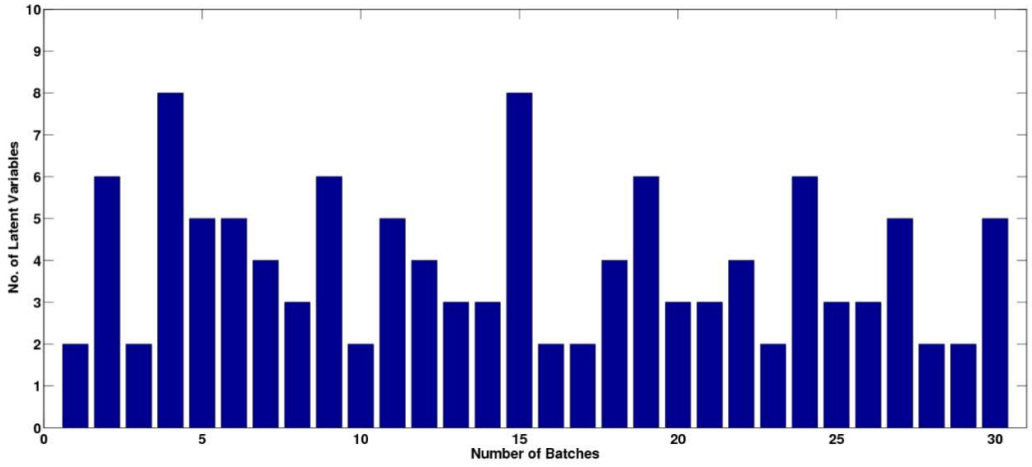


Figure 22: Numbers of Principal Components kept in the updated PCR models

Figure 23 shows the plot of performance index against the number of batches without disturbance.

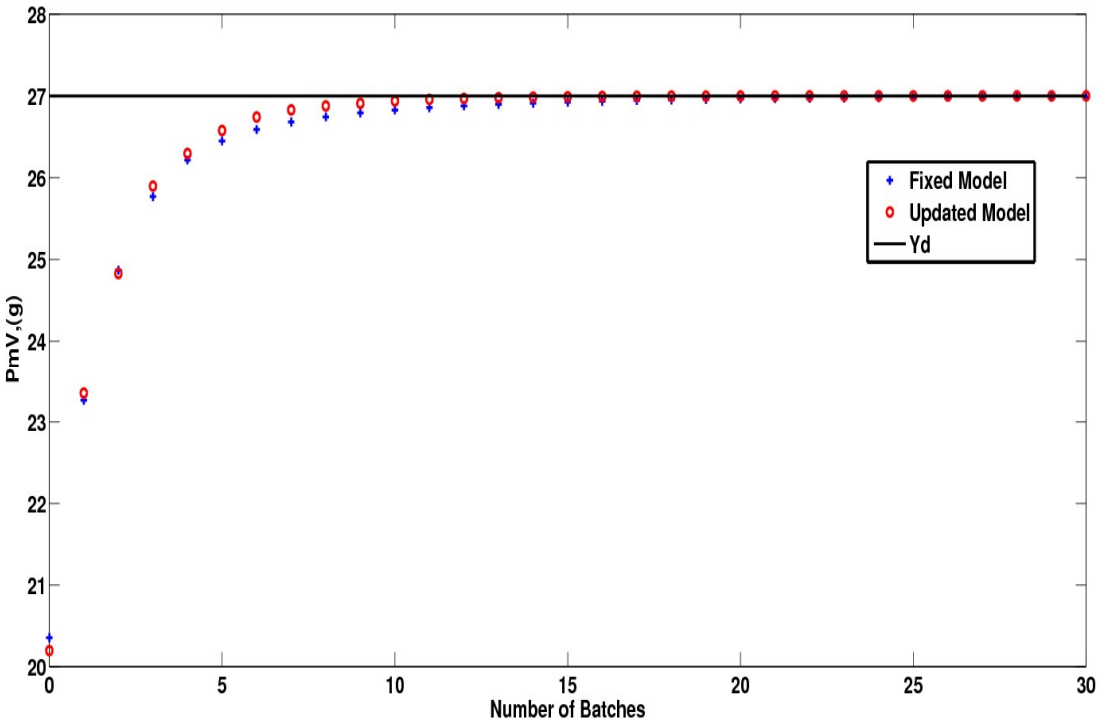


Figure 23: ILC Performance based on PLS Model without Disturbance

Similar to that in PCR, it can be seen in this figure that for the fixed model, the product was able to converge to the desired trajectory at about the 14th batch. However, the updated model gave a better performance. It can be noticed that both start off at same pace for the first two batches but from the 3rd batch, the updated model moves faster than that of the fixed model to the desired trajectory. The fixed and updated models reach the desired convergence at the 14th and 9th batch respectively.

Figure 24 shows the plot of performance index against the number of batches with disturbance.

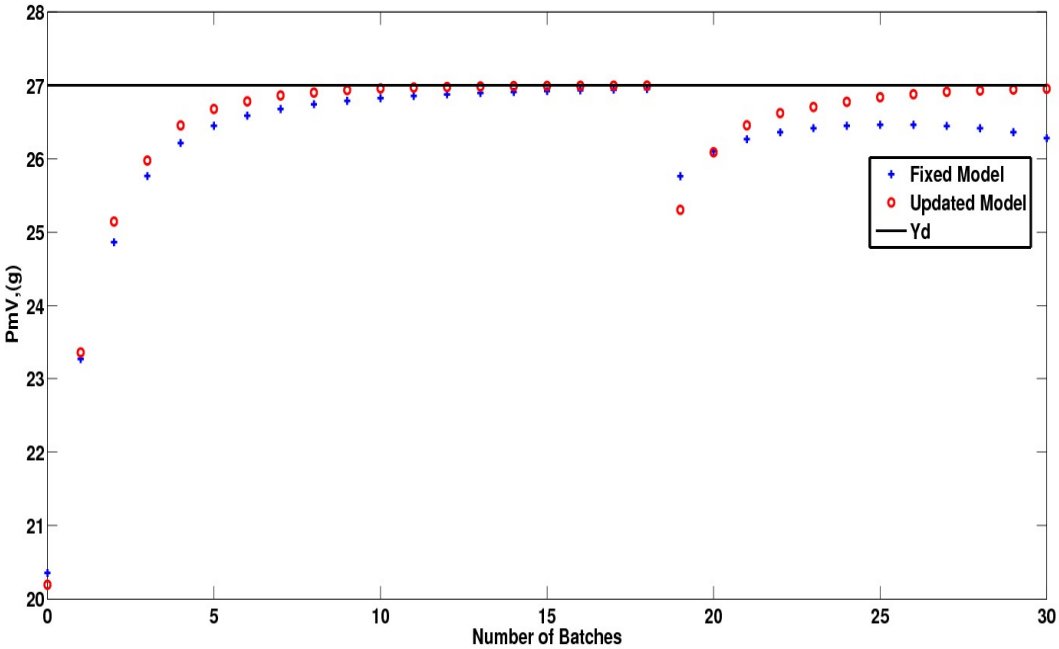


Figure 24: ILC Performance based on PCR Model with Disturbance

At the introduction of a disturbance at the 18th batch, the product moves away from the desired trajectory but

from the 18th batch the product moves from batch by batch to the desired trajectory at the 26th batch for the updated model. However, the performance index based on fixed model fails to converged to the desired trajectory. The tracking error plot is also shown in figure 25

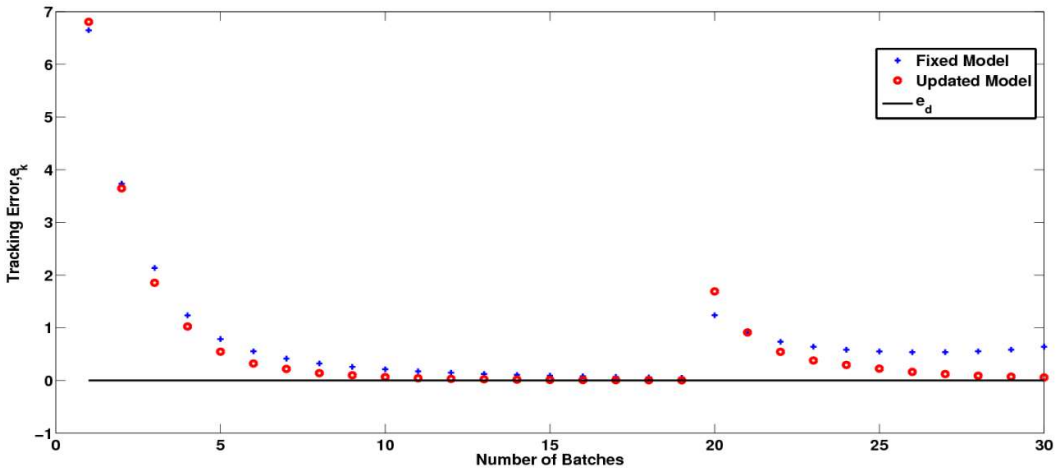


Figure 25: Evolution of Tracking Error for PLS Model

From this figure, it can be seen that the tracking error successfully converges to zero before and disturbance for the updated model. However, that of the fixed model acts otherwise as it fails to converge in the presence of disturbance.

Comparison of ILC Performance based on MLR, PCR and PLS Models

For the fixed model parameters, a comparative plot is shown in figure 26

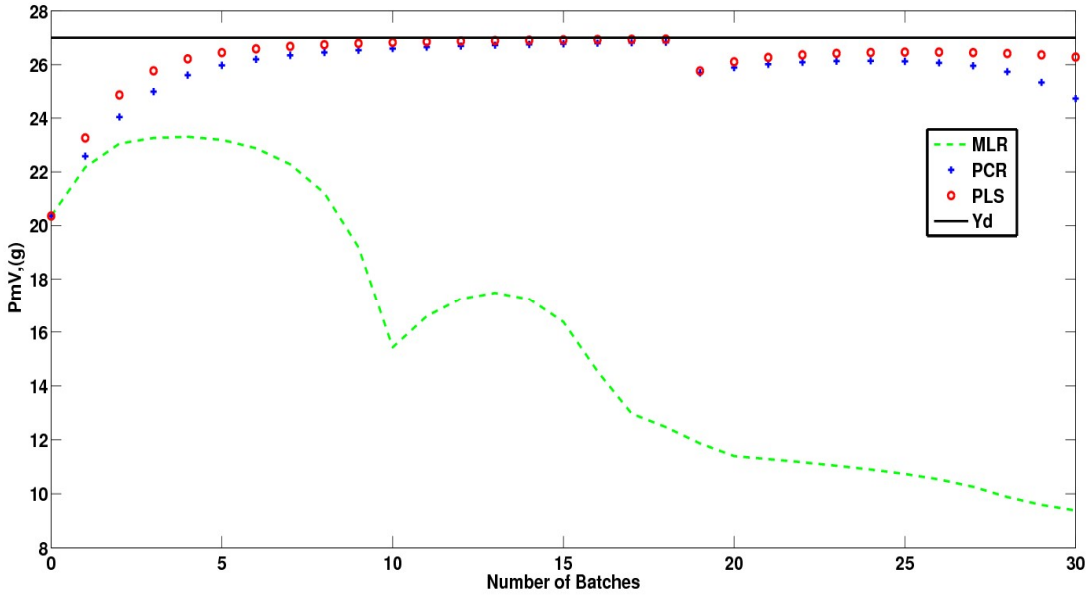


Figure 26: ILC Performance based on Fixed MLR, PCR and PLS Models

It can be seen from this figure that ILC based on all three regression models fail to reach the target after the introduction of disturbance. However, PLS and PCR gave a better performance as the product in each case was able to converge to the desired reference trajectory before the presence of a disturbance. It can also be seen from the figure that PLS model gives the best performance. A comparative plot is also shown for the updated models in figure 27.

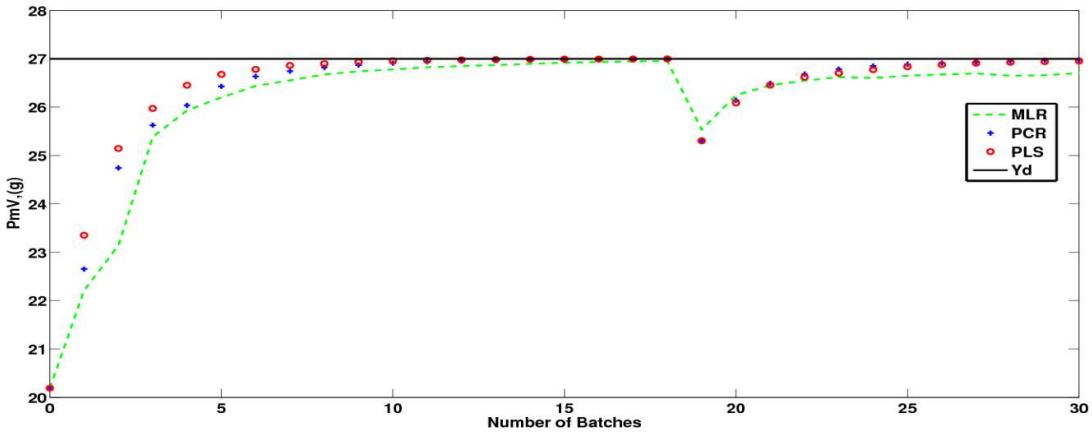


Figure 27: ILC Performance based on updated MLR, PCR and PLS Models

From this figure, the performance index based on the three models converged to the desired trajectory before disturbance with PLS giving the best performance followed by PCR and lastly MLR. However, at the introduction of disturbance at the 18th batch, ILC based on MLR model fails in achieving its objective of reaching the desired trajectory while PLS and PCR acts on the contrary. MLR is not ideal for process data with high collinearity. PLS and PCR models overcome this problem as can be seen in both figures. They are highly recommended when dealing with problem such as high data correlation.

Generally, it is obvious from these two plots that batch to batch updated ILC offers a better performance. By updating the model parameters, control policy and output after each batch the issue of the plant model mismatches and unknown disturbance is overcome.

Feed rate Profiles of Updated Model for Batch-to-Batch ILC

As earlier stated, the control variable in this work is the feed rate. This is manipulated by the controller such that the maximum amount of secreted protein is obtained. In other words, it is the pivot of the control mechanism. A brief look into the feed rates changes would give a clearer comprehension on ILC batch to batch performance shown in the previous figures. Since the ILC based on updated models gives a better performance, further analysis of its control profile for each of the regression methods is carried out. This analysis is based on changes in feed rate with and without the introduction of disturbance. Therefore, the first 17 batches are considered as without disturbance.

Control profile under Partial Least Squares model

After the use of PLS method, the feed rates for each batch against the batch time is plotted and shown in figure 28. It can be seen from this figure that as the batch time increases, the feed rate exponentially increases across each batch. However, at about a batch time of 9h, the feed rate drops significantly then increases to 1.5l/h for the rest of the batch time. A further look into the control profile with the introduction of disturbance is illustrated in figure 29. From this figure, it can be seen that the pattern remains the same. Better still, there is no significant difference between the feed rates from batch to batch. This is because the product quality quickly returns to tracking the desired trajectory after disturbance is introduced. This further shows that ILC based on PLS method is a good one.

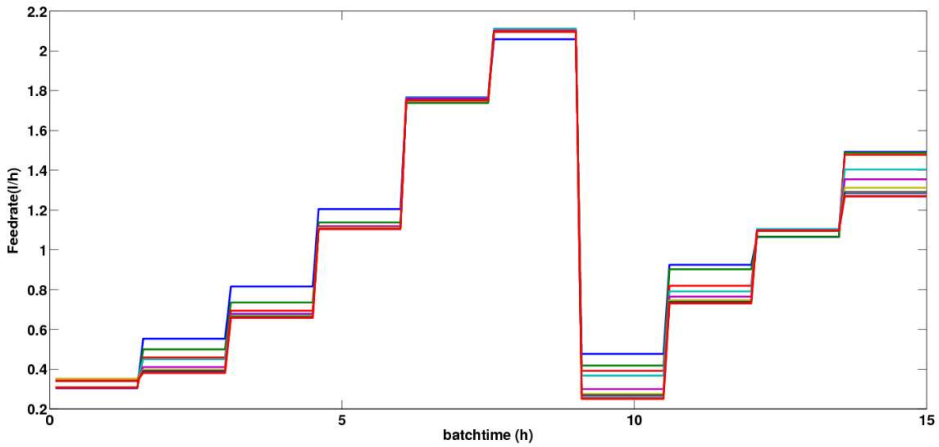


Figure 28: Control Profile based on PLS without Disturbance

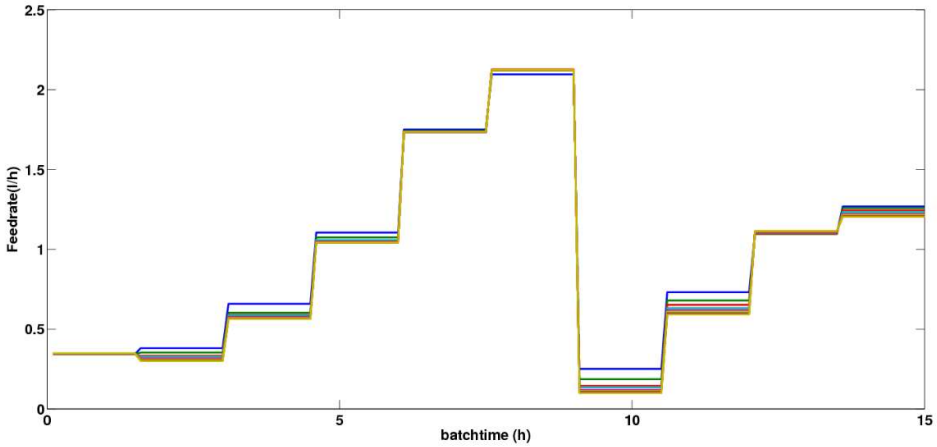


Figure 29: Control Profile based on PLS with disturbance

Control profile under Principal Component Regression Model

On applying PCR method, the feed rates for each batch against the batch time is plotted and shown in figure 30. This is similar to that seen in PLS. Figure 31 shows the control profile after disturbance is introduced. This also gives same pattern as that shown for PLS. However, those of PLS lie closer to each other.

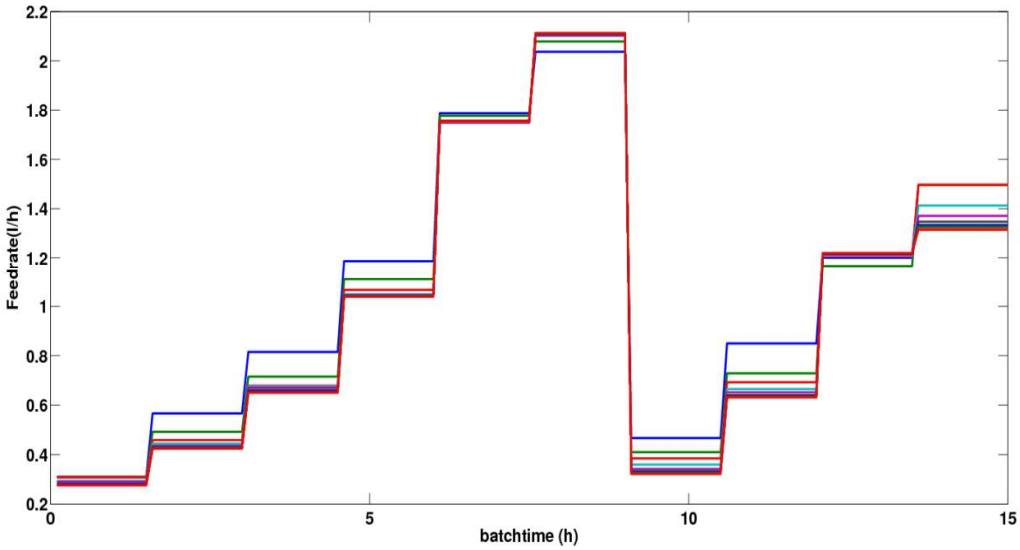


Figure 30: Control Profile based on PCR without Disturbance

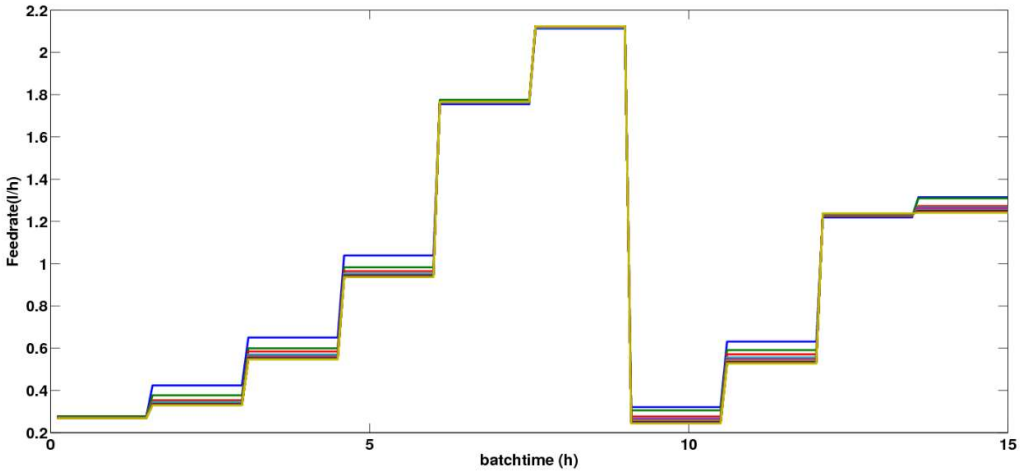


Figure 31: Control Profile based on PCR with disturbance

Control profile under Multiple Linear Regression Model

On applying MLR method, the feed rates for each batch against the batch time is plotted and shown in figure 32. From this figure, the control profile gives a somewhat different pattern compared to its counterparts. It maintains the same trend up till a batch time of 14h. From this batch time, some of the batches move in a descending manner as opposed to the normal trend. This depicts the inability of MLR to fully handle the process collinearity. It becomes worse with the introduction of disturbance. This is shown in figure 33. It can be seen in this figure that all the batches give an opposite trend from a batch time of 14h.

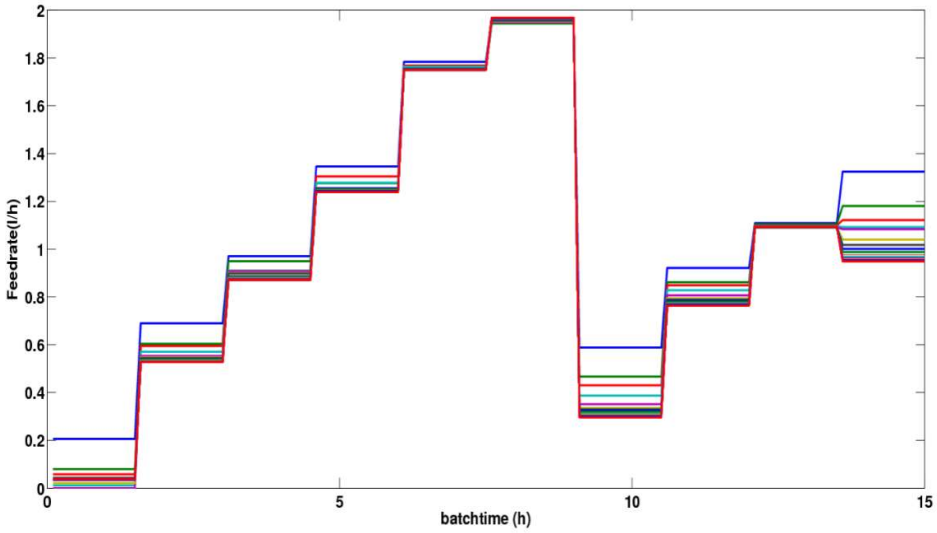


Figure 32: Control Profile based on MLR without Disturbance

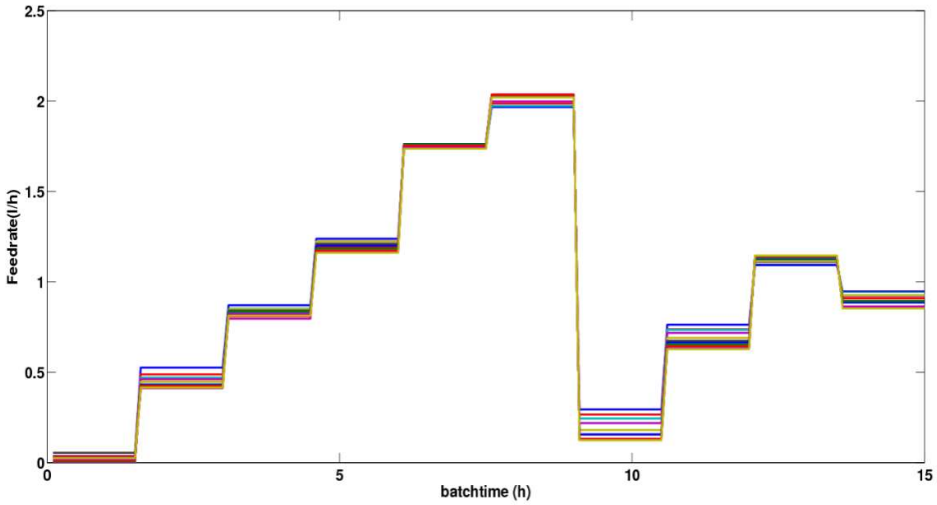
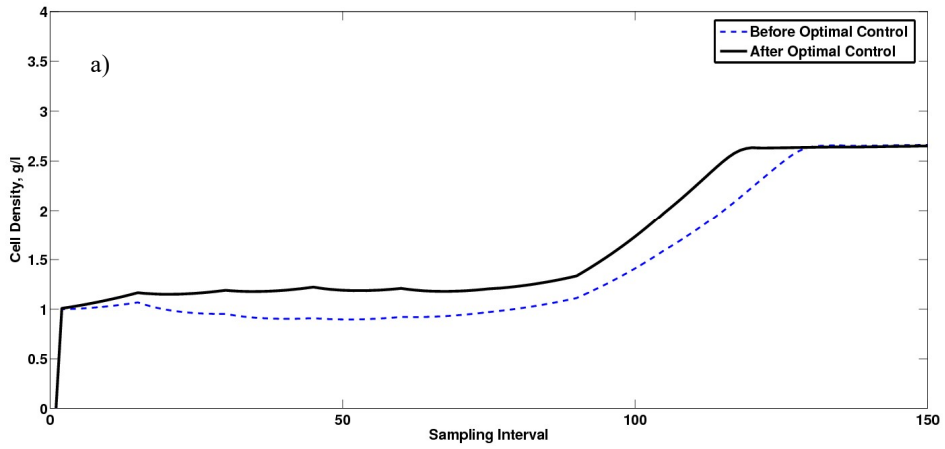


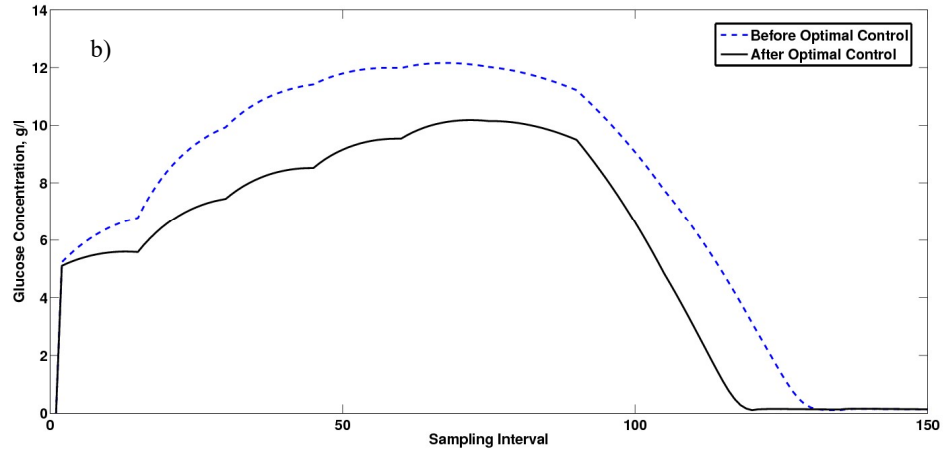
Figure 33: Control Profile based on MLR with Disturbance

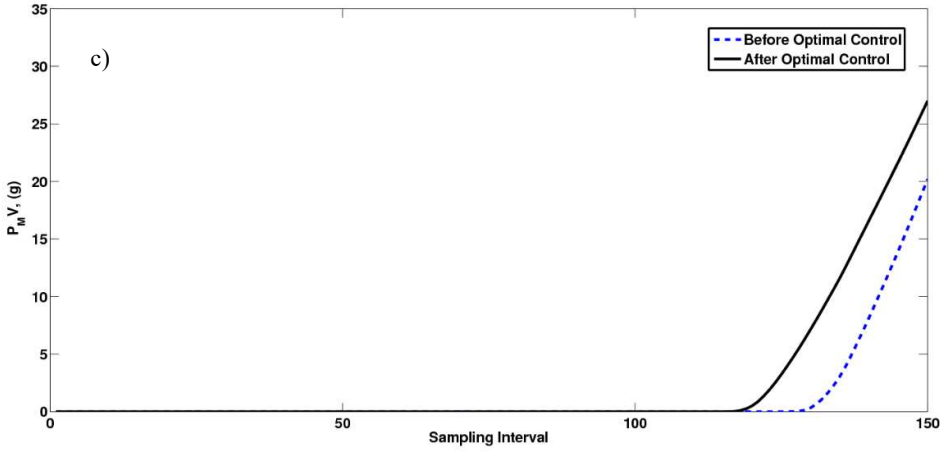
With the introduction of disturbance, it was seen that control profiles of PCR and PLS reserve their pattern after the introduction of disturbance unlike that of MLR. It is clear and evident that the performance of ILC based on updated PLS and PCR models are preferred options than that of MLR.

Comparison of State Variables before and after Optimal ILC

At the end of the fed batch fermentation process, it is expected that the maximum amount of secreted protein is obtained by determining the optimal control profiles. Figures 34a and 34b give a comparison of the state variable before and after optimization. Taking a closer look at figures 34a and b, it is seen that at the time when glucose diminishes the cell growth remains constant throughout the remaining batch time. It is also at this point that the product formation begins. It can be noticed that after optimization, the time at which the substrate diminishes reduces enabling a higher production of secreted and total protein before the end of batch time. This can be seen in figures 34c and 34d. The secreted and total protein is recorded at 20g and 26.5g before optimal control. However, after optimal ILC secreted and total protein is recorded at 27g and 33g.


a)
Before Optimal Control
After Optimal Control
Cell Density, g/l
Sampling Interval
0
0.5
1
1.5
2
2.5
3
3.5
4
0
50
100
150


b)
Before Optimal Control
After Optimal Control
Glucose Concentration, g/l
Sampling Interval
0
2
4
6
8
10
12
14
0
50
100
150


c)
Before Optimal Control
After Optimal Control
$P_M V_x$ (g)
Sampling Interval
0
5
10
15
20
25
30
35
0
50
100
150

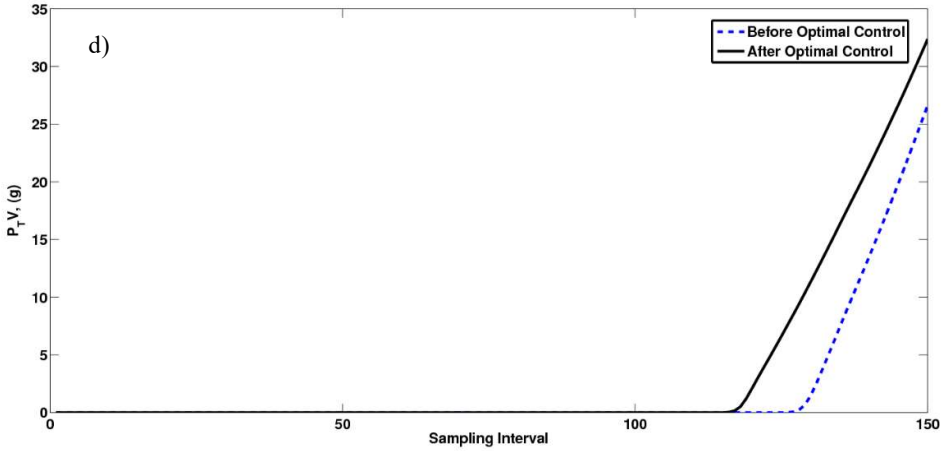


Figure 34: Profiles of State Variables

Figure 35 shows the optimal control profile based on PLS model. This follows the same trend with that in literature (Xiong and Zhang, 2003) obtained from mechanistic model and neural networks. A bit difference is observed in the maximum product quality obtained in using ILC and the mechanistic model because of the approximation of baseline control policy used in this work. It is worth noting that in using ILC for the case study a maximum product quality is obtained without violating the constraint placed on the feed rate profile.

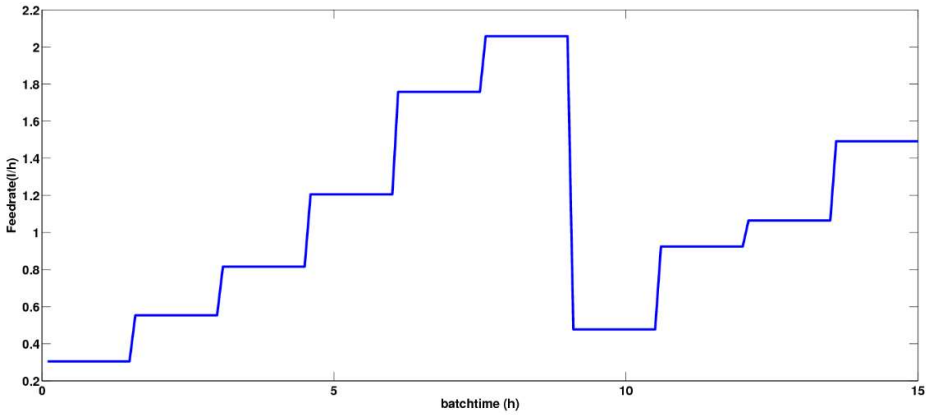


Figure 35: Optimal Control Profile

From the results above, it can be seen that the weighting matrices are important parameters in the overall performance of the controller. Therefore, careful selection of weighting matrices is considered in this work. In the light of that, batch to batch ILC performance using different values of R were compared and R was chosen as 100I while Q was set to 1. Amongst the three linear models, ILC based on PLS model gave the best performance followed by PCR. Optimal ILC based on MLR model gave unsatisfactory performance especially in the case of fixed model parameters. This can be explained by the high correlation of process data existing between batch stages. The general performance of batch-to-batch ILC using updated models gives an indication that the issue associated with model mismatches and unknown disturbances can be overcome. It is worth noting that the optimal control policy obtained from the updated model has been shown to be quite close to that obtained in using mechanistic model shown in section 3. This goes further to prove that ILC can be used as a simple alternative when mechanistic models become difficult to develop

6. Conclusion

In this work batch to batch Iterative Learning Control on fed batch fermentation process using linearised models has been considered. ILC uses the information from a previous of a batch process to improve the product quality from batch to batch. The tracking error and control input from the previous batch are used in generating the input for the current batch. This forms the basis for the optimal ILC law. The tracking error under the optimal ILC law is expected to converge to zero as it moves from batch to batch. In order to eliminate the non-linearity in the process, deviation of the input and output trajectories from their nominal trajectories were established and using

these perturbed variables, the linear models' parameters were then obtained using Multi Linear Regression (MLR), Principal Component Regression (PCR) and Partial Least Squares (PLS) method. In addressing the issues of model mismatches and unknown disturbances, the control policy was updated from batch to batch by using the previous batch as the reference batch. In addressing the issue of process collinearity between control policies at different stages in a batch, PLS and PCR were used to identify the linearised models.

The application of the proposed optimal ILC to a simulated fed batch reactor for the production of secreted protein gave satisfactory results based on tracking the desired trajectory and convergence of tracking error to zero. This gives an indication that ILC technique, a simple but yet an effective optimal control strategy can be used in eliminating the problems of plant model mismatches and process variations thus achieving optimal control in batch processes.

References

Aiba S., Shoda, M., and Nagatani M. (1968). "Kinetics of Product Inhibition in Alcohol Fermentation," Biotechnology and Bioengineering. Vol. 10, pp 845-846.

Arimoto S., Kawamura S. and Miyazaki F. (1984). "Bettering Operation of Robots by Learning". Journal of Robotic Systems. Vol. 1, no. 1, pp 123-140.

Cai Z. (2009). "Iterative Learning Control: Algorithm Development and Experimental Benchmarking". PhD thesis. School of Electronics and Computer Science, University of Southampton.

Cinar A., Parulekar S.J., Undey C. and Birol G. (2003). "Batch Fermentation: Modeling: Monitoring, and Control". Taylor & Francis Inc. United States. Pp. 6-7.

HENSON, M.A. (2006) "Exploiting cellular biology to manufacture high-value products", IEEE control systems magazine, vol. 26, pp. 54 – 62.

Jewarathnam, J. (2013). "Batch-to-batch Iterative Learning Control of a Fed-Batch Fermentation Process". PhD Thesis, School of Chemical Engineering and Advanced Materials, Newcastle University, UK. p 9. [Online]. Available at: https://theses.ncl.ac.uk/dspace/handle/10443/1901. Accessed on 22nd July, 2015.

Lee B. B., Tham, H. J. and Chan E. S. (2007). "Fed-batch Fermentation of Lactic Acid Bacteria to Improve Biomass Production: A Theoretical Approach". Journal of Applied Sciences, vol, 7, pp. 2211-2215.

Lee J., Lee S. Y., Park S., Middelberg A. P. J. (1999). "Control of fed batch fermentation". Biotechnology Advances, vol. 17, issue 1, pp 29-48.

Lee K.S. and Lee J.H. (1997). "Model predictive control for nonlinear batch processes with asymptotically perfect tracking." Computers and Chemical Engineering, vol. 21, pp. 873-879.

Moore K.L. (2006). "An introduction to iterative learning control." E ES 504/604A Seminar, Colorado School of Mines.

Park S. and Ramirez W.F. (1988). "Optimal production of secreted protein in fed-batch reactors". A.I.Ch.E. journal, vol. 34, pp. 1550-1558.

Van Impe J.F. and Bastin G. (1995). Optimal adaptive control of fed-batch fermentation processes. Control Engineering Practice, vol. 3, no. 7, pp. 939-954.

Xiong Z. and Zhang J. (2003). "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.

Yamanè, T., Shimizu S. (1984). "Fed-batch Techniques in Microbial Processes". Advances in Biochemical Engineering/Biotechnology, vol. 30, pp. 147-194.

.