Multivariate Batch to Batch Optimisation of Fermentation Processes Incorporating Validity Constraints

DOI:
10.1016/j.jprocont.2016.07.002

Document Version
Accepted author manuscript

Multivariate Batch to Batch Optimisation of Fermentation Processes Incorporating Validity Constraints

Carlos A. Duran-Villalobos

School of Electrical and Electronic Engineering, The University of Manchester, M13 9PL, UK

(duranvillalobos.carlos@gmail.com)

Barry Lennox

School of Electrical and Electronic Engineering, The University of Manchester, M13 9PL, UK

(barry.lennox@manchester.ac.uk)

David Lauri

School of Electrical and Electronic Engineering, The University of Manchester, M13 9PL, UK

(dalaupl@gmail.com)

Abstract: This paper presents an innovative optimisation technique, which utilises an adaptive Multiway Partial Least Squares (MPLS) model to track the dynamics of a batch process from one batch to the next. Utilising this model, an optimisation algorithm solves a quadratic cost function that identifies operating conditions for the subsequent batch that should increase yield. Hard constraints are shown to be required when solving the cost function to ensure that batch conditions do not vary too greatly from one batch to the next. Furthermore, validity constraints are imposed to prevent the PLS model from extrapolating significantly when determining new operating conditions. The capabilities of the proposed technique are illustrated through its application to two benchmark fermentation simulations, where its performance is shown to compare favourably with alternative batch-to-batch optimisation techniques.

Keywords: Partial Least Squares, Batch control, Batch-to-batch, Adaptive control, Validity constraints, Data-driven modelling.

1. INTRODUCTION

Batch processes, and fermentation systems in particular, tend to exhibit highly non-linear characteristics. These non-linearities can be introduced by changes that affect the process from one batch to the next, for example changes in raw material properties or equipment changes or may be a result of the complex dynamics of the chemical or biological materials [1]. These changes to the process are typically ignored by recipe-based systems in use on the majority of industrial fermentation plants and as a consequence there can be considerable variation in product quality from batch-to-batch. As regulatory authorities, such as the Food and Drugs Agency (FDA) introduce more stringent requirements, it is crucial that control systems are introduced that ensure product consistency despite variations in process characteristics [2]. Model based control systems are now used routinely to regulate continuous processes in the presence of process variability. However, as a consequence of the difficulties in obtaining a model that adequately describes the process dynamics, the application of such technology to batch processes is rare [3].

A variety of modelling techniques have been proposed to describe the dynamics of batch processes. These include mechanistic techniques [4] and linear and non-linear empirical techniques. Empirical models are typically easier to develop and more accurate for large-scale systems, where there is an incomplete understanding of the fundamental dynamics of the process. For batch processes, the most frequently applied empirical techniques are based on Multiway Partial Least Squares (MPLS). MPLS was originally designed to detect and locate abnormal conditions, such as sensor faults, within batch processes [5–8] and it is particularly well suited to modelling batch processes where product quality measurements are only available at the end of the batch. Such processes are the subject of this work.

In recent years there have been several studies that have integrated multivariate statistical models, such as MPLS, within control systems to regulate product quality and improve yield. Flores-Cerrillo and MacGregor [6] for example, integrated an MPLS model within a Model Predictive Control (MPC) strategy to regulate operating conditions within an emulsion polymerisation process to improve product consistency in the presence of disturbances. Wan et al [7] extended this approach and

demonstrated that if a disturbance model was used within the controller then control performance could be improved significantly. A similar approach was used in [8], where a constrained control strategy was applied to reduce product quality variation resulting from unmeasured disturbances in an exothermic chemical batch reactor.

Further studies in to the use of multivariate statistical models to regulate batch processes have been conducted by several research groups, including [9–13]. Whilst this work has been demonstrated to be successful, multivariate techniques are not without their limitations. For example, many multivariate methods assume fixed batch lengths, which can introduce problems. Whilst there exist many processes where batch lengths vary considerably the case studies considered in this work are fermentation processes where it is common for batch lengths to be fixed. The extension of multivariate statistical techniques to processes with variable batch lengths is an area of considerable interest; see for example [14], which involved the successful application of MPLS to an industrial process where batch length varied between 8 and 24 hours and [15] which proposed the use of dynamic time warping to allow MSPC techniques to be applied a polymerisation process.

In the studies described above, the control systems were applied to regulate end-point product quality by adjusting the manipulated variables in the process. In such systems, the conditions within the process should not differ considerably from one batch to the next and hence linear models are often sufficient to model the process dynamics. However, the focus of the present work was to maximise product quality or yield from one batch to the next, in a process known as *batch-to-batch optimisation*.

In batch-to-batch optimisation, a model of the process is used at the start of the batch to compute the trajectory of the manipulated variable (or Manipulated Variable Trajectory – MVT) over the full batch length that it is expected will ensure that the end-point product quality meets its target. This target may change from batch-to-batch to, for example, continuously increase yield. To maximise product quality, it is likely that the conditions within the batch process, and as a consequence the dynamics of the system, will change considerably. Linear techniques, such as MPLS will have difficulties in tracking

the non-linear characteristics of the process under such conditions and hence alternative techniques have been investigated for batch-to-batch optimisation. Lee [16,17], for example proposed a MPC technique which used characteristics of Iterative Learning Control (ILC) to achieve batch to batch optimisation. Their results showed a clear increase in yield from one batch to the next when applied to a jacketed semi-batch reactor. Similarly, in [18] an ILC strategy was formulated to ensure a tracking performance was achieved in a semi-batch chemical reactor. This strategy used a dynamic model based on energy balance equations and an MPC based optimization strategy. Another example [19] consisted of using a sliding mode control scheme to increase the growth rate in a fermentation process. The primary distinction with this approach was that it required minimum knowledge of the on-line process parameters by relying on a partial state feedback law derived from a reference model. The sliding mode scheme was then used to overcome model uncertainties.

Despite its limitations attempts have been made to incorporate MPLS within batch to batch optimisation schemes. In [20], for example, the authors proposed and successfully implemented a batch to batch optimisation technique that used a hybrid model to predict the final particle size distribution (PSD) in a cobalt oxalate synthesis process. The hybrid model was a combination of a simplified first principle model and an MPLS model; the latter being used to correct errors in the mechanistic model. An alternative approach, proposed by Camacho [21], involved using an evolutionary optimisation technique that identified MVTs that would improve product yield in the forthcoming batch. The MVTs were determined by consideration of the gradient identified from an MPLS model that described the dynamic characteristics of the batch. The MPLS model was adapted from one batch to the next. The proposed technique led to a significant increase in yield when applied to a simulated fermentation process and compared very favourably with knowledge based techniques.

An important advantage of the MPLS based optimisation approach proposed in [21] is its low computational complexity. However, a limitation with the approach is that the technique does not consider that the model is only valid around the operating conditions used to calibrate the model. In the technique proposed in this paper, the validity of the MPLS model is considered as the new MVT is

determined. This ensures that the identified MVT will not upset the process considerably from one batch to the next, ensuring a gradual and more robust change to process conditions from one batch to the next.

A number of techniques have been suggested for ensuring the validity of PLS models. These techniques have typically been formulated to provide confidence limits on the predictions made by the models. For example, in [22] a method was proposed for determining confidence limits for PLS estimates that used a combination of the Mean Squared Error (MSE) of the PLS model and the value of Hotelling's T^2 statistic. The limits were combined with Squared Prediction Error (SPE) charts in [6] to produce an integrated technique for quality prediction, fault diagnosis and monitoring of batch processes.

Confidence intervals were also used in [23], where an innovative Latent Variable Model Predictive Control (LVMPC) strategy for continuous processes was presented. The authors included the use of normalized restrictions in the cost function to avoid the tuning of a weighting factor used in [6]. In addition, a quadratic error term for the model, frequently employed in monitoring and fault diagnosis, was included in the cost function to ensure the model remained valid. This work was extended in [24] where the validity of the model, measured using both the T^2 and Square Prediction Error (SPE) statistics were introduced as a hard constraint within the control system.

The work in this article formulates an alternative batch-to batch optimisation approach that presents a QP problem that is solved in the MVT space. This is in contrast to alternative formulations found in the literature which solve the QP problem in the Latent Variable (LV) space [6,7,9,24].The drawback of the LV approach is that once the optimized points are found, it is necessary to compute the real MVT by inverting the PLS model, which can cause actuation changes that are detrimental to the yield if the PLS model is not sufficiently constrained. In addition, the proposed design uses validity restrictions inside the MVT optimization to limit the solution of the QP problem to the region within

which there is confidence in the predictions made by the PLS model. Similar validity restrictions have been used before for MPC in the literature [11,23], but not to batch-to-batch optimization.

The work presented in this paper firstly proposes a novel batch-to-batch optimisation technique, which adapts a MPLS model from one batch to the next so that it is able to track the conditions within the process. This model is then used within a control system that iteratively adjusts the MVT to increase yield of the batch process. The benefits of introducing validity constraints within the control system are then demonstrated through its application to two benchmark simulation studies.

Section 2 of this paper formulates the mathematical theory and control methodology that is used in the proposed approach. Section 3 then describes the two benchmark batch fermentation simulations that the proposed batch-to-batch control technique is applied to. The results obtained using these simulations are then discussed and compared with those obtained using previously proposed techniques. Finally, the conclusions from the work are presented in section 4.

2. METHODOLOGY

This section provides a brief overview of MPLS modelling and the approach by which an MPLS model can be integrated within a batch-to-batch optimisation strategy. The nomenclature used in this section is summarised in Appendix A. Further details of PLS and MPLS can be found in [3] and [22] respectively.

2.1 PLS Modelling

PLS is a multivariate statistical technique often used to approximate systems where correlated measurements create difficulties when more traditional regression techniques, such as ordinary least squares are applied. In PLS, a multiple linear regression model is found by projecting predictor, $\boldsymbol{X}$, and response, $\boldsymbol{Y}$, variables into a new space, known as the Latent Variable (LV) space. The LV space is determined such that a limited set of orthogonal vectors A explains the maximum covariance between $\boldsymbol{X}$ and $\boldsymbol{Y}$. The dimension A of the LV space is usually selected by cross-validation to ensure that the

resulting model provides a robust prediction of $\boldsymbol{Y}$ [25]. PLS is described mathematically in equations 1 and 2.

$$\boldsymbol{X} = \boldsymbol{T}\boldsymbol{P}^{\boldsymbol{T}} + \boldsymbol{E} \quad (1)$$

$$\boldsymbol{Y} = \boldsymbol{T}\boldsymbol{Q}^{\boldsymbol{T}} + \boldsymbol{F} \quad (2)$$

The predictor matrix, $\boldsymbol{X}$ is decomposed into the scores and loading matrices, $\boldsymbol{T}$ and $\boldsymbol{P}$ respectively. The vectors in the matrix $\boldsymbol{T}$ contain the value of each sample in the LV space, while the vectors in the matrix $\boldsymbol{P}$ contain the projection of each column of $\boldsymbol{X}$ in the LV space. The vectors in the residual matrix, $\boldsymbol{E}$ represent the difference between the decomposed and original variables and similarly, the response matrix $\boldsymbol{Y}$ is decomposed into the matrices of scores $\boldsymbol{T}$**,** loadings $\boldsymbol{Q}$ and residual matrix, $\boldsymbol{F}$. In this work it is assumed that measurements of the response variable are only available at the end of the batch. In other words, on-line measurements of the quality variable are not available within the batch. This is typical for many batch processes, particularly fermentation systems.

A complication introduced with batch processes is that the measurements are typically stored in a 3-dimensional data array, with dimensions corresponding to measured variables (J), time intervals (K) and batch number (I). Before PLS can be applied to such data it is necessary to transform it in to a 2-dimensional matrix. In [22] an unfolding technique referred to as Unfolded or MPLS, was proposed and is now used routinely for modelling batch processes. The result of the MPLS transformation is shown in Equation 3 and Figure 1.

$$\boldsymbol{X_{3D}} \in \mathbb{R}^{I \times J \times K} \rightarrow \boldsymbol{X_{2D}} \in \mathbb{R}^{I \times JK} \quad (3)$$

2.1.1 Model Adaptation

The purpose of the MPLS model in this work is to track the dynamics of a batch process as the operating conditions are varied from batch-to-batch and to then use this model to optimise the process. As MPLS is a linear technique it will be unable to accurately represent the dynamics of the process over different operating conditions and it will therefore be necessary for the model to adapt to the localised batch conditions. In this work, the adaptive mechanism proposed in [25] was used to update

the MPLS model at the end of each batch. This technique uses a forgetting factor, λ, to ensure that the model forgets the behaviour of historical batches but remembers the most recent batches. The inclusion of the forgetting factor is essential as without it linear MPLS would not be able to track the changing characteristics of the batch. To adapt the MPLS model, the $\boldsymbol{X}$ and $\boldsymbol{Y}$ matrices are appended with data from the previous batch, as shown in equation 4, where $\boldsymbol{x}_k$ and $\boldsymbol{y}_k$ are vectors containing the trajectories of the predictor and response variables for batch k and a new model is then identified by applying MPLS to these matrices. Note that to forget past batches, λ needs to be positive and smaller than 1. In this paper λ is defined as a scalar, but it could be defined as a weighting vector with smaller values associated with the oldest batches and increasing values for the more recent batches.

$$\boldsymbol{X_k} = \begin{bmatrix} \lambda \boldsymbol{X_{k-1}} \\ \boldsymbol{x}_k \end{bmatrix} \; and \; \boldsymbol{Y_k} = \begin{bmatrix} \lambda \boldsymbol{Y_{k-1}} \\ \boldsymbol{y}_k \end{bmatrix} \tag{4}$$

A drawback with this approach to adapting the MPLS model is that the size of the $\boldsymbol{X}$ and $\boldsymbol{Y}$ matrices will increase with each batch, consequently increasing the computational and storage requirements. To overcome this drawback, [26] showed that the same model is obtained if PLS is applied to the matrices defined in equation 5. Applying PLS to these matrices has the advantage that the size of $\boldsymbol{X}$ and $\boldsymbol{Y}$ remains unchanged and hence the computational and storage requirements are not as demanding. Once again, the matrices will be updated at the end of each batch; this is described in further detail in Section 2.4.

$$\boldsymbol{X_k} = \begin{bmatrix} \lambda \boldsymbol{P_{k-1}^T} \\ \boldsymbol{x}_k \end{bmatrix} \; and \; \boldsymbol{Y_k} = \begin{bmatrix} \lambda \boldsymbol{Q_{k-1}^T} \\ \boldsymbol{y}_k \end{bmatrix} \tag{5}$$

When selecting a suitable value for λ, it is necessary to consider the number of batches of data, N, that are to be *remembered* by the model. The approximate relationship between λ and N is given by equation 6 [25]. The forgetting factor λ must be chosen such that the number of batches N is relevant to the conditions around which the process is currently operating.

$$N = \frac{1}{1-\lambda} \tag{6}$$

Where $0 < \lambda \leq 1$

If *N* is chosen to be too large, then the model will not adapt fast enough to follow substantial changes in process dynamics. In contrast, if *N* is chosen to be too small, then the model will not consider enough past batches to obtain a useful model. In this work, *N* was specified to be equal to 3, which gave acceptable results. It is acknowledged that the most suitable value of *N* is likely to be problem dependent. However, in extensive testing it was found that the results were not improved significantly by increasing *N*.

2.2 MVT Optimisation

Following the identification of the MPLS model, it was used to calculate a suitable MVT for the forthcoming batch. This MVT was determined using a similar approach to that proposed in [6]. However, in this work the cost function was formulated in the real MVT space, rather than in the latent variable (LV) space. The reason for this is that the constraints in the LV space can produce erratic control behaviour[27].

The cost function used in this work to identify the MVT is presented in Equation 7.

$$\min_{\Delta \boldsymbol{u}} \left[\left(\hat{\boldsymbol{y}} - \boldsymbol{y}_{sp} \right)^T \left(\hat{\boldsymbol{y}} - \boldsymbol{y}_{sp} \right) + \Delta \boldsymbol{u}^T \boldsymbol{M} \Delta \boldsymbol{u} \right] \quad s.t. \begin{cases} \hat{\boldsymbol{y}} = \boldsymbol{t_k} \boldsymbol{Q}^T \\ \boldsymbol{lb} \leq \boldsymbol{u_n} + \Delta \boldsymbol{u} \leq \boldsymbol{ub} \end{cases} \tag{7}$$

Where $\Delta \boldsymbol{u}$ is the adjustment that is to be made to the MVT from the previous batch to minimise the error between the desired and predicted end-point quality, $\boldsymbol{y}_{sp}$ and $\hat{\boldsymbol{y}}$ respectively. The diagonal matrix of weights $\boldsymbol{M}$ is used to moderate the change in the manipulated variable throughout the batch.

The MVT is equal to the sum of the nominal value, $\boldsymbol{u_n}$, and $\Delta \boldsymbol{u}$ and is constrained by physical limitations. These constraints are represented by the lower and upper bound vectors, $\boldsymbol{lb}$ and $\boldsymbol{ub}$

respectively. Finally, the score vector $\boldsymbol{t_k}$ for the new batch, k, can be obtained using the projection weight matrix $\boldsymbol{V}$ applied to the predictors $\boldsymbol{X}$ as shown in equations 8 and 9.

$$\boldsymbol{T} = \boldsymbol{XV} \tag{8}$$

$$\boldsymbol{t_k} = \boldsymbol{x_k V} \tag{9}$$

Where $\boldsymbol{x_k}$.is given as follows:

$$\boldsymbol{x_k} = [\boldsymbol{x_p} \quad \boldsymbol{u_n} + \Delta\boldsymbol{u} \quad \boldsymbol{x_f}] \tag{10}$$

Where $\boldsymbol{x_p}$ represents the initial conditions of the process variables in the batch. Solving equation 7 requires predictions of the end-point quality to be made, which itself requires the future values of all the process variables, $\boldsymbol{x_f}$. As the future process measurements are unknown at the start of the batch, missing data algorithms were used to estimate their values. Several missing data estimation techniques have been proposed [13,28]. However, the most suitable technique for these types of problem, as demonstrated by [13] and suggested in [6] is the Projection to the Modal Plane (PMP) method [27].

The PMP method estimates the score vector by regressing the known variables on the plane defined by the loading matrix, $\boldsymbol{P}$. The score can then be estimated by regressing the vector of input variables $\boldsymbol{x_p}$ and the nominal values of the MVT $\boldsymbol{u_n}$ into the plane defined by the loading matrix $\boldsymbol{P}$ as shown in equation 11.

$$\widehat{\boldsymbol{t_k^T}} = \left(\boldsymbol{P_p^T P_p}\right)^{-1} \boldsymbol{P_p^T} [\boldsymbol{x_p} \quad \boldsymbol{u_n}]^T \tag{11}$$

Where the matrix of known loadings $\boldsymbol{P_p^T}$ contains the rows of $\boldsymbol{P^T}$ that account for the known vectors of predictors ($\boldsymbol{x_p}$ and $\boldsymbol{u_n}$), and the matrix of future loadings $\boldsymbol{P_f^T}$ contains the rows which are related to the unknown variables ($\boldsymbol{x_f}$). The estimated score $\widehat{\boldsymbol{t_k}}$ can then be used to obtain an estimate of the future measurements as shown in equation 12.

$$\boldsymbol{x_f} = \widehat{\boldsymbol{t_k}} \boldsymbol{P_f^T} \tag{12}$$

The effect that the change $\Delta \boldsymbol{u}$ has on the future estimates within $\boldsymbol{x_f}$ must also be considered. To do this, the estimated score $\widehat{\boldsymbol{t_k}}$ in equation 11 can be replaced as shown in equation 13.

$$\widehat{\boldsymbol{t_k}} = [\boldsymbol{x_p} \quad \boldsymbol{u_n} + \Delta \boldsymbol{u}]\left(\boldsymbol{P_p^T P_p}\right)^{-1}\boldsymbol{P_p^T} \tag{13}$$

The effect of the PMP estimation can be included in the QP formulation by substituting equation 13 into equation 7, as shown in equation 14.

$$\widehat{\boldsymbol{y}} = [\boldsymbol{x_p} \quad \boldsymbol{u_n} + \Delta \boldsymbol{u} \quad \widehat{\boldsymbol{t_k}}\boldsymbol{P_f^T}]\boldsymbol{V}\boldsymbol{Q}^T \tag{14}$$

The variables that are dependent on changes in the MVT can then be separated from equation 14 as shown in equation 15. In this equation the matrix of loadings $\boldsymbol{P_u^T}$ only considers the rows of $\boldsymbol{P^T}$ that correspond to the MVT. Similarly, the matrix of weights $\boldsymbol{V_u}$ accounts only for the rows corresponding to the MVT, and the matrix of weights $\boldsymbol{V_f}$ accounts only for the rows corresponding to $\boldsymbol{x_f}$.

$$\begin{aligned}\widehat{\boldsymbol{y}} = \underbrace{\left[\boldsymbol{x_p} \quad \boldsymbol{u_n} \quad [\boldsymbol{x_p} \quad \boldsymbol{u_n}]\left(\boldsymbol{P_p^T P_p}\right)^{-1}\boldsymbol{P_p^T P_f^T}\right]}_{\boldsymbol{\eta}}\boldsymbol{V}\boldsymbol{Q}^T \; + \\ \Delta \boldsymbol{u}\underbrace{\left(\left(\boldsymbol{P_p^T P_p}\right)^{-1}\boldsymbol{P_u^T P_f^T V_f} + \boldsymbol{V_u}\right)}_{\boldsymbol{V_{uf}}}\boldsymbol{Q}^T\end{aligned} \tag{15}$$

Equation 15 can be used to reformulate the cost function of equation 7 as shown in equation 16.

$$\begin{aligned}&\min_{\Delta \boldsymbol{u}} \quad \left\{\left(\widehat{\boldsymbol{y}} - \boldsymbol{y}_{sp}\right)^T\left(\widehat{\boldsymbol{y}} - \boldsymbol{y}_{sp}\right) + \; \Delta \boldsymbol{u}^T \boldsymbol{M} \Delta \boldsymbol{u}\right\} \\ &s.t. \begin{cases} \widehat{\boldsymbol{y}} = \left(\boldsymbol{\eta V} + \Delta \boldsymbol{u V_{uf}}\right)\boldsymbol{Q}^T \\ \boldsymbol{lb} \leq \boldsymbol{u_n} + \Delta \boldsymbol{u} \leq \boldsymbol{ub} \end{cases}\end{aligned} \tag{16}$$

Finally, equation 16 can be expressed as a QP problem as shown in equation 17, which is the mathematical expression used to obtain the MVT in approach proposed in this paper.

$$\min_{\Delta \boldsymbol{u}} \quad \left[\frac{1}{2}\Delta \boldsymbol{u}^T \boldsymbol{H} \Delta \boldsymbol{u} + \boldsymbol{f}^T \Delta \boldsymbol{u}\right] \tag{17}$$

$$s.t.\begin{cases} \boldsymbol{H = V_{uf} Q^T Q V_{uf}^T + M} \\ \boldsymbol{f^T = (\eta V Q^T - y_{sp}) Q V_{uf}^T} \\ \boldsymbol{lb \leq u_n + \Delta u \leq ub} \end{cases}$$

2.3 Validity constraints (VC)

The control methodology presented in this article is heavily dependent on the quality of predictions, and it is, therefore, important to consider the validity of the model when determining the MVT. In the literature, validity measures have been used as a weighting within the cost function to prevent erratic changes in the score vector [6,23]. Unfortunately using weights within the cost function does not ensure that the constraint will be respected. To address this, in this work, hard constraints were applied to the cost function using the approach adopted in [24]. However, rather than applying the constraints in the LV space, they were applied in the real space; Equation 18 shows the addition of the hard validity constraints in the MVT optimisation defined in equation 17.

$$\min_{\boldsymbol{\Delta u}} \ [\boldsymbol{\Delta u^T H \Delta u + f^T \Delta u}]$$

$$s.t.\begin{cases} \boldsymbol{H = V_{uf} Q^T Q V_{uf}^T + M} \\ \boldsymbol{f^T = (\eta V Q^T - y_{sp}) Q V_{uf}^T} \\ \boldsymbol{lb \leq u_n + \Delta u \leq ub} \\ J_t(\boldsymbol{\Delta u}) \leq L_t \\ J_e(\boldsymbol{\Delta u}) \leq L_e \end{cases} \tag{18}$$

Where the score validity indicator J_t and the residuals validity indicator J_e for batch $\boldsymbol{k}$ must be less than their pre-defined limits L_t and L_e. These limits define how far the scores of the batch can move away from the scores of the calibration data and are chosen to have a value of 1 if the scores are to be constrained within the calibration range or above 1 if some extrapolation is to be allowed. The validity indicator J_t is based on the Hotelling's statistic T^2 and provides a measure of the deviation of the score $\boldsymbol{t_k}$ from the region covered by the identification dataset. Equations 19 and 20 provide the mathematical definition of J_t.

$$J_t = \frac{\boldsymbol{t_k (S_a^2)^{-1} t_k^T}}{J_{tmax}} \tag{19}$$

$$\boldsymbol{t_k = \eta V + \Delta u V_{uf}} \tag{20}$$

Where the diagonal matrix of covariance $\boldsymbol{S_\alpha^2}$ contains the covariance of each LV in the identification dataset and the normalization variable J_{tmax} is the value of the T^2 statistic taken at a 95% confidence limit [24]. The validity indicator J_e provides a quadratic measure of the error between the predictor vector, $\boldsymbol{x_k}$ for the current batch and its value when reconstructed from the scores in the LV space. Equations 21 and 22 show the definition of the validity indicator J_e.

$$J_e = \frac{\boldsymbol{e_k e_k^T}}{J_{emax}} \tag{21}$$

$$\boldsymbol{e_k = \eta\left(I - VP^T\right) + \Delta u(I - V_{uf}P^T)} \tag{22}$$

Where the normalization variable J_{emax} is equal to the value of the square of the error, $\boldsymbol{ee^T}$ at the 95% confidence limit [24].

2.4 Iterative control

The purpose of the iterative control scheme introduced in this work is to gradually increase the yield from one batch to the next. As discussed earlier, this requires the MPLS model to adapt to the most recent batch conditions. This section describes how the model is adapted and the process optimised. An overview of the full batch optimisation scheme is provided in Figure 2.

Stage 1:

With the process operating in open loop and a Pseudo-Random Binary Sequence (PRBS) added to the nominal MVT, data is collected from several batches. The PRBS is required to excite the process, allowing a MPLS model to be identified. The PRBS can be added to a nominal MVT that was either based on a previously identified trajectory, the trajectory of a 'golden batch' or if necessary, a vector of zeros. The amplitude of the PRBS was specified to be sufficiently low that it did not significantly alter the behaviour of the process, but high enough for the characteristics of the process to be identifiable from background noise. In this work the nominal trajectory was taken as a vector of zeros

as this was considered to be the worst-case scenario. For an industrial process it is highly likely that an MVT that produces an acceptable amount of product would be available. If such an MVT is available then the benefit in using it as the initial MVT is that fewer batches would typically be required to optimise the process. Further details of this are available in [29]

Stage 2:

Once the data was collected, an MPLS model was identified as described in section 2.1. For industrial applications, the number of batches of data required to identify the model should be minimised. In this work it was found that good results were obtained when data from only 3 batches were used to identify the initial MPLS model.

Stage 3:

The initial conditions for the current *k*th batch were collected, and the matrices required to define the optimisation function, equation 18, formed.

Stage 4:

The target end-point quality, $\boldsymbol{y_{sp}}$ in equation 17 was defined and equation 18 was solved using QP to obtain the MVT for the current batch. $\boldsymbol{y_{sp}}$ can either be set to a specific value or it can be incrementally adjusted from one batch to the next. In this work, the value of $\boldsymbol{y_{sp}}$ remained fixed for each of the case studies. The specific value of $\boldsymbol{y_{sp}}$ that was used was based on a-priori knowledge of each of the processes. However, the results were very similar when the value of $\boldsymbol{y_{sp}}$ was increased slightly from one batch to the next.

In [21,30,31], it was suggested that the identified MVT be passed through a low-pass filter before being applied to the process. In this work, filtering the identified MVT proved to be beneficial and hence in each case study the MVT was passed through a zero-phase, low pass Finite Impulse Response (FIR) filter with a cut-off frequency of 10% of the maximum frequency (Nyquist

frequency). Experimental results showed that there was little difference when the maximum frequency was varied between 3% and 10%.

Stage 5:

As the data collected from the *k*th batch will be used to adapt the model it was necessary to introduce further excitation in to the process. This was achieved by adding a low-amplitude PRBS to the MVT determined in stage 4. In this article the amplitude of` the PRBS was specified to be equal to 3% of the mean of the amplitude of the nominal MVT measured for the initial identification set of batches. However, it was found that there was not significant difference in the results if the amplitude of the PRBS was varied between 1% to 5% of this amplitude. As an example, Figure 3 shows the MVT that resulted from solving equation 18 and this same MVT after it has been smoothed and the PRBS added.

Stage 6:

Data was collected from the batch until it terminated and then the MPLS model was updated, as described in Section 2.1.1. Typically, the maximum number of LVs that can be utilised in the initial model was insufficient to enable the MPLS model to fully describe the process. This was because the number of batches used to identify the initial model was limited [32], consequently the model was updated using equation 4 to increase the size of the $\boldsymbol{X}$ and $\boldsymbol{Y}$ matrices. If the number of LVs was not limited then equation 5 was used and the number of LVs kept constant. The number of LVs with the smallest mean squared prediction error (MSEP) or the point at which the remaining LVs were relatively small was used as an estimator in leave-one-out cross validation to determine the number of LVs of the model.

3. RESULTS

In this article two benchmark simulations of fed-batch fermentation processes were used to validate the proposed optimisation technique. This section first introduces the two case studies and this is followed in sections 3.3 and 3.4 with a summary of the results that were obtained.

3.1 Saccharomyces Cerevisiae (Saccha)

The first case study utilised a simulation of a fed-batch production process of Saccharomyces Cerevisiae (Saccha). This simulation models the growth of Saccha, which is type of yeast that is useful in the production of human consumption fermentation-based products, on glucose and ethanol and includes 11 model reactions and 9 mass balance dynamic equations. Full details are available in [33]. The initial conditions that were used in this work are provided in Table 1.

Table 1 Initial Conditions

Variables	**Initial condition**
Glucose concentration	0 (g/l)
Pyruvate concentration	0 (g/l)
Acetaldehyde concentration	0 (g/l)
Acetate concentration	0 (g/l)
Ethanol concentration	0 (g/l)
Biomass concentration	1 (g/l)
Active cell material	0.3
Acetaldehyde dehydrogenase	0.0075
Volume	7 (l)

The objective in terms of optimisation of this process was to maximise the final biomass concentration obtained from each batch. Biomass concentration is principally effected by manipulating the trajectory of glucose that is fed to the reactor. The simulation parameters that were used in this work, and listed in Tables 1 and 2, matched those used in [21].

Table 2 Saccha simulation parameters

Simulation Parameter	**Value**
Batch total time	10 h
Sample interval	0.1 h
Control action start	0 h
Output noise	1%
Initial conditions variability	3%
Final volume	Less than 9(l)
Hard constraints in the MVT (l/h)	$0.0 \leq and \leq 0.6$

The final volume restriction in Table 2 imposes an additional constraint in the QP optimisation problem. This constraint was included in the cost function by considering the mean, $\boldsymbol{mean}_u$, and standard deviation, $\boldsymbol{std}_u$, of the manipulate variable as shown in equation 23.

$$\boldsymbol{t_i}\,\overline{\boldsymbol{u}}^{\boldsymbol{T}} \leq V_{max} - V_{ini}$$

$$\text{Where} \qquad \overline{\boldsymbol{u}}^{\boldsymbol{T}} = \boldsymbol{diag}(\boldsymbol{std}_u)\widehat{\boldsymbol{u}}^{\boldsymbol{T}} + \boldsymbol{mean}_u{}^T \qquad (23)$$

$$\widehat{\boldsymbol{u}}^{\boldsymbol{T}} = \boldsymbol{u}_{\boldsymbol{n}}^{\boldsymbol{T}} + \Delta\boldsymbol{u}^{\boldsymbol{T}}$$

Where the terms $\boldsymbol{t_i}$ is a vector of ones multiplied by the scalar corresponding to the sample interval, V_{max} is the final accepted volume and V_{ini} is the initial volume. Equation 23 can be redefined by the inequality constraint shown in equation 24.

$$\boldsymbol{A} * \Delta\boldsymbol{u}^{\boldsymbol{T}} \leq \boldsymbol{b}$$

$$\text{Where} \quad \boldsymbol{A} = \boldsymbol{t_i}\boldsymbol{diag}(\boldsymbol{std}_u) \qquad (24)$$

$$\boldsymbol{b} = V_{max} - V_{ini} - \boldsymbol{t_i}\left(\boldsymbol{mean}_u{}^T + \boldsymbol{diag}(\boldsymbol{std}_u)\boldsymbol{u}_{\boldsymbol{n}}^{\boldsymbol{T}}\right)$$

Figure 4 shows the final volume that was measured from 100 batches when the volume constraint was imposed within the cost function. This figure shows that the 9 l volume constraint is occasionally exceeded. The reason for this is that the constraint does not consider the small PRBS that is added at the end of the MVT optimisation. A method to avoid violating this constraint is included in equation 24. However, this was not included in the work described in this article as it was felt that exceeding this constraint very slightly would not be a problem in an industrial process.

3.2 Penicillin production (Pensim)

The second case study used in this article was a benchmark simulator of a penicillin fermentation process described in [34]. The simulation, named Pensim, has been used in many articles as a tool for evaluating control designs [7,35,36]. The simulation consists of 5 input variables, 9 process variables and 5 output variables, shown in Table 3. This table also shows the amplitude of the PRBS signals that were added to the set-points of the low-level PID controllers that were used to regulate process

variables, such as temperature and glucose feed rate. The objective for this case study was to maximize the final penicillin concentration by manipulating the MVT of the glucose feed rate in the minimum number of batches. As highlighted by [37], the relationship between penicillin and glucose feed rate is highly non-linear.

Table 3 Pensim simulation variables

Input Variable	**Initial condition**	**PRBS Amplitude**
Aeration rate	8 (l/h)	0.3 (l/h)
Agitator power	30 (W)	1 (W)
Glucose feed rate	0.045 (l/h)	3%
Glucose feed temperature	296 (K)	0.5 (K)
Acid flow rate	0 (ml/h)	0.03 (ml/h)
Temperature	298 (K)	0.03 (K)
Volume	100 (l)	-
Temperature	297 (K)	-
Generated heat	0 (cal)	-
pH	5	-
Substrate concentration	15 (g/l)	-
Biomass concentration	0.1 (g/l)	-
Penicillin concentration	0 (g/l)	-
Dissolved oxygen	1.16 (g/l)	-
Carbon Dioxide concentration	0.5 (mmol/l)	-

The simulation was set-up with the standard configuration as described in [7]. As with many industrial processes [35], the process was switched from batch to fed-batch operation after 45 hours. Hard constraints were also imposed on the glucose feed rate, which was restricted to between 0 l/h and 0.2 l/h.

3.3 Results and comparison with relevant literature

The results presented in this section were obtained by running multiple experiments for each of the two simulations. For the Saccha simulation each experiment considered the results from 100 batches and for the Pensim simulation, each experiment considered 30 batches. Each batch contained random variability in the initial conditions as defined in [21] and [35] for the Saccha and Pensim simulations respectively. The reason for using different numbers of batches per experiment was so that the results from this work could be compared with those of [21] for Saccha and those of [7] for Pensim. The initial dataset for both simulations consisted of 3 batches with PRBS added to nominal MVTs. To

provide comparisons with other studies, the initial MVT for the Saccha process was a vector of zeros while for Pensim it was a nominal trajectory of 0 before 45 hours and 0.045 l/h after this time.

The vector of initial conditions $\boldsymbol{x_p}$ was obtained from the initial states of the process variables for the Saccha simulation, and during the initial batch phase of the Pensim simulation, where there was no glucose fed in to the reactor. Figures 6 and 7 show the results obtained when the proposed batch-to-batch optimisation technique was applied to the Saccha simulator. Figure 5 shows that the final biomass concentration increased steadily from about 2 g/l to 11.3 g/l and after approximately 15 batches levels off at this output concentration. Figure 6 shows the consistency of the proposed approach when it was applied to 100 experiments on the Saccha simulator. This figure shows that the final biomass concentration increased consistently to approximately 11.3 g/l in each experiment. The standard deviation for the end-point biomass concentration, measured over the 100 experiments was 0.2 g/l.

The resulting end-point biomass concentration compares very favourably with other results applied to this simulator. In [21] for example, their batch-to-batch optimisation technique increased biomass concentration to 10.74 g/l, with a standard deviation of 0.16 g/l, after approximately 60 batches. Although the work wasn't specifically attempting to maximise batch yield, [33] and [38] recorded end-point biomass concentrations of approximately 10 g/l.

Figure 7 shows how the trajectory of the substrate feed changes during a typical experiment. This figure shows that the trajectory changes considerably from the first through to the final batch. The feeding law presented in this graph does not include the additional PRBS that was applied to these trajectories.

Figures 8 shows how the end-point penicillin concentration increases over 100 batches for the Pensim simulator following the introduction of the proposed batch-to-batch optimisation system. In this figure

the presented results are the average taken over 30 experiments. These results show that the end-point penicillin concentration increases to approximately 2.11 g/l.

Figure 9 shows how the MVT evolves over 100 batches for a single experiment. This figure shows that the MVT for the first batch switches to fed-batch mode after approximately 45 hours. After this the feeding rate increases gradually until it reaches a value approaching the actuator constraints. The feed rate then reduces towards the end of the batch. If the feed rate remains high through to the end of the batch then the concentration of glucose in the fermenter will increase and as a consequence the amount of penicillin that is produced will reduce.

The average value of the culture volume obtained at the end of the batch was approximately120 l, which means that the average quantity of penicillin produced from each batch was approximately 253 g. This result compares favourably to those presented in [34], where the final penicillin production was 106 g and those resulting from the evolutionary optimisation technique presented in [35] which produced 160 g of penicillin.

3.3.2 Introduction of Validity Constraints

The results presented in this section provide a comparison of the results obtained when the Validity Constraints (VC) introduced in section 2.3 were introduced. Figure 10 shows the final biomass concentration obtained for the Saccha simulation when various limits were placed on the values of J_e and J_t. These limits were chosen as relevant examples of the effect of using validity constraints in the proposed optimisation scheme.

The results presented in figure 10 show that the effect of introducing the validity constraints is to reduce the rate at which biomass increases during the first 20 batches. This is to be expected as the constraints will have the effect of limiting the allowable change in MVT from one batch to the next, which will slow the rate of convergence of the optimisation strategy. It can also be observed that in most cases, the introduction of validity constraints results in a slight reduction in the final biomass

concentration. This is because the introduction of the validity constraints has the effect of limiting the solution space over which the optimisation algorithm searches over and hence it may not identify the same solution as was obtained with no validity constraints.

Figure 11 shows the final penicillin concentration when varying validity constraints were applied to the Pensim simulator. As with the Saccha simulator, this figure shows that the batch-to-batch optimisation strategy converges at a slower rate when the constraints are introduced.

Although the introduction of the validity constraints has the effect of slowing convergence and reducing the end-point yield slightly, they have the positive benefit of reducing variation in the results and hence improving consistency. For example, variation was minimised when the constraint, $J_e < 4$ was introduced; the standard deviation when this constraint was implemented was 0.125 g/l for Saccha and 0.141 g/l for Pensim, which compares with 0.19 g/l and 0.20 g/l when no validity constraints were considered. It is anticipated that the validity constraints would be of significant benefit with any real applications. In such applications it is likely to be desirable for only minimal changes to be made to the MVT from one batch to the next. The precise value that the validity constraints should take has not been determined in this work and is the subject of future research.

4. CONCLUSIONS

This article proposed an innovative batch-to-batch optimisation technique that can be used to improve yield of batch processes. The technique identifies an MPLS model of a batch process, which is then used within an optimisation function to identify the trajectory of a manipulated variable in the process, so that the target end-point quality, such as biomass concentration is achieved. By constraining the changes allowable in the identified trajectories of the manipulated variables and by updating the MPLS model at the end of each batch, the yield of the batch is gradually increased from batch to batch.

The capabilities of the proposed technique were demonstrated through application to two benchmark fed-batch fermentation simulators: Saccharomyces Cerevisiae and Pensim. For both simulators, the proposed batch-to-batch optimisation technique was found to increase yield considerably over a limited number of batches. For example, for the saccharomyces cerevisiae process, yield was increased almost 5-fold over fewer than 20 batches.

Validity constraints were introduced in to the batch-to-batch optimisation technique to ensure that the optimisation algorithm restricted its search space to the region where the MPLS was valid. The introduction of these constraints resulted in a slower rate of convergence and in some case a slight reduction in final yield. However, the constraints did improve consistency, which would be a major requirement for any industrial application.

ACKNOWLEDGEMENTS

This research was partially supported by the Mexican Government (CONACyT 214965 and SEP 43901). The authors would like to thank PhD Jose Camacho Paez for providing us with suitable parameters for the saccharomyces cerevisiae simulator and to Qiaolin Yuan for her support with some of the methods utilised in this article.

REFERENCE

[1] D. Bonvin, B. Srinivasan, D. Hunkeler, Control and optimization of batch processes: improvement of process operation in the production of specialty chemicals, IEEE Control Systems Magazine. (2006) 34–45.

[2] J. Zhang, J. Nguyan, Batch to batch iterative learning control of a fed-batch fermentation process using linearised models, 10th International Conference on Control, Automation, Robotics and Vision, 2008: pp. 745–750.

[3] H. Martens, Reliable and relevant modelling of real world data: a personal account of the development of PLS Regression, Chemometrics and Intelligent Laboratory Systems. 58 (2001) 85–95.

[4] S. Goldrick, S. Andrei, D. Lovett, G. Montague, B. Lennox, The development of an industrial-scale fed-batch fermentation simulation, Journal of Biotechnology. 193 (2015) 70–82.

[5] C. Ündey, S. Ertunç, A. Çinar, Online batch/fed-batch process performance monitoring, quality prediction, and variable-contribution analysis for diagnosis, Industrial & Engineering Chemistry Research. 42 (2003) 4645–4658.

[6] J. Flores-Cerrillo, J.F. MacGregor, Control of batch product quality by trajectory manipulation

using latent variable models, Journal of Process Control. 14 (2004) 539–553.

[7] J. Wan, O. Marjanovic, B. Lennox, Disturbance rejection for the control of batch end-product quality using latent variable models, Journal of Process Control. 22 (2012) 643–652.

[8] J. Chen, K.-C. Lin, Batch-to-batch iterative learning control and within-batch on-line control for end-point qualities using MPLS-based dEWMA, Chemical Engineering Science. 63 (2008) 977–990.

[9] J. Flores-Cerrillo, J.F. MacGregor, Latent variable MPC for trajectory tracking in batch processes, Journal of Process Control. 15 (2005) 651–663.

[10] H. Yu, J. Flores-Cerrillo, Latent Variable Model Predictive Control for Trajectory Tracking in Batch Processes: Internal Model Control Interpretation and Design Methodology, Industrial & Engineering Chemistry Research. (2013).

[11] D. Laurí, B. Lennox, J. Camacho, Model predictive control for batch processes: Ensuring validity of predictions, Journal of Process Control. 24 (2014) 239–249.

[12] S. Aumi, B. Corbett, T. Clarke-Pringle, P. Mhaskar, Data-driven model predictive quality control of batch processes, AIChE Journal. 59 (2013) 2852–2861.

[13] F. Arteaga, A. Ferrer, Dealing with missing data in MSPC: several methods, different interpretations, some examples, Journal of Chemometrics. (2002) 408–418.

[14] O. Marjanovic, B. Lennox, D. Sandoz, K. Smith, Real-time monitoring of an industrial batch process, Computers & Chemical. (2006).

[15] A. Kassidas, J.F. Macgregor, P. a Taylor, Synchronization of batch trajectories using dynamic time warping, AIChE Journal. 44 (1998) 864–875.

[16] K. Lee, I. Chin, H. Lee, Model predictive control technique combined with iterative learning for batch processes, AIChE Journal. 45 (1999) 2175–2187.

[17] K.S. Lee, J.H. Lee, Iterative learning control-based batch process control technique for integrated control of end product properties and transient profiles of process variables, Journal of Process Control. 13 (2003) 607–621.

[18] M. Mezghani, G. Roux, M. Cabassud, B. Dahhou, M.. Le Lann, G. Casamatta, Robust iterative learning control of an exothermic semi-batch chemical reactor, Mathematics and Computers in Simulation. 57 (2001) 367–385.

[19] E. Picó-Marco, J. Picó, H. De Battista, Sliding mode scheme for adaptive specific growth rate control in biotechnological fed-batch processes, International Journal of Control. 78 (2005) 128–141.

[20] S. Zhang, F. Wang, D. He, R. Jia, Batch-to-batch control of particle size distribution in cobalt oxalate synthesis process based on hybrid model, Powder Technology. 224 (2012) 253–259.

[21] J. Camacho, J. Pico, A. Ferrer, Self-tuning run to run optimization of fed batch processes using unfold PLS, AIChE Journal. 53 (2007).

[22] P. Nomikos, J.F. MacGregor, Multi-way partial least squares in monitoring batch processes, Chemometrics and Intelligent Laboratory Systems. 30 (1995) 97–108.

[23] D. Laurí, J. a. Rossiter, J. Sanchis, M. Martínez, Data-driven latent-variable model-based predictive control for continuous processes, Journal of Process Control. 20 (2010) 1207–1219.

[24] D. Laurí, J. Sanchis, M. Martínez, a. Hilario, Latent variable based model predictive control: Ensuring validity of predictions, Journal of Process Control. 23 (2013) 12–22.

[25] B.S. Dayal, J.F. MacGregor, Recursive exponentially weighted PLS and its applications to adaptive control and prediction, Journal of Process Control. 7 (1997) 169–179.

[26] S. Joe Qin, Recursive PLS algorithms for adaptive data modeling, Computers & Chemical Engineering. 22 (1998) 503–514.

[27] H. Lin, The Application of Multivariate Statistical Analysis and Batch Process Control in

Industrial Processes (Doctoral thesis), Universtity of Manchester, 2010.

[28] P. Nelson, P. Taylor, Missing data methods in PCA and PLS: Score calculations with incomplete observations, Chemometrics and Intelligent Laboratory Systems. 35 (1996) 45–65.

[29] C.A. Duran Villalobos, Run-to-Run Modelling and Control of Batch Processes (Doctoral Thesis), The University of Manchester, 2015.

[30] C. Duran-Villalobos, B. Lennox, Iterative Learning Modelling and Control of Batch Fermentation Processes, 10th IFAC International Symposium on Dynamics and Control of Process Systems, 2013: pp. 511–516.

[31] J. Camacho, D. Lauri, B. Lennox, M. Escabias, M. Valderrama, Evaluation of smoothing techniques in the run to run optimization of fed-batch processes with u-PLS, Journal of Chemometrics. 29 (2015) 338–348.

[32] D. Laurí, B. Lennox, Expanded PLS algorithm: Modeling of batch processes, Chemometrics and Intelligent Laboratory Systems. 132 (2014) 111–123.

[33] F. Lei, M. Rotbøll, S.B. Jørgensen, A biochemically structured model for Saccharomyces cerevisiae., Journal of Biotechnology. 88 (2001) 205–21.

[34] A. Cinar, A modular simulation package for fed-batch fermentation: penicillin production, Computers & Chemical Engineering. 26 (2002) 1553–1565.

[35] H. Zhang, Z. Zhang, Evolutionary optimization of a fed-batch penicillin fermentation process, Computer Communication Control and Automation. (2010) 403–406.

[36] H. Zhang, B. Lennox, Integrated condition monitoring and control of fed-batch fermentation processes, Journal of Process Control. 14 (2004) 41–50.

[37] L. Yan, B. Lennox, The application of nonlinear partial least square to batch processes, 10th IFAC International Symposium on Dynamics and Control of Process Systems. (2013) 289–294.

[38] H. Pham, G. Larsson, Growth and energy metabolism in aerobic fed-batch cultures of Saccharomyces cerevisiae: Simulation and model verification, Biotechnology and Bioengineering. 60 (1998) 474–482.

Appendix: Nomenclature

$\boldsymbol{e_k} \in \mathbb{R}^{1 \times JK}$	residuals of predictors vector	$\boldsymbol{V_{uf}} \in \mathbb{R}^{K \times A}$	weights matrix after PMP
$\boldsymbol{E} \in \mathbb{R}^{I \times JK}$	residual matrix of predictors	$\boldsymbol{V} \in \mathbb{R}^{JK \times A}$	weights matrix of the PLS model
$\boldsymbol{F} \in \mathbb{R}^{I \times no}$	residual matrix of responses	$\boldsymbol{X} \in \mathbb{R}^{I \times JK}$	predictors matrix
$J_e \in \mathbb{R}$	residuals validity indicator	$\boldsymbol{x_k} \in \mathbb{R}^{1 \times JK}$	predictors vector of the current batch
$J_t \in \mathbb{R}$	score validity indicator	$\boldsymbol{x_p} \in \mathbb{R}^{1 \times (in)}$	initial conditions vector
$L_e \in \mathbb{R}$	residuals validity limit	$\boldsymbol{x_f} \in \mathbb{R}^{1 \times (\mathrm{J} - in - K)}$	unmeasured variables vector
$L_t \in \mathbb{R}$	score validity limit	$\boldsymbol{Y} \in \mathbb{R}^{I \times no}$	responses matrix
$\boldsymbol{lb} \in \mathbb{R}^{1 \times K}$	lower bound constraints vector	$\hat{\boldsymbol{y}} \in \mathbb{R}^{1 \times no}$	end-point quality prediction
$\boldsymbol{M} \in \mathbb{R}^{K \times K}$	weight matrix of the MVT adjustments	$\boldsymbol{y}_k \in \mathbb{R}^{1 \times no}$	end-point quality of the current batch
$N \in \mathbb{R}$	Number of batches effectively used for λ	$\boldsymbol{y}_{sp} \in \mathbb{R}^{1 \times no}$	end-point quality set-point
$\boldsymbol{P} \in \mathbb{R}^{A \times JK}$.	loading matrix of predictors		
$\boldsymbol{P_p} \in \mathbb{R}^{A \times JK(f)}$.	loading matrix of known predictors	Symbols	
$\boldsymbol{P_f} \in \mathbb{R}^{A \times JK(p)}$.	loading matrix of future predictors	$\Delta \boldsymbol{u} \in \mathbb{R}^{1 \times K}$	adjustments in the MVT vector
$\boldsymbol{P_u} \in \mathbb{R}^{A \times JK(u)}$.	loading matrix of MVT predictors	$\lambda \in \mathbb{R}$	Forgetting factor
$\boldsymbol{Q} \in \mathbb{R}^{A \times no}$	loading matrix of responses	$\boldsymbol{\eta} \in \mathbb{R}^{1 \times JK}$	Measured and estimated predictors
$\boldsymbol{T} \in \mathbb{R}^{I \times A}$	scores matrix		
$t_i \in \mathbb{R}$	Sample interval	Index	
$\boldsymbol{t_k} \in \mathbb{R}^{1 \times A}$	score vector of the current batch	$in \in \mathbb{R}$	number of initial conditions measurements
$\boldsymbol{u_n} \in \mathbb{R}^{1 \times K}$	nominal MVT vector	$no \in \mathbb{R}$	number of output variables
$\boldsymbol{ub} \in \mathbb{R}^{1 \times K}$	upper bound constraints vector	$A \in \mathbb{R}$	number of Latent Variables
$V_{ini} \in \mathbb{R}$	Initial volume	$I \in \mathbb{R}$	number of batches in the identification
$V_{max} \in \mathbb{R}$	Maximum volume	$J \in \mathbb{R}$	number of input variables
$\boldsymbol{V_f} \in \mathbb{R}^{K(f) \times A}$	weights matrix of future predictors	$K \in \mathbb{R}$	Time intervals
$\boldsymbol{V_u} \in \mathbb{R}^{K \times A}$	weights matrix of MVT	$k \in \mathbb{R}$	Current batch

Figure captions

Figure 1

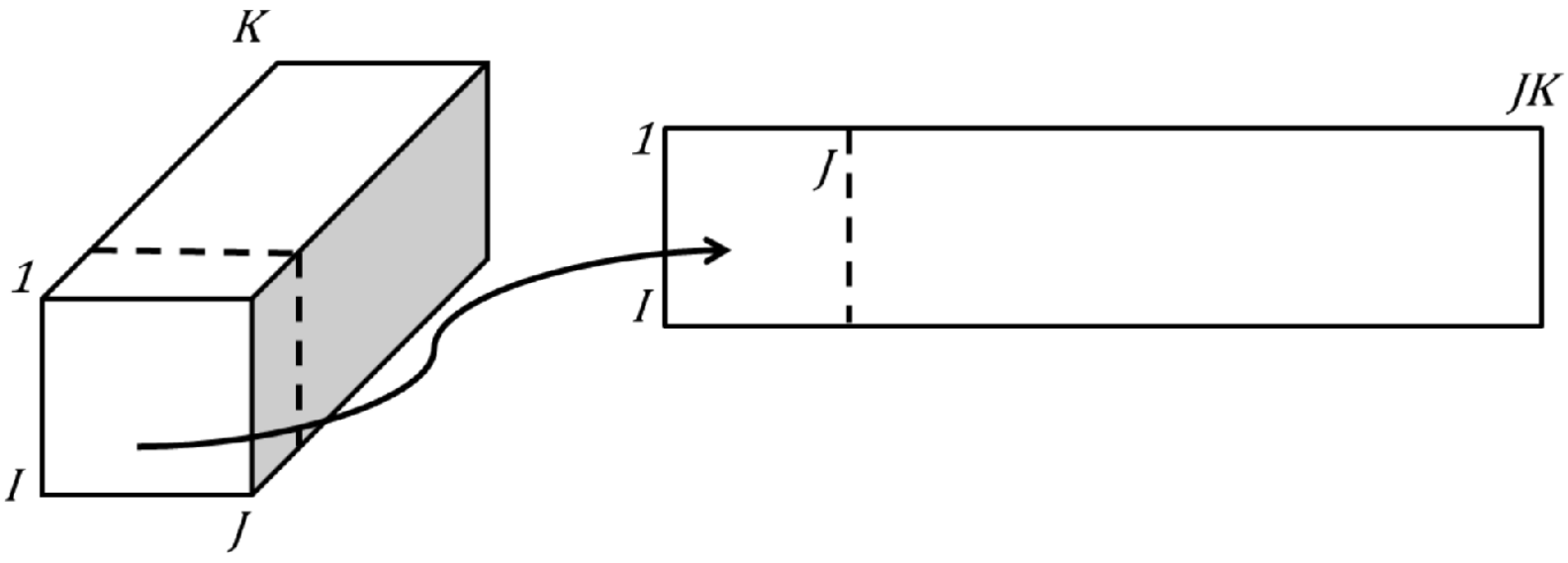


Figure 2

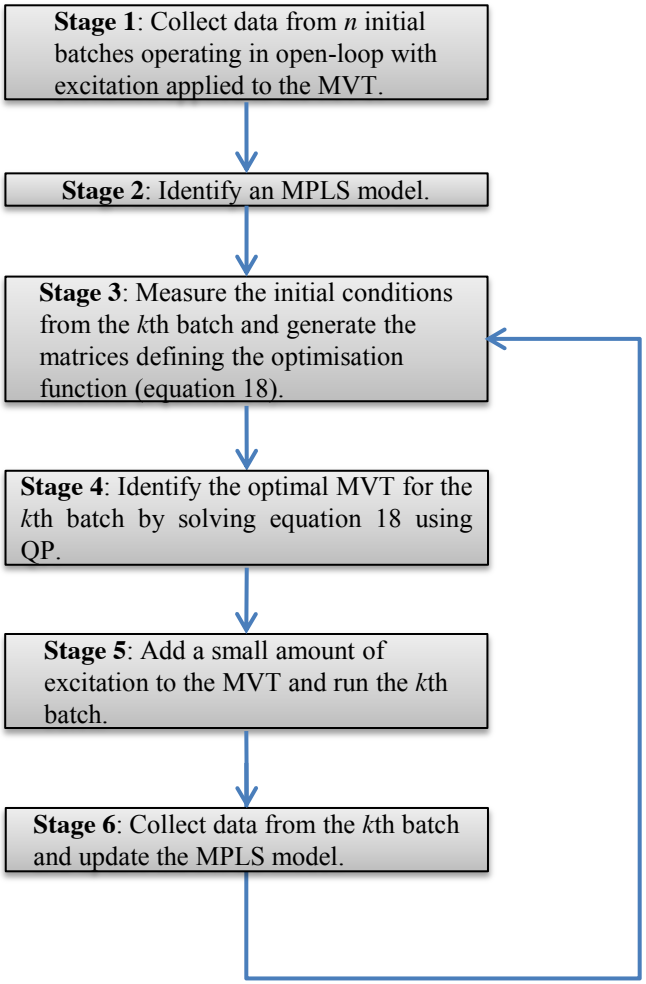

Stage 1: Collect data from *n* initial batches operating in open-loop with excitation applied to the MVT.
Stage 2: Identify an MPLS model.
Stage 3: Measure the initial conditions from the *k*th batch and generate the matrices defining the optimisation function (equation 18).
Stage 4: Identify the optimal MVT for the *k*th batch by solving equation 18 using QP.
Stage 5: Add a small amount of excitation to the MVT and run the *k*th batch.
Stage 6: Collect data from the *k*th batch and update the MPLS model.

Figure 3

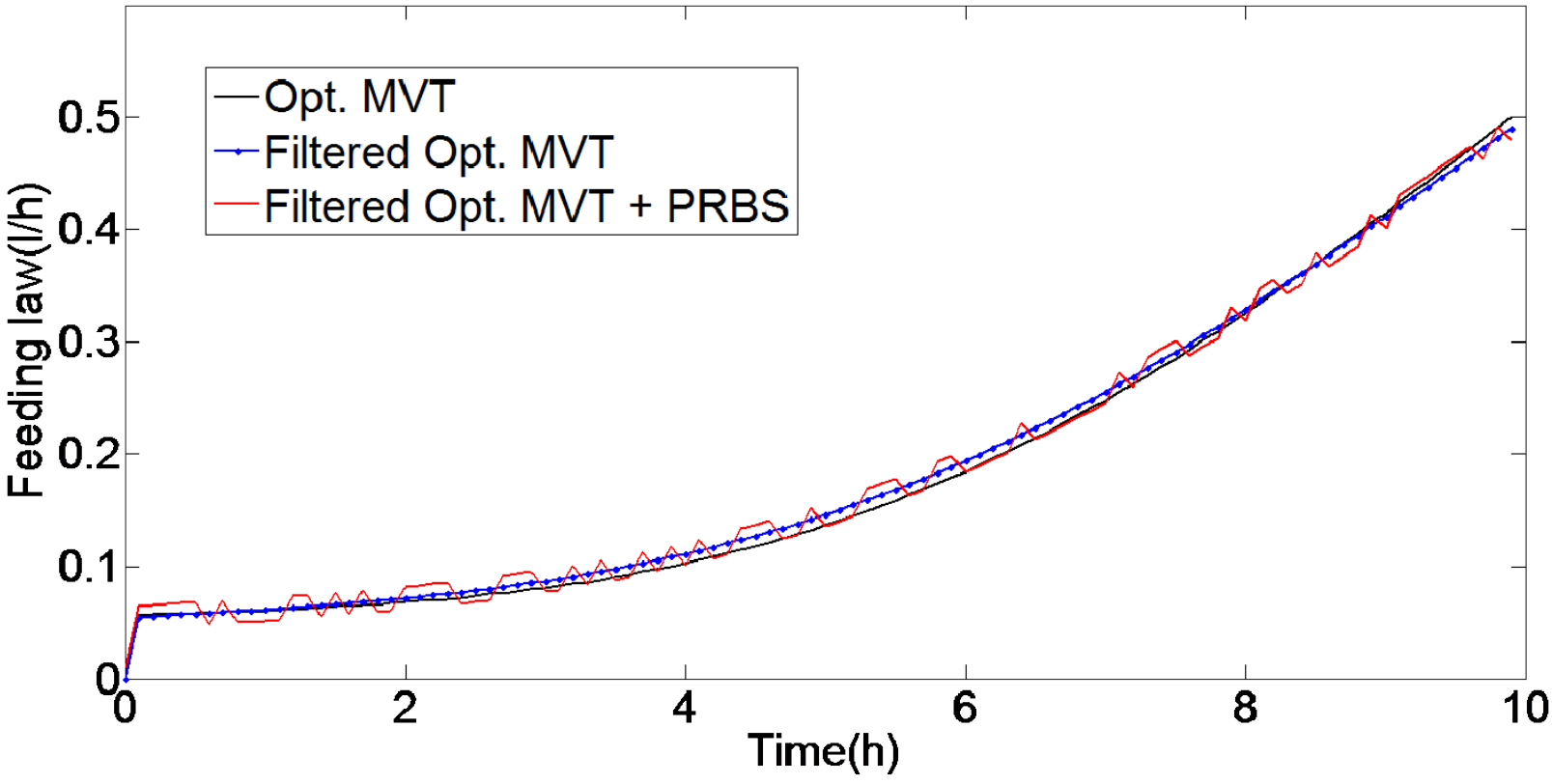

Opt. MVT
Filtered Opt. MVT
Filtered Opt. MVT + PRBS
Feeding law(l/h)
0
0.1
0.2
0.3
0.4
0.5
0
2
4
6
8
10
Time(h)

Figure 4

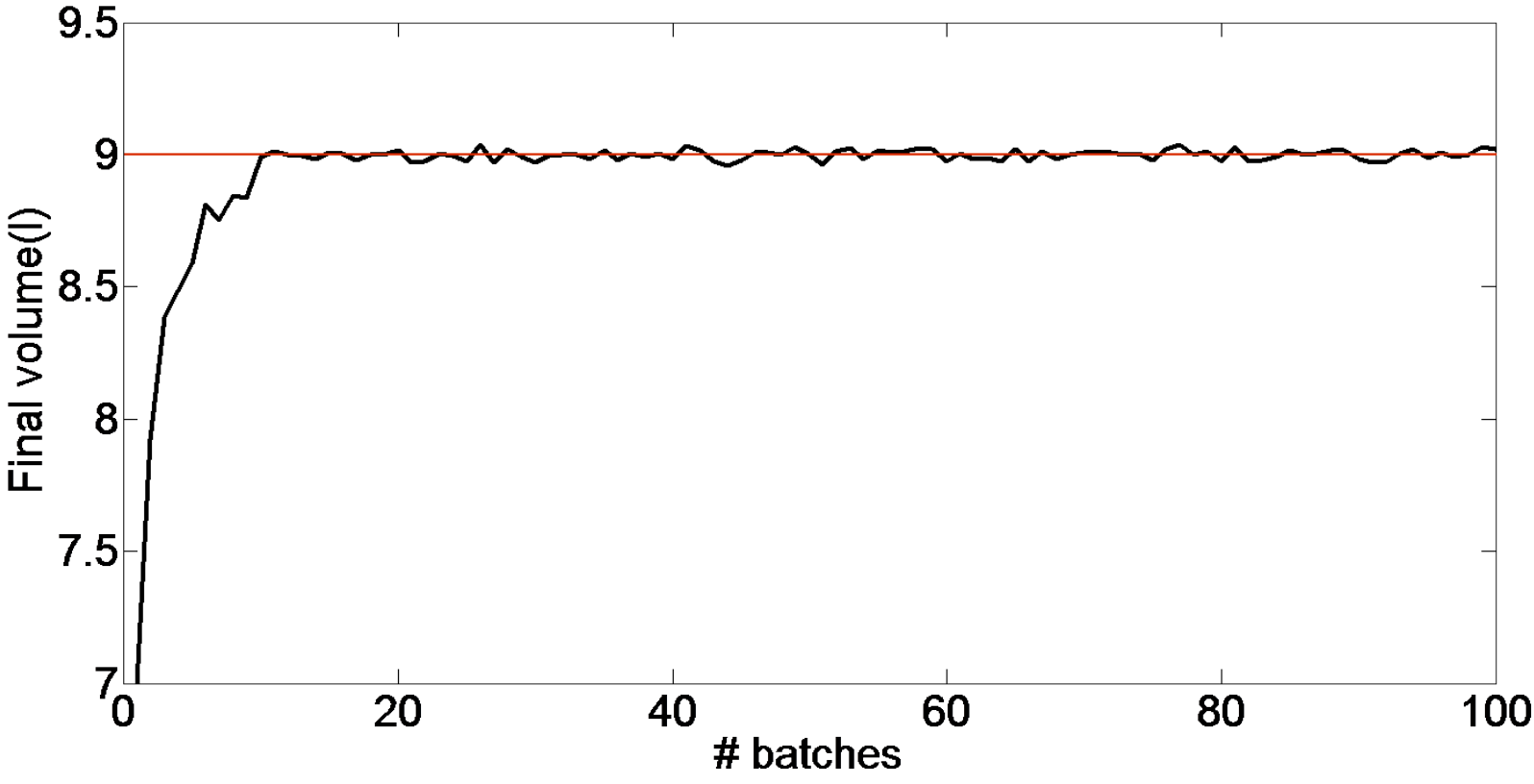


Figure 6

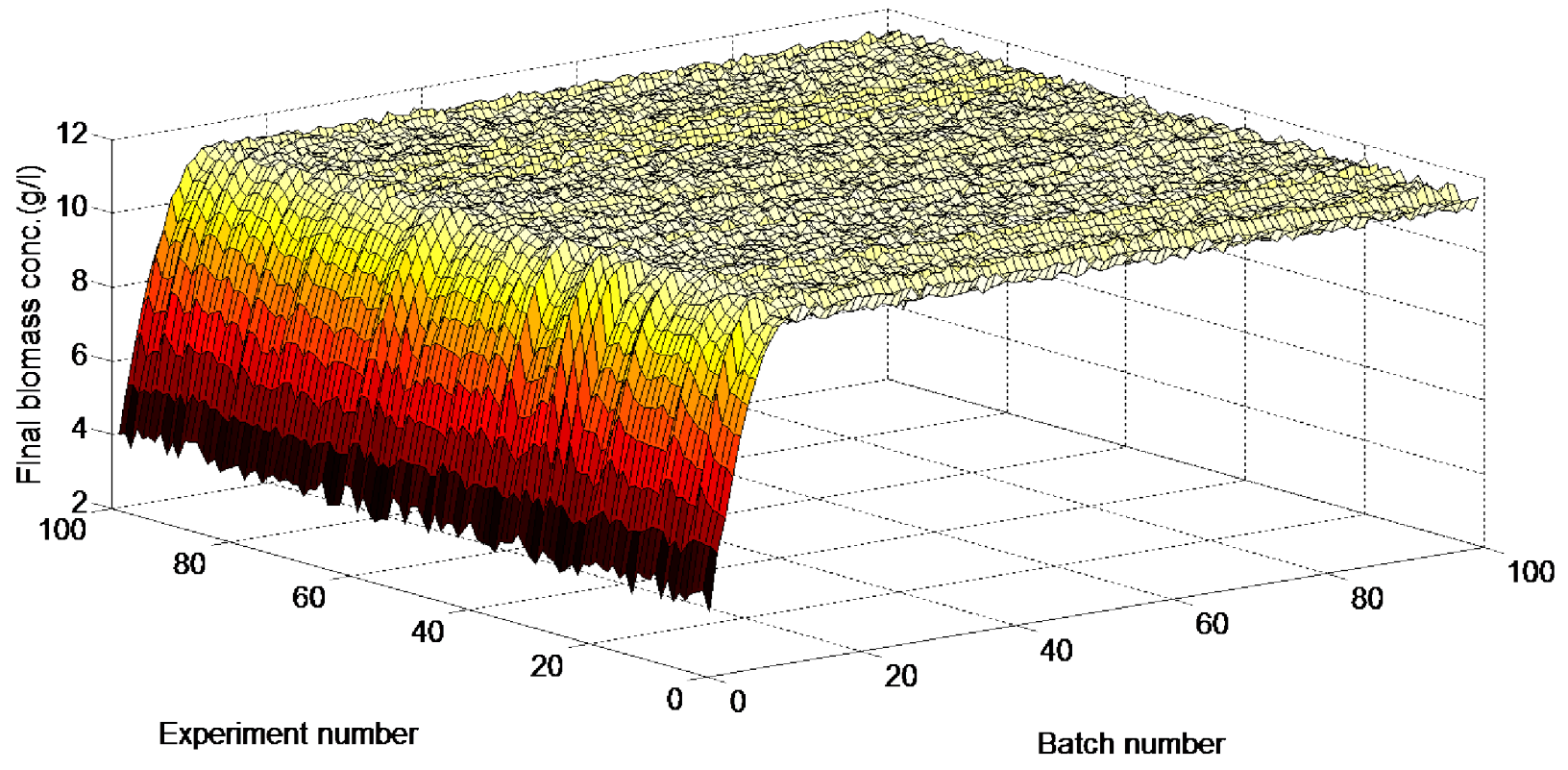

Final biomass conc.(g/l)
12
10
8
6
4
2
100
80
60
40
20
0
Experiment number
0
20
40
60
80
100
Batch number

Figure 7

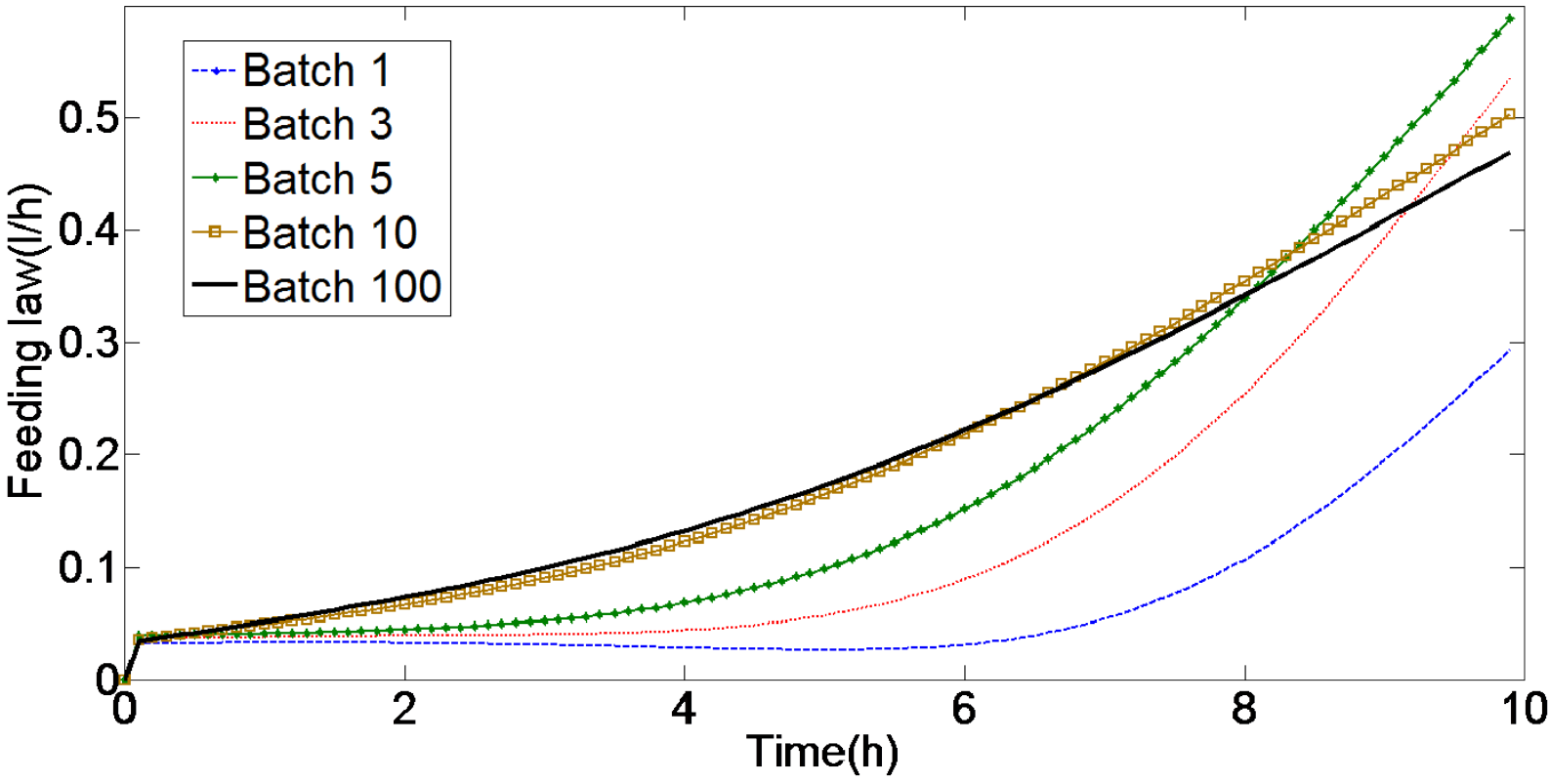

Batch 1
Batch 3
Batch 5
Batch 10
Batch 100
Feeding law(l/h)
0
0.1
0.2
0.3
0.4
0.5
0
2
4
6
8
10
Time(h)

Figure 8

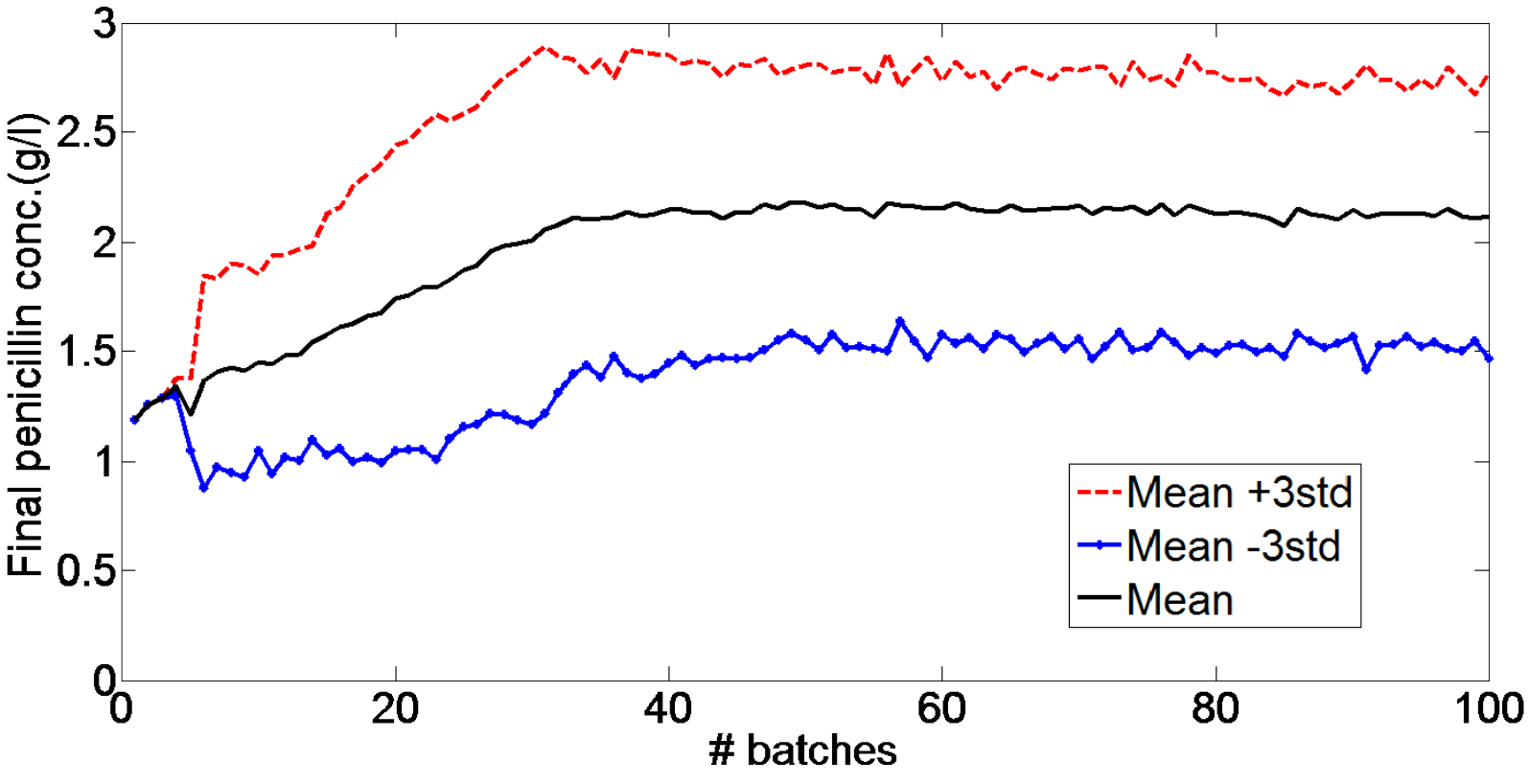

Final penicillin conc.(g/l)
batches
Mean +3std
Mean -3std
Mean
0
0.5
1
1.5
2
2.5
3
0
20
40
60
80
100

Figure 10

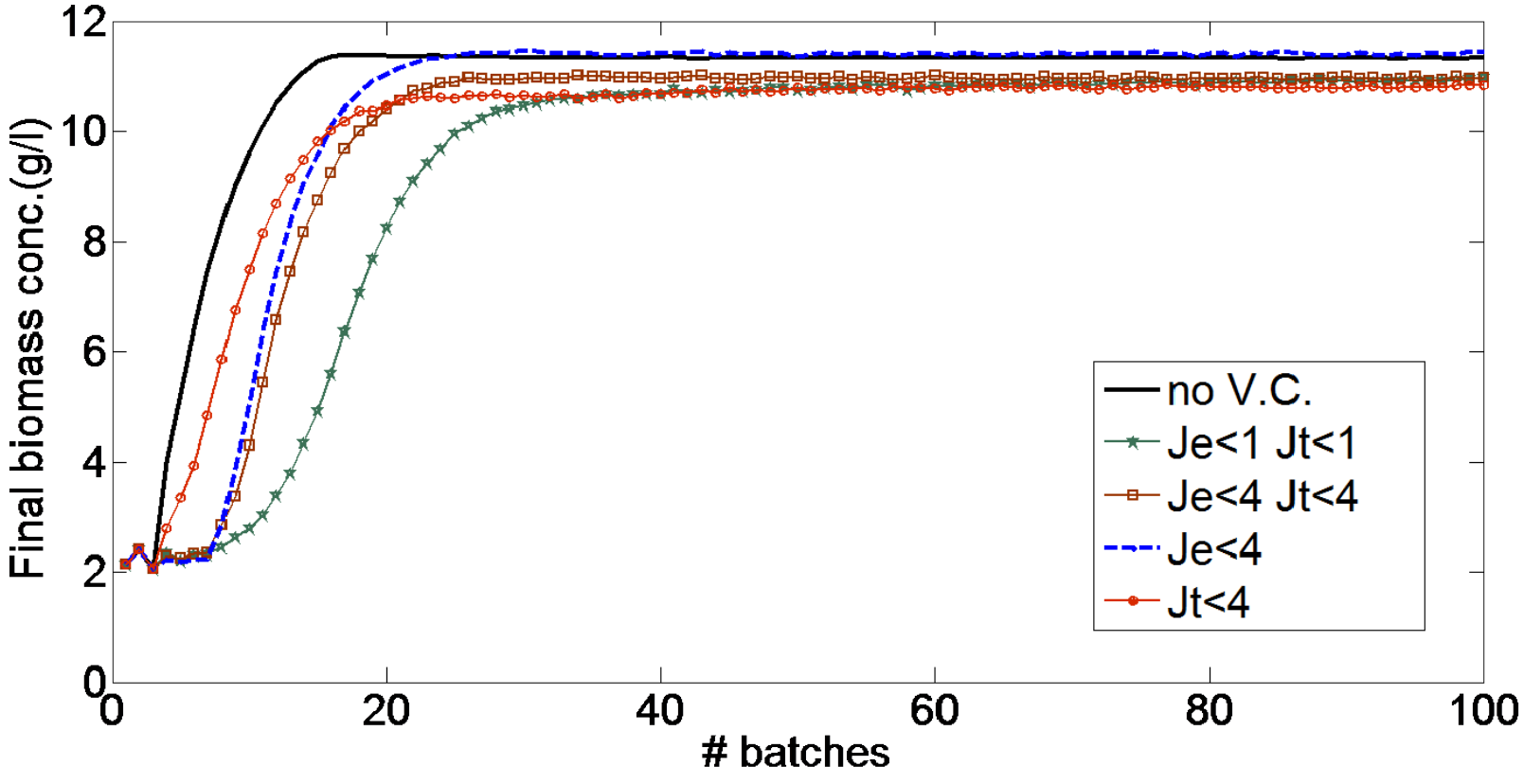

Final biomass conc.(g/l)
batches
0
2
4
6
8
10
12
0
20
40
60
80
100
no V.C.
Je<1 Jt<1
Je<4 Jt<4
Je<4
Jt<4

Figure 11

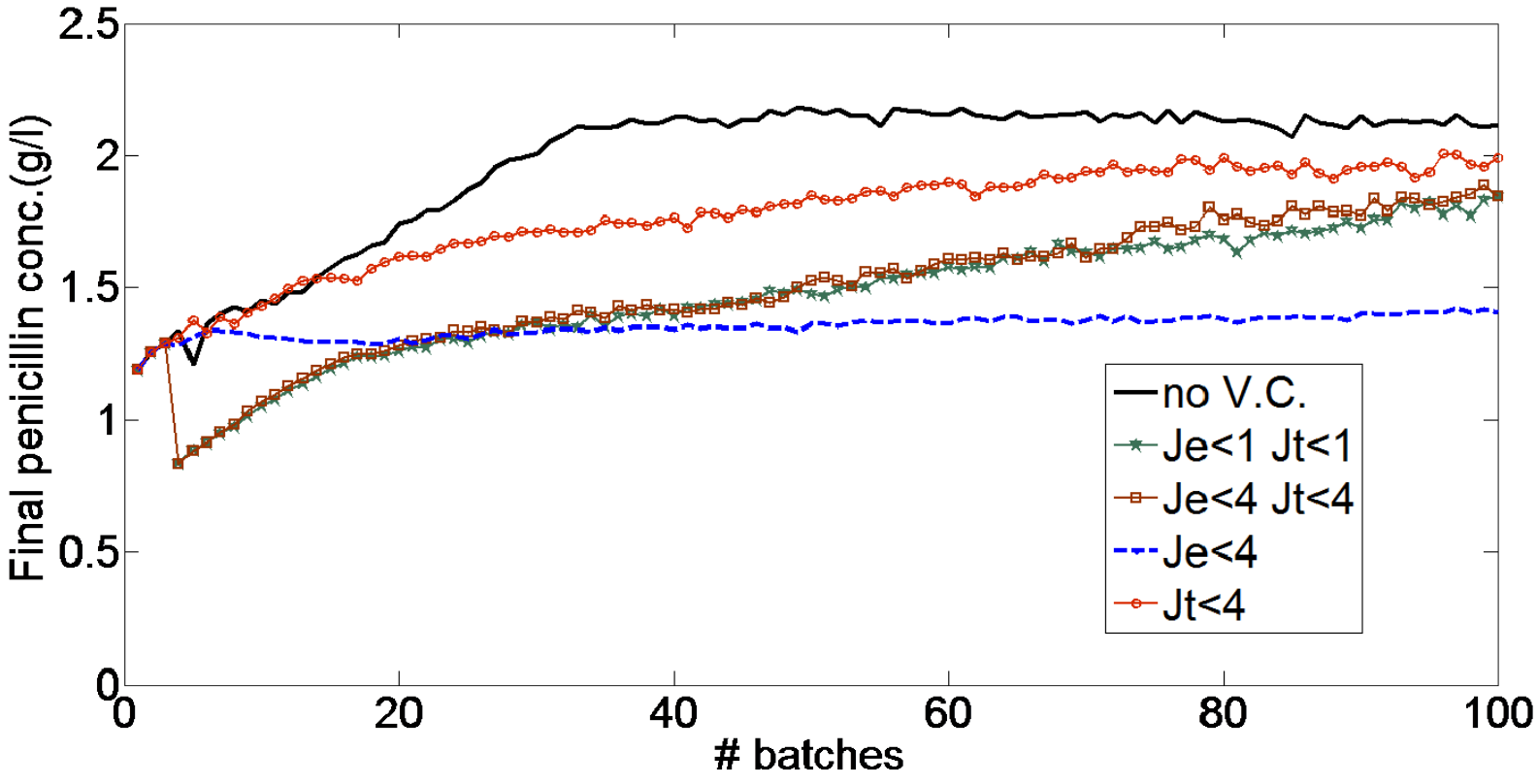

Final penicillin conc.(g/l)
2.5
2
1.5
1
0.5
0
0
20
40
60
80
100
batches
no V.C.
Je<1 Jt<1
Je<4 Jt<4
Je<4
Jt<4

Figure 5

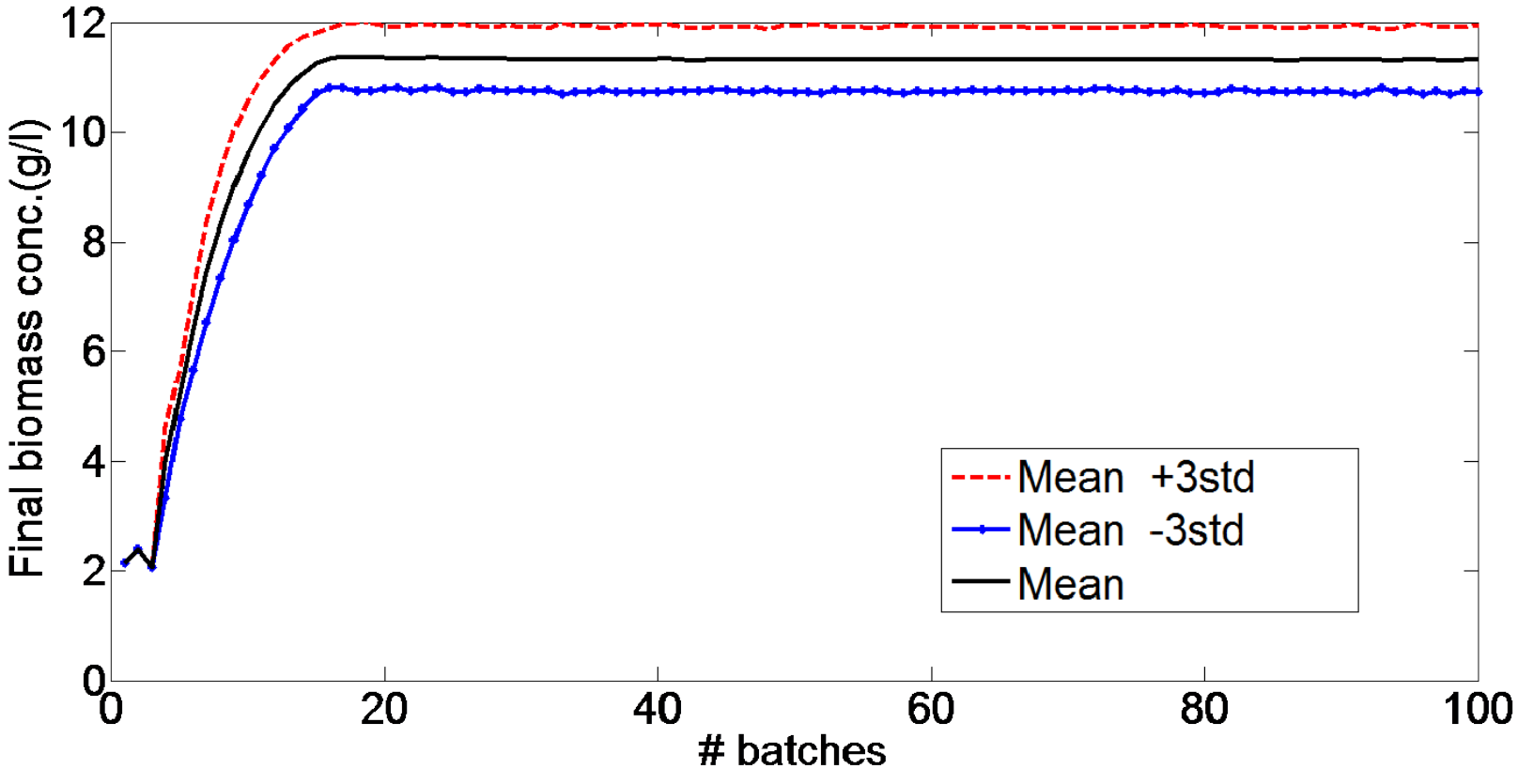
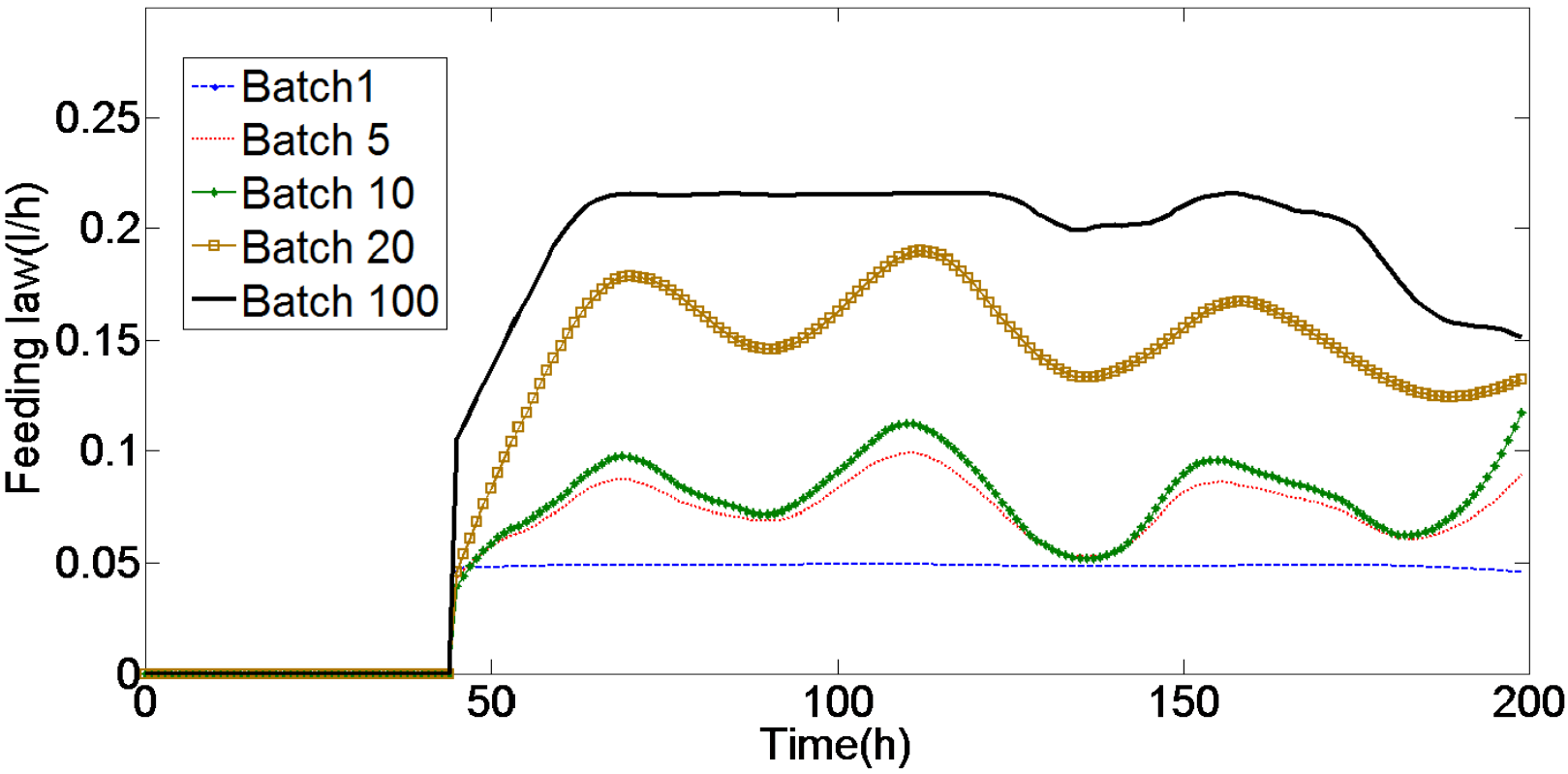

Final biomass conc.(g/l)
batches
0
2
4
6
8
10
12
0
20
40
60
80
100
Mean +3std
Mean -3std
Mean

Figure 9


Batch1
Batch 5
Batch 10
Batch 20
Batch 100
Feeding law(l/h)
0
0.05
0.1
0.15
0.2
0.25
0
50
100
150
200
Time(h)