Iterative Learning based Grey Box modelling of Batch Processes


*Du-Pont Research Centre, Hyderabad, India (e-mail: kondeks@rediffmail.com)
**Indian Institute of Technology, Bombay, Mumbai, India (e-mail: mohitpoddar35@gmail.com)
***McMaster University, Hamilton, Canada (e-mail: patelsr@mcmaster.ca)
****Indian Institute of Technology - Bombay, Mumbai, India (e-mail: ravigudi@iitb.ac.in)

Abstract:
Performance control of batch processes using model based approaches requires the deployment of high fidelity models. However, due to a variety of reasons that stem from short process development cycles and the consequent poor first principles knowledge about the process, such accurate models are difficult to develop. Another daunting aspect of any modeling activity is related to a lack of structural knowledge about the parameter-state dependencies. Furthermore, the changing dynamics resulting from scale-up in manufacturing could render the models developed at the lab scale to be obsolete. All of the above pose a requirement to update the model on a timely basis so as to achieve tight control of the batch processes. In this paper, we propose the use of (i) an iterative learning based model refinement approach that is based on the principles of continuous improvement paradigm of GMP, and (ii) grey box models to represent the cause-effect relationships, to address the above challenges. The proposed iterative learning based estimation approach has been validated on simulations and experimental data from representative batch processes.

1. INTRODUCTION

Batch processes hold an important position in the chemical industry, especially in the manufacturing of specialty chemicals, pharmaceutical products, and polymers. Optimal control of batch processes is essential to produce consistent quality product in the presence of various disturbance. Batch processes exhibit several peculiar features that are not seen in continuous processes. The most conspicuous of these is the dynamic nature of operation in which variable relationships change over a relatively large range throughout the course of a batch; additionally such relationships could also be nonlinear. Moreover, a typical batch process consist two or more process phases (due to induction or transition to fed-batch) as well as stages. Due to presence of these processing stages, the process dynamics could exhibit sharp discontinuities. Finally, batch processes operate over finite (but variable) durations; this feature therefore necessitates that during online operation, any optimizing control strategy needs to be designed for the case of a shrinking time horizon. All of the above characteristics add more complexities in the task of performance control of batch processes.

The above mentioned complex nonlinear dynamics render linear control strategies to be largely ineffective; non-linear model based controllers can be expected to perform better. For these controllers, accurate process models are usually required for the purposes of defining optimal control policies for the batch runs. Also, the reference trajectory can be more aggressively designed and achieved if the model is accurate. Furthermore, it is well known that, from stability and performance viewpoints, it is mandatory to have high fidelity models in the shrinking horizon control formulation.

Generally, batch processes are represented by mathematical models which can be classified as knowledge-based models, data-based models or combinations of both. Knowledge-based models, also referred to as mechanistic, first principles, or white-box models, are based on a description of the process phenomena and underlying physics and chemistry. Given the short process development cycles and the paucity of data, these models are difficult to develop and also do not scale up. However, they do contain relevant structural information and are therefore interpretable and have reasonably good extrapolation properties depending on the scale of application. Data-based or black-box models, on the other hand, represent the variable relationships using an empirical model structure, the parameters of which are estimated from the data. Reasonably accurate models are generally easy to obtain without detailed knowledge of the system, but they may suffer from weak extrapolation properties. The choice for either of the above model types depends on the modeling goal and the available process knowledge and data. Another class of model called grey box models have been proposed and with respect to modeling rigor, these fall in between first principles and black box models. In the grey box model, the non-linear structural relations which are exactly known from the physics of the problem are retained (e.g. mass balances and energy balances etc), but empirical models are used for the unknown relationships (e.g. reaction rates dependencies, enzyme mechanism dependencies etc). Again, due to short process development cycles and the resulting paucity of experimental data, the estimation of the parameters in the empirical models is always a concern. The use of parsimonious representations and continuous model refinement approach could help address this concern.

For repetitive processes/tasks, iterative learning approaches have recently found increasing research interest. For batch processes as well, this learning paradigm has found
significant acceptance for control of the batch trajectories along specified desired trajectories. Starting from a nominal control policy, the iterative learning control (ILC) algorithm takes advantage of repetitive nature of batch process, and learns from the inter-batch tracking errors. The nominal control policy is continuously updated at the end of every batch run so as to achieve convergence between the batch state and the reference trajectories. Lee and Lee (1) have provided a detailed review of ILC for batch process control. To perform the state estimation and parameter identification, adaptive observers have also been extensively studied (Bastin et al. 1988, Friendland 1997). It is important to recognize however that the design of optimal reference trajectories is dependent on the availability of a high-fidelity model, and therefore there is a strong incentive for iterative refinement of the batch model, so as facilitate the overall optimization of the batch process.

In this paper, we propose an approach based on iterative learning to continuously update a suitable parameterized grey box model using inter-batch prediction error trajectories. Inspired from the successes in the iterative learning control approaches that leverage batch-to-batch information to minimize control errors, and recognizing the duality that exists between estimation and control, this paper proposes an iterative learning estimation (ILE) approach to identify parameters of non-linear dynamic batch system using observations from batch to batch. The ILE algorithm proposed in this paper learns from information from batch to batch by minimizing the prediction errors. Two key novelties proposed in the paper include (i) the proposition of a grey box model to account for the poorly known dependence of the parameters on the states, and (ii) use of functional space approximation methods towards parsimonious representation of the nonlinear dependence of the batch parameters on the states. It is shown that the proposed method can exploit information available from relatively abundant laboratory scale research data as well as plant data from previous batches to train the model. The proposed approach is validated extensively using simulations as well as experimental data involving a *Pichia Pastoris* fermentation.

The paper is organized as follows: The ILE methodology is discussed in the next section. The dual relationship between ILC and ILE is also presented. The framework of ILE algorithm is presented in section 3. Finally, validation results obtained using the proposed method when applied on the above mentioned fed-batch fermentation process is discussed in section 4, followed by the conclusions.

### 2. ITERATIVE LEARNING APPROACH

As discussed earlier, due to the repetitive nature of batch processes, it is possible to improve the operation of the next batch using the information of the current and previous batch runs. ILC learns iteratively using the past information and refines the control inputs to improve the current operation. It updates the parameters of the control policy such that the deviation of actual process output from the reference trajectory will be minimized. As discussed in the introduction, different approaches have used observations within batch to estimate either states or parameters. However, information from archived batch runs can be exploited in different ways using iterative learning towards enhancing the overall productivity of the batch operation. Depending on the time horizon of the trajectories considered, either the development of short-term intra-batch adaptive observers could be achieved. Alternatively, on a longer basis, the development of a continuous refinement approach that uses complete batch records for the updating of dynamic models of the batch could be targeted. In this section, such an iterative learning estimation (ILE) approach is proposed to identify and update parameters of the non-linear dynamic system using observations from batch to batch. As the ILE method learns from data from previous runs, it can easily exploit abundant laboratory scale research data as well as plant data from previous batches to train and refine the model.

#### 2.1 ILE Methodology

Consider a system represented by

$$\dot{x} = f(x,u,\theta)$$  \hspace{1cm} (1)

where $x_{n+1}$ represent system states, $u_{n+1}$ represent control inputs, $\theta_{n+1}$ represent system parameters. The output variables $y_{n+1}$ can be related to the system states, control inputs and system parameters as:

$$y = g(x,u,\theta)$$  \hspace{1cm} (2)

**Process Model:** The system output can be related to control inputs for batch process duration of $N$ instants as follows:

$$Y^k = GU + Y^0$$  \hspace{1cm} (3)

where $Y$ is the system output vector of dimension $Nm \times 1$; $G$ is the input-output relationship (process model) matrix of dimension $Nm \times Nq$; $U$ is the control input vector of dimension $Nq \times 1$; $Y^0$ is the initial output vector of dimension $Nm \times 1$

$$y^k = \begin{bmatrix} y^k(1) \\ y^k(2) \\ \vdots \\ y^k(N) \end{bmatrix}, \quad U^k = \begin{bmatrix} u^k(1) \\ u^k(2) \\ \vdots \\ u^k(N) \end{bmatrix}, \quad Y^0 = \begin{bmatrix} y^0(1) \\ y^0(2) \\ \vdots \\ y^0(N) \end{bmatrix}$$  \hspace{1cm} (4)

where $k$ defines the batch run

Let us define the reference trajectory as:

$$y^{ref} = \begin{bmatrix} y_{r}(1) \\ y_{r}(2) \\ \vdots \\ y_{r}(N) \end{bmatrix}$$  \hspace{1cm} (5)

The error between the actual plant output and the reference trajectory after $k^{th}$ batch run can be defined as:

$$e^k_c = y^k - y^{ref}$$  \hspace{1cm} (6)

The iterative control algorithm changes the control input $U$ such that the error $e_c$ progressively approaches zero with increasing number of batches. It uses a first order or a higher
order learning update mechanism to update the control input 
$U$. The most commonly used, first order learning equation can be written as,

$$U^k = U^{k-1} + H_c e_c^{k-1}$$  \hspace{1cm} (7)$$

where $H_c$ is learning filter matrix of dimension $N_q \times N_m$ generally chosen as $G' \prod$ or can be tuned to achieve arbitrary rate of convergence of $e_c^k$.

For repetitive processes of finite duration $N$, the ILC claims to achieve

$$|e_c^k| \rightarrow 0 \text{ as } k \rightarrow \infty$$

i.e. the norm of the control error trajectory goes to zero with an increase in the $k$, the number of batches. Taking inspiration from ILC, in this paper we propose an iterative learning based approach to estimate the model parameters for the batch processes. Here, the objective is to update the system model parameters such that the actual plant output will match with the model predictions. Since the algorithm learns iteratively from previous runs to update model parameters, the method is termed as Iterative Learning Estimation (ILE).

The iterative learning estimation (ILE) can be looked as a dual of ILC as follows:

<table>
<thead>
<tr>
<th>ILC</th>
<th>ILE</th>
</tr>
</thead>
<tbody>
<tr>
<td>Manipulation relationship $Y = GU$</td>
<td>Affine parameter relationship $Y = P\theta$</td>
</tr>
<tr>
<td>Minimize tracking error $</td>
<td>e_c^k</td>
</tr>
<tr>
<td>Control update relationship $U^k = U^{k-1} + H_c e_c^{k-1}$</td>
<td>Parameter update relationship $\theta^k = \theta^{k-1} + H_p e_p^{k-1}$</td>
</tr>
</tbody>
</table>

where, $P$ relates the system parameters to process output. The dimension of $P$ is $N_m \times N_p$; $\theta$ is the system parameter vector at $k$th batch run of dimension $N_q \times 1$, $H_c$ is the gain matrix.

The error $e_p$ is defined as,

$$e_p^{k-1} = Y_M^k - \hat{Y}_M^k$$ \hspace{1cm} (10)$$

where, $\hat{Y}_M^k$ is the model predicted output vector estimated at $\theta^{k-1}$; $Y_M^k$ is the actual plant output vector at $k$th batch run of size $N \times m$

**Special case – Linear system:** If the relationship between the process parameters and the output i.e. $P$ is known and is nonsingular then it can be shown that the iterative learning approach will converge to the true parameters in exactly one iteration if $H_c = P^{-1}$.

Multiplying Eq. (9) by $P$,

$$P\theta^k = P\theta^{k-1} + PH_p \left( Y_M^k - \hat{Y}_M^k \right)$$ \hspace{1cm} (11)$$

using $PH_p = I$, $P\theta^k = \bar{Y}_p^k |_{\theta^k}$ and $P\theta^{k-1} = \bar{Y}_p^{k-1} |_{\theta^{k-1}}$

$$\bar{Y}_p^{k-1} |_{\theta^{k-1}} = \bar{Y}_p^{k-1} + \bar{Y}_M^{k-1} - \bar{Y}_p^{k-1} PH_p = I$$ \hspace{1cm} (12)$$

$$\bar{Y}_p^k |_{\theta^k} = Y_M^k$$ \hspace{1cm} (13)$$

In the absence of accurate process model $P^i$, $H_p$ can be tuned to achieve arbitrary rate of convergence as follows:

**Gain Matrix ($H_p$) derivation:**

The dynamic balance equation for any batch process can be represented in the following general form as:

$$\frac{dX}{dt} = f(X, \theta) + g(X, u)$$ \hspace{1cm} (14)$$

If the above equation is discretized using a simple Euler discretization at $\theta$ time instant,

$$X^{i+1} - X^i = f(X^i, \theta) + g(X^i, u^i)$$ \hspace{1cm} (15)$$

The model parameter ($\theta$) is assumed to be constant over entire batch operation.

As $f(x_i, \theta)$ is non-linear in $\theta$, it can be linearized at $\theta_i$ the parameter vector estimated at $k$th iteration:

$$f(X^i, \theta) = f(X^i, \theta_i) + \frac{df(X^i, \theta)}{d\theta} (\theta - \theta_i)$$ \hspace{1cm} (16)$$

After substituting Eq. (16) in Eq. (15) and writing it for $k$th batch as follows

$$X_i^{i+1} - X_i^i = f(X_i^i, \theta_i) + \frac{df(X_i^i, \theta_i)}{d\theta} (\theta - \theta_i) + g(X_i^i, u_i^i) \Delta t$$ \hspace{1cm} (17)$$

If system is discretized in $n$ intervals, the above equation can be rewritten for all discretized points in matrix form as:

$$\begin{bmatrix}
X_i^1 \\
X_i^2 \\
\vdots \\
X_i^{i+1}
\end{bmatrix} = \begin{bmatrix}
f(X_i^1, \theta_i) - \frac{df(X_i^1, \theta_i)}{d\theta} \theta_i + g(X_i^1, u_i^1) \\
f(X_i^2, \theta_i) - \frac{df(X_i^2, \theta_i)}{d\theta} \theta_i + g(X_i^2, u_i^2) \\
\vdots \\
f(X_i^{i+1}, \theta_i) - \frac{df(X_i^{i+1}, \theta_i)}{d\theta} \theta_i + g(X_i^{i+1}, u_i^{i+1})
\end{bmatrix} \Delta t$$ \hspace{1cm} (18)$$

$$Y = b + D\theta$$
The relationship is linear in terms of model parameter \( (\theta) \) which can be estimated using linear least square analysis as,

\[
\hat{\theta} = \left[ D^T W D \right]^{-1} D^T W (y - b) = P D^T W (y - b) \tag{19}
\]

As new sets of data arrive, the model parameters can be recalculated. The matrix inversion is computationally very inefficient, so the model parameter is updated based on the new information rather than recalculation. The recursive form of parameter estimation can then be derived as,

\[
\hat{\theta}_{k+1} = \hat{\theta}_k + H_{k+1} \left( \tilde{y}_{k+1} - b_{k+1} - D_{k+1} \hat{\theta}_k \right) \tag{20}
\]

where \( H_{k+1} = P_{k+1} D_{k+1}^T W_{k+1} \), and \( P_{k+1}^{-1} = P_k^{-1} + D_k^T W_k D_k + D_k^T W_{k+1} D_{k+1} \)

The configuration of the iterative learning estimation algorithm is shown in Fig.1.

**Fig. 1: Iterative Learning Estimation Algorithm**

### 2.2 Parsimonious Representation of Parameter – State dependency

Often times the parameter vector is dependent on the states in a non linear fashion which is typically unknown. In this paper, we propose to represent this unknown dependence using orthonormal basis functions to bring in parsimony in represent. As discussed in the case study, these basis functions offer a rich but compact way of representing the non linear dependency.

### 3. CASE STUDY

The application of proposed grey-box modeling approach using iterative learning estimation is illustrated with a fed-batch *Pichia-Pastoris* fermentation process. This fermentation is the workhorse of the recombinant protein industry, and the protein under consideration is the HCG protein production.

Fermentation runs were conducted with a batch phase growth on glycerol for approximately 24 hours followed by a fed-batch phase also on glycerol to achieve exponential growth for about 8-14 hours. After these two phases, the recombinant protein was induced by switching to growth on methanol as substrate for about 40-60 hours of batch time.

Data from a total of 11 fermentation runs were considered for the evaluation of the grey box modeling approach.

#### 3.1 Model Development

Data from the above fermentation runs were used to build the following two models:

**First Principles Model**

The dynamic mass balances for the biomass, the substrate and the broth volume can be written as,

\[
\frac{d(X)}{dt} = \mu X \\
\frac{d(S)}{dt} = -\frac{\mu}{Y_{X/S}} X V + F S_f \\
\frac{d(V)}{dt} = F
\]

where \( X \) is the biomass concentration (g/L), \( V \) is the volume (L), \( F \) is the substrate feed rate (L/hr), \( S \) is the substrate concentration (g/L); \( S_f \) is the feed substrate concentration (g/L), and \( \mu \) is the specific growth rate (hr⁻¹).

For the first principles models, the dependence of the specific growth rate was assumed to be exactly known and described by,

\[
\mu = \frac{\mu_{\text{max}} S}{K_s + S + \frac{S^2}{K_i}} \tag{22}
\]

The parameters of the above dependencies were estimated from the data using procedures outlined in Konde (2008).

**Grey Box Model**

The key difference in the grey box model representation when compared with the first principles model is the description of the parameter dependencies. While the mass balances are assumed to be exactly known, the dependence of the specific growth rate on the culture states (substrate concentrations) is assumed to be described by the following empirical relationship,

\[
\mu = a_0 L_0 + a_1 L_1 + \cdots + a_9 L_9 \tag{23}
\]

where, \( L_0, L_1, \ldots, L_9 \) are orthogonal polynomial functions, which are the function of \( S \) (substrate concentration) and are defined as follows.

\[
L_0 = 1, \quad L_1 = S, \quad L_2 = \frac{3S^2 - 1}{2}, \ldots,
\]

\[
L_9 = \frac{12155S^9 - 25740S^7 + 18018S^5 - 4620S^3 + 315S}{128}
\]

Due to orthogonal nature of the polynomials, the above representation provides a compact relationship structure and so requires less number of data to estimate the model parameters.

Traditionally, the process model is validated by comparing the predicted and actual output variables trajectories. Unlike previous approach, here the proposed grey-box modeling approach offers a compact and efficient way to model the fermentation process.
approach is validated using an online trajectory optimization study. In this study, the optimal trajectories of input variables are defined by subjecting the model to optimization algorithm. Therefore, in the optimization study, the model is evaluated over the entire region in order to obtain the best optimal trajectory. Thus it provides an efficient way to validate the model as the trajectory definition will not be optimal in true sense if the model is not accurate over the entire operation regions.

3.2 Results and Discussions

The process model is represented by grey box model (Eq. 21,23) where all known mass balances are retained in their original form but only the unknown kinetic relation (specific growth rates) are represented by polynomial functions. The grey-box model parameters are estimated using experimental data such that the error between actual and predicted output will be minimized. The model parameters are updated iteratively as the new data from a completed batch arrives. Fig 2 shows the evolution of parameters with number of iterations. The number of iteration represents number of number of new batch runs that arrive on completion of a batch. It can be seen from Fig 2 that the model parameters progressive converge as soon as additional information from a completed batch is available. The parameters converge to constant values after the 5th iteration, i.e. at the completion of the 5th batch.

Fig. 2: Evaluation of parameter

Fig. 3 shows how the model is refined iteratively to match the true dynamics. It can be seen from Fig. 3 that after the first iteration, the predicted value of output variables, biomass (X) and substrate (S) trajectories are not in good agreement with the actual trajectories. As the iterations progress, the model gets refined in each iteration and after 6th iteration the model output exactly matches with the true values.

Fig. 3: Performance of ILE on experimental data for Pichia-Pastoris Fermentation

To validate the grey-box model the following two additional case studies have also been performed via simulations

Case 1: Effect of Initial substrate level

The initial glycerol substrate level is decreased from 40 g/L to 20 g/L. The grey-box model is used to predict the culture states and output. The predicted values of the outputs (substrate concentration and biomass concentration profiles) are compared against actual output, here generated from the first principle (FP) models. Fig. 4(a) to 4(c) shows that predicted value matches very accurately with FP models. For 4(d) shows the substrate feeding policy during fed-batch phase.

Fig. 4: Model validation: Effect of initial substrate level

Case 2: Effect of feeding policy

In this case the substrate feeding policy is changed and model outputs are predicted using grey-box model. The initial
substrate level is kept constant at 40 g/L. Fig. 5 shows comparison of predicted vs actual output variables profiles. Fig. 5(a) and (b) shows the actual and predicted output variables profiles while Fig. 5(c) and (d) shows the corresponding feeding strategy. Depending on feeding policy, the fermentation process shows different dynamics. It can be seen from the Fig. 5(a) and (b) that predicted output match very closely with actual values, which confirms the ability of grey box model to capture dynamics of process.

These validations confirm the efficiency of ILE algorithm for parameter estimation.

Fig. 5: Model validation: Effect of feeding policy
To check the fidelity of model, optimal trajectory studies are also carried out using derived grey-box model. Fig 6 shows the result of optimal control case study.

Fig 6 : Model validation : Optimal trajectory study
The Genetic Algorithm (GA) based optimization framework is used derive for optimal trajectory. Fig 6 shows that the optimal trajectory derived using grey-box model matches very well with those derived from first principle based process model.

4. CONCLUSIONS
A method for grey-box modeling for batch process is proposed. The complex unknown kinetics are represented by orthogonal functions set. The technique based on iterative learning principle is proposed to estimate the grey-box model parameters. To minimize the online computation load for parameter estimation, a recursive form of the parameter update equation is derived. Finally, the proposed approach is validated using experimental data and optimal control study of *Pichia-Pastoris* fed-batch fermentation process. Further work in this area is currently being focused towards integration of the proposed ILE and the well established ILE algorithm. This integrated approach is expected to facilitate re-defining of the reference trajectory based on ILE updating followed by aggressive tracking of this re-define trajectories using ILC. This will be subject matter of future publication (Poddar et. al. 2010).

REFERENCES