Article

Design of Dynamic Experiments in Modeling for Optimization of Batch Processes

Ernesto C. Marti#nez, Mariano D. Cristaldi, and Ricardo J. Grau

Ind. Eng. Chem. Res., **2009**, 48 (7), 3453-3465• DOI: 10.1021/ie8000953 • Publication Date (Web): 09 March 2009

PROCESS DESIGN AND CONTROL

Design of Dynamic Experiments in Modeling for Optimization of Batch Processes

Ernesto C. Martínez,*,† Mariano D. Cristaldi,‡ and Ricardo J. Grau‡

INGAR (CONICET−UTN), Avellaneda 3657, Santa Fe S3002 GJC, Argentina, and INTEC (CONICET−UNL), Guemes 3450, Santa Fe 3000, Argentina

Finding optimal operating conditions fast with a scarce budget of experimental runs is a key problem to speeding up the development of innovative products and processes. Modeling for optimization is proposed as a systematic approach to bias data gathering for iterative policy improvement through experimental design using first-principles models. Designing dynamic experiments that are optimally informative in order to reduce the uncertainty about the optimal operating conditions is addressed by integrating policy iteration based on the Hamilton−Jacobi−Bellman optimality equation with global sensitivity analysis. A conceptual framework for run-to-run convergence of a model-based policy iteration algorithm is proposed. Results obtained in the fed-batch fermentation of penicillin G are presented. The well-known Bajpai and Reuss bioreactor model validated with industrial data is used to increase on a run-to-run basis the amount of penicillin obtained by input policy optimization and selective (re)estimation of relevant model parameters. A remarkable improvement in productivity can be gain using a simple policy structure after only two modeling runs despite initial modeling uncertainty.

1. Introduction

The best use of a model through proper handling of its inherent uncertainty is a recurrent issue in the vast literature related to optimization methods for batch processes.[1−22] There are two extreme assumptions which can be made regarding modeling uncertainty and available measurements in model-based dynamic optimization.[14] One of the idealized situations is the *perfect model* assumption. In this case, it is assumed that it is feasible to find an optimal parametrization of a structurally correct model comprised of all thermodynamic and kinetic relationships required to comprehensively describe the batch process dynamics of interest. Under this postulate there is no need for optimal operation to measure and feedback key variables from the batch process. The nominal optimal policy derived from the perfect model is applied "open loop" since it is robust enough to compensate in advance for any source of process variability and to anticipate the detrimental effect of any disturbance. The other extreme situation is the *comprehensive instrumentation* condition. This condition refers to the case where all state variables of the process can be readily measured online with sufficient accuracy and frequency. If everything that matters can be properly measured, it can be easily argued that the need for an accurate first-principles model is lessened and measurement-based dynamic optimization[6] is a much more appealing alternative.

Unfortunately, neither of the above idealized assumptions is valid in batch industrial environments.[1−3] Thus, migration from laboratory conditions to production runs is often made with high levels of uncertainty about the best input policy. *Modeling for optimization*[3−5] has become of crucial concern as the limited number of costly runs performed during process development and scale-up leave little room for accurate modeling of batch processes. One central concern in modeling for optimization is how experimental design in the model development life cycle should be addressed considering poor knowledge about phenomena involved, poor reproducibility of batch run outcomes, and modeling bias inherited from laboratory scale experimentation. Modeling for optimization constitutes thus a shift from the traditional modeling approach based on the separation between a process model and its intended use for simultaneous model development and process optimization. In the latter case, the purpose of modeling is to bias experimentation and data gathering to help achieve a near-optimal operating condition. Different model hypotheses comprising structure and parametrization upon selectively obtained data are postulated, refined, and discharged in the search for iterative improvement of the operating policy. Each new experiment in the sequence of batch runs must be able to bring new data to significantly reduce the uncertainty regarding optimal operation.

In modeling for optimization, it is convenient to differentiate between two building blocks for model identification: (i) first-principles hypotheses about the inner workings (conservation equations, constitutive laws, etc.) of the process and (ii) data bias that characterizes the actual (observed) behavior of the process under study through parameter identification. As a result, model development for batch process optimization cannot be entirely knowledge-driven or measurement-driven alone. Instead, the strategy for model development should necessarily be one that takes the best of both worlds. One avenue for doing this in a reaction system (the most ubiquitous for batch systems), initially proposed by Georgakis et al.,[7,8] is *tendency modeling*. A "tendency model" is a low-order, nonlinear, dynamic model that approximates the stoichiometry and kinetic relationships of a process using the available plant data along with fundamental knowledge of the process characteristics. The model structure and parameters are incrementally updated as more data become available.

The main use of a tendency model is to determine a direction toward the optimum. For this to be feasible, the model should

* To whom correspondence should be addressed. Tel.: +54 (342) 4534451. Fax: +54 (342) 4553439. E-mail: ecmarti@santafe-conicet.gov.ar.

† INGAR.

‡ INTEC.

10.1021/ie8000953 CCC: $40.75
Published on Web 03/09/2009

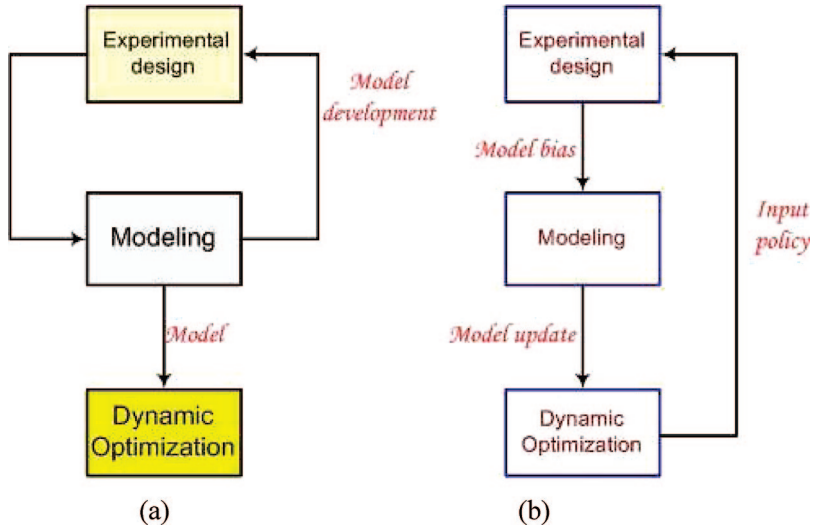

Figure 1. Integration of model development with input policy optimization. (a) Traditional approach. (b) Modeling for optimization approach.

be able to extrapolate to operating conditions quite different from those initially used for model identification. First-principles and constitutive laws incorporated while building the model hopefully provide this capability by constraining the degrees of freedom available for data fitting. Since tendency modeling was first proposed,[7,8] a number of new developments and successful applications have been reported.[9−20] The issue of model structure identification from process data has attracted a great deal of attention.[10−17] Singular value decomposition[10,11] and structured target factor analysis[13,18−20] have proved to be very successful tools for many examples. Also, a methodology to quantify the impact of model parametric uncertainty on the uncertainty of the predicted optimal operating condition has been developed.[15,18−20]

In order to achieve the goal of optimal operation of batch processes in the face of modeling uncertainty, a number of requirements are imposed on modeling for optimization to make an impact on industrial practice. A key issue is how to design dynamic experiments that are optimally informative in order to reduce the uncertainty about the most profitable input policy. As data gathering is biased toward the most profitable operating condition, model parameters and structure may conveniently be updated. Despite the importance of this problem, there is no previous work on the development of experimental design techniques addressing the more specific objective of model development for optimization. The problem of increasingly biasing data gathering in modeling for optimization is formulated here as follows: How does one adjust the time-varying controls, initial conditions, and length of each dynamic experiment to generate the maximum amount of information for the purpose of significantly reducing the uncertainty regarding the location of the process optimum? The notion of a dynamic experiment[23] highlights the fact that some control variables are time-varying during the experiment, which rules out using standard experimental design techniques such as response surface methodology. In this work, systematic design of dynamic experiments is aimed at not reducing parametric uncertainty comprehensively but at the more specific objective of process optimization rather than accurate modeling.

2. Modeling for Optimization

Most optimization techniques are model based, and since reliable models are rarely available for batch processes, seeking to operate them optimally is a difficult problem to solve.[1,3,27] In the attempt to compensate for any process−model mismatch, optimal operation under uncertainty requires using measurements from carefully designed experiments to improve on a run-to-run basis from an initial input policy. The standard procedure consists of iteratively using new measurements to increasingly bias model parameter estimation and later resorting to the updated model for policy improvement.[24] The underlying idea of "modeling for optimization"[3,27] is to identify an imperfect model that allows computing inputs that are nearly optimal for the process. In contrast, the goal of parameter precision in detailed process modeling is to come up with a model that, for given inputs, can accurately predict the outputs of the process over a wide range of operating conditions.[25] Since the utility of model for a system or process must be assessed with regard to a purpose, in modeling for optimization the model is understood as a means to find a near-optimal operation policy despite incomplete understanding of process dynamics and uncontrollable disturbances affecting the state evolution of a batch. Thus, the process model is not an end in itself as it is in kinetic studies[26] where experiments are designed to reduce parametric uncertainty in an accurate model.[23,25]

The main objective of modeling for optimization is stated here as designing informative experiments to guide the search for the optimum operating policy with minimum experimental effort based on tendency models. The model identification strategy chosen for reducing parametric uncertainty will influence greatly the number of modeling runs, the cost involved, and the length of time required to accomplish the objective of near-optimal operation. Ideally, data gathering and model identification need to be progressively biased toward the most profitable operating policy by selectively reducing the uncertainty in model parameters that affect significantly the chosen performance index. As expected, such bias will give rise to a model parametrization which is less capable of providing accurate performance predictions when the process is run far from normal operating conditions. Accordingly, model identification bearing in mind optimization differs significantly from increasing parameter precision in detailed model development. It is worth noting that if there is process−model mismatch due to structural errors, as is often the case in batch processes, seeking to improve parameter precision may become a futile and very costly undertaking from the point of view of input optimization.[24]

As shown in Figure 1, modeling for optimization proposes an entirely different approach for relating experimental design with model-based dynamic optimization.[5] Traditionally (see

Figure 1a), an accurate process model is developed first and, only once the model parameters have been comprehensively validated, dynamic optimization is undertaken. Consequently, a great deal of experimentation is spent aiming to reduce both structural and parametric uncertainty by obtaining data over a wide range of operating conditions which permit derivation and testing of an accurate model hypothesis. This course of action is acceptable only if choosing far from optimal operating policies is not an issue from the safety or economy point of view and when the budget for experimentation is not tightly constrained in terms of time and money. However, due to an incomplete understanding of the inner workings of the process and limited room for exploring dangerous or low performance operating conditions, model development for optimization should be conveniently integrated with policy iteration to quickly bias data gathering toward near-optimal operation as shown in Figure 1b. Experimental design for optimization thus has a very different scope and objectives to satisfy for uncertainty reduction as emphasized in the next section. It is worth noting the shift from experimental design for comprehensive parameter precision to the design of experiments seeking to increasingly reduce uncertainty for run-to-run policy improvement based on tendency models.

As was pointed out by Srinivasan and Bonvin in ref 24, model parameters may converge to values that minimize errors in model outputs, but there is no guarantee that this parameter precision give rises to accurately predicting the performance index and constraints of the optimization problem. As a result, if the objective and constraints of the optimization problem are not predicted properly, the solution obtained by optimizing the model does not necessarily optimize the process. To address this issue, it was proposed in ref 24 to modify the objective function of the identification problem to include the cost function and the constraints of the optimization problem. The weights of the various terms in the extended objective function are based on Lagrange multipliers. Optimization bias can also be introduced by including optimality conditions in the parameter estimation problem, as has been proposed by Zhang and Forbes.[27] This solution is somewhat similar to a well-known approach used to solve the "dual control" problem in the area of system identification and adaptive control (see ref 24 and references therein). Even though modifying the estimation problem may help reduce the *performance prediction mismatch* the issue still remains of persistent excitation of inputs used in modeling runs. This problem becomes much harder as the number of parameters and their degree of uncertainty are increased. To this aim, it is proposed here that data sampling along each dynamic experiment must account for the sensitivity of performance prediction when selectively reducing parametric uncertainty. Also, persistent excitation in modeling for optimization requires addressing the tradeoff between exploitation (optimization) and exploration (identification).[4]

Optimizing a real process using the iterative approach of Figure 1b poses the question of convergence of the input policy toward an optimum. Assuming the model has the right structure, it can be shown that the performance prediction mismatch can be narrowed down to the intrinsic variability of the process and a local near-optimal policy will be found (see section 5). However, due to noisy data and optimization constraints, the performance response surface is typically multimodal and nonsmooth, which makes it very difficult to guarantee global optimality of the iterative identification−optimization loop in Figure 1b. Lacking the correct model structure, the performance prediction mismatch may not be reduced to the intrinsic process variability, yet the iterative approach will converge to a significantly improved but not necessarily plant optimal policy (see section 5). A possibly avenue to overcome policy suboptimality would be resorting to a library of alternative tendency models for experimental design which allows including explicitly a safe exploration dimension in each iteration of the modeling for optimization strategy.

3. Experimental Design for Optimization

In modeling for optimization, it will be assumed hereafter that initially the model predictive capability of a performance index J along state trajectories induced by alternative operating policies is qualitatively correct but quantitatively uncertain due to model parametrization errors and data bias. Model discrimination to handle uncertainty regarding model structure in modeling for optimization will be addressed in a forthcoming paper. Thus, only model parameters will be updated as new data are gathered in policy evaluation experiments. Modeling for optimization thus revolves around iteratively improving the input policy based on the proper design of dynamic experiments upon which process performance is incrementally improved on a run-to-run basis and parametric uncertainty is increasingly reduced by an appropriate selection of model parameters to be reestimated.

3.1. Model-Based Policy Iteration. In what follows let us assume that the dynamic behavior of the batch process under study is modeled by the set of ordinary differential equations (ODEs)

$$\frac{\mathrm{d}x}{\mathrm{d}t} = f(x(t), \mathscr{P}(w,t), \theta); \quad 0 \le t \le t_\mathrm{f},\ x(0)\text{: given} \qquad (1)$$

and the optimization objective to be minimized is

$$J(t_\mathrm{f}, x) = h(x(t_\mathrm{f})) + \int_0^{t_\mathrm{f}} g(x, \mathscr{P}(w,t))\,\mathrm{d}t \qquad (2)$$

where $x(t)$ is an n_s-dimensional vector of time-dependent state variables, w is an m-dimensional vector of parameters for the input policy $\mathscr{P}$, $\theta \in \Theta$ is a p-dimensional vector of model parameters, and t_f is the final time of a batch run which in turn may also be optimized. The initial definition of the parameter space Θ is defined by some a priori knowledge or belief. The function g is the instantaneous cost function along the state trajectory defined by a given policy parametrization, whereas the function h is the specific cost for the final state of the batch run. It is worth noting that eq 2 defines the *cost-to-go* of a policy parametrization in the dynamic programming jargon.[28] Accordingly, a policy defined by the set of parameters w_2 with value J_2 is better than (or preferred to) a policy defined by w_1 with value J_1 if and only if $J_2 < J_1$. The sensitivity of process performance to policy parametrization is a central issue for designing optimally informative dynamic experiments to bias data gathering in modeling for optimization and when deciding which subset of model parameters should be reestimated using data gathered in the current iteration.

For a given model parametrization θ, the optimal policy parametrization $\mathscr{P}(w^*,t)$ for the deterministic continuous-time optimal control problem defined by eqs 1 and 2 should satisfy the well-known *Hamilton−Jacobi−Bellman* (HJB), which is a sufficient optimality condition:[28,29]

$$0 = \min_{w\in\Omega}\left\{ g(x(t), \mathscr{P}(w^*,t)) + \frac{\partial J^*(t,x)}{\partial t} + \left[\frac{\partial J^*(t,x)}{\partial x}\right]^\mathrm{T} f(x, \mathscr{P}(w^*,t), \theta)\right\} \qquad (3)$$

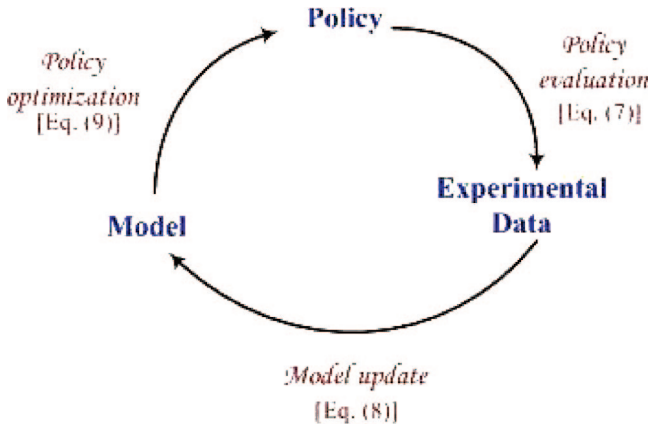

Figure 2. Model-based policy iteration.

for all t, x, along with the boundary condition $J^*(t_f,x) = h(x)$ for all x.

The HJB optimality condition is a partial differential equation which must be satisfied for all time−state pairs (t, x) by the cost-to-go function J^* corresponding to the optimal policy parametrization w^*. For any suboptimal policy $\mathscr{P}(\hat{w},t)$ there exists an error term which can be characterized by rewriting the right-hand side of eq 3 as follows:

$$0 < g(\hat{x}(t), \mathscr{P}(\hat{w}, t)) + \frac{\mathrm{d}(J(t, \hat{x}(t)))}{\mathrm{d}t} \tag{4}$$

where $\mathrm{d}(\cdot)/\mathrm{d}t$ denotes the total derivative of J with respect to t. By integrating the right-hand side of the above expression over $t \in [0,t_f]$, the Bellman residual BR is defined:

$$\mathrm{BR}(\hat{\theta}, \hat{w}) = \int_0^{t_f} g(\hat{x}(t), \mathscr{P}(\hat{w}, t))\, \mathrm{d}t + h(\hat{x}(t_f) - J(0, x(0)) > 0 \tag{5}$$

By subtracting from the $\mathrm{BR}(\hat{\theta},\hat{w}^*)$ for the special case where $\hat{\theta} \neq \theta_{\mathrm{real}}$ and $\hat{w}^*$ is the estimated "optimal" policy parametrization based on an imperfect model from the $\mathrm{BR}(\theta_{\mathrm{real}},w^*)$ (which has the minimum possible value of zero for the optimal policy $\mathscr{P}(w^*,t)$), the Bellman residual can be rewritten as

$$\begin{aligned}\mathrm{BR}(\theta, \hat{w}^*) - \mathrm{BR}(\theta_{\mathrm{real}}, \hat{w}^*) = \int_0^{t_f} [g(\hat{x}(t), \mathscr{P}(\hat{w}^*, t)) - \\ g(x_{\mathrm{real}}(t), \mathscr{P}(\hat{w}^*, t))]\, \mathrm{d}t + [h(\hat{x}(t_f) - \\ h(x_{\mathrm{real}}(t_f))] = \mathrm{BR}(\theta, \hat{w}^*) = J(t_f, \hat{x}) - J^*(t_f, x_{\mathrm{real}}); \quad \text{for any } t_f\end{aligned} \tag{6}$$

Based on eq 6, the optimal policy $\mathscr{P}(w^*,t)$ can be found by iteratively minimizing the performance prediction error resulting from applying an estimated optimal policy to the real process using a model-based policy iteration approach, as shown in Figure 2. Each iteration starts with a given policy $\mathscr{P}(\hat{w}^*,t)$ from the previous iteration which is evaluated in a specifically designed dynamic experiment (see section 3.2) by calculating the performance prediction error E using data from samples and a model with a given parameter vector θ as follows:

$$E(\theta) = \sum_{r=1}^{\mathrm{sp}} (J^*(t_r, \hat{x}) - J(t_r, x_{\mathrm{obs}}))^2 \tag{7}$$

As long as the performance error E is greater than a predefined tolerance (which accounts for measurement noise), there exists some room for policy improvement by reducing further the performance prediction mismatch defined in eq 7. It is worth noting that in order to minimize the performance prediction error, data gathering and model update should reduce selectively parametric uncertainty by defining at each iteration which are the most sensible sampling times along with the subset of model parameters that are most influential for performance prediction. Let us denote by $\hat{\theta}_l^n$ the subset of parameters for which uncertainty is going to be selectively reduced using measurements $y^n(k)$, $j = 1,...,$ sp gathered in the nth evaluation experiment with sp samples. The following least-squares minimization problem is solved to reestimate $\hat{\theta}_l^n$:

$$\theta_l^n = \arg\min \sum_{r=1}^{\mathrm{sp}} ||y^n(r) - \hat{y}^n(r)|| \tag{8}$$

s.t.

$$\frac{\mathrm{d}\hat{x}^n}{\mathrm{d}t} = f(\hat{x}^n(t), \mathscr{P}((\hat{w}^*)^{n-1}, t), \hat{\theta}^{n-1});$$

$$0 \le t \le t_f,\ \hat{x}^n(0) = \text{given}$$

$$\hat{y}^n(k) = Y(\hat{x}^n, \hat{\theta}^n); \quad k = 1, ..., \mathrm{sp}$$

Based on the updated process model the operating policy, is then reoptimized by solving

$$(\hat{w})^n = \arg\min_{\hat{w}\in\Omega} \hat{J}(t_f, \hat{x}^n) = \arg\min_{\hat{w}\in\Omega} h(\hat{x}^n(t_f)) + \int_0^{t_f} g(\hat{x}^n, \mathscr{P}(w, t))\, \mathrm{d}t \tag{9}$$

s.t.

$$\frac{\mathrm{d}\hat{x}^n}{\mathrm{d}t} = f(\hat{x}^n(t), \mathscr{P}(w, t), \hat{\theta}^n); \quad 0 \le t \le t_f,\ \hat{x}^n(0) = \text{given}$$

$$S(\hat{x}^n, \hat{\theta}^n, w) \le 0; \qquad T(\hat{x}^n(t_f), \hat{\theta}^n) \le 0$$

where S and T are path and terminal constraints whereas Ω corresponds to the constraint set for policy parameters. The reader is referred to the seminal work of Srinavasan et al.[31] and references therein for techniques available to solve the dynamic optimization problem in eq 9.

As soon as the performance prediction mismatch E can be driven to zero, policy iteration will converge to the optimal policy $\mathscr{P}(w^*,t)$ which gives rise to optimal performance. Should model structure and parameters be perfectly known a priori for the problem being addressed, model-based policy iteration is able to provide the optimal policy parametrization once the first iteration has been completed. As model parametric uncertainty is significantly high for batch processes, an optimally informative dynamic experiment for policy evaluation must be designed in each iteration step. To deal with the issue of enough input excitation[24] for data gathering in the policy evaluation experiment, (re)estimation is only done for the subset θ_l^n of model parameters and the functional form of the policy $\mathscr{P}$should allow some room for time-varying controls. To determine which parameters are in the subset θ_l^n, global sensitivity analysis[32] is used as explained below. The evaluation−identification−optimization cycle is shown in Figure 2 with due reference to relevant equations in the text.

3.2. Global Sensitivity Analysis. Global sensitivity analysis (GSA)[32−34] takes into account the fact that parametric uncertainty in complex models can propagate, compensate, or suffer many kinds of interactions which may affect the *output* of interest (i.e., the performance index J) in different ways. GSA is a variance-based technique that decomposes model *output* variability as a combination of uncertainty from each ith independent *input* factor and its interactions with other factors. This decomposition attempts to rank the

importance of uncertainty sources by mean of sensitivity indices. Briefly, let us suppose that the value of a model *output* of interest y is estimated with an uncertainty due to a set of k independent parameters x_i, $i = 1, 2,..., k$. Furthermore, those parameters can interact among them and as a whole influence model output y. The unconditional variance $V(y)$ of y is decomposed as follows:

$$V(y) = \sum_i V_i + \sum_i \sum_{j>i} V_{ij} + ... + V_{ij...k} \tag{10}$$

$$V_i = V_{x_i}(E_{x_{-i}}(y|x_i))$$
$$V_{ij} = V_{x_i x_j}(E_{x_{-ij}}(y|x_i x_j)) - V_{x_i}(E_{x_{-i}}(y|x_i)) - V_{x_j}(E_{x_{-j}}(y|x_j)) \tag{11}$$

where V_i is the amount of the total variance in the model response which can be explained only because of ith parameter values and it is known as the main effect term for the ith parameter; V_{ij} is the amount of variability generated due to the interaction between the ith and jth parameters. Note that, in computing V_i, it is necessary to compute and integrate over x_{-i} (all factors except x_i) and then a new integral over the marginal distribution of x_i to finally know the conditional variance V_i. The objective of applying GSA is to rank factors to know how $V(y)$ would be reduced if some of those factors were fixed in their true values. Accordingly, a first measure of the fraction of $V(y)$ which accounts for the uncertainty of x_i is the so-called first-order sensitivity index Si_i defined as

$$Si_i = \frac{V_i}{V} \tag{12}$$

Estimators for Si_i's can be obtained following different approaches, and here they have been computed by Sobol's method[32] which has been recently improved by Saltelli et al.[33,34] In this method a quasi-random sampling in the multidimensional space spanned by the parameter space Θ is used in order to find the sensitivity indices which ensures exploration over the whole range of variation for all *input* factors. To facilitate assessing the significance of sensitivity indices computed using Monte Carlo simulations, it can be useful to normalize each Si as follows:

$$Si_i^n = \frac{Si_i - \min(Si_j)}{\sum_j Si_j}; \quad j = 1, ..., k \tag{13}$$

3.3. Optimal Sampling. Due to the a priori significant uncertainty about model parameters, their values should be estimated selectively using data gathered in policy evaluation experiments. Based on the optimal policy parametrization from the previous iteration in the nth policy iteration of Figure 2, optimal sampling times ψ^{opt} along a batch run must be calculated to bring new information to selectively reduce parametric uncertainty which affects the most the value estimation of the performance index J. Assuming model parameters are set to $\hat{\theta}^{n-1} \in \Theta^{n-1}$, the issue of optimal sampling is related to calculating at which times $\psi^{opt} \in \Psi$ in a dynamic experiment the values of measured process variables are most informative in modeling for optimization assuming that each policy iteration should help reduce the performance prediction mismatch E. To this end, the following optimization problem is solved:

$$\psi^{opt} = \max_{\psi \in \Psi} \det|M(\hat{\theta}^{n-1}, \mathscr{R}((\hat{w}^*)^{n-1}, t)\psi)|, \; M = Q^T Q$$
$$Q = \begin{pmatrix} Si_{1l}^n & \cdots & Si_{1k}^n \\ \vdots & \ddots & \vdots \\ Si_{ml}^n & \cdots & Si_{mk}^n \end{pmatrix} \tag{14}$$

where each entry of the matrix Q, Si_{ij}^n, measures the sensitivity of the performance index at the ith sampling time with respect to jth parameter of the operating policy. The number of samples taken along each run will be defined in accordance with the budget for processing samples and bearing in mind that this number should be at least equal to the number of parameters defining the input policy. This formulation for optimal sampling in modeling for optimization has been inspired by a recent proposal in optimal design of experiments using global sensitivity analysis.[30]

Figure 3 provides a comprehensive summary of the proposed methodology for experimental design in modeling for optimization. At each iteration a dynamic experiment is designed around the previous iteration policy $\mathscr{P}((\hat{w}^*)^{n-1}, t)$ and optimal sampling times ψ^{opt} are calculated by solving eq 14. The experiment is carried out and new data are collected. Based on these data the subset $\hat{\theta}_{\not}^{n-1}$ of model parameters is reestimated to give $\hat{\theta}_{\not}^{n}$ which reduces parametric uncertainty in such a way that $\Theta^n \subset \Theta^{n-1}$. With the updated model a new policy parametrization $\mathscr{P}((\hat{w}^*)^n, t)$ is defined and a new iteration begins. The identification−optimization cycle is continued until no performance improvement is obtained and the defining parameters for the calculated input policy converge.

4. Case Study: Fed-Batch Fermentation of Penicillin G

Penicillin production is an established benchmark in fermentation processes for testing new approaches in modeling, optimization, and control of novel bioprocesses.[35,36] Penicillin and biomass are obtained at the expense of substrates (S) such as glucose and organic nitrogen compounds. Since the concentrations of viable (v) and death (d) biomass (X), penicillin (P), and glucose are routinely measured, they are chosen as the descriptive state variables along with broth volume (V), which varies with time t in this fed-batch bioprocess. The policy optimization problem is defined in terms of the final amount of penicillin obtained. Model equations for a unstructured tendency model of a fed-batch bioreactor are given below, whereas uncertainty intervals of model parameters are given in Table 1.[36] The column labeled "in silico" bioreactor corresponds to the (assumed unknown) parameters for the simulation model which were used to generate the incoming data provided by designed dynamic experiments.

$$\frac{dV(t)}{dt} = F_{in} - F_{evap}$$
$$\frac{dX_v}{dt} = \mu X_v - K_d X_v - \frac{F_{in} - F_{evap}}{V} X_v; \quad \mu = \frac{\mu_{max} S}{K_x X_v + S}$$
$$\frac{dX_d}{dt} = K_d X_v - k_{lys} X_d - \frac{F_{in} - F_{evap}}{V(t)} X_d$$
$$\frac{dS}{dt} = -\sigma X_v + \langle S_{in} F_{in} \rangle - \frac{F_{in} - F_{evap}}{V(t)} S;$$
$$\sigma = \frac{\mu}{Y_{xs}} + \frac{\pi}{Y_{ps}} + \zeta; \quad \zeta = \frac{\zeta_{max} S}{K_s + S}$$
$$\frac{dP}{dt} = \pi X_v - \frac{F_{in} - F_{evap}}{V(t)} P - K_h P; \quad \pi = \frac{\pi_{max} S}{K_p + S} \tag{15}$$

The fed-batch bioreactor needs some discharges of culture medium in order to maintain both viability and axenity of the

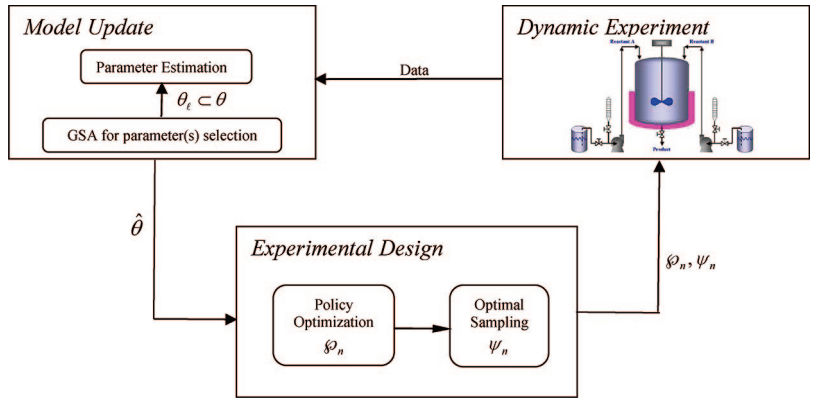

Figure 3. Modeling for optimization cycle.

Table 1. Model Parameters with Their Initial Uncertainty Ranges[36]

parameter	symbol	units	initial range	"in silico" bioreactor
maximum specific biomass growth rate	μ_{max}	h^{-1}	0.12−0.17	0.13
growth saturation constant	K_x	g of substrate/g of DW	0.006−0.4	0.131
cellular death rate	K_d	h^{-1}	0.005−0.01	0.006
cellular lysis constant	k_{lys}	h^{-1}	0.00001−0.008	0.0008
substrate to biomass yield	Y_{xs}	g of DW/g of substrate	0.40−0.58	0.52
substrate to penicillin yield	Y_{ps}	g of penicillin/g of substrate	0.4−1	0.97
maximum specific penicillin production rate	π_{max}	h^{-1}	0.003−0.015	0.011
saturation constant for penicillin production	K_p	g L^{-1}	0.00001−0.0002	0.0001
maximum specific substrate consumption rate for maintenance	ζ_{max}	h^{-1}	0.014−0.029	0.02
saturation constant for maintenance	K_s	g L^{-1}	0.00001−0.0002	0.0001
penicillin hydrolysis rate constant	K_h	h^{-1}	0.002−0.01	0.002

penicillin producing fungi strain since those drain-offs avoid any possibility of mutations and productivity reduction. Such discharges must be made at some specific moments along the production run and with certain frequency all along the run duration. Moreover, the model takes into account a culture medium evaporation rate F_{evap} which is set as constant in the present study.

At any time t, the input policy $\mathscr{A}(w,t)$ is defined by a vector w of parameters corresponding to two different degrees of freedom for process optimization. A subset of the entries of the vector w corresponds to inputs that can be modified from run to run but are time-invariant in a given run such as the substrate feeding concentration, the discharge frequency, the discharge volume, and the time corresponding to the first discharge. The remaining entries are parameters which are used here for describing the profile of time-varying control variables such as the feeding rate. In the latter case, a key issue is the mathematical description to be used to provide ample room for different variability patterns within economic and safety constraints with a minimum number of independent parameters. In the past there have been various approaches to implement bioreactor feeding policies which can be defined as constant, piecewise constant,[23] piecewise continuous,[36] or totally continuous functions of time.[37] In this work, the feeding rate profile is described using inverse polynomials of low order with respect to time. Inverse polynomials resort to a small number of parameters to define time trajectories which are quite flexible for modeling a rich variety of continuous feeding patterns for bioreactor optimization. It is worth noting that the methodology proposed in section 3 is by no means limited by the family of mathematical functions used to describe time-varying controllable input controls. However, bioreactor dynamics slowly unfolds cell responses to environmental changes which require using smooth profiles for time-varying control inputs.

The ranges of variation for each parameter in the inverse polynomial describing the feeding profile have been defined from an exploratory analysis of alternative parametrizations that produce a time-varying in-flow rate which is constrained to the interval [0,10] L h^{-1} at any time in a batch run. It is noteworthy that this family of functions has been chosen to eliminate problems found in GSA implementations when there are correlation and dependency in the set of parameters defining the feeding profile. This input dependency is known as multicollinearity and introduces significant errors in sensitivity indices when the effects of policy parameters in the objective function (J) are computed. This is the very case when for example the feeding profile is modeled as a piecewise constant or a polynomial spline function. Before considering more elaborate feeding profiles, it seems worth trying to get the most of a constant feeding rate.

The upper and lower bounds for input policy parameters assuming a constant feeding rate are given in Table 2 along with optimum values for first and second iterations of the proposed methodology. Also, the predicted values for policy parametrizations are shown together with their standard deviations [std(J)] based on the model updated with kinetic parameters' intervals recomputed after each data gathering experiment in the modeling for the optimization cycle of Figure 3. The "true" optimal policy parametrization has been obtained using the in silico model parametrization when solving eq 9 for the chosen functional form of the operating policy.

As can be seen in Table 2, the estimation of penicillin production using optimal policy parameters after the first iteration is quite uncertain and is even greater than the estimation for the second iteration. As new data are available to reestimate some kinetic parameters, the confidence in policy values increases even though policy parameters remain unchanged.

Table 2. Input Policy Parameterization Using a Constant Feeding Rate

parameter	initial value	lower bound	upper bound	optimum (first iteration)	optimum (second iteration)	"real" optimum
feeding rate [L h^{-1}]	4	0	10	5.73	5.73	5.73
t_0 [h]	0	0	24	0	0.025	0.024
t_f [h]	240	200	300	240	240	240
substrate feed concn [g L^{-1}]	240	200	300	300	300	300
time first discharge [h]	24	24	48	24	24	24
discharge volume [L]	60	30	80	80	80	80
discharge frequency [h]	24	24	60	24	24	24
initial volume [L]	600	500	700	500	500	500
penicillin obtained, J (kg)	24.38			36.85	34.68	40.16
std(J) (kg)				8.89	3.89	

Table 3. Sensitivity Indices for Model Kinetic Parameters Assuming a Constant Feeding Profile

model parameter	Si^n (first iteration)	Si^n (second iteration)
π_{max}	0.6110	0.2402
K_h	0.2456	0.0198
Y_{ps}	0.0509	0.1654
ζ_{max}	0.0463	0.3501
Y_{xs}	0.0214	0.1234
k_{lys}	0.0000	0.0008
μ_{max}	0.0000	0.0008
K_x	0.0002	0.0008
K_d	0.0246	0.0978
K_s	0.0000	0.0010
K_p	0.0000	0.0000

The optimal sampling times for data gathering in both iterations have been computed by solving the optimization problem defined by eq 4, and the results obtained are as follows:

first iteration:

$$\psi^{opt}(h)\text{:}17, 34, 141, 157, 173, 189, 205, 221, 233, 239$$
$$\det |M| = 2.57 \times 10^{-38}$$

second iteration:

$$\psi^{opt}(h)\text{:}21, 27, 33, 143, 160, 176, 193, 210, 228, 240$$
$$\det |M| = 3.40 \times 10^{-39}$$

Data gathered in each experiment done on the in silico bioreactor are used to (re)estimate selectively model parameters in accordance with their sensitivity indices Si^n in Table 3.

In the first iteration the sensitivity indices reflect that most of the initial uncertainty in policy value predictions is due to π_{max} and K_h, so they could be reestimated using data gathered in the first experiment. In the second iteration sensitivity indices in Table 3 highlight that a further reduction in the uncertainty of policy value predictions can be achieved by using data from the second experiment to reestimate π_{max}, Y_{ps}, ζ_{max}, Y_{xs}, and K_d. After two experiments the initial model uncertainty has been reduced to

first iteration:

$$\pi_{max} = 0.0104 \pm 0.0010$$
$$K_h = 0.0022 \pm 0.0003$$

second iteration:

$$\pi_{max} = 0.0098 \pm 0.0008$$
$$Y_{ps} = 0.94 \pm 0.17$$
$$\zeta_{max} = 0.0143 \pm 0.0002$$
$$Y_{xs} = 0.47 \pm 0.04$$
$$K_d = 0.0061 \pm 0.0012$$

In a first attempt to increase the amount of penicillin obtained by using a constant feeding rate, let us try finding an optimal feeding rate profile modeled using a linear inverse polynomial. A linear inverse polynomial has the form

$$F_{in} = \begin{cases} 0, & t < t_0 \\ A\dfrac{t}{B + t}, & t \geq t_0 \end{cases} \tag{16}$$

The upper and lower bounds for parameters A and B assuming this pattern of variation for the feeding rate are given in Table 4 along with optimum policy parameters after the first and second iterations of the proposed methodology. Also, predicted values for J are shown along with their standard deviations [std(J)] using the model updated with intervals for kinetic parameters recomputed after each iteration. In Figure 4 the resulting "optimum" feeding profile obtained is compared with the feeding profile used by Menezes et al.,[36] who has reported a penicillin G production of 24 kg. For each iteration GSA has been used to identify which subset of kinetic parameters should be reestimated based on normalized first-order sensitivity indices (Si^n) (see Table 5 for details).

As can be seen in Table 4, the estimated amount of penicillin produced after the first iteration was significantly increased with regard to the initial policy, a continuous approximation to the one used by Menezes et al.[35] However, this potential improvement is rather uncertain. In the second iteration this rise is more evident since std(J) is reduced enough so as to provide a reliable estimation for J. It is also worth noting that the second iteration is mostly a confirmatory result of the first one since the policy's parameters have been slightly changed whereas the value of the J is practically the same. It is noteworthy that the more elaborate feeding policy does not provide a significant increase in productivity as might be expected. Before attempting a continuous feeding profile that better reflects the one proposed in ref 35, some iteration results are discussed.

In accordance with the proposed algorithm in Figure 3, before the current operating policy is actually implemented in the in silico bioreactor, the optimization problem of eq 4 must be solved to determine the optimal sampling times for the next dynamic experiment. Sampling times and values for the objective M in the first two iterations are (see Figures 5, 6, and 7)

first iteration:

$$\psi^{opt}(h)\text{:}5, 14, 19, 36, 183, 187, 192, 198, 202, 232$$
$$\det |M| = 3.83 \times 10^{-34}$$

second iteration:

$$\psi^{opt}(h)\text{:}20, 121, 129, 146, 160, 178, 184, 205, 213, 221$$
$$\det |M| = 3.31 \times 10^{-36}$$

In the first iteration the resulting optimal sampling times suggest that initial uncertainty about the performance index is mainly due to uncertainty in parameters π_{max} and K_h which are reestimated

Table 4. Results Obtained Using a Feeding Rate Profile Modeled as a Linear Inverse Polynomial

parameter	initial value	lower bound	upper bound	optimum (first iteration)	optimum (second iteration)	"real" optimum
A [L h^{-1}]	6	3	10	4.88	4.03	3.89
B [h]	20	0	200	13.88	1.71	0
t_0 [h]	0	0	24	0	0	0
t_f [h]	240	200	300	240	240	240
substrate feed concn [g L^{-1}]	240	200	300	300	300	300
time first discharge [h]	24	24	48	25	48	48
discharge volume [L]	60	30	80	80	80	80
discharge frequency [h]	24	24	60	53.8	60	60
initial volume [L]	600	500	700	500	500	500
penicillin obtained, J, [kg]	27.76			34.00	34.31	34.93
std(J) (kg)				5.53	2.12	

using data gathered in the first dynamic experiment and the corresponding uncertainty is consequently reduced. In the second iteration the optimal sampling strategy reveals that the remaining uncertainty in the performance index prediction can significantly be reduced if the resulting data are used to reestimate parameters Y_{ps}, ζ_{max}, K_p, and K_d. The selective reduction in parametric uncertainty after only two iterations results in a remarkable convergence to the actual values used for simulations in the in silico bioreactor as follows.

first iteration:

$$\pi_{max} = 0.01105 \pm 0.00018$$
$$K_h = 0.00200 \pm 0.00014$$

second iteration:

$$Y_{ps} = 0.8 \pm 0.3$$
$$\zeta_{max} = 0.018 \pm 0.005$$
$$K_p = 0.00014 \pm 0.00008$$
$$K_d = 0.0059 \pm 0.0007$$

In Figures 5−7 the model's predictions and "experimental data" (which have been obtained from the in silico bioreactor experiments where a 10% variability has been added to simulate noise at sampling times) are compared for state variables in the first and second iterations when a feeding rate profile shaped as a linear inverse polynomial is used.

An intrinsic limitation of the linear inverse polynomial with parameters constrained as shown in Table 4 is that the maximum feeding rate is always located at the end of the run. To better

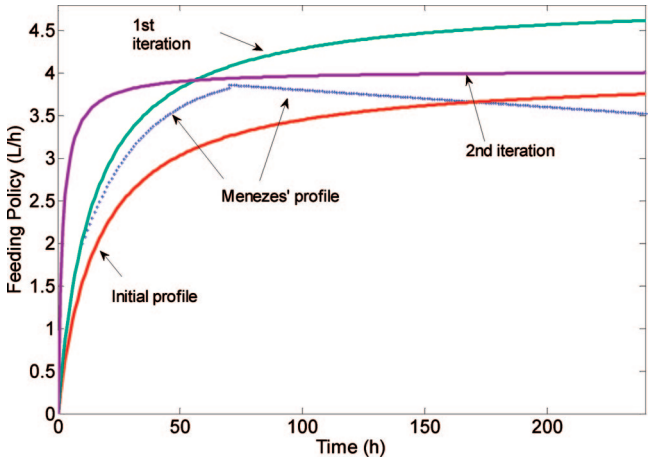

Figure 4. Feeding policies implemented by Menezes (*) and computed with the proposed methodology in the initial condition, first iteration, and second iteration using a linear inverse polynomial.

Table 5. Sensitivity Indices for Kinetic Model Parameters Assuming a Linear Inverse Polynomial for Modeling the Feeding Rate Profile

model parameter	Si^n (first iteration)	Si^n (second iteration)
π_{max}	0.7410	0.0294
K_h	0.1062	0.0009
Y_{ps}	0.0567	0.3916
ζ_{max}	0.0201	0.2104
Y_{xs}	0.0079	0.0175
k_{lys}	0.0035	0.0013
μ_{max}	0.0030	0.0042
K_x	0.0030	0.0000
K_d	0.0030	0.1225
K_s	0.0005	0.0805
K_p	0.0000	0.1418

approximate the feeding rate profile used by Menezes et al.,[36] the following quadratic inverse polynomial is proposed:

$$F_{in} = \begin{cases} 0, & t < t_0 \\ \dfrac{At}{1 + Bt + Ct^2}, & t \geq t_0 \end{cases} \quad (17)$$

In Table 6 upper and lower bounds and optimal values for operating policy parameters are shown for the first and second iterations along with predicted values for the objective function (J) and its standard deviation [std(J)] computed using GSA. Also, in Figure 8 the obtained optimal feeding profile is compared with the one used by Menezes et al.[36] Assuming a quadratic inverse polynomial is used to describe the feeding rate profile, it is possible to provide room for a decreasing feeding rate in the final stage of a fermentation run (see Figure 8).

As can be seen from Table 6, when a feeding rate profile having an intrarun maximum is used, the proposed input policy improvement methodology is able to improve substantially the amount of penicillin obtained in a bioprocess run. It can argued though that the variability in estimating the value J of a policy parametrization is a bit greater than when a linear inverse polynomial feeding profile is used, but the increase in penicillin production is quite a remarkable improvement. Moreover, it is worth noting that the algorithm has found optimal values for all parameters in the operating policy, except those of the feeding profile, just in the first iteration. Once again, in the second iteration the improvement in J is reflected not in its mean value but in the reduced uncertainty with which the improvement is predicted.

The optimal sampling strategy for both iterations is given below together with the corresponding M values:

first iteration:

$$\psi^{opt}(h): 15, 21, 39, 55, 147, 164, 181, 198, 215, 233$$
$$\det |M| = 8.06 \times 10^{-42}$$

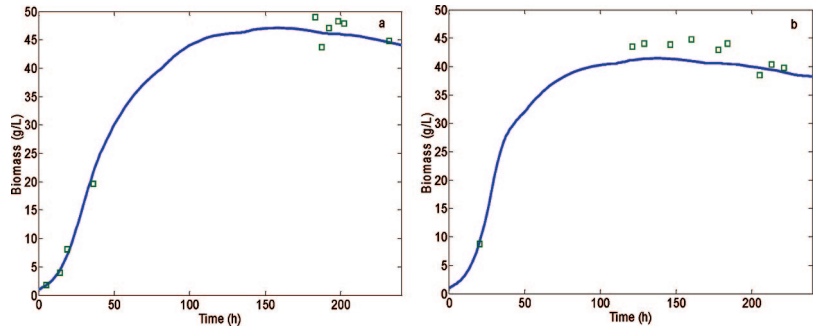

Figure 5. Sampling times and model prediction in first (a) and second (b) iterations for biomass concentration for a feeding policy modeled as an inverse linear polynomial.

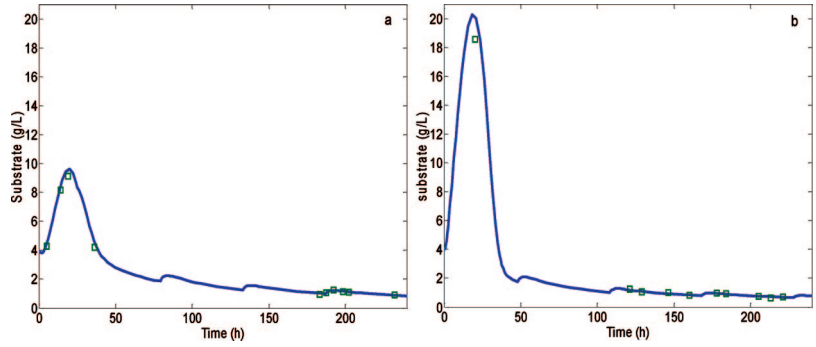

Figure 6. Sampling times and model fitting in first (a) and second (b) iterations for substrate concentration for a feeding policy modeled as an inverse linear polynomial.

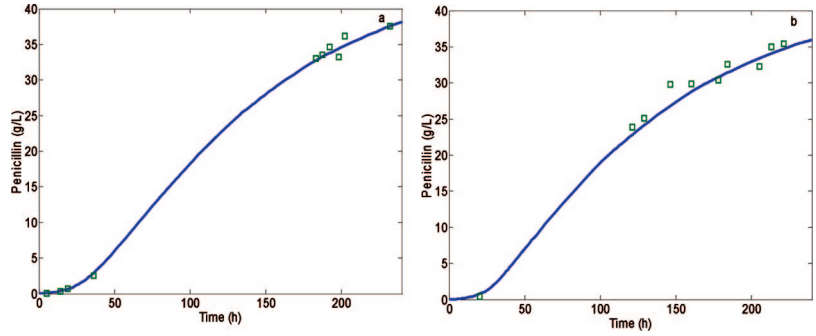

Figure 7. Sampling times and model fitting in first (a) and second (b) iterations for penicillin concentration for a feeding policy modeled as an inverse linear polynomial.

Table 6. Results Obtained Using a Feeding Rate Profile Described as a Quadratic Inverse Polynomial

parameter	initial value	lower bound	upper bound	optimum (first iteration)	optimum (second iteration)	"real" optimum
A [L h^{-2}]	0.6882	0	4.13	2.0649	2	2.0287
B [h^{-1}]	0.1431	0.1	0.86	0.2061	0.2627	0.2000
C [h^{-2}]	0.0002	−0.0008	0.0012	0.0012	0.0006	0.0012
t_0 [h]	0	0	24	0	0	0
t_f [h]	240	200	300	240	240	240
substrate feed concn [g L^{-1}]	240	200	300	300	300	300
time first discharge [h]	24	24	48	24	24	24
discharge volume [L]	60	30	80	80	80	80
discharge frequency [h]	24	24	60	24	24	24
initial volume [L]	600	500	700	500	500	500
penicillin obtained, J (kg)	28.04			41.84	40.05	41.27
std(J) [kg]				2.0649	2	

second iteration:

$$\psi^{opt}(h){:}20, 121, 129, 146, 160, 178, 184, 205, 213, 221$$

$$\det |M| = 3.31 \times 10^{-36}$$

In Table 7 normalized sensitivity indices Si^n obtained using GSA are summarized for kinetics parameters for the two iterations. Once again the most relevant parameters in the first iteration were π_{max} and K_h, which vividly show that policy value sensitivity is invariant when uncertainty is mainly due to a few kinetic parameters. It is worth noting that π_{max} and K_h have been included again in the set of parameters to be reestimated in the second iteration after the

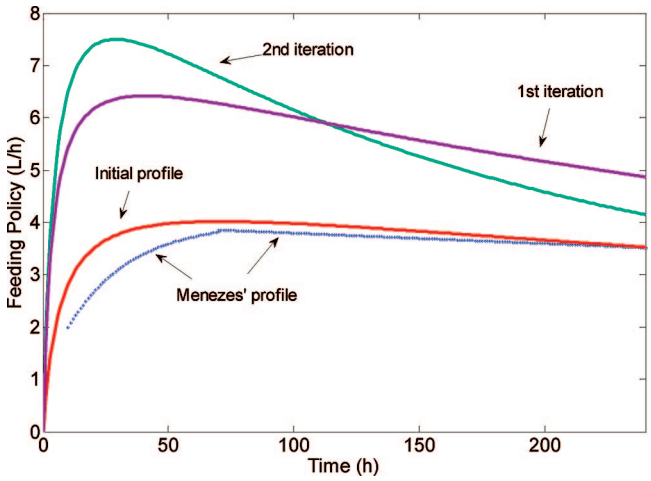

Figure 8. Feeding policies implemented by Menezes (*) and computed with the proposed methodology in the initial condition, first iteration, and second iteration assuming a quadratic inverse polynomial.

Table 7. Sensitivity Indices for Kinetic Model Parameters Assuming a Quadratic Inverse Polynomial For Modeling the Feeding Rate Profile

model parameter	Si^n (first iteration)	Si^n (second iteration)
π_{max}	0.7206	0.1331
K_h	0.1812	0.3114
Y_{ps}	0.0216	0.1749
ζ_{max}	0.0325	0.1588
Y_{xs}	0.0167	0.1111
k_{lys}	0.0000	0.0002
μ_{max}	0.0000	0.0002
K_x	0.0000	0.0002
K_d	0.0272	0.1099
K_s	0.0000	0.0000
K_p	0.0001	0.0002

first iteration because their level of uncertainty is still high enough to affect the variability of the performance index prediction. Parameter values for both iterations are given by

first iteration:

$$\pi_{max} = 0.013 \pm 0.002$$
$$K_h = 0.004 \pm 0.002$$

second iteration:

$$K_h = 0.004 \pm 0.002$$
$$Y_{ps} = 0.97 \pm 0.10$$
$$\zeta_{max} = 0.014 \pm 0.010$$
$$\pi_{max} = 0.013 \pm 0.002$$
$$Y_{xs} = 0.42 \pm 0.10$$
$$K_d = 0.005 \pm 0.003$$

In Figures 9, 10, and 11 model predictions and "experimental data" (which have been obtained from in silico experiences with an added 10% variability) of state variables are compared for both iterations when a quadratic inverse polynomial feeding rate profile is used.

As the last attempt to find a continuous feeding profile that can provide a further improvement in penicillin production, the following cubic inverse polynomial was used.

Intervals of variation for the four parameters have been determined to allow a maximum feeding rate in the initial stage of a run and an intrarun minimum. In Table 8 upper and lower bounds and optimal values for operating policy parameters for the first and second iterations are shown along with values for the objective function (J) predicted with the model and its standard deviation [std(J)] computed using GSA. Furthermore, in Figure 12 optimal trajectories computed as "optimal" feeding profiles are depicted together with the initial profile used to carry the policy optimization.

Unfortunately, as shown in Table 8, no further gain in penicillin production is obtained using this more complicated feeding profile. It can be argued that constraints imposed on parameters A, B, C, and D are too restrictive, but they have been used to assess a qualitatively different pattern for the feeding rate from those associated with quadratic inverse polynomials. It should be acknowledged that Menezes et al.[36] somehow discovered by trial and error that their feeding profile (see Figure 4) had the right pattern, but they did not attempt to optimize it. Accordingly, results obtained (Table 8) reinforce the idea that feeding patterns like those in Figure 12 are not able to generate an increase in penicillin production compared to quadratic inverse polynomials.

To end this case study, some important remarks regarding the implementation of the proposed model-based policy iteration strategy need to be made. First, the proposed methodology is aimed at using tendency models for the batch process to be optimized. Typically, such models have a rather small number of parameters, say between 5 and 12, which can be reliably estimated using a rather small number of samples taken from each modeling run. Also, the number of parameters used to describe the operating policy is assumed small enough (e.g., no more than 20) that computational costs of GSA can be maintained on the order of some hours with standard desktop computational power (actually approximately 16 h per iteration for the presented case study using a PC equipped with an Intel Pentium 2 core duo processor with a speed of 2 GHz). Complex operating policies with hundreds of parameters can still be used, but computational costs of using GSA will increase significantly. This is not a big problem in general since the number of iterations is small, but computational costs should be taken into account when analyzing the rationale behind a given functional form used to define the operating policy. The issue of a total lack of persistent excitation when time-varying controls are not used deserves a word of caution. As shown in Table 2 for the constant feed rate, using a policy without the required persistent excitation may slow down or even prevent convergence to the optimal policy.[39] Finally, there is the issue of characterizing the initial parametric uncertainty. For the presented case study there exists plenty of available literature[36] which allows defining an initial uncertainty on a rather sound basis. For innovative batch processes and bioprocesses, such knowledge often will not be available and the best practice for comprehensive policy optimization is to choose rather ample intervals for each parameter (say 50% around its nominal value). This will not present a problem for model-based policy iteration, yet the number of iterations for convergence may increase as required for such a level of parametric uncertainty.

5. Conceptual Framework for Run-to-Run Convergence

Convergence analysis of iterative identification−optimization schemes based on imperfect models is a very difficult problem to solve as has been pointed out in some previous works.[27,38,39] However, for a model-based policy iteration algorithm based on increasingly reducing parametric uncertainty, it is feasible to characterize conditions for convergence in simple, yet conclusive terms as follows.

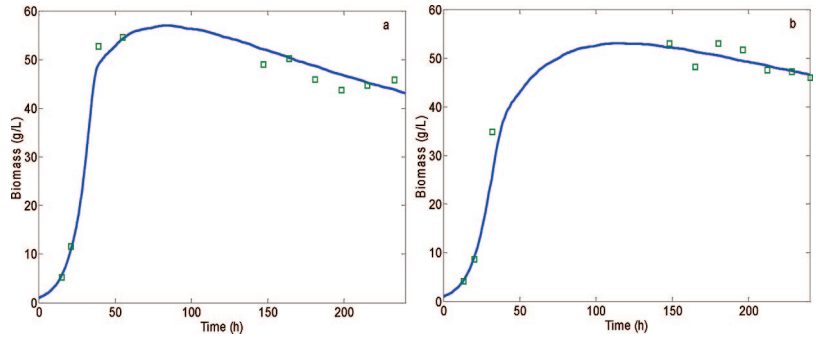

Figure 9. Sampling times and model fitting in first (a) and second (b) iterations for biomass for a feeding policy modeled as an inverse quadratic polynomial.

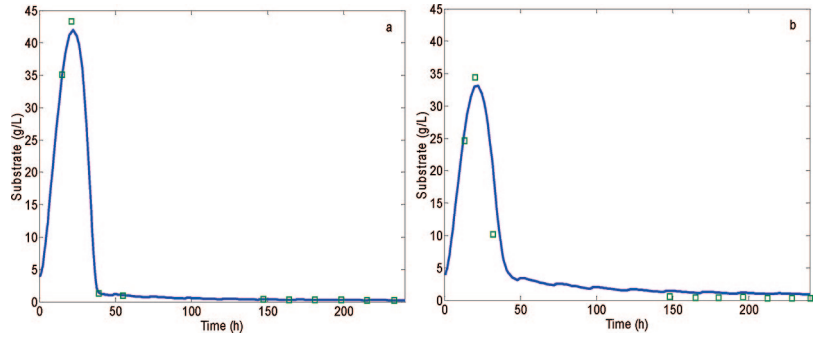

Figure 10. Sampling times and model fitting in first (a) and second (b) iterations for substrate concentration for a feeding policy modeled as an inverse quadratic polynomial.

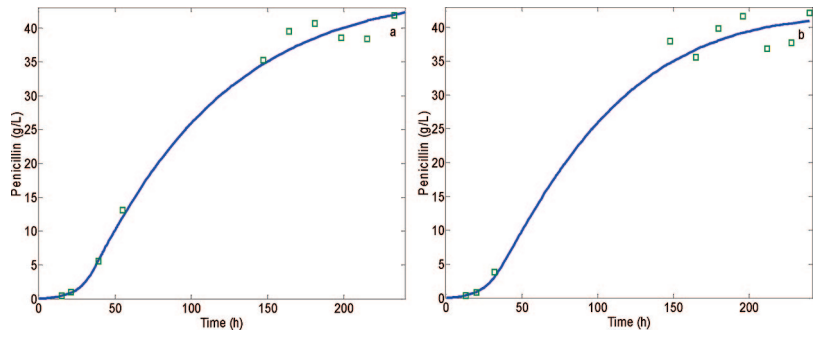

Figure 11. Sampling times and model fitting in first (a) and second (b) iterations for penicillin concentration for a feeding policy modeled as an inverse quadratic polynomial.

$$F_{\text{in}} = \begin{cases} 0, & t < t_0 \\ \dfrac{At}{1 + Bt + Ct^2 + Dt^3}, & t \geq t_0 \end{cases} \tag{18}$$

Consider the problem of minimizing the performance prediction error E: $\Theta \rightarrow \mathscr{R}$ for a given model parametrization and its corresponding optimal policy, where Θ is a bounded Euclidean space over which any model parametrization is confined to as parametric uncertainty is increasingly reduced on a run-to-run basis. Let Γ denote the HJB optimality condition, and let $\Sigma \subset \Theta$ denote the set of model parametrizations for which Γ is satisfied. Also, let ξ: $\Theta \rightarrow \mathscr{R}^-$ be a non-positive-valued function such that $\xi(\theta) = 0$ if and only if $\theta \in \Sigma$. Such a function is called an *optimality function associated with* Γ (see ref 40, p 19, for details). This function ξ must be defined to provide a quantitative measure of the extent to which a model parametrization along with its related optimal policy satisfies the condition Γ, namely the HJB equation in (3). The natural choice when defining an optimality function for convergence analysis of a policy iteration algorithm is resorting to the modulus of the Bellman residual defined in eq 5 for the special case where $\hat{w} = \hat{w}^*(\theta)$ corresponds to the optimal policy parametrization based on a model with parameter vector θ:

$$\xi(\theta) = -|\text{BR}(\theta, \hat{w}^*(\theta))|; \quad \theta \in \Theta \tag{19}$$

Definition 5.1. *Shrinking Set* Θ. As the number n of runs (iterations) increases, the parameter space Θ which characterizes model uncertainty shrinks toward an accumulation point $\tilde{\theta}$ such that

$$\lim_{n \rightarrow \infty} \Theta_n = \tilde{\theta} \tag{20}$$

where the parameter space for any finite n always satisfies $\Theta^{n+1} \subset \Theta^n$.

Table 8. Initial and Optimum Values for the Policy Value, J and std(J) Using a Cubic Inverse Polynomial Feeding Policy

parameter	initial value	lower bound	upper bound	optimum (first iteration)	optimum (second iteration)	"real" optimum
A [L h^{-2}]	0.2048	0.2151	0.3648	0.3648	0.3647	0.3648
B [h^{-1}]	−0.0451	−0.0682	−0.0373	−0.0458	−0.0458	−0.0458
C [h^{-2}]	1.378×10^{-3}	1.833×10^{-3}	2.666×10^{-3}	1.833×10^{-3}	1.833×10^{-3}	1.833×10^{-3}
D [h^{-3}]	-4.444×10^{-6}	-6.032×10^{-6}	-1.565×10^{-6}	-5.672×10^{-6}	-5.649×10^{-6}	-5.676×10^{-6}
t_0 [h]	0	0	24	1.72	2.7	0.1464
t_f [h]	240	200	300	240	240	240
substrate feed concn [g L^{-1}]	240	200	300	300	300	300
time first discharge [h]	24	24	48	24	24	24
discharge volume [L]	60	30	80	54.61	54	54.74
discharge frequency [h]	24	24	60	24	24	24
initial volume [L]	600	500	700	500	500	500
penicillin obtained, J [kg]	34.89			38.68	35.60	37.67
std(J) [kg]				5.13	1.72	

Definition 5.2. *Sufficient Descent.* A model-based policy iteration algorithm working over a shrinking parameter space Θ has the property of sufficient descent with respect to the chosen optimality function ξ if for every $\delta > 0$ there exists $\eta > 0$ such that, for every $n = 1, 2,...$, and for every iteration point (model parametrization) $\hat{\theta}^n$ computed by the algorithm, if $\xi(\hat{\theta}^n) < -\delta$, then

$$E(\hat{\theta}^{n+1}) - E(\hat{\theta}^n) < -\eta \tag{21}$$

where $\hat{\theta}^{n+1}$ is the next iteration point.

Now, run-to-run convergence can be stated as follows (see ref 41 for details).

Proposition 5.1. *Suppose there exists a constant $D \in R$ such that $E(\theta) \geq D$ for every $\theta \in \Theta$. If a model-based policy iteration algorithm has the property of sufficient descent, then for any infinite sequence $\{\hat{\theta}^n\}_{n=1}^{\infty}$, it computes, $\lim_{n\to\infty} \xi(\hat{\theta}_n) = 0$.*

Proof. The proof is immediate from Definition 5.2. □

Run-to-run convergence of a policy iteration algorithm to an optimal policy which satisfies the HJB is thus guaranteed as long as the model structure allows driving the performance prediction mismatch to zero. Hence, convergence to the optimal policy can only be ensured if the model has the correct structure and parameter reestimation at each iteration is carried out by global minimization of the prediction error surface. If there exist modeling errors which prevent reducing to zero the performance prediction mismatch E, algorithm convergence to a policy is still guaranteed based on Definition 5.1.

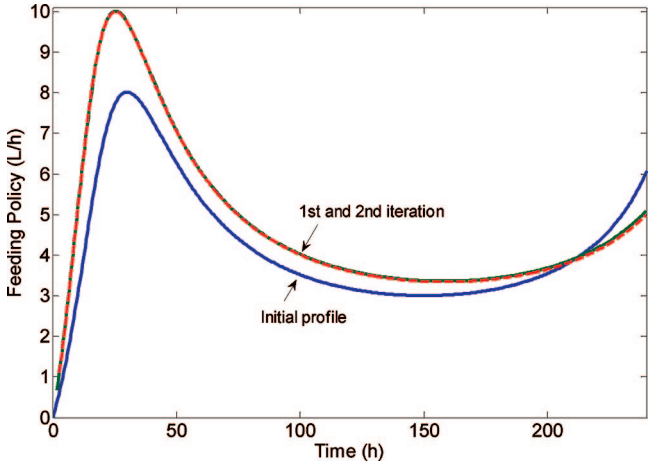

Figure 12. Feeding policies for the initial condition and first and second iterations assuming the feeding rate profile is shaped as an inverse cubic polynomial.

6. Concluding Remarks

This paper has presented a systematic procedure for designing dynamic experiments in modeling for optimization aimed at selectively reducing parametric uncertainty by iteratively improving the input policy. To this end the traditional separation between modeling and optimization has been changed to the modeling for optimization cycle where model development is tightly integrated with dynamic optimization. Global sensitivity analysis has been used to formulate the optimal sampling in each dynamic experiment as an optimization problem whose solution provides the optimal sampling times at which the performance objective is most sensitive to changes in the policy parameters. Once new data are available, global sensitivity analysis is used to determine the subset of model parameters that should be reestimated. Following model update, the input policy parametrization is recalculated by dynamic optimization and a new iteration begins. A case study related to penicillin G production has been used to illustrate the proposed approach, and results obtained are very encouraging. Convergence analysis of the proposed model-based policy iteration strategy has been stated in a novel conceptual framework under the assumption that modeling errors are the only consequence of parametric uncertainty. The objective of future work is to account for structural uncertainty in modeling for optimization, namely by integrating model discrimination in the proposed methodology for experimental design.

Acknowledgment

The kind support received from Prof. Saltelli and Prof. Tarantola to overcome problems found when implementing GSA is gratefully acknowledged.

Nomenclature

Symbols

A = feeding profile parameter [(L h^{-1}) if a linear or (L h^{-2}) if a quadratic inverse polynomial profile is assumed]
B = feeding profile parameter (h^{-1})
C = feeding profile parameter (h^{-2})
D = feeding profile parameter (h^{-3})
DW = dry weight
$f(\cdot)$ = functional form of the model
F_{in} = inlet flow rate (L h^{-1})
F_{evap} = outlet flow rate due evaporation (L h^{-1})
$g(\cdot)$ = reward function along the state trajectory
$h(\cdot)$ = specific reward for the final state of the batch run
J = performance index to be maximized
M = sensitivity indices with respect to operating policy parameters w
P = penicillin concentration (as potassium salt) (g of PenGK L^{-1})

$\mathscr{P}(\cdot)$ = input policy
S = substrate concentration (g L^{-1})
S_{ij} = first-order sensitivity index at the ith sampling time with respect to jth parameter of the operating policy
Si_k^n = normalized sensitivity index Si_k
t = time (h)
t_0 = initial time to beginning feed (h)
t_f = final time of a batch run (h)
V = culture broth volume (L)
w = m-dimensional vector of policy parameters
$x(t)$ = n_s-dimensional vector of process state variables
X = biomass concentration (g of DW L^{-1})
X_d = death biomass concentration (g of DW L^{-1})
X_v = viable biomass concentration (g of DW L^{-1})

Greek Symbols

μ = specific biomass growth rate (h^{-1})
$\theta \in \Theta$ = p-dimensional vector of model parameters
$\psi^{opt} \in \Psi$ = optimal sampling strategy

Literature Cited

(1) Terwiesch, P.; Agarwal, M.; Rippin, D. W. T. Batch unit optimization with imperfect modeling: a survey. *J. Process Control* **1994**, 238–258.

(2) Schenker, B.; Agarwal, M. Prediction of infrequently measurable quantities in poorly modeled processes. *J. Process Control* **1995**, 329–339.

(3) Bonvin, D. Optimal operation of batch reactors: a personal viewpoint. *J. Process Control* **1998**, 355–368.

(4) Martinez, E. C. Batch process modeling for optimization using reinforcement learning. *Comput. Chem. Eng.* **2000**, 1187–1193.

(5) Martínez, E. C.; Wilson, J. A. Evolutionary optimization of batch process systems using imperfect models. *Proc. Indian Natl. Sci. Acad.* **2003**, 403–428.

(6) Kadam, J. V.; Schlegel, M.; Srinivasan, B.; Bonvin, D.; Marquardt, W. Dynamic optimization in the presence of uncertainty: from off-line nominal solution to measurement-based implementation. *J. Process Control* **2007**, 389–398.

(7) Filippi, C.; Graffe, J.; Bordet, J.; Villermaux, J.; Barney, J.; Bonte, P.; Georgakis, C. Tendency modeling of semibatch reactors for optimization and control. *Chem. Eng. Sci.* **1986**, 913.

(8) Filippi, C.; Bordet, J.; Villermaux, J.; Marchal-Brassey, S.; Georgakis, C. Batch reactor optimisation by use of Tendency models. *Comput. Chem. Eng.* **1989**, 35–47.

(9) Uhlemann, J.; Cabassud, M.; LeLann, M.; Borredon, E.; Cassamatta, G. Semibatch reactor optimization and control for the production of furfural. *Chem. Eng. Sci.* **1994**, 3169–3191.

(10) Hamer, J. Stoichiometric interpretation of multireaction data; application to fed-batch fermentation data. *Chem. Eng. Sci.* **1989**, 2363–2374.

(11) Bonvin, D.; Rippin, D. Target factor analysis for the identification of stoichiometric models. *Chem. Eng. Sci.* **1990**, 3417–3426.

(12) Prinz, O. Chemometric Methods for Investigation of Chemical Reaction Systems. Ph.D. Thesis, ETH, Zurich, Switzerland, 1992.

(13) Tsobanakis, P. Tendency Modelling and Optimisation of Fed-batch Fermentations. Ph.D. Thesis, Lehigh University, 1994.

(14) Georgakis, G. Modern tools of process control: The case of black, gray and white models. *Entropie* **1995**, 34–48.

(15) Fotopoulos, J. Quantification of Model Uncertainty and its Impact on the Optimisation and State Estimation of Batch Processes via Tendency Modeling. Ph.D. Thesis, Lehigh University, 1996.

(16) Amrhein, M. Reaction and flow variants/invariants for the analysis of chemical reaction data. Ph.D. Thesis, EPFL, Lausanne, Switzerland, 1998.

(17) Amrhein, M; Srinivasan, B.; Bonvin, D. Target factor analysis of reaction data: use of data pretreatment and reaction invariant relationships. *Chem. Eng. Sci.* **1999**, 579–592.

(18) Fotopoulos, J.; Georgakis, C.; Stenger, H. G., Jr. Uncertainty issues in the modelling and optimisation of batch reactors with Tendency models. *Chem. Eng. Sci.* **1994**, 5533–5547.

(19) Fotopoulos, J.; Georgakis, C.; Stenger, H. G., Jr. Effect of the process-model mismatch on the optimisation of the catalytic epoxidation of oleic acid using Tendency models. *Chem. Eng. Sci.* **1996**, 1899–1908.

(20) Fotopoulos, J.; Georgakis, C.; Stenger, H. G., Jr. Use of Tendency models and their uncertainty in the design of state estimators for batch reactors. *Chem. Eng. Process.* **1998**, 545–558.

(21) Terwiesch, P. Cautious on-line correction of batch process operation. *AIChE J.* **1995**, 1337–1340.

(22) Terwiesch, P.; Agarwal, M. Robust input policies for batch reactors under parametric uncertainty. *Chem. Eng. Commun.* **1995**, 33–52.

(23) Asprey, S.; Macchietto, S. Designing robust optimal dynamic experiments. *J. Process Control* **2002**, 545–556.

(24) Srinivasan, B.; Bonvin, D. Interplay between identification and optimization in run-to-run optimization schemes. *Proceedings of the American Control Conference Anchorage, AK, May 8−1 0, 2002*; pp 2174−2179.

(25) Franceschini, G.; Macchietto, S. Model-based design of experiments for parameter precision: state of the art. *Chem. Eng. Sci.* **2008**, in press.

(26) Sedrati, Y.; Cabassud, M.; Lelann, M. V.; Casamatta, G. Sequential Experiment Design Strategy for Kinetic Parameter Estimation. *Comput. Chem. Eng.* **1999**, S427−S430.

(27) Zhang, Y.; Forbes, J. F. Design Cost: A Systematic Approach to Technology Selection for Model-Based Real-Time Optimization Systems. *Comput. Chem. Eng.* **1996**, 717–734.

(28) Bellman, R. E. *Dynamic Programming*; Princeton University Press: Princeton, NJ; 1958.

(29) Bertsekas, D. P. *Dynamic Programming and Optimal Control*, 1st ed.; Athena Scientific: Nashua, NH, 1995; Vol. 1, Chapter 3.

(30) Rodriguez-Fernandez, M.; Kucherenko, S.; Pantelides, C.; Shah, N. Optimal experimental design based on global sensitivity analysis. *Comput.-Aided Process Eng.* **2007**, 63–68.

(31) Srinivasan, B.; Palanki, S.; Bonvin, D. Dynamic optimization of batch processes I. Characterization of the nominal solution. *Comput. Chem. Eng.* **2003**, 1–23.

(32) Sobol', I. M. Sensitivity analysis for non-linear mathematical models. *Math. Model. Comput. Exp.* **1993**, 407–414.

(33) Saltelli, A. Making best use of model valuations to compute sensitivity indices. *Comput. Phys. Commun.* **2002**, 280–297.

(34) Saltelli, A.; Ratto, M.; Tarantola, S.; Campolongo, F. Sensitivity Analysis Practices. Strategies for Model-Based Inference. *Reliab. Eng. Syst. Saf.* **2006**, 1109–1125.

(35) Bajpai, R. K.; Reuss, M. A mechanistic model for penicillin production. *J. Chem. Technol. Biotechnol.* **1980**, 332–344.

(36) Menezes, J. C.; Alves, S. S.; Lemos, J. M. Mathematical Modelling of Industrial Pilot-Plant Penicillin-G Fed-Batch Fermentations. *J. Chem. Technol. Biotechnol.* **1994**, 123–138.

(37) Li, X. B.; Zhao, G. R.; Zuan, Y. J. A strategy of phosphorus feeding for repeated fed-batch fermentation of penicillin G. *Biochem. Eng. J.* **2005**, 53–58.

(38) Brdys, M.; Roberts, P. D. Convergence and optimality of modified two-step algorithm for integrated system optimisation and parameter estimation. *Int. J. Syst. Sci.* **1987**, 1305–1322.

(39) Srinivasan, B.; Bonvin, D. Convergence analysis of iterative identification and optimization schemes. *Proceedings of the ACC, 2003, Denver, CO*.

(40) Polak, E. *Optimization: Algorithms and Consistent Approximations*; Springer: New York, 1997.

(41) Polak, E.; Wardi, Y. A study of minimizing sequences. *SIAM J. Control Optim.* **1984**, 599–609.

Received for review January 18, 2008
Revised manuscript received November 10, 2008
Accepted January 19, 2009

IE8000953