ELSEVIER

Computers and Chemical Engineering 24 (2000) 1187–1193

Computers & Chemical Engineering

www.elsevier.com/locate/compchemeng

Batch process modeling for optimization using reinforcement learning

E.C. Martinez *

Instituto de Desarrollo y Diseño, Consejo Nacional de Investigaciones Cientificas y Técnicas, Avellaneda 3657 3000, Argentina

Abstract

Imperfect and incomplete understanding of reaction kinetics compounded with uncontrollable variations not only prevent achieving an optimal operation of batch and semi-batch reactors, but also give rise to potential risks of violating product end-use properties, ecological or safety constraints. This paper proposes a sequential experiment design strategy based on reinforcement learning to accomplish the specific goal of *modeling for optimization* in batch reactors by making the most effective use of cumulative data and an approximate model. Reactor operating condition is incrementally improved over runs by integrating together estimation of a probabilistic measure of success using an imperfect model and a gradient-based approach so as to trade off *exploitation* with *exploration*. An improved operating policy is found by incrementally shrinking the region of interest for policy parameters. The solution strategy focuses on 'learning by doing' using a *value* function that accounts for endpoint performance and feasibility. Simulation results reveal the robustness of reinforcement learning to parametric and structural modeling errors. © 2000 Elsevier Science Ltd. All rights reserved.

Keywords: Batch process; Reaction kinetics; Modeling for optimization

1. Introduction

The increased emphasis on pharmaceutical and fine chemicals in the Chemical Process Industries is demanding tools and methodologies to support all modeling activities that are specifically geared to batch product and process development, scaling up and optimization (Bonvin, 1998; Shah, Samsatli, Sharif, Borland & Papageorgiou, 1999; Stephanopoulos, Ali, Linninger & Salomone, 1999). Adjusting operating conditions of the reaction system is always critical for profitability. *Modeling for optimization* is then becoming a key activity as the innovative nature of products and reduced time-to-market are limiting the number of runs performed at the small scale of operation which is typical of the engineering laboratory. As a result, the migration to production runs is often made with high levels of uncertainty and risk. The main drawbacks that arise whilst scaling up operating strategies from lab conditions to commercial-size units are the high costs of not achieving minimal batch-to-batch variations in product quality and process performance. Batch processing of specialty chemicals imposed tight constraints on impurities in the final product. Also, imperfect and incomplete understanding of the kinetic phenomena involved compounded with intra- and inter-run variations not only prevent achieving an optimal operation of batch processes, but also give rise to potential risks of violating product end-use properties, ecological or safety constraints.

A sequential experiment design strategy for determining the most profitable operating conditions can be designed to follow either a *greedy* or an *active learning* approach (Raju & Cooney, 1998; Martinez, 1999). In the first case, the sequence of runs is simply the result of attempting to maximize the *exploitation* of current knowledge (e.g. a tendency model) and data available. That is, pure exploitation merely chooses operating conditions for the next run bearing in mind solely policy optimization without any care on systematically reducing the uncertainty about the location of the true optimum. No attempt is made to influence the generation of data from the reactor in order to circumvent uncertainty by selecting the most informative operating conditions. This is, in general, shortsighted as the opti-

* Corresponding author. Fax: + 54-342-4553439.
E-mail address: ecmarti@alpha.arcride.edu.ar (E.C. Martinez)

PII: S0098-1354(00)00354-9

mization search might quickly becomes stuck in a sub-optimal solution. In this work, modeling for optimization in batch processes is approached using *reinforcement learning* (Sutton & Barto, 1998) to decide where to explore next, so that we can identify a near optimal operating policy with a small data set in the face of uncertainty.

2. Modeling for optimization

In 'modeling for optimization' any plausible model describing the kinetic behavior is merely a *means* to a clear-cut *end*, namely to help systematically improve and optimize the reactor operating condition in the face of uncertainty. The process model here is not an end in itself as it is in kinetic studies (Sedrati, Cabassud, Lehann & Casamatta, 1999). The main objective of estimating model's parameters is instead to help locate the optimum operating condition as precisely as possible with minimum experimental effort. The strategy chosen to obtain sequentially the experimental data needed for this purpose will influence greatly the number of modeling runs, the cost involved and the length of time required to accomplish the objective stated above. Ideally, operating conditions need to be progressively biased towards the most profitable operating conditions for the process. However, due to uncertainties present, this can only be done confidently if exploration of apparently less promising conditions is also systematically practiced. Therefore, the generation of data from the batch process should be influenced *actively* to bring the most meaningful information from each experimental run.

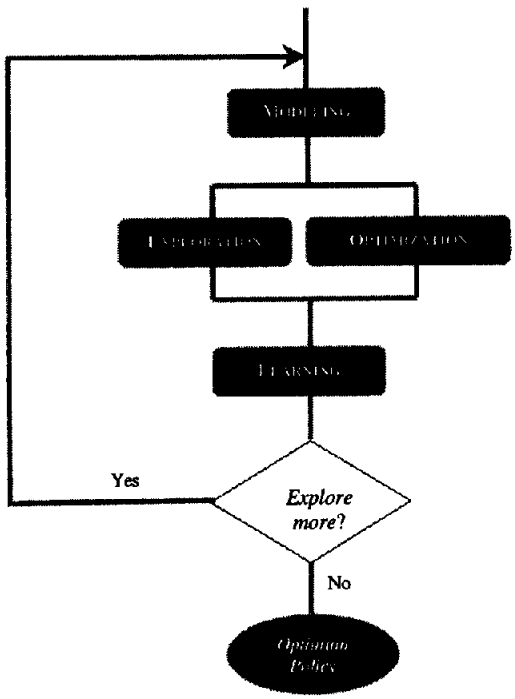

Fig. 1. Modeling for optimization using learning.

The distinctive requirement of modeling for optimization is that the process model should allow quantify, for each operating condition, a local approximation to the gradient direction towards the optimum operating policy. To overcome the difficulties of uncertainty and imperfect process models the introduction of 'learning by doing' is proposed here. Modeling for optimization using learning, proposes an entirely different approach to relate model development with process optimization. Traditionally, modeling is done first and, after a good model is available, optimization is then undertaken. However, for learning to be useful, modeling and optimization should be tightly integrated within an inner loop that also includes exploration as shown in Fig. 1. The learning block allows compensating for both structure and parameter modeling errors by doing a posteriori analysis of the actual value of a given operating policy regardless of its optimality.

3. Optimal operation under uncertainty

Optimal operation of batch processes is typically based on a performance function $J(x_f)$ that involve the final state x_f of each run, which in turn also needs to satisfy a set of endpoint constraints $g(x_f) \geq 0$. For a given operating policy $u = (u_1, u_2, \ldots, u_r)^T$, unknown initial conditions and uncontrollable intra-run variations make batch run outcome x_f quite uncertain and very difficult to model accurately. Even at laboratory conditions, end-point reproducibility as low as 5% is common (Terwiesch, Agarwal & Rippin, 1994). Significant variability of each run outcome does make difficult to measure the actual quality of using a given combination of policy parameters in terms of the chosen objective function. Also, outcome variance gives rise the problem of assessing the probability of satisfying endpoint constraints for each operating policy defined over the feasible region for policy parameters. The problem of *learning* improved operation using outcomes of successive runs consists of finding a operating policy that incrementally approximates the greedy policy u^*, which solves:

$$\underset{u}{\text{Optimize}}\ E[J(x_f)] \tag{1}$$

$$x_f = \chi(x_0, u) + \text{noise (known through experiments)}$$

Subject to:

$$g(x_f) \geq 0, \text{ endpoint constraints}$$

$$u_i^L \leq u_i \leq u_i^U, \quad u_i \in u, \quad \text{initial ROI}$$

where $E[\cdot]$ is the mean value operator, and J is a figure of merit for the final state that may involve not only the expected performance but also its variability. The mapping χ represents a priori unknown relationship describing the averaged effect of each operating policy and batch initial condition on the final state. Thus, the actual influence of a given u on x_f progressively unfolds as more replications for each operating policy are carried out. Changes in x_0 are mainly the result of inter-run variations such as catalyst activity, reactant/solvent recycling or raw material quality. The term 'noise' denotes unknown variations or disturbances acting on the process whilst a batch is being processed, e.g. poor temperature control or inadequate mixing. This noisy component of x_f is typically due to intra-run variations resulting in different process behaviors affecting outcome reproducibility in a biased way. Thus, the stochastic nature of noise is often non-Gaussian and skewed, which makes it very difficult to use standard statistical techniques for plant data analysis.

4. Reinforcement learning

Reinforcement learning (RL) can be defined as *learning what to do*, i.e. how to map perceptions and outcomes in a sequence to take decisions or actions so as to maximize an externally provided scalar reward signal (Sutton & Barto, 1998). Inherently related to the RL problem is the notion of a sequence of decisions under uncertainty such that for each stage:

1. The optimality of alternative decisions is upper-bounded by previous decisions, and
2. the goodness of the current decision will limit the optimality of the final solution.

In 'modeling for optimization' each RL decision stage means choosing a set of policy parameters that hopefully provides the information that is needed to pinpoint a near-optimal policy after a certain number of runs have been made. The degree of optimality of the final policy found is the cumulative reward we are seeking to maximize each time the operating conditions for the next run are decided. To this end, a number of requisites need to be accomplished. First, the solution algorithm should be designed to need only a few runs. Secondly, it should be very robust to modeling errors (structure and parameters). Also, the planning of runs derived from the RL algorithm should be geared to make the process generate data that are most informative for its global improvement, not only local optimization. Finally, and because of the uncertainty involved in Eq. (1), greedily seeking for u^* is too cumbersome a task. Hence, the more practical goal of determining a promising region of interest for the reactor operating conditions will be adopted here.

From the standpoint of modeling for optimization there exist two issues, feasibility and performance, that need to be integrated together when assessing the true *value* or *utility* of the solution found to the problem (Eq. (1)) and each intermediate decision. For feasibility, the value of each operating policy should measure the probability of satisfying endpoint constraints (Terwiesch, Ravemark, Schenker & Rippin, 1998). The probability of completing successfully a batch run is estimated here analytically using an approximated model. Statistical analysis of the uncertainty associated to model's parameters is used to assess the risk of endpoint constraint violation. Cubature integration techniques or other quadrature formulas aimed at efficient sampling are readily available for doing this calculation. The maximum achievable value of $\Pr_u[g(x_f) \geq 0]$ will depend on the magnitude of uncontrollable disturbances affecting each run outcome, i.e. x_f. For performance, the value of each policy should account for the expected value of $J(x_f)$. The quality of experimental data generated whilst modeling for optimization should then be measured in terms of the increasing capability of the resulting model to help assess the averaged *value* of operating policies.

The RL solution strategy proposed here shrinks systematically the region of interest (ROI), for each policy parameter u_i, $i = 1, \ldots, r$, over which process optimization is constrained to take place. Initially, ROI_0 corresponds to the overall feasible region for defining a policy as defined in Eq. (1). By cutting away unpromising regions within ROI_0, the algorithm is aimed at generating a sequence of ROI_1, ROI_2, ..., $\mathrm{ROI}_{\mathrm{iter}}$, $\mathrm{ROI}_{\mathrm{iter}+1}$, ..., such that

$$\mathrm{ROI}_{\mathrm{iter}} \supseteq \mathrm{ROI}_{\mathrm{iter}+1} \tag{2}$$

Eventually, a final ROI is obtained for which all possible operating policies are non-superior to each other. For this subset of policies, different values for x_f can only be explained on the basis of the noise component. As a result, further improvements can result only from reducing the variance of noise and, measurements permitting, by policy improvement through on-line (intra-run) optimization that compensates for changes in the batch initial conditions x_0.

5. Exploration versus exploitation

Let's consider that each policy u has an expected endpoint *value* given that this policy has been tried infinitely. In a practical sense, this true value can be approximated by averaging the $J^e(x_f)$ actually obtained when policy u is chosen, weighted by a model-based estimation of $\Pr_u = [g(x_f) \geq 0]$. Sampled estimations of values aim to approximate the true endpoint value $Q(u)$ for a given policy u defined as follows

$$Q(u) = \left(\frac{1}{r}\sum_{e=1}^{r\to\infty} J^e(x_f)\right) \times \Pr_u[g(x_f) \geq 0] \quad (3)$$

The RL problem solving approach is based on incrementally improving the estimation of the endpoint value for each policy u previously tried so as to make a sound decision for choosing the operating policy for the next run. Because of the uncertainty involved, value estimation creates the need for selective exploration during learning. In this work, exploration refers to either choosing again (replicating) one of the already tried policies or trying an entirely new one. In the latter case, deciding where over ROI, exploration is deemed necessary demands the use of a function approximation technique (e.g. regression or neural network) to extrapolate/interpolate the values of control policies to unseen operating conditions. Hence, at any time there is a least one policy within ROI, whose estimated value seems to be the greatest. We call this a *greedy* policy. If this policy is chosen for next run, it can be said that that we are exploiting our current knowledge of the values of control policies.

Exploitation is the right thing to do assuming that value estimates are sufficiently good and data are abundantly available all over ROI. However, if uncertainty is such that there would be other operating policies whose true values are in fact higher that the current greedy policy, then exploration is called for. Since it is not possible both to explore and to exploit with any single policy selection a conflict between these orthogonal objectives arises. To obtain more reward, we must prefer control policies that have been already tried and found to be effective in terms of the endpoint objective and constraints. But to discover such policies, we must try operating conditions that have not been selected before. Exploitation is needed to obtain more reward whilst exploration is needed to find better policies to exploit in the future. The dilemma is that neither exploitation nor exploration can be exclusively pursued whilst doing modeling for optimization without failing to satisfy the endpoint constraints. Having said this, the solution algorithm need to be designed to selectively try informative control policies, and progressively favors those that appear to be the best, upon actual experience. Also, and because of uncertainty, each operating policy must be tried a minimum number of times (say two or three) to gain a reliable estimation of its expected endpoint value. As model accuracy improves, the need for replicating a given policy diminish and value estimation can be made quite reliably even when it has been tried only once.

The balance between exploration and exploitation is achieved by varying the policy selection probabilities as a graded function of estimated values. The policy having the highest value according to current knowledge is still given the maximum selection probability, but all previously tried policies are ranked according to their value estimates. This is called the *softmax* criterion where the probability of selecting a certain u in the next run is defined as

$$\frac{e^{Q(u)/\tau}}{\sum_{b=1}^{n} e^{Q(b)/\tau}} \quad (4)$$

where n is the number of alternative policies to choose from (including the greedy one) and τ is a positive parameter called the temperature. High temperatures cause all the n policies to be nearly equiprobable. As the temperature is lowered, policies with higher values have greater chances of being selected. In the limit as $\tau \to 0$, the bias towards pure exploitation is total, i.e. the probability of selecting the greedy policy tends to 1, and no room for exploration is left.

6. Sequential experiment design strategy

Given as input a data set DS_{iter} from previous batch runs, the following steps are carried out iteratively.

6.1. Ranking

Each policy $u_j, j = 1, 2, \ldots$, for data points included in DS_{iter}, is given a figure of merit based on its current value to identify where cutting away an unpromising, yet feasible sub-space of the policy parameters u_i $i = 1, \ldots, r$ is deemed appropriate.

6.2. Cutting

Let $(u_{min} \to Q(u_{min}))$ be the data point that is predicted to be the worst within DS_{iter}. To define a cut of ROI_{iter}, all we need is an estimation of the direction of the steepest gradient $\nabla_u Q(u)$ at $u = u_{min}$. This requires using some function approximation technique to extrapolate/interpolate the values of policies over ROI_{iter}. There might be several approaches to follow for this purpose, e.g. using a regression model, a neural network or a fuzzy system. Because of the scarcity of data, the use of a simple linear (in the parameters) regression model is considered enough here:

$$\hat{Q} = X^T\beta + \varepsilon \quad (5)$$

where $X = X(u)$ is a $1 \times m$ matrix function of the control variables with rank m, β is a m-vector of unknown parameters and ε is normally distributed scalar variable with zero mean and variance σ^2 , is sufficient for our purpose. Moreover, since DS_{iter} often will consist of very few, weirdly distributed data points, simply fitting a quadratic is proposed here:

$$\hat{Q}_j(u_j) = c + b^T u_j + 1/2 u_j^T A u_j \quad (6)$$

where c, b and A, correspond to the set of fitting parameters to be obtained by the least-squares criterion. The local gradient at $u = u_{min}$ is then easy to calculate as $\nabla Q = b + Au_{min}$. A cut to DS_{iter} is made using the half-plane perpendicular to ∇Q so that:

$$ROI_{iter+1} = ROI_{iter} \cap \{u | (u - u_{min}) \cdot \nabla Q \geq 0\} \quad (7)$$

For the next cut, the data points to be considered are reduced to: $DS_{iter+1} = DS_{iter} - (u_{min} \rightarrow Q(u_{min}))$.

Ranking/cutting is continued this way until either one of the two following conditions apply:

1. There is not enough data present in DS_{iter+1} to determine all fitting parameters in c, b and A. The immediate result of insufficient data is matrix inversion problems;
2. the confidence interval around $Q(u^*)$, for the estimated optimum operating policy $u^* = -A^{-1}b$, exceeds $\|Q(u_{max}) - Q(u_{min})$ corresponding to best and worst points in DS_{iter+1}, respectively. Once no more cuts are allowed, proceeds to the *planning* step below.

6.3. Planning

At this stage, the crucial decision of defining the policy u_{next} to be used in the next experiment needs to be made. From the point of view of exploitation, it seems attractive to consider the greedy decision of choosing as the operating policy for the next run $u_{next} = u^* = -A^{-1}b$. However, this excessive bias towards exploitation is unsound unless the information gathered in previous runs provides enough evidence to support that $Pr_u[g(x_f) \geq 0]$ is high enough all over the reduced ROI, and the confidence interval around u^* is sufficiently small. Otherwise, the next run operating policy should be decided on the grounds of seeking for a trade off between pure exploitation against exploring further the reduced ROI, provided by the *cutting* stage above. From the point of view of exploration, we consider the question 'where over the reduced ROI, should one explore further in order to achieve the maximum reduction in the uncertainty about the location of the optimal policy?' To answer this, concepts of optimum experiment design (OED) theory are needed. According to the OED, the next run policy should be chosen so as to minimize the variance of value estimations for operating conditions over the final ROI.

6.3.1. Example

Consider the chemical reaction system conducted in an isothermal semi-batch reactor, which behaves according to the true mechanism:

$$A + B \underset{k_2}{\overset{k_1}{\Leftrightarrow}} I \xrightarrow{k_3} C; \quad B + B \xrightarrow{k_4} D \quad \text{(order 2.5)}$$

This simple problem characterizes an important class of industrial situations (Young, 1980; Terwiesch et al., 1998). At the chosen operating temperature, the true kinetic parameters (which are unknown) have the following *mean* values: $k_1 = 1.1355$, $k_2 = 5.0$, $k_3 = 4.5239$ and $k_4 = 3.5880$, which correspond to perfect temperature control. In industrial-size units, temperature control will never be perfect. As a result, the final state of each run will show unsystematic variations. Since the side-reaction yielding the undesired species D will be favored at high concentrations of B, it is better not to add all B initially. On the other hand, to easy product formation via the short-lived intermediate I, the equilibrium reaction should be forced towards the right. Accordingly, the system is operated in a semi-batch mode where a stream of pure B ($[B]_{feed} = 0.2$ mol l^{-1} is added to a 1000 l vessel which initially contains 0.2 mol l^{-1} of A and no B, and is filled to 50%.

The production objective is, through proper addition of reactant B, to convert as much as possible of the expensive reactant A to the desired product, C, in a maximum time of 120 min. Thus, $J(x_f) = [C]_f \times V_f$. The specifications on the final state of each batch are as follows. For *safety* reasons in downstream processing, the final concentration of unreacted B cannot be greater than 0.0035 mol l^{-1}. Also, product *purity* limits the final concentration of undesired side product, D, to be no more than 0.018 mol l^{-1}; and process *profitability* demands obtaining at least 0.058 moles of C at the end of the batch. Operating conditions are defined here as feed addition rate $0 \leq F \leq 12.0$ [l min^{-1}] and the duration of the semi-batch period $0 \leq t_1 \leq t_{final} = 120$. Even though more elaborated operating strategies can be used only this simple feeding policy is considered here for the sake of clarity and space. Since the optimal solution should necessarily be within a ROI, such that $Pr_u[g(x_f) \geq 0]$ is as high as possible, there is no point in reducing the overall batch time to less than the available 120 min. As a result, only two variables, F and t_1, are left for defining the reactor operating policy.

For the present case study, it is assumed that for each modeling run the only available measurements are the final concentrations of species B, C and D. Since the short-lived intermediate I cannot be observed, a postulated mechanism that was thought to agree well to the data from modeling runs is:

$$A + B \xrightarrow{\hat{k}_1} C; \quad B + B \xrightarrow{\hat{k}_2} D.$$

Using this imperfect model and standard regression technique, a given data set allows estimating $\hat{k}_1$ and $\hat{k}_2$ to account for both the kinetic behavior and variations due imperfect temperature control.

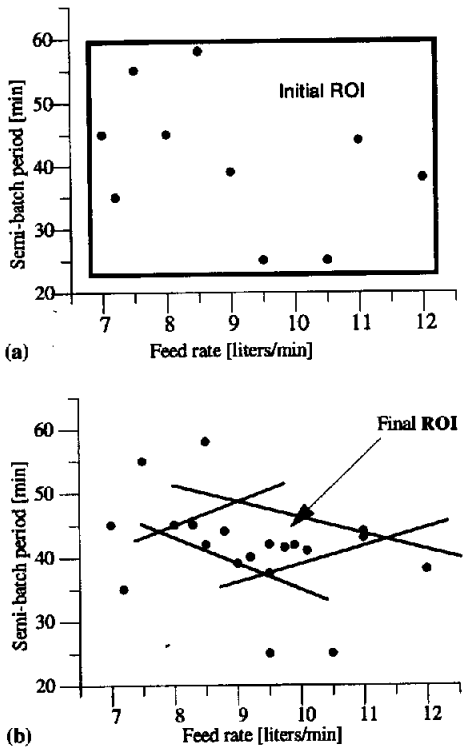

Fig. 2. (a) Initial set of operating policies. (b). The reduced ROI, after ten new policies have been explored/exploited.

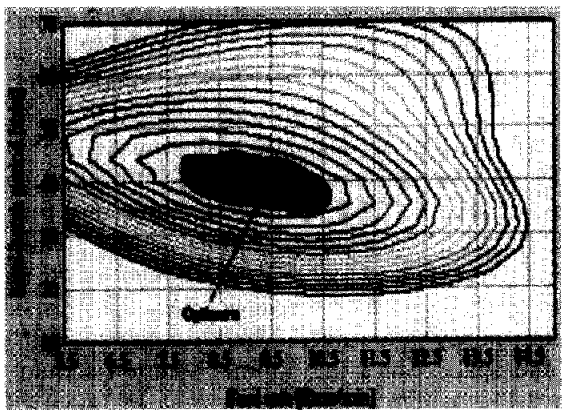

Fig. 3. Probability of success contour plots.

6.4. Results

The initial data set was made up of the ten policies shown in Fig. 2(a), each of which was tried twice. Note that the initial policies were judiciously chosen in a ROI, were the policy has some chances of satisfying the endpoint constraints. After ten new policies have been incorporated into the data set, the ROI, has been drastically reduced as shown in Fig. 2(b). Fig. 3 depicts the probability of success estimated using the imperfect model and data collected. The value function obtained is shown in Fig. 4. These results were obtained using an exponential cooling procedure as follows. After an entirely new control policy for the next run was actually implemented and its resulting information added to DS_0, the temperature parameter τ was decreased geometrically with the cumulative number of control policies z explored while learning using:

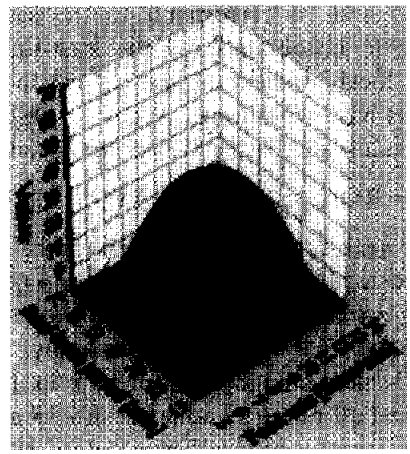

Fig. 4. Value function for the example.

$$\tau(z) = p(1-p)^z,\ z \in \{0, 1, 2, \ldots\} \quad (8)$$

with $p = 0.25$ to allow more exploration.

7. Final remarks

The precise meaning and distinctiveness of batch process modeling for optimization has been discussed along with the importance of introducing a *learning* dimension in the problem-solving strategy. The synergy between exploration and exploitation (optimization) has been then considered instrumental to deal successfully with ubiquitous uncertainty and imperfect kinetic models whilst deciding the next-run operating conditions. A sequential experiment design strategy based on reinforcement learning has been presented.

References

Bonvin, D. (1998). Optimal operation of batch reactors. *Journal of Process Control, 8*, 355–368.

Martinez, E. C. (1999). Improving cost and timeliness of batch process scale-up using active reinforcement learning. In Proceedings of the Focapd99 Conference, Breckenridge CO, USA. Paper no. A03.

Raju, G. K., & Cooney, C. L. (1998). Active learning from process data. *American Institute of Chemical Engineering Journal, 44*, 2199–2211.

Sedrati, Y., Cabassud, M., Lehann, M. V., & Casamatta, G. (1999). Sequential experiment design strategy for kinetic parameter estimation. *Comprehensive Chemical Engineering, 23*, S427–S430.

Shah, N., Samsatli, N.J., Sharif, M., Borland, J., & Papageorgiou, L. (1999). Modeling and optimization for pharmaceutical and fine chemical process development. In Proceedings of the Focapd99 Conference, Breckenridge CO, USA. Paper no. I05.

Stephanopoulos, G., Ali, S., Linninger, A., & Salomone, E. (1999). Batch process development: from reactions to manufacturing systems. *Comprehensive Chemical Engineering, 23*, S975–S984.

Sutton, R. S., & Barto, A. G. (1998). *Reinforcement learning; an introduction*. Cambridge, MA: MIT.

Terwiesch, P., Agarwal, M., & Rippin, D. W. T. (1994). Batch unit optimization with imperfect modeling: a survey. *Journal of Process Control, 4*, 238–258.

Terwiesch, P., Ravemark, D., Schenker, B., & Rippin, D. W. T. (1998). Semi-batch process optimization under uncertainty: theory and experiments. *Comprehensive Chemical Engineerinng, 22*, 201–213.

Young, M. J. (1980). Semi-batch reactions offer optimal yield. *Processing, 3*, 27–33.