RUN-TO-RUN OPTIMIZATION VIA CONTROL OF GENERALIZED CONSTRAINTS

B. Srinivasan[*], **C. J Primus**[†], **D. Bonvin**[*], **and N. L. Ricker**[†]

[*] *Institut d'Automatique, Ecole Polytechnique Fédérale de*
Lausanne, CH-1015 Lausanne, Switzerland
[†] *Department of Chemical Engineering,*
University of Washington, Seattle, Washington, USA

Abstract: Run-to-run optimization methodologies exploit the repetitive nature of batch processes to determine the optimal operating policy in the presence of uncertainty. In this paper, a parsimonious parameterization of the inputs is used and the decision variables of the parameterization are updated on a run-to-run basis using a feedback control scheme which tracks signals that are invariant under uncertainty. In this run-to-run framework, terminal constraints of the optimization problem and cost sensitivities constitute the invariant signals. The methodology is adapted to improve the cost function from batch to batch without constraint violation.

Keywords: Dynamic optimization, Batch processes, Run-to-Run optimization, Batch-to-Batch optimization, Constraint control.

1. INTRODUCTION

Batch and semi-batch processes are of great importance to the fine chemicals industry. Three characteristics that differentiate batch processes from continuous processes are: (i) unsteady-state operation, (ii) limited operation time, and (iii) the repetitive nature. Dynamic optimization of batch processes has received increased attention since, in the face of competition, it is a natural choice for reducing production costs. In addition, industry can see the potential of process optimization for improving product quality and meeting safety and environmental regulations (Bonvin, 1998).

Classically, optimal control theory has been utilized in the literature to calculate the input profile for specific batch processes (Ray, 1981). In most cases, the proposed implementation has been open-loop. However, open-loop implementation of the optimal input trajectory may not lead to optimal performance due to uncertainty in initial conditions and model parameters, and to process disturbances.

Traditionally, batch processes have been operated with very little instrumentation. However, in the last two decades, sensor technology has improved considerably for the purpose of monitoring the production on-line (Nichols, 1988). For instance, the measurement of melt index and monomer concentration by IR spectroscopy is fairly standard in the polymer industry. Therefore, these measurements can be used effectively to cope with uncertainty. In this way, the focus of optimization is shifting from a *model-based* framework to a *measurement-based* framework.

Since batch processes are intended to be run repeatedly, it is logical to exploit this for process optimization. The goal of batch-to-batch optimization is to find iteratively the optimal operating conditions in the presence of uncertainty, while performing the smallest number of suboptimal runs and preferably no unacceptable runs.

There have been several proposals reported in the literature that take advantage of batch-to-batch similarities for input profile optimization of batch processes (Filippi-Bossy *et al.*, 1989; Zafiriou and Zhu, 1990; Fotopoulos *et al.*, 1994). Run-to-run optimization is also of interest in the semiconductor and related industries (Scheid *et al.*, 1994; Zafiriou *et al.*, 1995; Adivikolanu and Zafiriou, 1998; Lee *et al.*, 2000). On the one hand, model-free approaches such as evolutionary operation (Box, 1957; Box and Draper, 1987; Clarke-Pringle and MacGregor, 1998) or MultiSimplex (MultiSimplex, 1999) use many batch runs to converge to an optimal solution. On the other hand, model-based approaches could converge faster but they suffer from the lack of accuracy of the model, especially when the model needs to be identified in the presence of noise (Martinez, 2000). As a compromise, a scheme that only necessitates the availability of a simplified model will be proposed in this paper.

Constraints play an important role in optimization. In continuous processes, the optimal operating policy is often determined by constraints (Maarleveld and Rijnsdorp, 1970). The same is true for batch processes, but the dependence of the input profile on the constraints is considerably more involved (especially terminal constraints, i.e., constraints which depend only on the final condition of the batch). Typical terminal constraints arise from selectivity considerations, where the concentration of particular species (e.g. side products) must be less than specified values to facilitate down-stream processing or simply to avoid additional separation steps.

The run-to-run optimization methodology proposed here is based on the concept of invariants (Visser *et al.*, 2000; Bonvin *et al.*, 2001). The idea is to identify those characteristics of the optimal solution that are invariant with respect to uncertainty and provide them as references to a feedback control scheme. It will be shown in this paper that, in the run-to-run framework, the invariants correspond to the active terminal constraints and cost sensitivities with respect to input parameters. Thus, a key issue in achieving optimality is to keep the terminal constraints active despite uncertainty.

Section 2 introduces the problem of measurement-based optimization while Section 3 reviews the available run-to-run optimization schemes. The invariant-based optimization is introduced in Section 4, and the optimization via regulation of invariants is discussed in Section 5. The results are illustrated on a simulated chemical reaction system with selectivity constraints in Section 6, and Section 7 concludes the paper.

2. MEASUREMENT-BASED OPTIMIZATION OF BATCH PROCESSES

The study will be restricted to a typical problem structure frequently encountered in batch process optimization. In most batch chemical processes, the manipulated inputs are flow rates that enter the system equations in an affine manner. Examples of typical manipulated inputs include flow rates of hot and cold fluids, and flow rates of reactants in the semi-batch scenario. The corresponding systems are called affine-in-input or control-affine systems. Furthermore, the batch objective involves meeting certain specifications only at the *end* of the batch. Typical cost functions are the maximization of yield or selectivity at final time. Thus, the objective function depends only on the final state. Without loss of generality, the final time, t_f, will be assumed to be fixed, since the case of free final time can be transformed into one where the final time is fixed.

2.1 *Problem Formulation*

Let us assume that the parameters of the model, θ, are unknown but do not vary from batch to batch. This assumption is made since typically only this type of uncertainty can be handled in a run-to-run framework. The methodology could also be applied to the case where θ varies 'slowly' from batch to batch. However, fast batch-to-batch variations and random variations within the batch cannot be handled with the run-to-run schemes presented in this paper.

To cope with this uncertainty, the run-to-run optimization utilizes measurements taken from one batch operation to improve the operating policy in subsequent batches. The fact that batch operation is repetitive is exploited to converge to the optimal operating policy over several batch runs. The run-to-run terminal-cost optimization problem of control-affine systems in the presence of uncertainty and measurements can be formulated mathematically as follows:

$$\begin{aligned}
\min_{u^k(t)} J^k &= \phi(x^k(t_f)) && (1)\\
s.t.\ \dot{x}^k &= f(x^k,\theta) + g(x^k,\theta)\ u^k + d^k && (2)\\
x^k(0) &= x_0\\
y^k &= h(x^k,\theta) + v^k\\
S(x^k,u^k) &\le 0\\
T(x^k(t_f)) &\le 0\\
\text{given}\ \ & y^j(i),\ \forall\, i,\ \forall j < k
\end{aligned}$$

where J^k is the cost function, $x^k(t) \in \Re^n$ are the states, $u^k(t) \in \Re^m$ are the inputs, and $d^k(t)$ is the process noise of the k^{th} batch. Let $y(t) \in \Re^p$ be the combination of states that can be measured, $y^k(i)$ the i^{th} measurement in time taken during the k^{th} batch, and $v^k(t)$ the measurement noise. $f : \Re^n \rightarrow \Re^n$ and $g : \Re^n \rightarrow \Re^{n \times m}$ describe the system dynamics, S a σ-dimensional vector of path constraints (input bounds and state constraints), T a τ-dimensional vector of terminal constraints, and ϕ the cost function. The objective is to utilize the measurements from the previous $(k-1)$ batches to handle the uncertainty in θ and find the optimal input policy for the k^{th} batch.

2.2 *Piecewise Analytic Characterization of the Optimal Solution*

The solution of terminal-cost optimization of control-affine systems has the following properties (Bryson, 1999):

- The inputs are in general discontinuous.
- The inputs are analytic in the intervals in between discontinuities. The time at which an input switches from one interval to another is called a *switching time.*
- Two types of arcs (nonsingular and singular) are possible between switching instants. In a nonsingular arc, the corresponding input is determined by a path constraint, whilst the input lies in the interior of the feasible region in a singular arc (Palanki *et al.*, 1993).

The pieces described above are sequenced in an appropriate manner to obtain the optimal solution. Though analytical expressions are available for either type of arcs, the inputs (especially in singular arcs) can be considered to be piecewise constant or linear for ease of implementation. Thus, the type and sequence of arcs, the switching times, and possibly a few variables corresponding to the singular intervals, completely parameterize the inputs. Determining the type and sequence of the arcs is, in general, a combinatorial problem, which annihilates the advantage of having an analytical solution in each arc.

So, what is proposed here is to use the standard numerical optimization on a simplified model of the system and the process engineer can interpret the numerical results to infer the type and sequence and of arcs (during which intervals the input bounds or path constraints are active, or whether the input is singular, and which terminal constraints are active). In this way, the shape of the optimal solution is available, the parameters for which will be adapted in a run-to-run manner as discussed later.

3. RUN-TO-RUN OPTIMIZATION SCHEMES

Let the decision variables of the input parameterization, i.e., the switching times and the variables describing the inputs in the singular intervals, be represented by π, and let π^k be the set of decision variables for the k^{th} batch run. The goal of the run-to-run optimization is to choose the decision variables in such a manner that, as k increases, the computed solution approaches the (unknown) optimal solution for the real system. Depending upon whether or not a model is used, various strategies for run-to-run optimization are possible as will be discussed in the following subsections.

3.1 *Model-free Evolutionary Optimization*

In this approach, no model is used, and the performance of a proposed input is evaluated experimentally. The procedure for evolutionary optimization is as follows:

(1) Parameterize the inputs using a finite number of decision variables π and choose the corresponding initial values.
(2) Run the batch with the given inputs and compute the performance index and the path and terminal constraints from the measurements.
(3) Sequentially perturb every component of π, each time re-running the batch in order to calculate the corresponding gradient and, thus, the search direction.
(4) Use an optimization algorithm (such as steepest descent) to update π. Repeat Steps 2–4 until convergence.

As can be seen, $\dim(\pi) + 1$ batch runs are necessary at each optimization iteration. Also note that the optimization algorithms that do not use gradient information typically converge more slowly, thereby requiring even more process runs.

3.2 *Evolutionary Optimization with Model-based Gradient*

The expensive part in the evolutionary optimization approach, in terms of the number of batch runs, is the calculation of the gradient (Step 3). The key idea in the model-based gradient approach is to use a dynamic model of the process, instead of an experimental run, to calculate the gradient. The model can be run as many times as the dimension of π to obtain the gradient. Another possibility is to use the Hamiltonian formulation, where the gradient is calculated from the states and adjoint variables. The states are measured or inferred from an experimental run. The adjoint variables are obtained from the model of the system by integrating the adjoint equations backward in time (Zafiriou and Zhu, 1990; Dong *et al.*, 1996).

3.3 *Optimization via Model Refinement*

This approach uses a model of the process for optimization and refines it using information gathered from previous batches. An optimization problem is solved before each batch run using the refined model. The procedure is as follows:

(1) Choose initial guesses for the parameters θ.
(2) Use the model and an optimization algorithm to obtain the optimal π.
(3) Run the batch with the optimal π.
(4) Use an identification algorithm and all the available measurements to obtain a new estimate for θ. Repeat Steps 2–4 until convergence.

3.4 *Discussion*

The model-free evolutionary optimization has the drawback of using numerous batch runs to calculate the gradient, whilst, when a model is used, the accuracy of the model becomes critical. If the model has to be refined, care should be taken to guarantee that the input is persistently exciting to uncover the parameters that have to be identified. This is normally not the case when the optimal input is implemented. Thus, there is a clear conflict between the identification and optimization objectives (Roberts and Williams, 1981). This conflict has also been studied in the adaptive control literature where it is referred to as the dual control problem (Wittenmark, 1995).

The scheme proposed in this paper attempts to resolve this conflict. The scheme lies in between the model-free evolutionary optimization and evolutionary optimization with model-based gradient. The gradient is obtained implicitly from the structure of the problem using the concept of invariants. Thus, no model is used for the implementation and no parameters need to be adapted. However, a *simplified model* is necessary to devise the scheme, namely to determine the shape of the inputs correctly.

4. INVARIANT-BASED OPTIMIZATION

The aim of this paper is to identify those important characteristics of the optimal solution which are invariant under uncertainty and provide them as references to a feedback control scheme. Thus, optimality is achieved by regulating the chosen references, and no numerical optimization is required from run-to-run.

To guarantee optimality despite uncertainty, the necessary conditions have to be satisfied even in the presence of uncertainty. Thus, the invariants simply correspond to the necessary conditions.

4.1 *Necessary Conditions of Optimality*

After parameterization of the inputs as $u(\pi, x, t)$, the optimization problem (1) can be written as:

$$\begin{aligned} \min_{\nu^k, \pi^k} \quad & \tilde{J}^k = \phi(x^k(t_f)) + (\nu^k)^T T(x^k(t_f)) \qquad (3) \\ s.t. \quad & \dot{x}^k = f(x^k, \theta) + g(x^k, \theta)\, u(\pi^k, x^k, t) + d^k \\ & x^k(0) = x_0 \\ & y^k = h(x^k, \theta) + v^k \\ & \nu^k \geq 0 \\ \text{given} \quad & y^j(i), \ \forall\, i, \ \forall j < k \end{aligned}$$

where ν^k are the τ-dimensional Lagrange multipliers for the terminal constraints. When the inputs are determined by path constraints, $u(\pi, x, t)$ in that arc is obtained by differentiating $S(x, u)$ with respect to time. Since the path constraints are implicit in the choice of $u(\pi, x, t)$, they do not appear explicitly in (3).

Without loss of generality, let all terminal constraints be active - the inactive constraints being simply removed from the optimization problem. Consequently, the number of decision variables arising from the parsimonious parameterization, n_π, need to be larger than τ in order to satisfy all terminal constraints.

An important assumption here is that the uncertainty does not affect the sequence of arcs and the set of active terminal constraints, though, the decision variables associated with the input parameterization can vary with the uncertainty. This assumption can be easily verified with a few numerical experiments.

The necessary conditions of optimality for (3) are:

$$T(x^k(t_f)) = 0 \ \text{ and } \ \frac{\partial \phi}{\partial \pi^k} + (\nu^k)^T \frac{\partial T}{\partial \pi^k} = 0 (4)$$

Next, those variations in π^k which affect the terminal constraints are separated from those which do not affect the terminal constraints. For this, a transformation $(\pi^k)^T \rightarrow [(\bar{\pi}^k)^T \ (\tilde{\pi}^k)^T]$ is sought such that $\bar{\pi}^k$ is a τ-dimensional vector and $\tilde{\pi}^k$ is of dimension $(n_\pi - \tau)$ with $\frac{\partial T}{\partial \tilde{\pi}^k} = 0$. This transformation is, in general, nonlinear and requires the solution of some partial differential equations. However, a linear transformation which satisfies these properties can always be found in the neighborhood of the optimum. Then, the necessary conditions for optimality of (3) are:

$$T(x^k(t_f)) = 0, \quad \text{and}$$
$$\frac{\partial \phi}{\partial \tilde{\pi}^k} = 0, \quad \frac{\partial \phi}{\partial \bar{\pi}^k} + (\nu^k)^T \frac{\partial T}{\partial \bar{\pi}^k} = 0 \qquad (5)$$

The constraints $T(x^k(t_f)) = 0$ determine the τ decision variables $\bar{\pi}^k$, whilst $\tilde{\pi}^k$ are determined from the optimality conditions $\frac{\partial \phi}{\partial \tilde{\pi}^k} = 0$. Thus, $\bar{\pi}^k$ are referred to as the nonsingular parameters (determined by the terminal constraints) and $\tilde{\pi}^k$ as the singular parameters (not determined by the terminal constraints). The Lagrange multipliers ν^k are calculated from $\frac{\partial \phi}{\partial \bar{\pi}^k} + (\nu^k)^T \frac{\partial T}{\partial \bar{\pi}^k} = 0$.

4.2 *Signals Invariant under Uncertainty*

From the necessary conditions of optimality, a set of signals $I^k = h(x^k(t_f))$ is constructed such that the optimum corresponds to $I^k = 0$, also in the presence of uncertainty. For the nonsingular parameters, the invariants correspond to the terminal constraints $h(x^k(t_f)) = T(x^k(t_f))$ and, for the singular parameters, to the cost sensitivities $h(x^k(t_f)) = \frac{\partial \phi}{\partial \tilde{\pi}^k}$.

4.3 *Sensitivity of the Cost*

The sensitivity of the cost to non-optimal operation is in general much larger for nonsingular parameters than for singular parameters. For a nonsingular parameter $\bar{\pi}_i^k$, $\frac{\partial \phi}{\partial \bar{\pi}_i^k}$ is nonzero since the necessary condition of optimality demands $\frac{\partial \phi}{\partial \bar{\pi}_i^k} + (\nu^k)^T \frac{\partial T}{\partial \bar{\pi}_i^k} = 0$. So, when $\bar{\pi}_i^k$ deviates from its optimal value, the change in cost is proportional to $\frac{\partial \phi}{\partial \bar{\pi}_i^k}$ which is non-zero. In contrast, for a singular parameter $\tilde{\pi}_i^k$, $\frac{\partial \phi}{\partial \tilde{\pi}_i^k} = 0$. So, a small deviation in $\tilde{\pi}_i^k$ will cause a negligible loss in cost. In summary, it is far more important to regulate the terminal constraints T than the sensitivities $\frac{\partial \phi}{\partial \tilde{\pi}^k}$. Consequently, it is often sufficient in practical situations to focus attention on the terminal constraints only.

5. OPTIMIZATION *VIA* CONTROL OF INVARIANTS

As shown in the previous section, optimality can be achieved by regulating the invariants *via* manipulation of the decision variables π corresponding to the input parameterization. The invariants are terminal constraints and sensitivities. Together, these are referred to as *generalized constraints*. As shown above, it is often sufficient to adapt the nonsingular parameters $\bar{\pi}^k$ and keep the singular parameters $\tilde{\pi}^k$ at the values determined by the numerical optimization using a nominal model.

It is also possible to consider the class of optimization problems where none of the parameters is singular. Then, in the absence of terminal constraints, the optimal operation corresponds to constant conditions (inputs on the bounds with no switching, such as the batch mode in chemical reaction systems). Therefore, for these cases, the optimization potential arises solely from the introduction of terminal constraints such as selectivity considerations.

For a given parameterization of the input, the optimal solution consists of choosing π^k such that all the invariants are zero (i.e., terminal constraints are active and $\frac{\partial \phi}{\partial \tilde{\pi}^k} = 0$). Tracking the invariants can also be interpreted as implicit gradient computation. Therefore, the deviations from the constraints represent a measure of non-optimality and gives the direction to update π^k. The procedure is as follows:

(1) Use prior knowledge or a simplified model to determine the active terminal constraints. Inactive constraints are removed from the optimization problem.
(2) Parameterize the input and make an initial guess for the parameters π^k, $k = 1$.
(3) Run the batch using the input corresponding to π^k. Compute the invariants I^k from the measurements at t_f.
(4) Update the input parameters using $\pi^{k+1} = \pi^k + G\, I^k$, with G being an appropriate gain matrix. Set $k = k + 1$ and repeat Steps 3–4 until convergence.

This procedure is similar to evolutionary optimization with the difference that additional batch runs are not required to compute the gradient. The gradient is obtained directly from the deviations of the invariants from zero. Thus, the method proposed does not require to perturb the system to calculate the gradient. So, the optimization *via* control of invariants considerably reduces the number of batch runs required for convergence.

Also, in the presence of disturbances, the gradient obtained using deviations of invariants from zero has a lesser variance (by a factor 2) than the gradient calculated using finite differences. This is due to the fact that the gradient calculation using

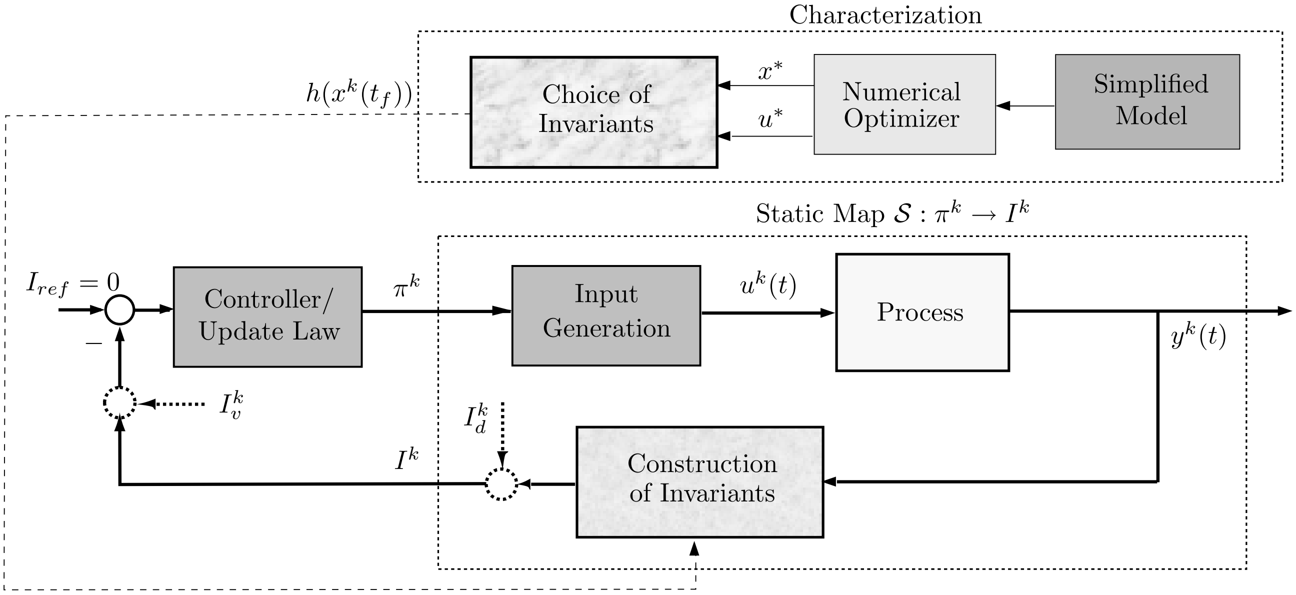

Fig. 1. Optimization *via* control of invariants

finite differences uses the difference of two quantities corrupted by noise, whilst that obtained from invariants uses the difference between a fixed value and a single noisy quantity.

Another attractive feature of this approach is its model-free implementation. However, a simple model is necessary to determine the active constraints on which the solution will lie.

The update law $\pi^{k+1} = \pi^k + G\ I^k$ can be implemented easily using a feedback controller. The system to be controlled is a *static* map represented by $\mathcal{S} : \pi^k \to I^k$ and the reference is $I_{ref} = 0$ (Figure 1). Note that $\mathcal{S}$ does not represent the dynamic system (2), but rather a static map between the input parameterization and the invariants. In this controlled system, the independent variable is the batch index k and not the time t. The controller, which has to be dynamic to avoid algebraic inconsistency, is the only dynamic element in the control scheme. Figure 1 also indicates the effects of process noise (I_d^k) and measurement noise (I_v^k) on the invariants.

The idea presented here is similar to that of tracking constraints using feedback for the sake of optimality (Maarleveld and Rijnsdorp, 1970). The particularities of the present work are: (i) invariants which constitute *terminal* constraints and cost sensitivities are considered, and (ii) the independent variable of the control loop is the batch number. Thus, the solution is implemented open-loop within each batch, and the feedback works over the successive batches.

The update law (Step 4) represents an integral controller. To improve the rate of convergence, a proportional or a derivative term can be added, thus resulting in a PI or PID controller, for the design of which standard controller design methodologies can be used. To reject the effect of the process noise I_d^k, a high-gain controller is required, whilst to be insensitive to the measurement noise I_v^k, the gain of the controller has to be as low as possible. Thus, a compromise has to be reached in the controller design.

Note that the static map $\mathcal{S}$ is a square system since, as shown in Section 4, the number of invariants is equal to the number of parameters. Even if the singular parameters and sensitivities are removed, the map $\bar{\mathcal{S}} : \bar{\pi}^k \to T^k$ is again square. However, the static map $\mathcal{S}$ varies considerably with the operating point, which complicates the controller design. Three possibilities exist for the design of controllers: (a) design a single centralized multi-input multi-output controller, (b) perform input-output pairing and design decentralized single-input single-output controllers for each input pair, and (c) use a decoupling scheme so that a series of single-input single-output controllers can be employed.

Once the solution has converged to the optimum, the control loop need not kept active, and the adaptation can be stopped. In this case, convergence to the optimum can be tested by looking at the error $I_{ref} - I^k$ or the adaptation can be stopped after a fixed number of batch runs.

Since constraint violations may mean that the batch is wasted, it is better to be sub-optimal than to violate the constraints. Thus, heuristics can be used to first render the solution feasible and then approach the constraint from within the feasible region. In addition, conservatism needs to be introduced to account for disturbances (process and measurement noise). For example, a backoff from the active constraint $I_{ref} = 0$ can be introduced by defining I_{ref} negative so that I^k remains negative despite disturbances (Morari and Zafiriou, 1989):

$$I_{ref} = -\max_z \left| \frac{I_d(z)}{1 + G(z)\,\mathcal{S}} \right| - \max_z \left| \frac{G(z)\,\mathcal{S}\,I_v(z)}{1 + G(z)\,\mathcal{S}} \right|$$

where $G(z)$ is the controller transfer function. Note that $I_d(z)$ and $I_v(z)$ are the z-transforms of the noise sequence I_d^k and I_v^k, respectively.

6. EXAMPLE

The methodology proposed here will be applied to a semi-batch reactor system for the acetoacetylation of pyrrole with diketene. The reaction system considered is described below and, for a more detailed description of the process and model, see Ruppen *et al.* (1998):

$$\begin{aligned} A + B &\rightarrow C \\ 2\,B &\rightarrow D \\ B &\rightarrow E \\ C + B &\rightarrow F \end{aligned}$$

where A: pyrrole, B: diketene, C: 2-acetoacetyl pyrrole, D: dehydroacetic acid, E: oligomers and F: undesired by-products. The optimization problem is

$$\begin{aligned} \max_{u(t)} J &= c_c(t_f)V(t_f) \\ s.t.\ \dot{c_a} &= -k_1 c_a c_b - (u/V)\,c_a \\ \dot{c_b} &= -k_1 c_a c_b - 2\,k_2 c_b^2 - k_3 c_b - k_4 c_b c_c \\ &\quad +(u/V)\,(c_{b_{in}} - c_b) \\ \dot{c_c} &= k_1 c_a c_b - k_4 c_b c_c - (u/V)\,c_c \\ \dot{c_d} &= k_2 c_b^2 - (u/V)\,c_d \\ \dot{V} &= u \\ & u_{min} \le u \le u_{max} \\ & c_b(t_f) - c_{b_{f,max}} \le 0 \\ & c_d(t_f) - c_{d_{f,max}} \le 0 \end{aligned}$$

where c_a, c_b, c_c and c_d are the concentrations of A, B, C, and D in mol/l, respectively. The feed consists of only the species B with concentration $c_{b_{in}}$. The goal is to maximize the number of moles of C at the final time, by manipulating the feedrate u (l/min), whilst satisfying the two terminal constraints on the concentrations of B and D. It is assumed that the concentrations of Species B and D are measured at final time.

k_1	0.053	l/mol min
k_2	0.128	l/mol min
k_3	0.028	min^{-1}
k_4	0.001	l/mol min
$c_{b_{in}}$	5	mol/l
$c_{b_{f,max}}$	0.025	mol/l
$c_{d_{f,max}}$	0.15	mol/l
u_{min}	0	l/min
u_{max}	0.002	l/min
t_f	250	min
c_{a_0}	0.72	mol/l
c_{b_0}	0.05	mol/l
c_{c_0}	0.08	mol/l
c_{d_0}	0.01	mol/l
V_0	1	l

Table 1. Parameter values and initial conditions

For the parameter values given in Table 1, the optimal solution can be computed numerically (Figure 6). The input can be characterized as having three intervals: (i) input at its upper bound, (ii) input being singular, approximated by a constant value, and (iii) input at its lower bound. From the optimal solution, the natural parameterization corresponds to the following set: t_m, switching time between the maximum input and the singular interval; u_s, the constant feedrate during the singular interval; and t_s, the switching time between the singular interval and the minimum input. The maximum amount of C obtained is 0.4884 mol. The optimal input parameters are $t_m = 11.04$ min, $u_s = 0.0012$ l/min, and $t_s = 224.38$ min.

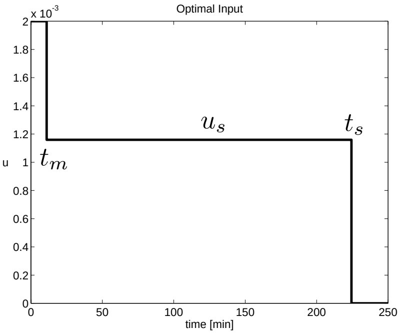

Fig. 2. Nominal computed optimal input (the singular arc is approximated by a constant value u_s)

Two degrees of freedom are necessary to meet the two terminal constraints. Thus, among the three parameters that have been chosen, one of them is singular. To find $\tilde{\pi}$, consider the influence of the three parameters on the two constraints. The gain matrix $\mathcal{M} : \pi \rightarrow T$, with $\pi = [t_m, u_s, t_s]^T$ and $T(x(t_f)) = [c_d(t_f) - c_{d_{f,max}}, c_b(t_f) - c_{b_{f,max}}]^T$ is given by:

$$\mathcal{M} = \begin{bmatrix} 0.6 \times 10^{-3} & 1.5 \times 10^{2} & 0.5 \times 10^{-3} \\ 1.9 \times 10^{-5} & 0.2 \times 10^{2} & 1.1 \times 10^{-3} \end{bmatrix}$$

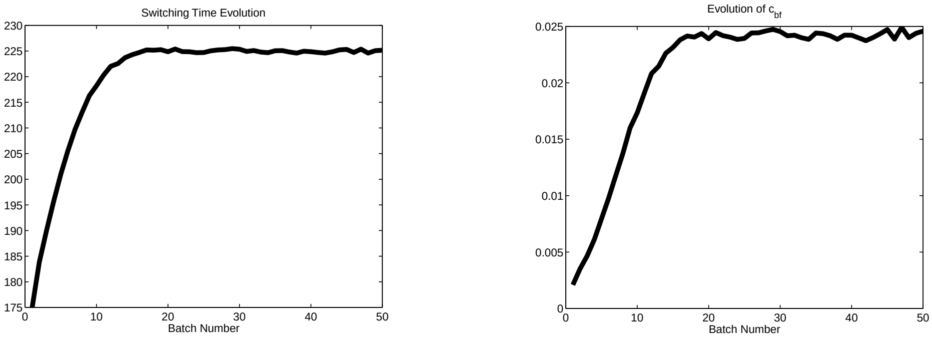

Fig. 3. Performance of $t_s \rightarrow c_b$ loop

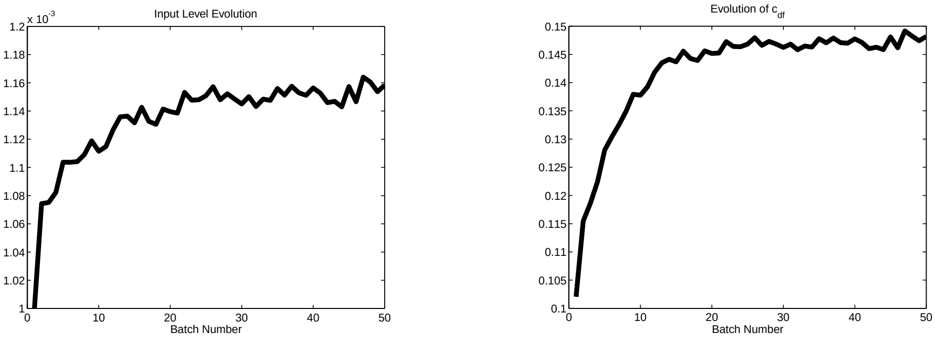

Fig. 4. Performance of $u_s \rightarrow c_d$ loop

In the neighborhood of the optimum, the singular parameter $\tilde{\pi}$ is given by the null space of $\mathcal{M}$ and corresponds to $\tilde{\pi} = t_m - 4 \times 10^{-6} u_s + 4.7 \times 10^{-2} t_s$. Since the contributions from u_s and t_s are negligible, t_m essentially acts as the singular parameter. Thus, t_m can be left constant at its numerically determined value of 11.04 min, and the other two, $\bar{\pi} = [u_s, t_s]^T$, are adapted using measurements in order to meet the two constraints. The static map $\bar{\mathcal{S}} : \bar{\pi} \rightarrow T$ obtained at the optimum solution corresponds to the last two rows of $\mathcal{M}$.

The scale-independent diagonal dominance of $\bar{\mathcal{S}}$ is tested using the relative gain array technique, which gives $\lambda = 1.054$ (Ogunnaike and Ray, 1994). This implies that two independent control loops could be constructed, with $c_b(t_f)$ being paired with t_s and $c_d(t_f)$ with u_s. PI-controllers tuned for the static map $\mathcal{S}$ were used for the $t_s \rightarrow c_b(t_f)$ and $u_s \rightarrow c_d(t_f)$ loops with proportional gains, $K_b = 100\,\mathrm{l\,min/mol}$, $K_d = 10^{-3}\ \mathrm{l}^2/\mathrm{mol\ min}$, and integral time constants, $T_{i_b} = 0.33$ min and $T_{i_d} = 2$ min, respectively.

The initial input parameterization $u_s = 0.001$ l/min and $t_s = 175$ min is used. The initial values are chosen in a conservative manner so that the optimal solution is approached from the safe side. A 5% zero-mean Gaussian multiplicative measurement noise is added to the concentration measurements of B and D. The results of a typical series of runs are shown in Figures 3-5. It can be seen that neither constraint is violated during the series of 50 batches, therefore producing usable product each time. Thus, the backoff introduced efficiently accounts for the measurement noise. Figure 5 illustrates the motivation for tracking terminal constraints for the purpose of optimization: as the constrained variables are pushed closer to their limits, the objective function is improved, in this case producing more of the desired product. It is important to note that the main improvement in the cost is realized in the first 10 batches. Clearly after 25 batch runs, the improvement in the cost is marginal and the adaptation can be stopped. The cost averaged over 100 realizations, each consisting of run-to-run adaptation over 50 batches, is $J = 0.4878$ mol. This small reduction from the nominal optimum of 0.4884 mol is due to the backoff introduced.

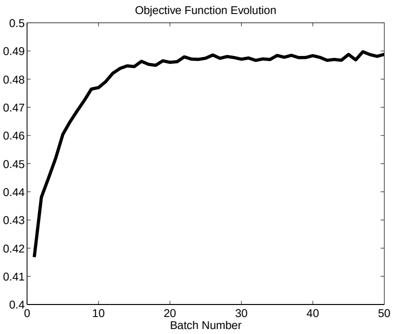

Fig. 5. Objective function evolution

7. CONCLUSION

This work has demonstrated the effectiveness of tracking terminal constraints for the run-to-run optimization of a class of batch processes. The update of the parameters required for optimization is realized using a simple control scheme. The update direction is obtained implicitly, thereby avoiding the need to excite the system for the estimation of the gradient.

The methodology has been adapted such that the production of the desired product increases with each progressing batch, without constraint violations. However, note that the technique proposed is not inherently immune to constraint violations. A proper choice of backoffs, initial conditions and controller tuning are required to ensure that none of the constraints is violated.

The design of the controller is an area which merits further research. The gain matrix $\mathcal{S}$ varies with the operating point, i.e., from one run to the next. If this variation is large, more sophisticated control methodologies are needed to avoid instability problems.

ACKNOWLEDGMENT

Financial support for the participation of C. Primus was provided by the University of Washington Center for Process Analytical Chemistry (CPAC).

8. REFERENCES

Adivikolanu, S. and E. Zafiriou (1998). Robust run-to-run control for semiconductor manufacturing. In: *American Control Conference*. Philadelphia. pp. 3687–3691.

Bonvin, D. (1998). Optimal operation of batch reactors - a personal view. *J. Process Contr.* **8**(5–6), 355–368.

Bonvin, D., B. Srinivasan and D. Ruppen (2001). Dynamic optimization in the batch chemical industry. In: *Chemical Process Control VI*. Tucson, AZ. p. Submitted.

Box, G. E. P (1957). Evolutionary operation: a method for increasing industrial productivity. *Applied Statistics* **6**, 81–101.

Box, G. E. P. and N. R. Draper (1987). *Empirical Model-building and Response Surfaces*. John Wiley, New York.

Bryson, A. E. (1999). *Dynamic Optimization*. Addison-Wesley, Menlo Park, California.

Clarke-Pringle, T. L. and J. F. MacGregor (1998). Optimization of molecular weight distribution using batch-to-batch adjustments. *Ind. Eng. Chem. Res.* **37**, 3660–3669.

Dong, D., T. J. McAvoy and E. Zafiriou (1996). Batch-to-batch optimization using neural networks. *Ind. Eng. Chem. Res.* **35**, 2269–2276.

Filippi-Bossy, C., J. Bordet, J. Villermaux, S. Marchal-Brassely and C. Georgakis (1989). Batch reactor optimization by use of tendency models. *Comp. Chem. Eng.* **13**, 35–47.

Fotopoulos, J., C. Georgakis and H. G. Stenger (1994). Uncertainty issues in the modeling and optimisation of batch reactors with tendency modeling. *Chem. Engng. Sci.* **49**, 5533–5548.

Lee, J., I. Chin, K. S. Lee, J. Choi and J. H. Lee (2000). Control of wafer temperature uniformity in rapid thermal processing using an optimal iterative learning technique. In: *IFAC ADCHEM'00*. Pisa, Italy. pp. 123–128.

Maarleveld, A. and J. E. Rijnsdorp (1970). Constraint control on distillation columns. *Automatica* **6**, 51–58.

Martinez, E. C. (2000). Batch process modeling for optimization and reinforcement learning. *Comp. Chem. Eng.* **24**, 1187–1193.

Morari, M. and E. Zafiriou (1989). *Robust Process Control*. Prentice-Hall, London.

MultiSimplex, AB. (1999). *User's Guide to MultiSimplex(r) 2.0*. Karlskrona, Sweden.

Nichols, G. D. (1988). *On-line Process Analyzers*. John Wiley, New York.

Ogunnaike, B. A. and W. H. Ray (1994). *Process Dynamics, Modeling and Control*. Oxford University Press.

Palanki, S., C. Kravaris and H. Y. Wang (1993). Synthesis of state feedback laws for end-point optimization in batch processes. *Chem. Engng. Sci.* **48**(1), 135–152.

Ray, W. H. (1981). *Advanced Process Control*. McGraw-Hill, New York.

Roberts, P. D. and T. W. C. Williams (1981). On an algorithm for combined system optimization and parameter estimation. *Automatica* **17**, 199–209.

Ruppen, D., D. Bonvin and D. W. T. Rippin (1998). Implementation of adaptive optimal operation for a semi-batch reaction system. *Comp. Chem. Eng.* **22**, 185–189.

Scheid, G. W., S. J. Qin and T. J. Riley (1994). Run-to-run optimization, monitoring, and control on a rapid thermal processor. In: *AIChE Annual Meeting*. Dallas, TX.

Visser, E., B. Srinivasan, S. Palanki and D. Bonvin (2000). A feedback-based implementation scheme for batch process optimization. *J. Process Contr.* **10**, 399–410.

Wittenmark, B. (1995). Adaptive dual control methods: An overview. In: *IFAC Symposium on Adaptive Syst. in Control and Signal Proc.*. Budapest. pp. 67–72.

Zafiriou, E. and J. M. Zhu (1990). Optimal control of semi-batch processes in the presence of modeling error. In: *American Control Conference*. San Diego, CA. pp. 1644–1649.

Zafiriou, E., H. W. Chiou and R. A. Adomaitis (1995). Nonlinear model-based run-to-run control for rapid thermal processing with unmeasured variable extimation. *Electrochem. Soc. Proc.* **95**(4), 18–31.