Dynamic Optimization of Batch Processes: II. Handling Uncertainty Using Measurements

B. Srinivasan, D. Bonvin, E. Visser, S. Palanki*
Institut d'Automatique, École Polytechnique Fédérale de Lausanne,
CH-1015 Lausanne, Switzerland.
*Florida State University, Tallahassee, Florida, USA.

June 21, 2001

Abstract

The main bottleneck in using optimization in industry is the way uncertainty is handled, which forms the subject of this series of two papers. The first part dealt with the characterization of the nominal solution and proposed an approach to separate the constraint-seeking and the compromise-seeking components of the inputs. This second part reviews various strategies for optimization under uncertainty, namely the robust and measurement-based optimization schemes. A novel scheme, labeled invariant-based optimization, is proposed, where optimality is achieved by tracking references that remain invariant under uncertainty. The different approaches are compared via the simulation of a bioreactor for penicillin production.

Keywords: Batch processes, Dynamic optimization, Robust optimization, Run-to-run optimization, Batch-to-batch optimization, On-line optimization, Measurement-based optimization.

1 Introduction

Optimization of batch processes has received attention recently because, in the face of growing competition, it represents a natural choice for reducing production costs, improving product quality, meeting safety requirements and environmental regulations. In the companion paper [51] of this two-part series, the analysis and numerical computation of the optimal solution was discussed based of a nominal process model, i.e., without considering the uncertainty resulting from model mismatch and disturbances.

In practical situations, however, an exact model can rarely be found with affordable effort [8]. In many cases, the stoichiometry and kinetics of reaction systems are insufficiently characterized, thereby leading to reaction lumping and to macroscopic modeling using time-varying parameters. Furthermore, since the model parameters are usually estimated from laboratory-scale experiments, they might vary in commercial-scale reactors due to differences in mixing characteristics, heat and mass transfer, and unmodeled dynamics that get magnified in large-scale reactors. In addition,

there could be uncertainty in loading conditions, especially when reactants are added as solids and mixed with the solvent after charging. This can lead to batch-to-batch variations in product quality and poor reproducibility. In the presence of uncertainty, the classical *open-loop* implementation of off-line calculated optimal inputs may not lead to optimal performance [13, 41, 50]. Moreover, constraint satisfaction, which becomes important in the presence of safety constraints, may not be guaranteed unless a conservative strategy is adopted [25, 55].

Two different types of optimization are generally available for handling uncertainty. The essential difference relates to whether or not *measurements* are utilized in the calculation of the optimal strategy. A robust optimization approach is typically used in the absence of measurements. When measurements are available, a measurement-based optimization approach can help adapt to process changes and disturbances. This classification is somewhat analogous to robust control and adaptive control for regulation problems.

In this paper, the available robust and measurement-based optimization strategies will be briefly reviewed. A novel optimization approach for batch processes will then be proposed, where optimality is achieved by using feedback along with an appropriate choice of reference signals, thereby avoiding the need for on-line numerical optimization. These reference signals are chosen such that they ensure optimality and, furthermore, are invariant under uncertainty.

The paper is written in a tutorial style. Section 2 reviews various optimization schemes that can handle uncertainty. The robust optimization framework is developed in Section 3, while Section 4 presents various measurement-based optimization schemes. The invariant-based optimization framework is developed in Section 5. A bioreactor example is provided in Section 6 to illustrate theoretical developments, and conclusions are drawn in Section 7.

2 Overview of Batch Process Optimization

Batch optimization problems typically involve both dynamic and static constraints and fall under the class of *dynamic optimization* problems. Possible scenarios in dynamic optimization are depicted in Figure 1. The first level of classification depends on whether or not uncertainty (e.g., variations in initial conditions, unknown model parameters, or process disturbances) is considered. The standard approach is to discard uncertainty, leading to a nominal solution that may not even be feasible, let alone optimal, in the presence of uncertainty.

The second level concerns the type of information that can be used to combat uncertainty. If measurements are not available, a conservative stand is required. In contrast, conservatism can be reduced with the use of measurements. In the next level, the classification is based on whether or not a model is used to guide the optimization. The different scenarios are discussed in detail next.

In addition, since batch processes are typically repeated over time, the optimization problem can also be formulated in terms of finding an improvement in cost over several batches. The goal of batch-to-batch (or run-to-run) optimization is then to iteratively find the optimal operating conditions in the presence of uncertainty, while performing only a few sub-optimal runs and preferably no unacceptable ones [20, 59, 21].

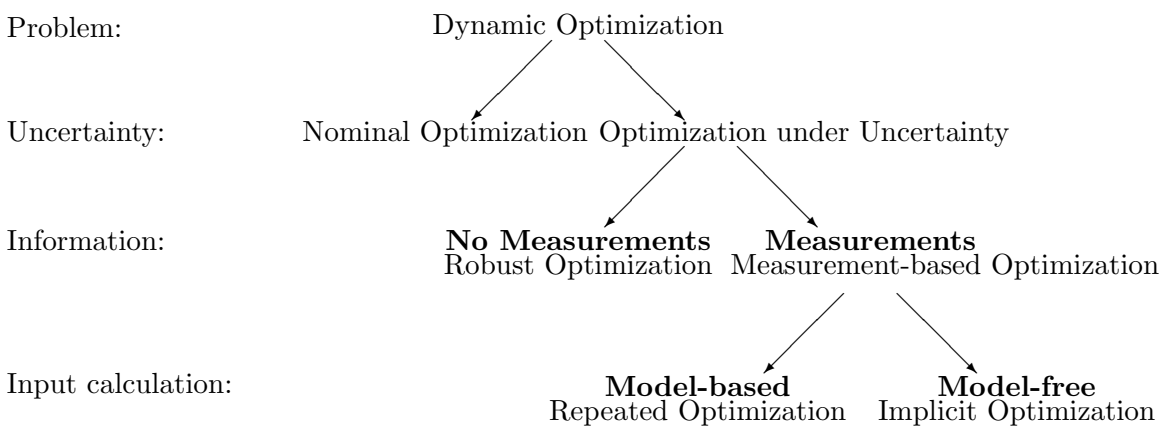

Figure 1: Dynamic optimization scenarios

2.1 Nominal Optimization

In nominal optimization, the uncertainty is simply discarded. The optimization objective typically corresponds to achieving a desired product quality at the most economical cost, or maximizing the product yield for a given batch time, i.e., the objective involves only specifications at the *end* of the batch. The terminal-cost optimization can be stated mathematically as follows (see Part I [51] for more information):

$$\min_{u(t)} J = \phi(x(t_f)) \tag{1}$$

$$s.t. \quad \dot{x} = F(x,u), \quad x(0) = x_0 \tag{2}$$

$$S(x,u) \leq 0, \quad T(x(t_f)) \leq 0 \tag{3}$$

where J is the scalar performance index to be minimized, x the n-vector of states with known initial conditions x_0, u the m-vector of inputs, F is a vector field describing the dynamics of the system, S the ζ-vector of path constraints (which include state constraints and input bounds), T the τ-vector of terminal constraints, ϕ a smooth scalar function representing the terminal cost, and t_f the final time. Without loss of generality, it will be assumed that the final time t_f is fixed.

Application of Pontryagin's Maximum Principle (PMP) to (1)–(3) results in the following Hamiltonian and adjoint equations [11]:

$$H = \lambda^T F(x,u) + \mu^T S(x,u) \tag{4}$$

$$\dot{\lambda}^T = -\frac{\partial H}{\partial x}, \quad \lambda^T(t_f) = \left.\frac{\partial \phi}{\partial x}\right|_{t_f} + \nu^T \left.\left(\frac{\partial T}{\partial x}\right)\right|_{t_f} \tag{5}$$

where $\lambda(t) \neq 0$ is the n-dimensional vector of adjoint states (Lagrange multipliers for the system equations), $\mu(t) \geq 0$ the ζ-dimensional vector of Lagrange multipliers for the path constraints, and $\nu \geq 0$ the τ-dimensional vector of Lagrange multipliers for the terminal constraints. The first-order necessary conditions for optimality are:

$$\frac{\partial H}{\partial u} = \lambda^T \frac{\partial F}{\partial u} + \mu^T \frac{\partial S}{\partial u} = 0 \tag{6}$$

$$\mu^T S = 0, \qquad \nu^T T = 0 \tag{7}$$

The solution of this nominal optimization problem, where uncertainty is ignored, is discussed in detail in the companion paper [51].

2.2 Optimization under Uncertainty

While the solution of the nominal optimization problem is important to elucidate the basic features of the implementation strategy (e.g. shape of the solution, number and types of arcs), it is necessary to modify this open-loop solution to account for uncertainty. The approaches used for handling uncertainty essentially differ in the type of process information available to compensate the effect of uncertainty.

- *Robust optimization:* In the absence of measurements, an open-loop off-line solution is sought that takes the uncertainty into account explicitly [53]. The uncertainty is dealt with by considering several possible values for the uncertain parameters. The optimization is performed by either considering the worst-case scenario or in an 'expected sense'. The input trajectories are then computed off-line once and utilized for all batches. However, this typically requires solving an optimization problem of dimension larger than without uncertainty [46]. The resulting solution is conservative but corresponds to the *best* possibility in the absence of measurements.
- *Measurement-based optimization:* Measurements are used to cope with uncertainty by adjusting the input profiles to account for both parametric uncertainty and disturbances. If off-line measurements are available, a *run-to-run optimization* approach can be used to account for parametric uncertainty. Process knowledge obtained from previous batches is used to update the operating strategy of the current batch. However, this approach does not account for the effect of process disturbances within the batch. When information is available in the course of the batch, an *on-line optimization* approach can be used. This approach, which seeks to optimize every batch run amidst uncertainty, is much less conservative than the robust optimization scheme and leads to a better cost.

The robust optimization approach is described in the next section and measurement-based optimization in Section 4.

3 Robust Optimization

3.1 Problem formulation

The nominal optimization problem described in (1)–(3) and solved in the companion paper [51] assumes the availability of a single (nominal) model that describes the process with sufficient accuracy. However, the model may carry a significant amount of uncertainty expressed in the form of uncertain model parameters, θ, and unknown disturbances, $d(t)$. Model parameters can be uncertain due to lack of knowledge or due to parameter variations resulting, for example, from reaction lumping. Disturbances such as changes in the quality of the utilities, variations or failure

in the dosing system also affect the process evolution. In general, disturbances can exhibit fast variations as opposed to parametric uncertainty which is time-invariant or slowly time-varying.

In the uncertain scenario, the terminal-cost optimization can be formulated as follows:

$$\min_{u(t)} J = \phi(x(t_f), \theta) \tag{8}$$

$$s.t. \quad \dot{x} = F(x, \theta, u) + d(t), \quad x(0) = x_0 \tag{9}$$

$$S(x, \theta, u) \leq 0, \quad T(x(t_f), \theta) \leq 0 \tag{10}$$

where θ is the vector of uncertain parameters, and $d(t)$ the unknown disturbance vector. In addition, the initial conditions x_0 could also be uncertain, mainly due to variations in reactant quality.

Robust optimization involves finding the optimal operating profile, where the cost function and the constraints are constructed by taking into account the modeling uncertainties explicitly [53]. The first task consists of characterizing the uncertainty, which can often be obtained from the parameter identification step. Depending upon the measurement error structure, (random noise or bounded error), the parametric uncertainty is either probabilistic or of the set membership type. When the parametric uncertainty is probabilistic, $\theta \in \Theta$ with the associated probability density function $p(\theta)$. With the set membership type of uncertainty, the only information available is $\theta \in \Theta$. In the latter case, Θ is essentially bounded, whereas this need not be the case in the former. In either case, the unknown constant or possibly time-varying parameters are described as random variables.

Similarly, the disturbance $d(t) \in \mathcal{D}$ is also either random noise with the associated probability density function $p(d)$ or simply bounded in the set $\mathcal{D}$. The main feature that distinguishes $d(t)$ from θ is the variation of the disturbance with time. Despite this fact, the ideas and tools used for handling parametric uncertainty can be easily extended to treat the presence of disturbances. Thus, for the sake of simplicity, parametric uncertainty will be considered below, and only brief comments will be added regarding the disturbance case.

Robust optimization has also been studied in the context of process design under uncertainty where the goal is to find design parameters which minimize an objective function [24, 40]. The techniques used therein are quite similar and references to the process design literature will be provided wherever appropriate.

3.2 Computation of the Probability Density Function for the States

Since the parameters and disturbances are random variables, the states are also random variables. Thus, to be able to calculate the cost function and the constraints, the probability density function of the state variables is necessary. Due to this additional calculation, the robust optimization problem is typically more involved than the solution of a similar problem with known parameters. The probability density function of $x(t)$ can be computed using one of the two approaches described below:

- *Discretization:* The probability density function $p(\theta)$ is discretized in the form of a numerical grid. Let D be the number of discretization points, and θ^j, $j = \{1, 2, \cdots, D\}$, the discrete values of θ with relative weighting w^j, $\sum_j w^j = 1$. The grid for θ can be chosen either

randomly (Monte-Carlo sampling) or through a systematic approach. The probability density function of the states is then obtained in a discrete fashion, i.e., $P[x = x^j] = w^j$, where x^j is obtained by simulating a copy of the system for the corresponding discretization point θ^j:

$$\dot{x}^j = F(x^j, \theta^j, u), \quad x^j(0) = x_0, \text{ for } j = \{1, \cdots, D\} \tag{11}$$

An augmented dynamic system of dimension nD needs to be integrated. However, such problems are ideal candidates for parallelization since the individual model evaluations are mutually decoupled [24]. In dealing with disturbances, the discretization should also be done in the time direction, i.e., different disturbance values during different time intervals should be considered. Such an approach is referred to as probability grid filtering [53].

- *Propagation:* If the probability density function of the states $x(t)$ for all time instants t can be assumed to follow (say) a gaussian distribution, the mean $E(x)$ and the standard deviation $\sigma(x)$ give complete information, i.e., the probability density function is described by $p(x) = \mathcal{N}(E(x), \sigma(x))$, where $\mathcal{N}$ is the normal distribution function. In such a case, $p(x)$ can be obtained by propagating only the mean $\bar{x} = E(x)$ and the standard deviation $\sigma(x)$. Also, it is assumed that the propagation of $\bar{x}$ can be computed from the system with the mean value of the parameter, $\bar{\theta} = E(\theta)$, and the dynamics of $\sigma(x)$ is determined from the linearization of (2) around the mean value of the parameter:

$$\dot{\bar{x}} = F(\bar{x}, \bar{\theta}, u), \qquad \bar{x}(0) = x_0 \tag{12}$$

$$\dot{\sigma}(x) = \left.\frac{\partial F}{\partial x}\right|_{\bar{x},\,\bar{\theta}} \sigma(x) + \left.\frac{\partial F}{\partial \theta}\right|_{\bar{x},\,\bar{\theta}} \sigma(\theta), \qquad \sigma(x(0)) = \sigma_0 \tag{13}$$

3.3 Expression for the Cost and the Constraints

Since the stochastic description of the states and model parameters cannot be used directly with the available numerical techniques [51], the probability, the expected value, or the extremum value of the random variable of interest (cost function and constraints) needs to be used in the formulation of the optimization problem. The possible formulations for the cost function and constraints are described below:

- *Probability:* Here, the probability of a random variable meeting a given set of specifications is considered [54]. This is natural for the constraints where the probability of constraint satisfaction should be greater than a specified confidence level c, $P\left[S(x, \theta, u) \leq 0\right] \geq c$. Also, in cases where the objective is to maximize the probability of the product meeting a given set of specifications (or, equivalently, minimizing the risk of an undesired outcome of the batch), the cost function can be a probabilistic measure, $J = P\left[\phi(x(t_f), \theta) \geq \eta\right]$, where η is an appropriately chosen threshold value.

  If the discretization approach is used, then the required probability is computed by picking those realizations where $\phi(x^j(t_f), \theta^j) \geq \eta$:

$$P\left[\phi(x(t_f), \theta) \geq \eta\right] = \sum_{j=1}^{D} I^j w^j, \quad \text{where } I^j = \begin{cases} 1 & \text{if } \phi(x^j(t_f), \theta^j) \geq \eta \\ 0 & \text{if } \phi(x^j(t_f), \theta^j) < \eta \end{cases} \tag{14}$$

  The discretization of θ can also be adapted as the optimization proceeds. For this, the region in θ where $\phi(x(t_f), \theta) \geq \eta$ or $S(x, \theta, u) \leq 0$ is identified. More discretization points

are provided in and around the feasible region to obtain a more accurate prediction of the required probability [26].

If the propagation approach is used to evaluate the probability density function $p(x)$, then the required probability is obtained by integrating $p(x(t_f))$ over the set where $\phi(x(t_f), \theta) \geq \eta$.

$$P\left[\phi(x(t_f), \theta) \geq \eta\right] = \int\limits_{\phi(x(t_f),\theta)\geq\eta} p(x(t_f))\ dx(t_f) \tag{15}$$

- *Expected value:* For the cost function, it is more common to consider the expected value of a random variable such as the expected product quality or quantity to be maximized, or the expected amount of undesired products to be minimized, $J = E\left[\phi(x(t_f), \theta)\right]$ [46]. Minimizing the variance of some target function $h(x(t_f), \theta)$ also falls under this framework since by formulating $\phi(x(t_f), \theta) = \left(h(x(t_f), \theta) - E\left[h(x(t_f), \theta)\right]\right)^2$, J becomes the variance of h. Expectations can also enter the optimization problem at the constraint level, a typical example being bounds on the variance of $h(x, \theta)$), $E\left[\left(h(x, \theta) - E\left[h(x, \theta)\right]\right)^2\right] \leq h_{max}$.

  The numerical evaluation of expectation-type functions is straightforward since they can be obtained as the weighted sum of results computed with the various parameter realizations, $E\left[\phi(x(t_f), \theta)\right] = \sum_{j=1}^{D} \phi(x^j(t_f), \theta^j) w^j$ or as the weighted integration of the probability density function, $E\left[\phi(x(t_f), \theta)\right] = \int \phi(x(t_f), \theta)\, p(x(t_f))\ dx(t_f)$.

- *Extremum value:* When Θ is bounded, the cost and the constraints can be treated for the worst (or, although rarely used, best) possible case. This is the only type of cost and constraints that can be used in the face of set membership type of uncertainty. The optimization of the cost becomes a min-max problem, $\min_{u(t)} \max_{\theta} \phi(x(t_f), \theta)$, [3]. The constraints are reformulated as $\max_{\theta} S(x, \theta, u) \leq 0$. Equivalent to the concept of variance, the difference $\max_{\theta} \phi(x(t_f), \theta) - \min_{\theta} \phi(x(t_f), \theta)$ can also be used as the cost or in the constraints.

  The calculation of extremum values is, in general, associated with the discretization approach. Care should be taken to ensure that the extrema are adequately represented by the discretization. While for nonlinear models the parameter values corresponding to an extremum can theoretically be found anywhere in Θ, experience with batch process models shows that the worst-case scenario is often found on the boundary of the set. Hence, in the case of interval uncertainty, where Θ is a hypercube, the corners need to be included in the discretization.

The optimization problem (8)–(10) can be transformed into a numerical (crisp) form using probabilities, expected values, or extremum values, for example as follows:

$$\begin{aligned} \min_{u(t)}\ J \quad &= \quad E[\phi(x(t_f), \theta)] \\ s.t. \qquad & \text{state probability density function computed from (11) or (12)–(13)} \\ & P\left[S(x, \theta, u) \leq 0\right] \geq c, \quad P\Big[T(x(t_f), \theta) \leq 0\Big] \geq c \end{aligned} \tag{16}$$

where c is the confidence level for constraints satisfaction. For Problem (16), any of the optimization methods described in [51] can be used. The choice of algorithm should be guided by criteria such as the number of states, the nature and the number of constraints, and the number of discrete parameter values.

3.4 Concept of Backoff

Let $\bar{\theta}$ be the nominal value of the parameters and $\bar{x}$ be the evolution of the states corresponding to the parameters $\bar{\theta}$. In the absence of uncertainty, the active path constraints are those with $S(\bar{x}, \bar{\theta}, u) = 0$. However, in the presence of uncertainty, the condition $P\left[S(x, \theta, u) \leq 0\right] = c$ determines the active constraints. For the probability of constraint satisfaction to be equal to the confidence level c, the nominal value $S(\bar{x}, \bar{\theta}, u)$ cannot be zero (i.e., right on the constraints), but has to be pushed inside the feasible region or backed off. Let $b_S > 0$ be the backoff required for constraint satisfaction in the presence of uncertainty, i.e., $S(\bar{x}, \bar{\theta}, u) + b_S = 0 \Rightarrow P\left[S(x, \theta, u) \leq 0\right] = c$. The backoff b_S is determined by the variance of $S(x, \theta, u)$ and, clearly, the larger the spread in $S(x, \theta, u)$, the larger the necessary backoff [56]. Thus, with the concept of backoff, the optimization problem (16) can be rewritten as:

$$\min_{u(t)} J = \phi(\bar{x}(t_f), \bar{\theta}) \tag{17}$$

$$s.t. \quad \dot{\bar{x}} = F(\bar{x}, \bar{\theta}, u), \quad \bar{x}(0) = x_0 \tag{18}$$

$$S(\bar{x}, \bar{\theta}, u) + b_S \leq 0, \quad T(\bar{x}(t_f), \bar{\theta}) + b_T \leq 0 \tag{19}$$

where $\bar{x}$ is the nominal value of the states and $b_S > 0$ and $b_T > 0$ are the backoffs appropriately chosen to account for uncertainty.

The concept of backoff close to that of the two stage stochastic programming approach [39] used for optimal process design. If the backoffs were known, then the formulation (17)-(19) could be solved to provide the robust optimal solution. However, the backoffs have to be consistent with the distributions of $S(x, \theta, u)$ and $T(x(t_f), \theta)$ along the computed optimum solution. This calls for an iterative procedure as described below:

1. Guess the initial backoffs b_S^0 and b_T^0, $k = 0$.
2. Compute the optimal solution $u^k(t)$ of (17)-(19) using backoffs b_S^k and b_T^k.
3. Compute the probability density function of the states for the optimal solution using (11) or (12)-(13) and thereby the distributions of $S(x^k, \theta, u^k)$ and $T(x^k(t_f), \theta)$.
4. Choose the backoff b_S^{k+1} such that $S(\bar{x}^k, \bar{\theta}, u^k) + b_S^{k+1} = 0 \Rightarrow P\left[S(x^k, \theta, u^k) \leq 0\right] = c$. Also, choose b_T^{k+1} accordingly.
5. Set $k = k + 1$ and repeat Steps 2-5 until convergence of b_S^k and b_T^k occurs or the maximum number of iterations allowed is reached.

The main disadvantage of this iterative approach is the absence of guaranteed convergence. However, if the procedure converges, it might be computationally attractive since only the state probability density function corresponding to the optimal solution needs to be computed.

A comparison of (8)-(10) and (17)-(19) shows that the only difference is the introduction of the backoffs b_S and b_T, which are clear indications of the amount of conservatism that is needed to handle the uncertainty. Also, using the definition of Lagrange multipliers [51], the loss in performance caused by the conservatism is given by $\int_0^{t_f} \mu^T b_S \, dt + \nu^T b_T$. Thus, the larger the backoff, the larger the loss in performance, thereby motivating the reduction of conservatism using measurements.

4 Measurement-based Optimization

The main disadvantage of robust optimization schemes is their conservative nature. The best that can be done in the absence of measurements is to try to be ready for all possible scenarios. However, in many batch processes, measurements are taken through: (i) on-line sensors such as thermocouples, pH probes, spectroscopic sensors, and/or (ii) off-line analytical methods such as HPLC and GC. These measurements can be used effectively to cope with uncertainty, thereby leading to less conservative optimization strategies.

4.1 Problem Formulation

The optimization problem in the presence of uncertainty and measurements can be formulated as follows:

$$\min_{u^k_{[t_l,t_f]}} \; J^k = \phi(x^k(t_f),\theta) \tag{20}$$

$$s.t. \quad \dot{x}^k = F(x^k,\theta,u^k) + d^k,\; x^k(0) = x^k_0 \tag{21}$$

$$y^k = h(x^k,\theta) + v^k \tag{22}$$

$$S(x^k,\theta,u) \le 0,\; T(x^k(t_f),\theta) \le 0 \tag{23}$$

$$\text{given} \quad y^j(i),\; i = \{1,\cdots,N\}\; \forall\; j = \{1,\cdots,k-1\}, \text{ and } i = \{1,\cdots,l\} \text{ for } j = k. \tag{24}$$

where $x^k(t)$ is the state vector, $u^k(t)$ the input vector, $d^k(t)$ the process disturbance, $v^k(t)$ the measurement noise, and J^k the cost function for the k^{th} batch. Let $y = h(x,\theta)$, a p-dimensional vector, be the combination of states that can be measured, $y^j(i)$ the i^{th} measurement taken during the j^{th} batch, and N the number of measurements within a batch. The objective is to utilize the measurements from the previous $(k-1)$ batches and the measurements up to the current time, t_l, of the k^{th} batch in order to tackle the uncertainty in θ and d^k and determine the optimal input policy for the remaining time interval $[t_l, t_f]$ of the k^{th} batch. If only off-line measurements are available, $N = 1$ and $t_l = 0$.

The formulation (20)–(24) is general in the sense that it does not address the question of how the measurements will be used for adaptation of the decision variables. Depending on how the measurements are utilized and whether or not a model is used, various schemes are possible as will be described next.

At this juncture, two classifications are possible. One is based on the methodology, i.e., the role of the model in the input calculation: (i) model-based repeated optimization, and (ii) model-free implicit optimization (presented in Figure 1). The other classification depends on the type of measurement available: (i) on-line measurements → on-line optimization, and (ii) off-line measurements → run-to-run optimization. To use one classification or the other is a question of preference, and here the classification based on the methodology is opted for. Then, for each methodology, the case of on-line measurements will be discussed first, and a subsection will be devoted on how the methodology can be modified to suit run-to-run optimization.

4.2 Model-based Repeated Optimization

In the on-line version of model-based repeated optimization, the optimal inputs are calculated by repeatedly solving (20)–(24) with the advent of every new measurement [47]. Measurements are used to estimate the current states and parameters. With the information on the current states serving as initial conditions, a model of the system is used to predict the evolution of the system and update the inputs towards the optimum. The calculated inputs are implemented until the next measurements are taken. As the estimation and optimization tasks are typically repeated over time, this scheme is often referred to as *repeated optimization* [2, 47, 42, 18]. This procedure is similar to model predictive control in the way the measurements are incorporated. One of the main disadvantages of this approach is that, if the measurements are frequent, the solution can be aggressive and often leads to switching between bounds resulting in a *chattering* solution [43].

In order to calculate the optimal inputs on-line, it is necessary to have a good estimate of the states and parameters. Instead of reoptimizing at every measurement instant, the states and parameters can be first estimated, and re-optimization can be undertaken only when the estimates deviate significantly from their predicted values. A variety of stochastic methods such as Extended Kalman Filtering [27, 48] and prediction error identification [33] as well as deterministic methods such as nonlinear least squares [45, 47] have been proposed in the literature to estimate the system states and parameters.

Though the problems of state and parameter estimation can be dealt with in a unified framework, there is fundamental difference between the two problems. The state estimation problem, which is linked to the disturbances $d(t)$, requires the observability condition, i.e., one must be able to reconstruct the states from the output and its derivatives. This condition is verified in most applications. On the other hand, the parameter estimation problem (determination of θ) requires persistency of excitation, i.e., the inputs should be varied sufficiently to uncover the unknown parameters. This might be a problem as discussed next.

4.2.1 Type of Model

The model used to assist the optimization can either be fixed or refined using measurements, the advantages and disadvantages of which are discussed next.

- *Fixed model [35, 1]*: If the model is not adjusted, it needs to be fairly accurate. This, however, is against the philosophy of the approach that assumes the presence of (considerable) uncertainty. If the uncertainty is only in the form of disturbances and not in the model parameters, it might be sufficient to use a fixed model. On the other hand, if the model is not accurate enough, the methodology will have difficulty converging to the optimal solution. Note that, since the measurements are used to estimate the states only (and not the parameters), there is no need for persistent inputs.
- *Refined model [18, 47, 38]*: When model refinement is used, the need to start with an accurate model is alleviated, but it is necessary to excite appropriately the system for estimating the uncertain parameters. However, the optimal inputs may not provide sufficient excitation. On the other hand, if sufficiently exciting inputs are provided for parameter identification, the resulting solution may not be optimal. This leads to a conflict between the objectives

of parameter estimation and optimization. This conflict has been studied in the adaptive control literature under the label *dual control problem* [44, 57].

4.2.2 Model-based Run-to-run Optimization

The run-to-run optimization is employed when only off-line measurements are available. Off-line measurements include measurements taken at the end of the batch (batch-end measurements) and, possibly, off-line analysis of samples taken during the batch. Such measurements are most common in industrial practice [9].

Off-line measurements enable the set-up of a batch-to-batch or *run-to-run optimization* approach that accounts for parametric uncertainty by exploiting the fact that batch processes are typically repeated. Process knowledge obtained from previous batches is used to update the operating strategy of the current batch. The objective is then to get to the optimum over a few batches. With this approach, it is possible to account for parametric uncertainties and disturbances that are repetitive in every batch. So, it is less conservative than the robust optimization scheme described in Section 3. However, since off-line measurements cannot be used to improve the current batch but only subsequent ones, random disturbances within the batch are not accounted for.

Two model-based run-to-run optimization schemes are discussed below, which vary fundamentally on how the measurements are incorporated into the adaptation procedure.

- *Measurements for model refinement [19, 21, 22]:*
  This approach uses a model of the process and refines it using information gathered from previous batches. The measurements obtained from earlier batch runs are used to identify the uncertain parameters. With the new set of parameters, the optimization problem is solved at the beginning of each batch run. The optimal input profiles are implemented in open-loop fashion as in the case of model predictive control.

- *Measurements instead of simulation [59, 16, 12, 32]:*
  Any numerical optimization procedure involves: (i) the choice of initial inputs, (ii) the calculation of the system states, the performance index J, and the constraints S and T, and (iii) the adaptation of the inputs towards the optimum (using for example gradient information). The steps (ii) and (iii) are then repeated until convergence. The particularity of run-to-run schemes is that the second step, which is usually calculated using a model, can be replaced by measurements, i.e., J, S, and T can be measured from an experimental run. However, when a gradient-based algorithm is used, a model is in general required to compute the gradient as in (6). So an approach can be devised where the states are measured from an experimental batch run, while the adjoints are obtained from a process model. The model parameters θ can be either fixed or adapted from run to run, while the disturbances $d(t)$ are ignored in the computation of the adjoints.

4.3 Model-free Implicit Optimization

One of the well-cited disadvantages of model-based schemes is the high computational burden involved in the on-line calculation of the optimal policy, especially for large-dimensional systems [15]. Furthermore, model-based implementation can be quite aggressive in the presence of frequent noisy measurements. Also, the disadvantages associated with a fixed model (possible poor prediction) and refined model (lack of persistency of excitation) are quite important.

The reason for these difficulties can be understood from the following analogy. In feedback control, the difference between the reference and measured signals is typically used to attenuate the effect of uncertainty [37]. Adaptive control, which is often used to deal with demanding control tasks, uses classical feedback control as the main building block and additional features for automatic tuning of controller parameters [4, 31]. The model parameters are estimated using input-output data (model refinement) and the controller parameters are updated accordingly. Instead of a standard adaptive control scheme, imagine a scheme where: (i) there is no direct feedback, (ii) the model is refined periodically, and (iii) the inputs are computed from the refined model. Such a scheme would be, by all means, less efficient than standard adaptive control techniques and has not even been reported in the control literature. Yet, the model-based approaches presented in the previous subsection for optimization under uncertainty are analogous to an adaptive control scheme without direct feedback and, thus, suffer from the aforementioned drawbacks.

Thus, an optimization approach where measurements are used *directly* to update the inputs towards the optimum, i.e., without using a model and explicit numerical optimization, would be preferable. Various possible options are discussed below.

4.3.1 Classification of Implicit Optimization Schemes

The classification here is based on whether measurements are used for interpolation between precomputed optimal values or simply compared to a reference.

- *Interpolation [30, 49]*: The optimal solution is worked out for different combinations of parameters and states as is done for the solution of the Hamilton-Jacobi-Bellman equation (dynamic programming) [29]. Note that an additional complication arises since the solution has to be discretized along the parameter values as well. The solution is then stored in some form or the other, e.g. using a neural network or a look-up table. From past data and current measurements, the point in the space of parameters and states that matches closely with the behavior of the system is obtained, and the corresponding solution is implemented.

  This is similar to the model-based approach, the main difference being that the estimation problem (interpolation) and the optimization problem (reading from the look-up table) are here implicit. The main drawback of this approach is the curse of dimensionality since it either requires a computationally expensive look-up table or a closed-form feedback law that is analytically expensive or impossible to obtain in many cases.
- *Reference tracking [23, 55]*: If the nominal model is available, it can be used to compute a numerical optimal solution. Numerical optimization provides information not only on the optimal evolution of the inputs $u^*(t)$, but also on the states $x^*(t)$. In the absence of distur-

bances, open-loop application of the inputs $u^*(t)$ or tracking of *any* of the states $x^*(t)$ will result in optimality.

However, in the presence of uncertainty, tracking some combinations of states can be better than open-loop application of the inputs. Also, implementation of tracking feedback controllers is numerically inexpensive and their tuning is quite simple.

The main bottleneck in incorporating reference tracking within the framework of optimization is the choice of the reference signals. In control problems, the references are typically user specified, whereas here the user provides the information *via* the formulation of the optimization problem. If the reference is not properly chosen, its tracking would be either infeasible or too conservative. Thus, it is important to track signals that are invariant under uncertainty.

Though there are many studies in the literature that use reference tracking, there is no systematic study on how to chose the references so as to guarantee optimality in the presence of uncertainty. As will be explained in the next section, this paper uses the concept of invariants to choose references, the tracking of which implies optimality.

4.3.2 Model-free Run-to-run Optimization

In model-free run-to-run optimization, the idea of using the measurements instead of model prediction is taken to the extreme. An experimental run is used to compute the cost J and the constraints S and T [14]. In addition, if a gradient-based optimization algorithm is used to update the decision variables, the gradient is also computed experimentally (as opposed to the model-based gradient discussed in Section 4.2.2). Every decision variable is perturbed and an experimental run is performed for each perturbation in order to obtain the gradient. Hence, for each optimization iteration, as many batch runs as there are decision variables in the input parameterization are necessary.

Such a procedure is termed evolutionary optimization in the literature [10]. Though the scheme presents the advantage of being model-free, it is experimentally expensive. In addition, the optimization algorithms that do not use gradient information converge slowly, thereby requiring even more process runs. Yet, evolutionary optimization works well when it is known *a priori* that the optimal solution is determined by path or terminal constraints. In such a case, the objective of the optimization algorithm is to take the system to the constraints. Since the direction of update is known, neither additional runs nor a process model are necessary to obtain the gradient [34, 52].

5 Invariant-based Optimization

In this section, an optimization scheme will be proposed where tracking feedback controllers form the basic elements. The reference trajectories are chosen such that they are invariant under uncertainty, i.e., they remain optimal despite uncertainty. If the structure of the optimal inputs is known, measurements can be used to directly update a few input parameters. Thus, the optimal inputs are determined directly from process measurements and not from a (possibly inaccurate) model. Also, the fact that batches are typically repeated over time can be used advantageously, thereby providing the possibility of on-line and/or batch-to-batch implementation.

5.1 Signals Invariant under Uncertainty

The core problem in such a methodology is the choice of reference signals that are invariant under uncertainty. Here, such a choice is based on the necessary conditions of optimality and constitutes the novelty in the proposed methodology. Thus, the optimization problem is translated into the satisfaction of selected constraints and the regulation of certain sensitivities around zero. Instead of treating $u(t)$ as a whole, the discontinuous optimal inputs are characterized in terms of specific arcs and switching times. This characterization is discussed in detail in [51] and is summarized below.

5.1.1 Characterization of the Optimal Solution

The dynamic optimization problem considered in (1)-(3) has two types of constraints: i) the path constraints impose bounds on the inputs and the states *during the batch*, and ii) the terminal constraints limit the outcome of the batch *at final time*. This gives rise to two corresponding types of objectives: (i) the instantaneous objectives, and (ii) the terminal objectives. In studies available in the literature, the importance of terminal objectives is almost always ignored. In contrast, the proposed method treats instantaneous and terminal objectives on equal footing.

For the problem (1)-(3), the optimal solution can be shown to have the following properties [9]:

1. The inputs are in general discontinuous. The time at which an input switches from one interval to another is called a *switching time*.
2. For each input, two types of arcs (constraint-seeking and compromise-seeking) are possible between switching instants. In a constraint-seeking arc, the input is determined by a path constraint, while in the other type of interval the input lies in the interior of the feasible region.
3. The switching instants can also be constraint-seeking or compromise-seeking depending on whether they are determined by terminal constraints or not.

A characterization of the optimal solution [51] can be performed that: i) treats the instantaneous and terminal objectives independently, and ii) separates the constraint-seeking variables from the compromise-seeking variables. Towards this end, the optimal inputs can be dissected into: i) the various arcs, $\eta(t)$, which cater to instantaneous objectives, and ii) the minimal parameterization π that is necessary to handle terminal objectives. Also, the constraint-seeking variables $\bar{\eta}(t)$ and $\bar{\pi}$ are separated from the compromise-seeking variables $\tilde{\eta}(t)$ and $\tilde{\pi}$.

In the presence of uncertainty, certain directions in $\eta(t)$ and π might change considerably while others do not change. For example, consider an optimal input that switches between u_{max} and u_{min} and an uncertainty that does not affect the sequence of arcs but only the switching time. Then, the inputs in the various arcs $\eta(t)$ do not change with uncertainty, while the switching time π does. So, two transformations $\eta(t)^T \rightarrow [\eta_f(t)^T \ \eta_v(t)^T]$, $\pi^T \rightarrow [\pi_f^T \ \pi_v^T]$ can be found where $\eta_f(t)$ and π_f represents the quantities (directions) that are fixed under uncertainty, while $\eta_v(t)$ and π_v are the

quantities that change with uncertainty. Only $\eta_v(t)$ and π_v need to be adapted using measurements. For the ease of notation, let $\eta_v(t) = \eta(t)$ and $\pi_v = \pi$.

In any case, what remains invariant under uncertainty is the fact that the necessary conditions of optimality have to be verified. The invariants take on different expressions for constraint-seeking and compromise-seeking arcs. In the following, it is assumed that the uncertainty is such that it does not affect the type and sequence of arcs (active path constraints) nor the set of active terminal constraints.

5.1.2 Invariants for Constraint-seeking and Compromise-seeking Input Directions

A set of signals $I^\eta(t) = h^\eta(x(t), u(t), t)$, referred to as *invariants* for the input directions, can be chosen such that optimality is achieved by tracking $I^\eta_{ref} = 0$. Note the dependence of h^η with respect to t, which indicates that h^η can be different in different intervals of the optimal solution.

Let $\bar{S}(x, u)$ denote the active constraints and $\bar{\mu}$ the corresponding Lagrange multipliers. Then, the necessary conditions of optimality are, $\bar{S} = 0$ and $\frac{\partial H}{\partial \eta} = \lambda^T \frac{\partial F}{\partial \eta} + \bar{\mu}^T \frac{\partial \bar{S}}{\partial \eta} = 0$. By definition, the constraint-seeking input directions $\bar{\eta}(t)$ push the system towards the constraints, while the compromise-seeking input directions $\tilde{\eta}(t)$ do not affect the active constraints, i.e., $\frac{\partial \bar{S}}{\partial \tilde{\eta}} = 0$. Thus, the necessary conditions of optimality read, $\bar{S} = 0$, $\frac{\partial H}{\partial \tilde{\eta}} = \lambda^T \frac{\partial F}{\partial \tilde{\eta}} = 0$, and $\frac{\partial H}{\partial \bar{\eta}} = \lambda^T \frac{\partial F}{\partial \bar{\eta}} + \bar{\mu}^T \frac{\partial \bar{S}}{\partial \bar{\eta}} = 0$. The first two equations are independent of $\bar{\mu}$ and the third one determines the value of $\bar{\mu}$. Thus, the advantage of separating the constraint-seeking and compromise-seeking input directions is that, by guaranteeing $\bar{S} = 0$ and $\lambda^T \frac{\partial F}{\partial \tilde{\eta}} = 0$, optimality can be achieved without the knowledge of the Lagrange multiplier $\bar{\mu}$.

The m input variables can be used to satisfy the m conditions $\bar{S} = 0$ and $\lambda^T \frac{\partial F}{\partial \tilde{\eta}} = 0$. Since the input directions $\tilde{\eta}(t)$ have no influence on $\bar{S}$, it is logical to use the constraint-seeking directions $\bar{\eta}$ to meet $\bar{S} = 0$ and the compromise-seeking directions $\tilde{\eta}$ to satisfy $\lambda^T \frac{\partial F}{\partial \tilde{\eta}} = 0$. So, the invariants along the constraint-seeking input directions are the active constraints themselves, $h^{\bar{\eta}}(x, u, t) = \bar{S}(x, u)$ and, for compromise-seeking input directions, the invariants are $h^{\tilde{\eta}}(x, u, t) = \lambda^T \frac{\partial F(x,u)}{\partial \tilde{\eta}}$. Note that the element that remains invariant despite uncertainty is the fact that optimal operation corresponds to $I^\eta_{ref} = 0$. However, the uncertainty does have an influence on the value of $I^\eta(t)$, and the inputs need to be adapted in order to guarantee $I^\eta_{ref} = 0$.

5.1.3 Invariants for Constraint-seeking and Compromise-seeking Parameters

In addition to the choice of invariants for the various arcs, it is important to choose the invariants for the parameters π. Following similar arguments, a set of signals $I^\pi = h^\pi(x(t_f))$ can be constructed such that the optimum corresponds to $I^\pi_{ref} = 0$, also in the presence of uncertainty. Clearly, the invariants arise from the conditions of optimality of the parameterized problem, i.e., $\bar{T}(x(t_f)) = 0$ and $\frac{\partial \phi}{\partial \pi} + \bar{\nu}^T \frac{\partial \bar{T}}{\partial \pi} = 0$, where $\bar{T}$ corresponds to the set of active terminal constraints. The reader is referred to [51] for more details.

Following the same arguments as in the case of input directions, and noting that the compromise-

seeking parameters are those with $\frac{\partial \bar{T}}{\partial \bar{\pi}} = 0$, the necessary conditions of optimality read: $\bar{T}(x(t_f)) = 0$, $\frac{\partial \phi}{\partial \bar{\pi}} = 0$, and $\frac{\partial \phi}{\partial \bar{\pi}} + \bar{\nu}^T \frac{\partial \bar{T}}{\partial \bar{\pi}} = 0$. So, for the constraint-seeking parameters, the invariants correspond to the active terminal constraints $h^{\bar{\pi}}(x(t_f)) = \bar{T}(x(t_f))$ and, for the compromise-seeking parameters, to sensitivities $h^{\tilde{\pi}}(x(t_f)) = \frac{\partial \phi(x(t_f))}{\partial \tilde{\pi}}$.

5.1.4 Choice of References

In summary, the invariants are determined as follows:

- For constraint-seeking arcs: $h^{\bar{\eta}}(x, u, t) = \bar{S}(x, u)$
- For compromise-seeking arcs: $h^{\tilde{\eta}}(x, u, t) = \lambda^T \frac{\partial F(x,u)}{\partial \tilde{\eta}}$
- For constraint-seeking parameters: $h^{\bar{\pi}}(x(t_f)) = \bar{T}(x(t_f))$
- For compromise-seeking parameters: $h^{\tilde{\pi}}(x(t_f)) = \frac{\partial \phi(x(t_f))}{\partial \tilde{\pi}}$

If the inputs are in the interior of the feasible region, then, to a first-order approximation, the deviation in cost is $\delta J = H_u \delta u$. Since by definition $H_u = 0$, any small deviation of u from the optimal trajectory will cause a negligibly small loss in cost. On the other hand, the Lagrange multipliers μ and ν capture the deviation in cost resulting from the path and terminal constraints not being active. So, to a first-order approximation, $\delta J = \int_0^{t_f} \mu^T \delta S \, dt + \nu^T \delta T$. Thus, in general, there is little to be gained/lost along the compromise-seeking variables, while it pays off to keep the path and terminal constraints active. In short, optimality despite uncertainty is approached by working close to the active constraints, i.e., where there is much to gain!

The core assumption in this approach is that the set of active constraints (both path and terminal) does not change in the presence of uncertainty. This means that the type and sequence of arcs that constitute the optimal solution, also in the presence of uncertainty, is known a priori. In fact, the type and sequence of arcs and the set of active terminal constraints is the only information required to set up the scheme.

5.2 Description of the Invariant-based Optimization Scheme

The proposed invariant-based optimization (IBO) scheme involves the following steps:

- *Determination of the structure of the optimal solution:* The structure of the optimal solution is described by the type and sequence of arcs and the set of active terminal constraints. These can be obtained in two ways: (i) educated guess by an experienced operator, or (ii) inspection of the solution obtained from numerical optimization using a simplified model. Quite often, experience dictates the qualitative shape of the inputs. Otherwise, a simplified (tendency) model of the process can be used to compute a numerical solution in which the various arcs are identified.
- *Choice of invariant references for adaptation under uncertainty:* In the presence of uncertainty, the numerical values of the inputs in the various arcs and the switching times might

change considerably. However, optimal operation corresponds to verifying the necessary conditions of optimality, also in the presence of uncertainty. Under the assumption that the type and sequence of arcs and the set of active terminal constraints remain unchanged, the necessary conditions of optimality translate into satisfying the active constraints $\bar{S}$ and $\bar{T}$ and regulating the sensitivities $\lambda^T \frac{\partial F}{\partial \bar{\eta}}$ and $\frac{\partial \phi}{\partial \bar{\pi}}$ around zero.

- *Tracking of invariants using measurements:* The structure given in Figure 2 is proposed to track the invariants by use of feedback. The invariants $I^{\eta}_{ref} = 0$ and $I^{\pi}_{ref} = 0$ are tracked with the help of path and terminal feedback controllers, respectively. The details concerning the design of the two controllers are discussed in the next subsection. The trajectory generator computes the current inputs $u(t)$ as a function of $\eta(t)$ and π that are generated by the path and terminal controllers. Note that the implementation is *model-free* and *measurement-based*.

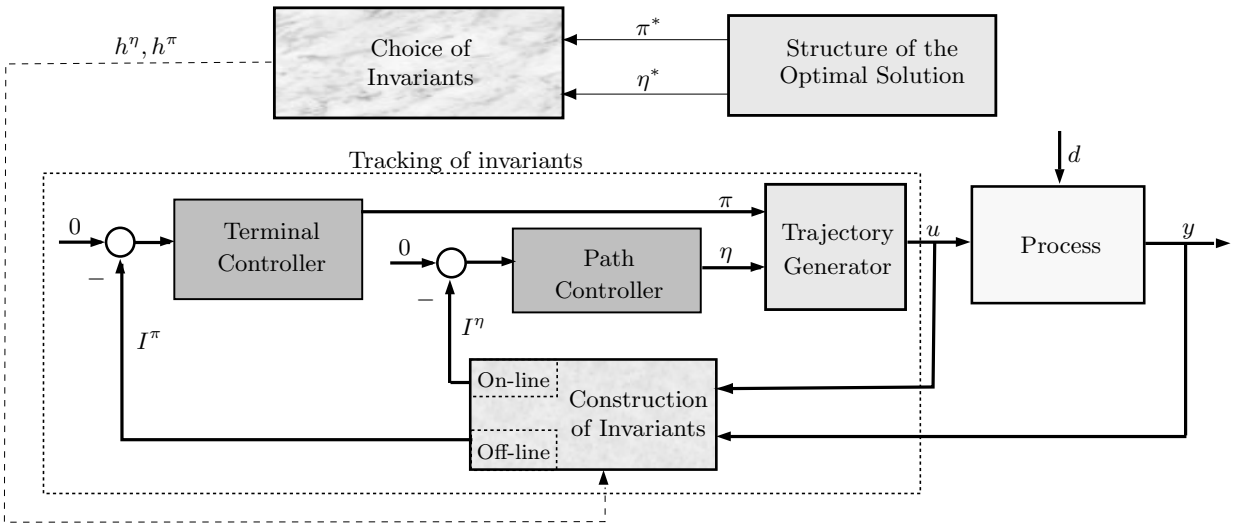

Figure 2: Invariant-based optimization

5.3 Design Aspects

The path controller is a PI-type controller that manipulates the various input arcs $\eta(t)$ to regulate $\bar{S}$ and $\lambda^T \frac{\partial F}{\partial \bar{\eta}}$ around zero. The terminal controller is also a PI-type controller that manipulates the input parameters π to regulate $\bar{T}$ and $\frac{\partial \phi}{\partial \bar{\pi}}$ around zero. Various design aspects are treated next.

- *Difference in time scale – on-line vs. off-line measurements:* In general, there is a difference in time scale between the path controller and the terminal controller. The path controller works within a batch using on-line measurements (running index is the batch time t) [7]. The terminal controller operates on a batch-to-batch basis using off-line measurements (running index is the batch number k) [52].

If on-line measurements are not available, the path controller is inactive. If off-line measurements of the path constraints are available, it is possible to use the path controller in a batch-to-batch mode so that the system will be closer to the path constraints during the next batch [36]. On the other hand, if it is possible to predict I^{π} from on-line measurements, it might be possible to use the terminal controller within the batch [58].

- *Centralized vs. decentralized controllers:* In general, the path controller is a multi-input multi-output controller. However, since $\tilde{\eta}$ does not affect $\bar{S}$, a first level input-output pairing can be performed, i.e, $\bar{\eta} \rightarrow \bar{S}$ and $\tilde{\eta} \rightarrow \lambda^T \frac{\partial F}{\partial \tilde{\eta}}$. Still $\bar{\eta}$, $\tilde{\eta}$, $\bar{S}$, and $\lambda^T \frac{\partial F}{\partial \bar{\eta}}$ are vector quantities. A further sensitivity analysis can be performed that leads either to an input-output pairing or the design of a decoupling matrix. As a result, appropriate decentralized controllers can be designed. A similar procedure is undertaken for the terminal controller.
- *Construction of invariants from measurements:* If I^{η} and I^{π} are not measured directly, they need to be reconstructed from the available measurements. In the case of constraint-seeking arcs and parameters, the invariants correspond to physical quantities (path or terminal constraints). Off-line measurements of terminal quantities are in general available. In most cases, a path constraint involves a variable that can be measured, or the constraint can be rewritten in terms of a quantity that can be measured. For example, the path constraint may correspond to a bound on temperature or pressure; or a constraint on heat removal can be rewritten as a constraint on the cooling temperature. In such cases, on-line measurement of the path constraint is directly available. On the other hand, if the path constraint cannot be measured directly, some type of inference [28, 17] or state estimation is necessary.

  For compromise-seeking arcs and parameters, the invariants are sensitivities. For computation of sensitivities, multiple process runs are typically required. Two runs with two different values of the decision variables are performed and the difference is used to compute the sensitivities. With this approach, as many runs as there are decision variables are necessary to compute all the sensitivities required for adaptation. Since $\tilde{\eta}(t)$ is infinite dimensional, it needs to be approximated using a finite number of decision variables before the sensitivities can be computed experimentally, while the parameters $\tilde{\pi}$ can be perturbed directly. An alternative is to use a model of the process just for sensitivity calculation as in [59], but such an approach is typically more involved. However, as discussed in Subsection 5.1.4, the sensitivity with respect to input variations in compromise-seeking arcs and parameters can often be neglected. In such a case, all compromise-seeking arcs and parameters can be kept at predetermined values, and only the constraint-seeking arcs and parameters are adjusted.

  It is clear from the above discussion that IBO is easiest to implement when: i) path and terminal constraints can be directly measured, and ii) compromise-seeking arcs are absent or can be approximated by simpler profiles.

5.4 Practical Applicability of IBO

If the selected structure of the optimal solution (type and sequence of arcs, active terminal constraints) is correct, i.e., it corresponds to that of the true (unknown) system, IBO will be capable of optimizing the true system. Thus, the applicability of IBO in practice will depend on: (i) the robustness of the proposed input structure with respect to uncertainty (modeling errors and disturbances), and (ii) the ability to measure the path and terminal constraints. These issues are briefly discussed next.

- *Role of the model:* If the structure of the optimal inputs cannot be obtained from experience or educated guess, then a process model is needed to determine the structure of the optimal inputs numerically. Thus, the goal of a process model is only to provide the correct structure for the optimal inputs. So, in contrast to model-based optimization approaches or what is sought for simulation purposes, there is no need for a detailed model or for accurate parameter values. The model simply needs to reflect the major tradeoffs specific to the optimization problem at hand. The parts of the model that do not address these effects can be discarded.

- *Disturbance rejection.* The presence of disturbances influences both $\eta(t)$ and π. Disturbances affecting $\eta(t)$ within the batch are rejected by the path controller. However, the effect of any disturbance within the batch on π cannot be rejected since the terminal controller only works on a batch-to-batch basis. Constant disturbances (e.g. raw material variations) can be rejected from batch to batch by the terminal controller.

- *Backoff from constraints.* In the presence of disturbances and parametric uncertainty that cannot be compensated for by feedback, the use of conservative margins, called backoffs, is inevitable to ensure feasibility of the optimization problem [56]. The presence of measurement errors also necessitates a backoff. Based on an estimate of the uncertainty, the probability density function of the state variables can be calculated. The margins are then chosen such that the spread of the states remains within the feasible region with a certain confidence level. Note that the margins typically vary with time.

  Due to the sensitivity reduction that is characteristic of feedback control, the conservatism can be reduced considerably in the proposed framework in comparison with the standard open-loop optimization schemes. The feedback parameters can be chosen so as to minimize the spread in the state variables resulting from uncertainty. The use of feedback becomes particularly important when the uncertainty tends to increase during a batch run. With reduced backoffs, the process can be driven closer to active constraints, thereby leading to improved performance.

6 Illustrative Example

In this section, the various optimization strategies discussed above are illustrated and compared through the simulated operation of a fed-batch bioreactor in the presence of uncertainty. This example, which represents a penicillin fermentation process, has the structure of Example 2 of the companion paper [51] wherein the nominal solution has been characterized. The numerical values, however, are different here as they have been adapted from [5].

- *Reactions:* $S \xrightarrow{X} X, \quad S \xrightarrow{X} P.$

- *Conditions:* Fed-batch, isothermal.

- *Objective:* Maximize the concentration of product P at a given final time.

- *Manipulated variable:* Feed rate of S.

- *Constraints:* Input bounds; upper limit on the biomass concentration, which is motivated by oxygen limitation typically occurring at large biomass concentrations.

6.1 Nominal Optimization

Variables and parameters: S: Concentration of substrate, X: Concentration of biomass, P: Concentration of product, V: Volume, u: Feed flowrate, S_{in}: Inlet substrate concentration, μ_m, K_m, K_i, ν: Kinetic parameters, and Y_x, Y_p: Yield coefficients.

Model equations:

$$\dot{X} = \mu(S)\, X - \frac{u}{V}\, X \qquad X(0) = X_o \tag{25}$$

$$\dot{S} = -\frac{\mu(S)\, X}{Y_x} - \frac{\nu\, X}{Y_p} + \frac{u}{V}\,(S_{in} - S) \qquad S(0) = S_o \tag{26}$$

$$\dot{P} = \nu\, X - \frac{u}{V}\, P \qquad P(0) = P_o \tag{27}$$

$$\dot{V} = u \qquad V(0) = V_o \tag{28}$$

with $\mu(S) = \frac{\mu_m\, S}{K_m + S + \frac{S^2}{K_i}}$ and the numerical values given in Table 1.

μ_m	0.02	$\frac{\text{l}}{\text{h}}$
K_m	0.05	$\frac{\text{g}}{\text{l}}$
K_i	5	$\frac{\text{g}}{\text{l}}$
Y_x	0.5	$\frac{\text{g[X]}}{\text{g[S]}}$
Y_p	1.2	$\frac{\text{g[P]}}{\text{g[S]}}$
ν	0.004	$\frac{\text{l}}{\text{h}}$
S_{in}	200	$\frac{\text{g}}{\text{l}}$

u_{min}	0	$\frac{\text{l}}{\text{h}}$
u_{max}	1	$\frac{\text{l}}{\text{h}}$
X_{max}	3.7	$\frac{\text{g}}{\text{l}}$
t_f	150	h

X_o	1	$\frac{\text{g}}{\text{l}}$
S_o	0.5	$\frac{\text{g}}{\text{l}}$
P_o	0	$\frac{\text{g}}{\text{l}}$
V_o	150	l

Table 1: Model parameters, operating bounds and initial conditions

Optimization problem:

$$\begin{aligned} \max_{u(t)} \quad & J = P(t_f) \\ s.t. \quad & (25) - (28) \\ & X(t) \le X_{max} \\ & u_{min} \le u \le u_{max} \end{aligned} \tag{29}$$

Types of arcs [51]: The compromise-seeking solution corresponds to being on the surface $S = S^* = \sqrt{K_i\ K_m}$, which can be differentiated once to obtain the input:

$$u_{comp} = \frac{V}{S_{in} - S} \left(\frac{1}{Y_x}\, \mu(S)\, X + \frac{1}{Y_p}\, \nu\, X \right) \Bigg|_{S=S^*} \tag{30}$$

The path constraint corresponds to $X = X_{max}$. The corresponding input can be obtained by differentiating the path constraint once:

$$u_{path} = \mu(S)\, V|_{X=X_{max}} \tag{31}$$

When $u = u_{path}$ is applied at $X = X_{max}$, the substrate dynamics become:

$$\dot{S} = -\frac{1}{Y_x}\ \mu(S)\ X_{max} - \frac{1}{Y_p}\ \nu\ X_{max}\ +\ \mu(S)\ (S_{in} - S) \tag{32}$$

Let S_e correspond to the equilibrium point of the internal dynamics (32). For the input to remain bounded, due to the instability of the internal dynamics, the biomass constraint has to be entered with $S = S_e = 0.0037\frac{\text{g}}{\text{l}}$.

Sequence of arcs [51]:

- The initial condition for S is chosen so that $S_o = S^* = 0.5\frac{\text{g}}{\text{l}}$. Due to this specific choice of initial condition, u_{comp} can be applied right from the start in order to increase X as quickly as possible (Figure 3).
- The input is then lowered to $u = u_{min}$ in order to reach $S = S_e$. The switching time between the second and the third interval should be chosen so that the conditions $X = X_{max}$ and $S = S_e$ occur at the same time instant.
- When $X = X_{max}$, the input is set to $u = u_{path}$.

The nominal optimal trajectories are given in Figure 3. The concentration of penicillin at the final time is maximized to 1.68 $\frac{\text{g}}{\text{l}}$.

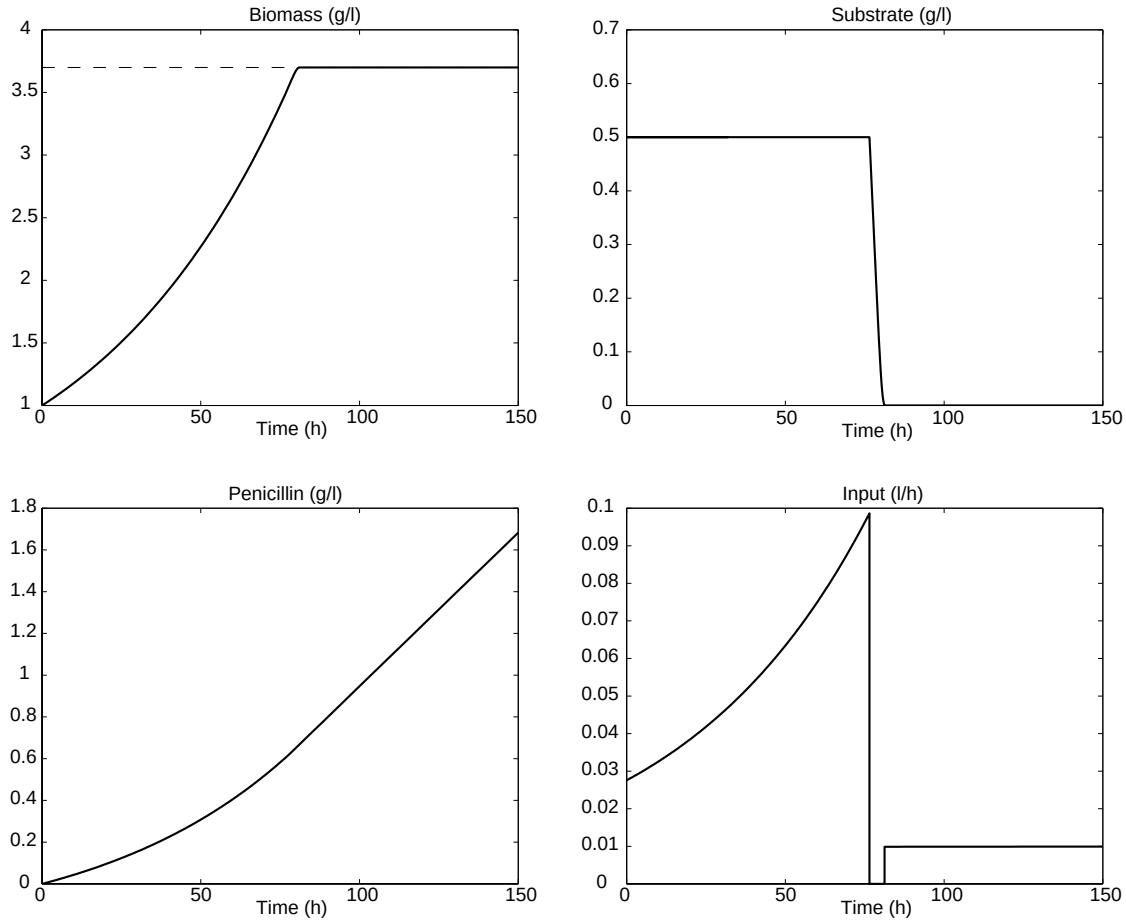

Figure 3: Nominal optimal trajectories for Problem (29)

6.2 Robust Optimization

Two types of uncertainty are considered:

- The parameter Y_x is uncertain in the range $0.3 - 0.5$.
- The substrate inlet concentration is normally distributed with mean $200\frac{\text{g}}{\text{l}}$ and standard deviation $25\frac{\text{g}}{\text{l}}$. S_{in} is assumed to change every hour.

The worst-case of Y_x, i.e., $Y_x^{max} = 0.5$ and the average value of S_{in}, i.e., $S_{in}^{av} = 200\frac{\text{g}}{\text{l}}$ are used to compute the conservative solution. The robust optimization problem is given below for a confidence level of 95%:

$$\begin{aligned} \max_{u(t)} \quad & J = \bar{P}(t_f) \\ s.t. \quad & \dot{\bar{x}} = F(\bar{x}, \bar{\theta}, u), \qquad \bar{x}(0) = x_o \\ & \bar{X}(t) - X_{max} + b_X \leq 0 \\ & u_{min} \leq u(t) \leq u_{max} \end{aligned} \tag{33}$$

where F represents the model equations (25)–(28). The backoff b_X is calculated such that $\bar{X}(t) - X_{max} + b_X \leq 0 \Rightarrow P[X(t) \leq X_{max}] \geq 0.95$. Assuming that the states are normally distributed, the propagation method, (12)–(13), is used to compute the probability density function of $x(t)$. The iterative algorithm described in Section 3.4 converges to $b_X = 0.19$.

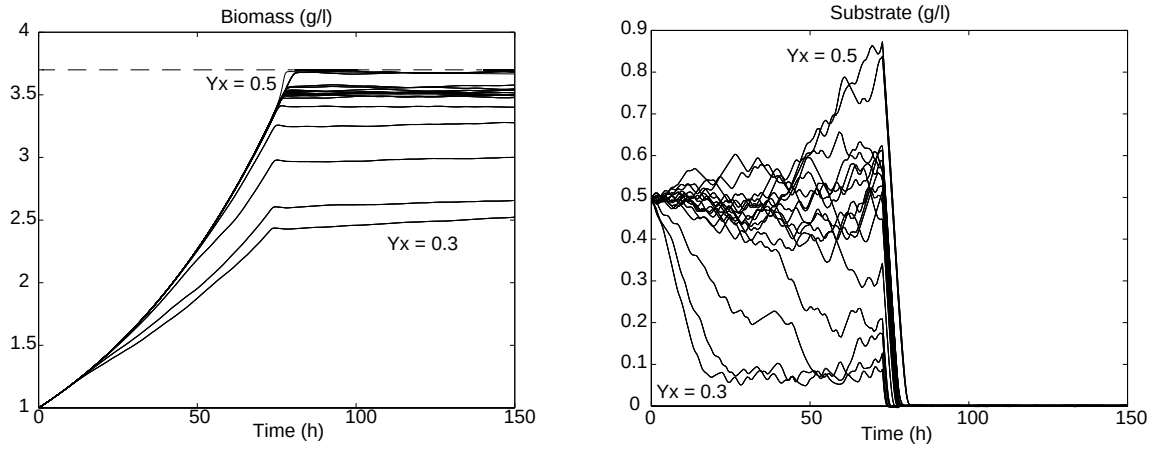

Figure 4: Open-loop implementation of robust input: Biomass and substrate profiles for different Y_x values and S_{in} realizations

The conservative solution is applied open-loop for various Y_x values and S_{in} realizations, the results being illustrated in Figure 4. Note that the conservative input is computed once with the nominal model and a necessary backoff. Due to the introduction of backoff, the feasibility of the biomass constraint is ensured (with a confidence level of 95%). Thus, the conservative input can be applied safely to the real process in an open-loop fashion. However, important oscillations of the substrate level result from the uncertainty, and even wash-out of the substrate may result for low Y_x values. The resulting loss in performance is important and will be analyzed later.

6.3 Measurement-based Optimization

Two different scenarios in the measurement-based optimization are considered: i) model-based repeated optimization, and ii) model-free implicit optimization.

6.3.1 Model-based Repeated Optimization

The strategy based on repeating optimization at periodic intervals is considered. A direct optimization approach is used with a piecewise constant input parameterization using 40 elements (CVP) [51]. The optimization routine is time-consuming and frequently converges to a local minimum; thus, it has to be restarted and manually guided to the optimal solution.

Full-state measurement is assumed and the current state acts as the initial condition for the dynamic system in the optimization problem. Also, under the assumption that most of the substrate is indeed consumed to produce biomass in the first part of the batch, the uncertain parameter Y_x is estimated from on-line measurements of biomass concentration and substrate consumption as: $Y_x = \frac{XV - X_0V_0}{S_{in}(V-V_0)+S_0V_0-SV}$ (stems from (25)–(28) with $\nu\, X = 0$).

6.3.2 Model-free Implicit Optimization

The invariant-based optimization scheme illustrates the model-free approach. It is implemented in the following manner.

- Along the first interval, a PI-controller is used to keep the system on the surface $S = S^* = 0.5\frac{\text{g}}{\text{l}}$, which corresponds to maintaining maximum growth rate of the biomass.
- In the second interval, the input is set to its lower bound $u_{min} = 0$.
- In the third interval, a biomass controller is used to track the biomass constraint X_{max}. The cascade structure (Figure 5) is necessary to stabilize the substrate internal dynamics and cope with the large time-scale difference between the biomass and substrate dynamics.

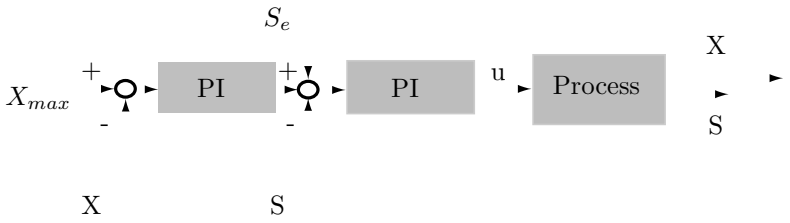

Figure 5: Biomass controller: $S_e = 0.0037\frac{\text{g}}{\text{l}}$ is the equilibrium value of the internal dynamics

Although the determination of the switching instant between the first and the second arcs is rather involved, a simple heuristic approach is taken here: The input is switched when $X = 0.96X_{max}$.

The input is then kept at u_{min} until the biomass constraint is reached ($X = X_{max}$). Any error in the choice of the first switching instant will be reflected as difference between S_e and S when $X = X_{max}$, but this error will be compensated by the controller structure presented in Figure 5.

Simulation results are presented in Figure 6. Note that the variations in the biomass and substrate profiles and thus the backoff from the biomass constraint are much smaller compared to the robust optimization case since feedback is used to cope with the uncertainty. However, this scheme necessitates the knowledge of biomass and substrate concentrations, which need to be either measured or estimated with appropriate observers [6].

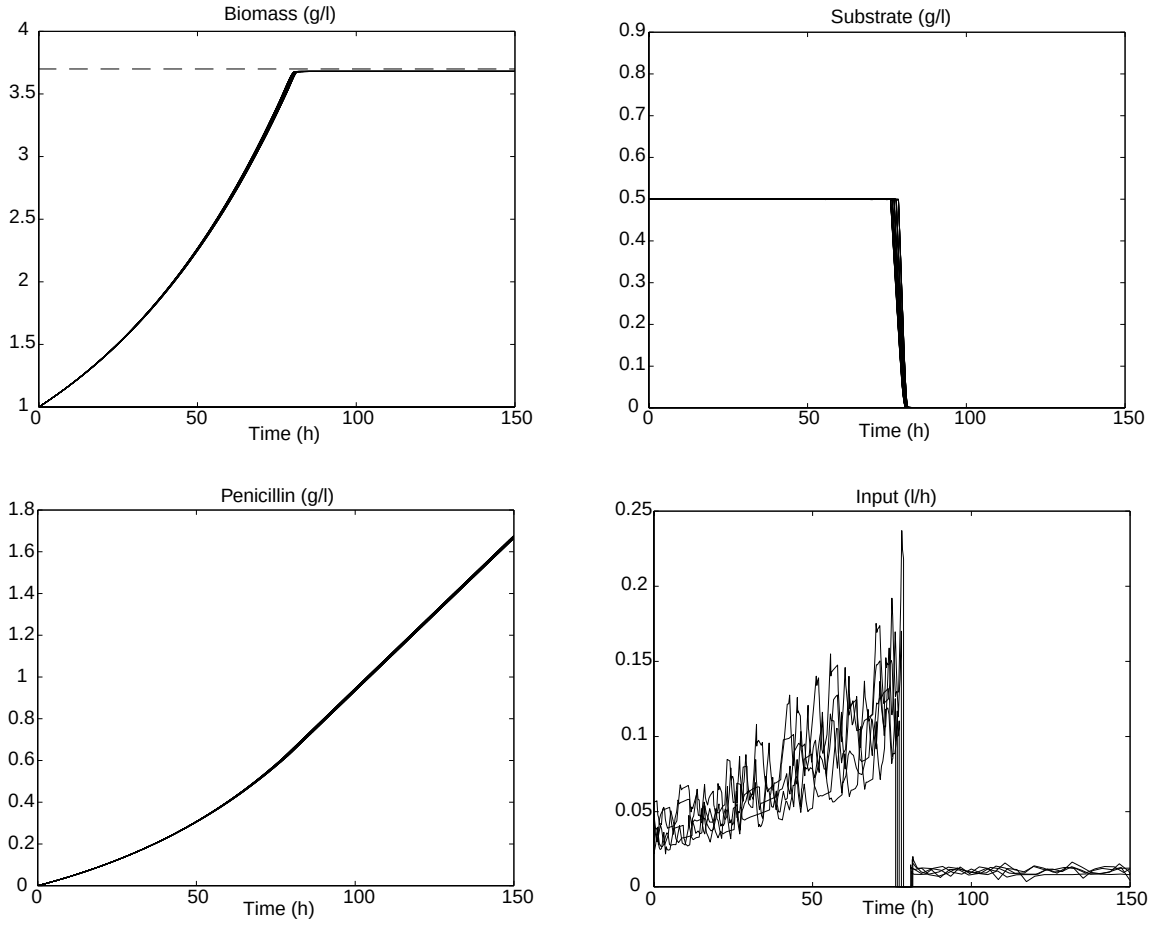

Figure 6: Invariant-based optimization for different Y_x values and S_{in} realizations

6.4 Performance Comparison

Table 2 illustrates the average performance of the various optimization strategies for different values of the uncertain parameter Y_x. For each optimization strategy and each value of Y_x, 50 different realizations of S_{in} are considered and the average cost is computed. The robust optimal trajectory is computed with the model parameters given in Table 1.

Though the parameter Y_x changes, the nominal cost calculated using the correct Y_x value remains

Y_x	Nominal Optimization	Robust Optimization	Repeated Opt. with n_o				IBO
			2	4	16	150	
0.3	1.67	1.26	1.50	1.57	1.64	1.66	1.66
0.35	1.67	1.36	1.55	1.60	1.65	1.67	1.67
0.4	1.68	1.46	1.60	1.63	1.65	1.67	1.67
0.45	1.68	1.55	1.62	1.64	1.65	1.68	1.68
0.5	1.68	1.63	1.63	1.64	1.66	1.68	1.68

Table 2: Performance comparison (product concentration in $\frac{g}{l}$) for various values of the uncertain parameter Y_x: Nominal Optimization $\rightarrow$ Solution where Y_x is known and no disturbances; Robust Optimization $\rightarrow$ Solution with $Y_x = 0.5$ and backoff to account for variations in S_{in} ($\sigma = 25\frac{g}{l}$); n_o = number of on-line optimizations; IBO $\rightarrow$ Invariant-based optimization.

fairly constant. This can be explained as follows: From (25)–(28), it is seen that the variation in Y_x only affects the dynamics of S which in turn can be compensated by adapting $u(t)$. So, after compensation, all the state evolutions, and also the objective function, are fairly close to those in the case $Y_x = 0.5$. This is clearly illustrated by a comparison of Figures 4 and 6: Figure 6 shows that variations in the input $u(t)$ are capable of compensating the uncertainty in Y_x and S_{in}.

The loss in performance observed in Table 2 compared to the nominal situation with Y_x known is due to two reasons:

1. *The introduction of backoff in order to remain feasible amidst perturbation in S_{in}*: The effect of backoff on the average performance can be observed from the last row in Table 2. Each optimization strategy deals with the uncertainty differently and thus the backoff is different. The backoff is largest for the open-loop application of the conservative profile (robust optimization) and is reduced progressively with increasing reoptimization frequency in the repeated optimization scheme. The smallest conservatism is obtained with the invariant-based optimization scheme. Thus, the average performance increases from the robust optimization over the repeated optimization to the invariant-based optimization scheme.
2. *Too low substrate concentrations resulting from an actually lower value of Y_x:* The effect of non-optimal S values ($S \neq S^*$) at the beginning of the batch is strongest in the case $Y_x = 0.3$, as can be seen in Figure 4. The biomass grows slower than expected by the model and, at the end of the first arc, there is a large offset from the biomass constraint. This causes a severe loss in performance of 24.5 % in the case of robust optimization. This loss in performance can be reduced with repeated optimization and, especially, invariant-based optimization. The improvement is accomplished by increasing the feed rate of substrate in the first part of the batch to bring the substrate concentration back to its optimal value $S = S^* = 0.5\frac{g}{l}$.

With the repeated optimization scheme, it is observed that a significant performance improvement is already obtained with a few reoptimizations. However, a total number of 150 optimizations (a reoptimization every hour) would be necessary to achieve the same performance as with the invariant-based optimization scheme. This performance is only slightly lower than the nominal performance without uncertainty. Hence, the invariant-based optimization scheme is able to achieve near-optimal performance by compensating the effect of unknown parameters and disturbances at the cost of being able to measure or estimate a few states.

7 Conclusions

Most techniques proposed in the literature for the optimization of dynamic processes are *model-based*, whilst accurate models of industrial processes are rarely available. Owing to uncertainty (model mismatch, parametric uncertainty, and disturbances), the open-loop implementation of an off-line computed optimal solution leads to either infeasible or highly sub-optimal operation. Hence, handling uncertainty becomes an important issue, especially in the presence of constraints related to quality and safety.

On the other hand, frequent process measurements, which have been made possible by recent developments in sensor technology, are now available in many industrial setups. Hence, the goal of the work presented here has been to analyze *measurement-based* optimization strategies to cope with uncertainty and make optimization more applicable industrially. The characterization step proposed in the companion paper is crucial to the ideas presented in this paper.

It is also possible perceive the proposed feedback-based optimization strategy from an industrial perspective. Classical PID control is the most popular technique used currently in industry, and trading it to attain optimality is unacceptable industrially. Hence, in contrast to most model-based optimization studies, this work attempts to use feedback control for the sake of optimality. In this sense, the approach has a greater industrial potential.

Acknowledgments

The authors would like to thank Dr. P. Terwiesch, ABB Corporate Research, Baden, Switzerland for his input on robust optimization and the National Science Foundation, Switzerland, grant number 21-46922.96, for supporting this research work.

References

[1] O. Abel, A. Helbig, W. Marquardt, H. Zwick, and T. Daszkowski. Productivity optimization of an industrial semi-batch polymerization reactor under safety constraints. *J. Process Contr.*, 10(4):351–362, 2000.

[2] O. Abel and W. Marquardt. A model predictive control scheme for safe and optimal operation of exothermic semi-batch reactors. In *IFAC DYCOPS-5*, pages 761–766, Corfu, Greece, 1998.

[3] M. Alamir and I. Balloul. Robust control of general constrainted batch processes. In *European Control Conference*, number F805, Karlsruhe, Germany, 1999.

[4] K. J. Aström and B. Wittenmark. *Adaptive Control*. Addison-Wesley, Reading, Massachusetts, 2nd edition, 1995.

[5] R. K. Bajpai and M. Reuss. Evaluation of feeding strategies in carbon-regulated secondary metabolite production through mathematical modelling. *Biotechnol. Bioengng.*, 13:717–738, 1981.

[6] G. Bastin and D. Dochain. *On-line Estimation and Adaptive Control of Bioreactors*. Elsevier, Amsterdam, 1990.

[7] C. Benthack. *Feedback-Based Optimization of a Class of Constrained Nonlinear Systems: Application to a Biofilter*. PhD thesis 1717, Swiss Federal Institute of Technology (EPFL), Lausanne, Switzerland, 1997.

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

[9] D. Bonvin, B. Srinivasan, and D. Ruppen. Dynamic optimization in the batch chemical industry. In *Chemical Process Control - 6*, Tucson, AZ, 2001.

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

[11] A. E. Bryson. *Dynamic Optimization*. Addison-Wesley, Menlo Park, California, 1999.

[12] I. S. Chin, K. S. Lee, and J. H. Lee. A unified framework for combined quality and tracking control of batch processes. In *14th IFAC World Congress*, volume N, pages 349–354, Beijing, P.R. China, 1999.

[13] W. B. Z. Chu and A. Constantinides. Modeling, optimization, and computer control of cephalosporin C fermentation process. *Biotechnol. Bioengng.*, 32:277–288, 1988.

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

[15] U. M. Diwekar. *Batch Distillation: Simulation, Optimal Design and Control*. Taylor and Francis, Washington, 1995.

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

[17] F. J. Doyle. Nonlinear inferential control for process applications. *J. Process Contr.*, 8(5–6):339–353, 1998.

[18] J. W. Eaton and J. B. Rawlings. Feedback control of nonlinear processes using on-line optimization techniques. *Comp. Chem. Eng.*, 14:469–479, 1990.

[19] C. Filippi, J. L. Greffe, J. Bordet, J. Villermaux, J. L. Barnay, B. Ponte, and C. Georgakis. Tendency modeling of semi-batch reactors for optimization and control. *Comp. Chem. Eng.*, 41:913–920, 1986.

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

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

[22] V. Garcia, M. Cabassud, M. V. Le Lann, L. Rigal, and G. Casamatta. A global strategy for the optimization of batch reactors: Application to lactic acid synthesis by the alkaline degradation of fructose. In *IMACS/IFAC Symposium on Mathematical Modelling and Simulation in Agricultural and Bio-Industries*, pages 63–68, Budapest, 1997.

[23] C. Gentric, F. Pla, M. A. Latifi, and J. P. Corriou. Optimization and nonlinear control of a batch emulsion polymerization. *Chem. Eng. Journal*, 75(1):31–46, 1999.

[24] I. E. Grossmann and R. W. H. Sargent. Optimum design of chemical plants with uncertain parameters. *AIChE J.*, 24:1021–1028, 1978.

[25] R. Gygax. Chemical reaction engineering for safety. *Comp. Chem. Eng.*, 43(8):1759–1771, 1988.

[26] K. P. Halemane and I. E. Grossmann. Optimal process design under uncertainty. *AIChE J.*, 29:425–433, 1983.

[27] A. H. Jazwinski. *Stochastic Processes and Filtering Theory*. Academic Press, New York, 1970.

[28] B. Joseph and C. Brosilow. Inferential control of processes. III. Construction of optimal and suboptimal dynamic estimators. *AIChE J.*, 24(3):500–508, 1978.

[29] D. E. Kirk. *Optimal Control Theory : An Introduction*. Prentice-Hall, London, 1970.

[30] M. Krothapally, B. Bennett, W. Finney, and S. Palanki. Experimental implementation of an on-line optimization scheme to batch PMMA synthesis. *ISA Trans.*, 38:185–198, 1999.

[31] I. D. Landau, R. Lozano, and M. M'Saad. *Adaptive Control*. Springer-Verlag, London, 1998.

[32] K. S. Lee, I. S. Chin, H. J. Lee, and J. H. Lee. Model predictive control technique combined with iterative learning for batch processes. *AIChE J.*, 45(10):2175–2187, 1999.

[33] L. Ljung. *System Identification: Theory for the User*. Prentice Hall, Englewood Cliff, NJ, 1987.

[34] A. Maarleveld and J. E. Rijnsdorp. Constraint control on distillation columns. *Automatica*, 6:51–58, 1970.

[35] E. S. Meadows and J. B. Rawlings. Model identification and control of a semibatch chemical reactor. In *American Control Conference*, pages 249–255, Boston, MA, 1991.

[36] K. L. Moore. *Iterative Learning Control for Deterministic Systems*. Springer-Verlag, Advances in Industrial Control, London, 1993.

[37] M. Morari and E. Zafiriou. *Robust Process Control*. Prentice-Hall, London, 1989.

[38] M. Noda, T. Chida, S. Hasebe, and I. Hashimoto. On-line optimization system of pilot scale multi-effect batch distillation system. *Comp. Chem. Eng.*, 24:1577–1583, 2000.

[39] C. C. D. Pai and R. R. Hughes. Strategies for formulating and solving two-stage problems for process design under uncertainty. *Comp. Chem. Eng.*, 11:695–706, 1987.

[40] E. N. Pistikopoulos and M. G. Ierapetritou. Novel approach for optimal process design under uncertainty. *Comp. Chem. Eng.*, 19:1089–1110, 1995.

[41] S. R. Ponnuswamy, S. L. Shah, and C. A. Kiparissides. Computer optimal control of batch polymerization reactors. *Ind. Eng. Chem. Res.*, 26:2229–2236, 1987.

[42] J. B. Rawlings, N. F. Jerome, J. W. Hamer, and T. M. Bruemmer. End-point control in semi-batch chemical reactors. In *IFAC DYCORD+'89*, pages 339–344, Maastricht, Netherlands, 1989.

[43] H. M. Robbins. A generalized Legendre–Clebsch condition for the singular cases of optimal control. *IBM Journal of Research and Development*, 1967.

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

[45] D. G. Robertson, J. H. Lee, and J. B. Rawlings. A moving-horizon based approach for least squares estimation. *AIChE J.*, 42(8):2209–2224, 1996.

[46] D. Ruppen, C. Benthack, and D. Bonvin. Optimization of batch reactor operation under parametric uncertainty - Computational aspects. *J. Process Contr.*, 5(4):235–240, 1995.

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

[48] A. P. Sage and C. C. White III. *Optimal Systems Control*. Prentice-Hall, Englewood Cliffs, 2nd edition, 1977.

[49] B. Schenker and M. Agarwal. On-line optimized feed switching in semi-batch reactors using semi-empirical dynamic models. *Control Eng. Practice*, 8(12):1393–1403, 2000.

[50] M. Soroush and S. Valluri. An approach to optimization and control of batch processes. In *American Control Conference*, pages 490–494, Baltimore, MD, 1994.

[51] B. Srinivasan, S. Palanki, and D. Bonvin. Dynamic optimization of batch processes: I. Characterization of the nominal solution. *Comp. Chem. Eng.*, page Submitted, 2001.

[52] B. Srinivasan, C. J Primus, D. Bonvin, and N. L. Ricker. Run-to-run optimization via generalized constraint control. *Control Eng. Practice*, page In Press, 2001.

[53] P. Terwiesch, M. Agarwal, and D. W. T. Rippin. Batch unit optimization with imperfect modeling - A survey. *J. Process Contr.*, 4:238–258, 1994.

[54] P. Terwiesch, D. Ravemark, B. Schenker, and D. W. T. Rippin. Semi-batch process optimization under uncertainty: Theory and experiments. *Comp. Chem. Eng.*, 22(1–2):201–213, 1998.

[55] O. Ubrich, B. Srinivasan, F. Stoessel, and D. Bonvin. Optimization of a semi-batch reaction system under safety constraints. In *European Control Conference*, pages F306.1–6, Karlsruhe, Germany, 1999.

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

[57] B. Wittenmark. Adaptive dual control methods: An overview. In *IFAC Symposium on Adaptive Syst. in Control and Signal Proc.*, pages 67–72, Budapest, 1995.

[58] Y. Yabuki and J. F. MacGregor. Product quality control in semi-batch reactors using mid-course correction policies. In *IFAC ADCHEM'97*, pages 189–194, Banff, Canada, 1997.

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