Available online at www.sciencedirect.com
SCIENCE DIRECT®
Computers and Chemical Engineering 27 (2002) 27–44
Computers & Chemical Engineering
www.elsevier.com/locate/compchemeng

Dynamic optimization of batch processes II. Role of measurements in handling uncertainty

B. Srinivasan [a], D. Bonvin [a,*], E. Visser [a], S. Palanki [b]

[a] Laboratoire d'Automatique, École Polytechnique Fédérale de Lausanne, CH-1015 Lausanne, Switzerland
[b] Florida State University, Tallahassee, FL, USA

Received 31 July 2000; accepted 22 April 2002

Abstract

The main bottleneck in using optimization at the industrial level is the presence of uncertainty in the form of model mismatch and disturbances. The way uncertainty can be handled constitutes 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 from the sensitivity-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 is proposed, where optimality is achieved by tracking the necessary conditions of optimality. The different approaches are compared via the simulation of a bioreactor for penicillin production.

Keywords: Batch processes; Dynamic optimization; Optimal control; Robust optimization; Run-to-run 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. The companion paper (Srinivasan, Palanki, & Bonvin, 2002) of this two-part series discussed the characterization of the optimal solution on the basis of a nominal model, i.e. without considering the uncertainty resulting from model mismatch and disturbances.

In practical situations, however, an accurate model can rarely be found with affordable effort (Bonvin, 1998). For example, the stoichiometry and kinetics of reaction systems are often insufficiently characterized, thereby leading to reaction lumping and macroscopic modeling using time-varying parameters. Furthermore, since the model parameters are usually estimated from laboratory-scale experiments, they might be inaccurate in commercial-scale reactors due to differences in mixing characteristics, heat and mass transfer. 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 (Chu & Constantinides, 1988; Ponnuswamy, Shah, & Kiparissides, 1987; Soroush & Valluri, 1994). Moreover, constraint satisfaction, which becomes important in the presence of safety constraints, may not be guaranteed unless a conservative strategy is adopted (Gygax, 1988; Ubrich, Srinivasan, Stoessel, & Bonvin, 1999).

Two main classes of optimization methods are available for handling uncertainty. The essential difference relates to whether or not *measurements* are used 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 classi-

* Corresponding author. Tel.: +41-21-693-3843; fax: +41-21-693-2574
E-mail address: dominique.bonvin@epfl.ch (D. Bonvin).

PII: S0098-1354(02)00117-5

fication is similar to that found in control problems with the robust and adaptive approaches.

For a measurement-based optimization strategy to be applicable, reliable measurements are necessary. Developments in chromatographic and spectroscopic sensors enable on-line estimation of chemical composition in reacting and non-reacting mixtures (Nichols, 1988; McLennan & Kowalski, 1995). As an example of new and promising developments, rugged NIR spectroscopic sensors can be used to monitor performance in industrial chemical processes. The physical measurements (absorbance or reflectance) extend over several hundred wavelengths and are rather sensitive to chemical composition. This has opened new avenues for using measurements to handle uncertainty.

This paper first reviews the available robust and measurement-based optimization strategies. A novel optimization approach for batch processes is then proposed, where optimality is achieved by tracking the necessary conditions of optimality that remain valid also in the presence of uncertainty.

The paper is organized as follows. Section 2 provides a brief overview of dynamic optimization methods and introduces the topic of optimization under uncertainty. The robust optimization framework is developed in Section 3, while Section 4 presents various measurement-based optimization schemes. The novel measurement-based optimization framework is developed in Section 5. A bioreactor example is provided in Section 6 to illustrate the 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 Fig. 1.

The first level of classification depends on whether or not uncertainty (e.g. variations in initial conditions, uncertain model parameters, or process disturbances) is considered. The standard approach is to discard uncertainty, which leads to a nominal solution that, however, may not even be feasible, let alone optimal, in the presence of uncertainty.

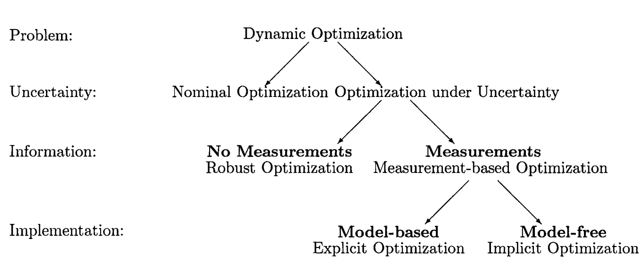

Fig. 1. Dynamic optimization scenarios.

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 at the implementation level. The different scenarios will be discussed in detail below.

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 (Srinivasan et al., 2002) for more information):

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

$$\text{s.t.} \quad \dot{\boldsymbol{x}} = \boldsymbol{F}(\boldsymbol{x}, \boldsymbol{u}), \quad \boldsymbol{x}(0) = \boldsymbol{x}_0, \tag{2}$$

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

where J is the scalar performance index to be minimized; $\boldsymbol{x}$, the n-dimensional vector of states with known initial conditions $\boldsymbol{x}_0$; $\boldsymbol{u}$, the m-dimensional vector of inputs; $\boldsymbol{F}$, a vector field describing the dynamics of the system; $\boldsymbol{S}$ the ζ-dimensional vector of path constraints (which include state constraints and input bounds); $\boldsymbol{T}$ the τ-dimensional vector of terminal constraints; ϕ, a smooth scalar function representing the terminal cost; and t_f the final time that is finite but can be either fixed or free (the more general case of a free final time is considered in Eq. (1)).

Application of Pontryagin's Maximum Principle (PMP) to Eqs. (1)–(3) results in the following Hamiltonian and adjoint equations (Bryson, 1999):

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

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

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

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

$$\mu^{\mathrm{T}} S = 0, \quad \nu^{\mathrm{T}} T = 0. \tag{7}$$

For the case of a free terminal time, an additional condition, referred to as the transversality condition, needs to be satisfied:

$$H(t_f) = 0. \tag{8}$$

The solution of this nominal optimization problem, where uncertainty is ignored, is discussed in detail in the companion paper (Srinivasan et al., 2002).

2.2. Optimization under uncertainty

While the solution of the nominal optimization problem is important to elucidate the qualitative features of the optimal solution (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, a solution is sought that can take the uncertainty into account explicitly (Terwiesch, Agarwal, & Rippin, 1994). The uncertainty is dealt with by considering several possible values for the uncertain parameters. The optimization is performed either by considering the 'worst-case scenario' or in some 'expected sense'. The input trajectories are then computed off-line once and utilized for all batches. However, this typically requires solving an optimization problem that includes many more differential equations than without uncertainty (Ruppen, Benthack, & Bonvin, 1995). The resulting solution is conservative, but it corresponds to the *best* strategy available in the absence of measurements.
- *Measurement-based optimization:* Measurements can be used to cope with uncertainty by adjusting the input profiles to account for both parametric uncertainty and disturbances. The measurement-based optimization methods can be classified according to whether or not a model of the plant is used for implementation. In the model-based schemes, a standard model-based optimization is repeated explicitly with the advent of every measurement, hence the name *explicit optimization*. The model may also be updated using fresh measurements. In contrast, in the model-free implementation schemes, the optimization is carried out implicitly (hence the name *implicit optimization*) by either tracking appropriate references or using prior information regarding the optimal solution.

The robust and measurement-based optimization approaches are described in Sections 3 and 4, respectively.

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 optimization in the presence of uncertainty is to find the optimal operating strategy iteratively, while performing only a few sub-optimal runs and preferably no unacceptable ones (Filippi-Bossy, Bordet, Villermaux, Marchal-Brassely, & Georgakis, 1989; Zafiriou & Zhu, 1990; Fotopoulos, Georgakis, & Stenger, 1994).

Thus, measurement-based optimization schemes can also be classified depending on the type of update used (inter-run or intra-run) and the type of information available (batch-end or on-line). If measurements are available off-line, 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 for the current batch. However, this approach does not account for the effect of process disturbances within the batch. When information is available during the batch, an *on-line optimization* approach can be used. This approach, which seeks to optimize every batch run amidst uncertainty, at the end of the batch leads to better performance.

3. Robust optimization

3.1. Problem formulation

The nominal optimization problem described in Eqs. (1)–(3) and solved in the companion paper (Srinivasan et al., 2002) 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 that can be expressed in the form of uncertain model parameters, $\boldsymbol{\theta}$, and unknown disturbances, $\boldsymbol{d}(t)$. Model parameters can be uncertain due to lack of knowledge or 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 problem can be formulated as follows:

$$\min_{t_f, \boldsymbol{u}(t)} J = \phi(\boldsymbol{x}(t_f), \boldsymbol{\theta}), \tag{9}$$

$$\text{s.t.} \quad \dot{\boldsymbol{x}} = \boldsymbol{F}(\boldsymbol{x}, \boldsymbol{\theta}, \boldsymbol{u}) + \boldsymbol{d}(t), \quad \boldsymbol{x}(0) = \boldsymbol{x}_0, \tag{10}$$

$$\boldsymbol{S}(\boldsymbol{x}, \boldsymbol{\theta}, \boldsymbol{u}) \le \boldsymbol{0}, \quad \boldsymbol{T}(\boldsymbol{x}(t_f), \boldsymbol{\theta}) \le \boldsymbol{0}, \tag{11}$$

where $\boldsymbol{\theta}$ is the vector of uncertain parameters, and $\boldsymbol{d}(t)$ the unknown disturbance (process noise) vector. In addition, the initial conditions $\boldsymbol{x}_0$ could also be uncertain, mainly due to variations in reactant quality (not shown here).

Robust optimization involves finding the optimal operating profiles when modeling uncertainties are considered explicitly in the optimization problem (Terwiesch et al., 1994). 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, $\boldsymbol{\theta} \in \Theta$ with the associated probability density function $p(\boldsymbol{\theta})$. With the set membership type of uncertainty, the only information available is $\boldsymbol{\theta} \in \Theta$. In the latter case, Θ is essentially bounded, whereas this need not be the case in the former.

Similarly, the disturbance vector $\boldsymbol{d}(t) \in \mathscr{D}$ is also either random noise with the associated probability density function $p(\boldsymbol{d})$ or simply bounded in the set $\mathscr{D}$. The main feature that distinguishes $\boldsymbol{d}(t)$ from $\boldsymbol{\theta}$ is the (possibly fast) variation of the disturbances 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.

3.2. Computation of the probability density function of the states

If 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 calculations, the robust optimization problem is typically more involved than the similar problem with known parameters. The probability density function of $\boldsymbol{x}(t)$ can be computed using one of the two approaches described next:

- *Discretization:* The probability density function $p(\boldsymbol{\theta})$ is discretized in the form of a numerical grid. Let D be the number of discretization points, and $\boldsymbol{\theta}^j$, $j = \{1, 2, \ldots, D\}$, the jth discrete values of $\boldsymbol{\theta}$ with relative weighting w^j, $\Sigma_j\, w^j = 1$. The grid for $\boldsymbol{\theta}$ can be chosen either randomly (Monte-Carlo sampling) or systematically. The probability density function of the states is then obtained in a discrete fashion, i.e. $P[\boldsymbol{x} = \boldsymbol{x}^j] = w^j$, where $\boldsymbol{x}^j$ is obtained by simulating a copy of the system for the discretization point $\boldsymbol{\theta}^j$:

$$\dot{\boldsymbol{x}}^j = \boldsymbol{F}(\boldsymbol{x}^j, \boldsymbol{\theta}^j, \boldsymbol{u}), \quad \boldsymbol{x}^j(0) = \boldsymbol{x}_0, \quad \text{for } j = \{1, \ldots, D\}. \tag{12}$$

  An augmented dynamic system of dimension nD needs to be integrated. However, this problem is an ideal candidate for parallelization since the individual model evaluations are mutually decoupled (Grossmann and Sargent, 1978). In dealing with time-varying disturbances, the discretization should also be done in the time direction, i.e. different disturbance values should be considered for different time intervals. Such an approach is referred to as probability grid filtering (Terwiesch et al., 1994).
- *Propagation:* If the probability density function of the states $\boldsymbol{x}(t)$ can be assumed to follow (say) a gaussian distribution, the mean $\boldsymbol{E}(\boldsymbol{x})$ and the standard deviation $\boldsymbol{\sigma}(\boldsymbol{x})$ give complete information, i.e. the probability density function is described by $p(\boldsymbol{x}) = \mathscr{N}(\boldsymbol{E}(\boldsymbol{x}), \boldsymbol{\sigma}(\boldsymbol{x}))$, where $\mathscr{N}$ is the normal distribution function. In such a case, $p(\boldsymbol{x})$ can be obtained by propagating only the mean $\bar{\boldsymbol{x}} = \boldsymbol{E}(\boldsymbol{x})$ and the covariance matrix $\boldsymbol{\Sigma} = \boldsymbol{\sigma}(\boldsymbol{x})\boldsymbol{\sigma}(\boldsymbol{x})^{\mathrm{T}}$. Also, it is assumed that the propagation of $\boldsymbol{x}$ can be computed from the mod with the mean value of the parameters, $\bar{\boldsymbol{\theta}} = \boldsymbol{E}(\boldsymbol{\theta})$, and the dynamics of $\boldsymbol{\Sigma}$ determined from the linearization of the state equations (10) around $\bar{\boldsymbol{x}}$ and $\bar{\boldsymbol{\theta}}$ as given below (Friedland, 1986):

$$\dot{\bar{\boldsymbol{x}}} = \boldsymbol{F}(\bar{\boldsymbol{x}}, \bar{\boldsymbol{\theta}}, \boldsymbol{u}), \quad \bar{\boldsymbol{x}}(0) = \boldsymbol{x}_0, \tag{13}$$

$$\dot{\boldsymbol{\Sigma}} = \left.\frac{\partial \boldsymbol{F}}{\partial \boldsymbol{x}}\right|_{\bar{\boldsymbol{x}},\bar{\boldsymbol{\theta}}} \boldsymbol{\Sigma} + \boldsymbol{\Sigma} \left.\frac{\partial \boldsymbol{F}}{\partial \boldsymbol{x}}\right|^{\mathrm{T}}_{\bar{\boldsymbol{x}},\bar{\boldsymbol{\theta}}} + \left.\frac{\partial \boldsymbol{F}}{\partial \boldsymbol{\theta}}\right|_{\bar{\boldsymbol{x}},\bar{\boldsymbol{\theta}}} \boldsymbol{\Sigma} \left.\frac{\partial \boldsymbol{F}}{\partial \boldsymbol{\theta}}\right|^{\mathrm{T}}_{\bar{\boldsymbol{x}},\bar{\boldsymbol{\theta}}}, \quad \boldsymbol{\Sigma}(0) = \boldsymbol{\Sigma}_0. \tag{14}$$

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 (Srinivasan et al., 2002), the probability, the expected value, or the extremum value of the random variables of interest (cost function and constraints) need to be used in the formulation of the optimization problem. The possible formulations for the cost function and the constraints are described next:

- *Probability:* The probability of a random variable meeting a given set of specifications is considered (Terwiesch, Ravemark, Schenker, & Rippin, 1998). This is natural for the constraints where the probabilities of constraint satisfaction should be greater than specified confidence levels $\boldsymbol{c}$, $\boldsymbol{P}[\boldsymbol{S}(\boldsymbol{x}, \boldsymbol{\theta}, \boldsymbol{u}) \le \boldsymbol{0}] \ge \boldsymbol{c}$. 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, $J = P[\phi(\boldsymbol{x}(t_{\mathrm{f}}), \boldsymbol{\theta}) \ge \eta]$, where η is an appropriately chosen threshold value.

With the discretization approach, the required probability is computed by picking those realizations where $\phi(\mathbf{x}^j(t_f), \boldsymbol{\theta}^j) \geq \eta$:

$$P[\phi(\mathbf{x}(t_f), \boldsymbol{\theta}) \geq \eta] = \sum_{j=1}^{D} I^j w^j, \quad \text{where } I^j = \begin{cases} 1 & \text{if } \phi(\mathbf{x}^j(t_f), \boldsymbol{\theta}^j) \geq \eta \\ 0 & \text{if } \phi(\mathbf{x}^j(t_f), \boldsymbol{\theta}^j) < \eta \end{cases}. \tag{15}$$

The discretization of $\boldsymbol{\theta}$ can also be adapted as the optimization proceeds. For this, the region of $\boldsymbol{\theta}$ where $\phi(\mathbf{x}(t_f), \boldsymbol{\theta}) \geq \eta$ or $\mathbf{S}(\mathbf{x}, \boldsymbol{\theta}, \mathbf{u}) \leq \mathbf{0}$ is identified. More discretization points are provided in and around the feasible region to obtain a more accurate prediction of the required probability (Halemane & Grossmann, 1983).With the propagation approach, the required probability is obtained by integrating $p(\mathbf{x}(t_f))$ over the set where $\phi(\mathbf{x}(t_f), \boldsymbol{\theta}) \geq \eta$:

$$P[\phi(\mathbf{x}(t_f), \boldsymbol{\theta}) \geq \eta] = \int_{\phi(\mathbf{x}(t_f), \boldsymbol{\theta}) \geq \eta} p(\mathbf{x}(t_f))\, d\mathbf{x}(t_f). \tag{16}$$

- *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 to be maximized, or the expected amount of undesired products to be minimized, $J = E[\phi(\mathbf{x}(t_f), \boldsymbol{\theta})]$ (Ruppen et al., 1995). Minimizing the variance of some target function $h(\mathbf{x}(t_f), \boldsymbol{\theta})$ also falls under this framework since, by formulating $\phi(\mathbf{x}(t_f), \boldsymbol{\theta}) = (h(\mathbf{x}(t_f), \boldsymbol{\theta}) - E[h(\mathbf{x}(t_f), \boldsymbol{\theta})])^2$, J becomes the variance of h. Expectations can also enter the optimization problem at the constraint level, a typical example being bounds of the variance of $h(\mathbf{x}, \boldsymbol{\theta})$, $E[(h(\mathbf{x}, \boldsymbol{\theta}) - E[h(\mathbf{x}, \boldsymbol{\theta})])^2] \leq h_{\max}$.The numerical evaluation of expectation-type functions is straightforward since they can be obtained as the weighted sum of the values computed from the various parameter realizations, $E[\phi(\mathbf{x}(t_f), \boldsymbol{\theta})] = \Sigma_{j=1}^{D} \phi(\mathbf{x}^j(t_f), \boldsymbol{\theta}^j) w^j$ or as the weighted integration of the probability density function, $E[\phi(\mathbf{x}(t_f), \boldsymbol{\theta})] = \int \phi(\mathbf{x}(t_f), \boldsymbol{\theta}) p(\mathbf{x}(t_f))\, d\mathbf{x}(t_f)$.
- *Extremum value:* When Θ is bounded, the cost and the constraints can be treated for the worst (or, although rarely used, the 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, $\min_{\mathbf{u}(t)} \max_{\boldsymbol{\theta}} \phi(\mathbf{x}(t_f), \boldsymbol{\theta})$ (Alamir & Balloul, 1999). The constraints are reformulated as $\max_{\boldsymbol{\theta}} \mathbf{S}(\mathbf{x}, \boldsymbol{\theta}, \mathbf{u}) \leq \mathbf{0}$. Equivalent to the concept of variance, the difference $\max_{\boldsymbol{\theta}} \phi(\mathbf{x}(t_f), \boldsymbol{\theta}) - \min_{\boldsymbol{\theta}} \phi(\mathbf{x}(t_f), \boldsymbol{\theta})$ can also be used as the cost or $\max_{\boldsymbol{\theta}} h(\mathbf{x}, \boldsymbol{\theta}) - \min_{\boldsymbol{\theta}} h(\mathbf{x}, \boldsymbol{\theta})$ 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 (9)–(11) can be transformed into a numerical (crisp) form using probabilities, expected values, or extremum values, for example as follows:

$$\min_{t_f, \mathbf{u}(t)} J = E[\phi(\mathbf{x}(t_f), \boldsymbol{\theta})], \tag{17}$$

s.t.

state probability density function computed from (12) or (13)–(14),

$$\mathbf{P}[\mathbf{S}(\mathbf{x}, \boldsymbol{\theta}, \mathbf{u}) \leq \mathbf{0}] \geq \mathbf{c}, \quad \mathbf{P}[\mathbf{T}(\mathbf{x}(t_f), \boldsymbol{\theta}) \leq \mathbf{0}] \geq \mathbf{c},$$

where $\mathbf{P}[\mathbf{S}(\mathbf{x}, \boldsymbol{\theta}, \mathbf{u}) \leq \mathbf{0}]$ are the probabilities of path constraint satisfaction taken independently at each point in time, $\mathbf{P}[\mathbf{T}(\mathbf{x}(t_f), \boldsymbol{\theta}) \leq \mathbf{0}]$ the probabilities of terminal constraint satisfaction, and $\mathbf{c}$ the confidence levels for constraint satisfaction. For PProblem (17), any of the optimization methods described in Srinivasan et al. (2002) can be used. The choice of algorithm is 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 $\boldsymbol{\theta}$ be the nominal value of the parameters and $\bar{\mathbf{x}}(t)$ be the evolution of the states corresponding to the parameters $\bar{\boldsymbol{\theta}}$. In the absence of uncertainty, the active path constraints are those with $\mathbf{S}(\bar{\mathbf{x}}, \bar{\boldsymbol{\theta}}, \mathbf{u}) = \mathbf{0}$. However, in the presence of uncertainty, the conditions $\mathbf{P}[\mathbf{S}(\mathbf{x}, \boldsymbol{\theta}, \mathbf{u}) \leq \mathbf{0}] = \mathbf{c}$ determine the active constraints. For the probabilities of constraint satisfaction to equal the confidence levels $\mathbf{c}$, the nominal values $\mathbf{S}(\bar{\mathbf{x}}, \bar{\boldsymbol{\theta}}, \mathbf{u})$ cannot be zero (i.e. right on the constraints), but have to be pushed inside the feasible region or backed off. Let $\mathbf{b}_S > \mathbf{0}$ be the backoffs required for constraint satisfaction in the presence of uncertainty, i.e. $\mathbf{S}(\bar{\mathbf{x}}, \bar{\boldsymbol{\theta}}, \mathbf{u}) + \mathbf{b}_S = \mathbf{0} \Rightarrow \mathbf{P}[\mathbf{S}(\mathbf{x}, \boldsymbol{\theta}, \mathbf{u}) \leq \mathbf{0}] = \mathbf{c}$. The backoffs $\mathbf{b}_S$ are determined from the variance of $\mathbf{S}(\mathbf{x}, \boldsymbol{\theta}, \mathbf{u})$ and, clearly, the larger the spread in $\mathbf{S}(\mathbf{x}, \boldsymbol{\theta}, \mathbf{u})$, the larger the necessary backoffs (Visser, Srinivasan, Palanki, & Bonvin, 2000). Thus, with the concept of backoff, the optimization problem (17) can be rewritten in the much simpler form:

$$\min_{t_f, \mathbf{u}(t)} J = \phi(\bar{\mathbf{x}}(t_f), \bar{\boldsymbol{\theta}}), \tag{18}$$

$$\text{s.t.} \quad \dot{\bar{x}} = F(\bar{x}, \bar{\theta}, u), \quad x(0) = x_0, \tag{19}$$

$$S(\bar{x}, \bar{\theta}, u) + b_S \leq 0, \quad T(\bar{x}(t_f), \bar{\theta}) + b_T \leq 0, \tag{20}$$

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

The concept of backoff introduced here is similar to that found in two-stage stochastic programming used for optimal process design (Pai & Hughes, 1987). If the backoffs were known, the formulation (18)–(20) 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 optimal solution. This calls for the following iterative procedure:

1) Guess the initial backoffs b_S^0 and b_T^0, $k = 0$.
2) Compute the optimal solution t_f^k, $u^k(t)$ of Eqs. (18)–(20) using backoffs b_S^k and b_T^k.
3) Compute the probability density function of the states for the optimal solution using Eq. (12) or Eqs. (13) and (14) 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 $S(\bar{x}^k, \bar{\theta}, u^k) + b_S^{k+1} = 0 \Rightarrow P[S(x^k, \theta, u^k) \leq 0] = c$. Also, choose b_T^{k+1} similarly.
5) Set $k = k+1$ and repeat Steps 2–5 until b_S^k and b_T^k have converged or the maximum number of allowed iterations 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 Eqs. (9)–(11) and Eqs. (18)–(20) 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 (Srinivasan et al., 2002), the loss in performance caused by this conservatism is given by $\int_0^{t_f} \mu^T b_S \, dt + v^T b_T$. Thus, the larger the backoffs, 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:

$$\min_{t_f^k, u^k([t_l^k, t_f^k])} J^k = \phi(x^k(t_f^k), \theta), \tag{21}$$

$$\text{s.t.} \quad \dot{x}^k = F(x^k, \theta, u^k) + d^k, \quad x^k(0) = x_0^k, \tag{22}$$

$$y^k = h(x^k, \theta) + v^k, \tag{23}$$

$$S(x^k, \theta, u^k) \leq 0, \quad T(x^k(t_f^k), \theta) \leq 0, \tag{24}$$

given

$$y^j(i), \quad i = \{1, \ldots, N\} \text{ for } j = \{1, \ldots, k-1\}, \quad i = \{1, \ldots, l\} \text{ for } j = k, \tag{25}$$

where $x^k(t)$ is the state vector, $u^k(t)$ the input vector, $d^k(t)$ the process disturbance vector, $v^k(t)$ the measurement noise vector, and J^k the cost function for the kth batch. Let $y = h(x, \theta)$, a p-dimensional vector, be the combination of states that can be measured, $y^j(i)$ the ith measurement taken during the jth 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^k, of the kth batch in order to tackle the uncertainty in θ and d^k and determine the optimal input policy for the remaining time interval $[t_l^k, t_f^k]$ of the kth batch. If only batch-end measurements are available, $t_l^k = 0$.

The formulation (21)–(25) is too general to be solved as such. Depending on how the measurements are utilized and whether or not a model is used, various schemes are possible as will be described next. In the explicit optimization schemes, the problem (21)–(25) is solved as two subsequent problems (estimation and optimization), while in the implicit optimization schemes, a feedback law links past measurements and future inputs.

Two classifications of measurement-based optimization schemes are possible. As hinted above, a first classification is based on the role played by the model at the implementation level, i.e. either model-based explicit optimization or model-free implicit optimization. This classification was presented in Fig. 1. The other classification depends on the type of measurement available, i.e. either on-line measurements available at the end ot the batch leading to on-line optimization, or off-line measurements avilable at the end of the batch leading to run-to-run optimization. The use of one or the other classification is a matter of choice, and here the first classification based on the methodology is chosen. Consequently, for each methodology, the case of on-line measurements leading to on-line optimization will be discussed first, and a subsection will deal with the case of off-line measurements leading to run-to-run optimization.

4.2. Explicit optimization

In the on-line version of model-based explicit optimization, the optimal inputs are calculated by repeatedly solving Eqs. (21)–(25) with the advent of every new measurement. The calculated inputs are implemented until the next measurement is taken. As the estimation and numerical optimization tasks are carried out repeatedly, this scheme is also referred to as *repeated optimization* (Abel & Marquardt, 1998; Ruppen, Bonvin, & Rippin, 1998; Rawlings, Jerome, Hamer, & Bruemmer, 1989; Eaton & Rawlings, 1990). This procedure is similar to model predictive control in the way the measurements are incorporated.

4.2.1. Estimation and optimization problems

The problem (21)–(25) is split into two subproblems. In the first subproblem (estimation), measurements are used to estimate the current states and parameters. In the second subproblem (optimization), the information on the current states serves as initial conditions, and the dynamic model is used to predict the evolution of the system and update the inputs towards the optimum.

- *Estimation problem:* The estimation problem can be formulated mathematically as follows:

$$(\boldsymbol{x}_0^*, \boldsymbol{\theta}^*, \boldsymbol{d}^*) = \arg\min_{\boldsymbol{x}_0, \boldsymbol{\theta}, \boldsymbol{d}} \sum_{j=1}^{k-1} \sum_{i=1}^{N} \|\boldsymbol{y}^j(i) - \hat{\boldsymbol{y}}^j(i)\|_Q + \sum_{i=1}^{l} \|\boldsymbol{y}^k(i) - \hat{\boldsymbol{y}}^k(i)\|_Q, \tag{26}$$

$$\text{s.t.} \quad \dot{\hat{\boldsymbol{x}}}^j = \boldsymbol{F}(\hat{\boldsymbol{x}}^j, \boldsymbol{\theta}, \boldsymbol{u}^j) + \boldsymbol{d}^j, \quad \boldsymbol{x}^j(0) = \boldsymbol{x}_0^j, \tag{27}$$

$$\hat{\boldsymbol{y}}^j = \boldsymbol{h}(\hat{\boldsymbol{x}}^j, \boldsymbol{\theta}), \tag{28}$$

given

$$\boldsymbol{y}^j(i), \quad i = \{1, \ldots, N\} \text{ for } j = \{1, \ldots, k-1\}, \quad i = \{1, \ldots, l\} \text{ for } j = k. \tag{29}$$

where an appropriate weighted norm $\|\cdot\|_Q$ of the error is minimized. $\hat{\boldsymbol{x}}$ and $\hat{\boldsymbol{y}}$ represent the estimates of states and outputs, respectively, and the superscript $(\cdot)^*$ is used to denote optimal values. Using $(\boldsymbol{x}_0^*, \boldsymbol{\theta}^*, \boldsymbol{d}^*)$, an estimate of the state vector at t_l^k, x_l^{k*} can be obtained by integrating Eq. (27) for $j = k$. To solve Eqs. (26)–(29), a variety of stochastic methods such as Extended Kalman Filtering (Jazwinski, 1970; Sage & White, 1977) and prediction error identification (Ljung, 1987) as well as deterministic methods such as nonlinear least-squares (Robertson, Lee, & Rawlings, 1996; Ruppen et al., 1998) have been proposed in the literature. 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 $\boldsymbol{d}(t)$, 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 $\boldsymbol{\theta}$) requires persistency of excitation, i.e. the inputs should be varied sufficiently to uncover the unknown parameters. This might cause a difficulty as discussed in the next subsection.
- *Optimization problem:* The second subproblem is the optimization problem that uses the estimated parameters and initial conditions to compute the optimal input profiles:

$$\min_{t_f^k, \boldsymbol{u}^k([t_l^k, t_f^k])} J^k = \phi(\boldsymbol{x}^k(t_f^k), \boldsymbol{\theta}^*), \tag{30}$$

$$\text{s.t.} \quad \dot{\boldsymbol{x}}^k = \boldsymbol{F}(\boldsymbol{x}^k, \boldsymbol{\theta}^*, \boldsymbol{u}^k), \quad \boldsymbol{x}^k(t_l^k) = \boldsymbol{x}_l^{k*}, \tag{31}$$

$$\boldsymbol{S}(\boldsymbol{x}^k, \boldsymbol{\theta}^*, \boldsymbol{u}^k) \le \boldsymbol{0}, \quad \boldsymbol{T}(\boldsymbol{x}^k(t_f^k), \boldsymbol{\theta}^*) \le \boldsymbol{0}. \tag{32}$$

Instead of re-optimizing at every measurement instant, re-optimization can be undertaken only when the estimates deviate significantly from their predicted values.

4.2.2. Types of model

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

- *Fixed model* (Meadows & Rawlings, 1991; Abel, Helbig, Marquardt, Zwick, & Daszkowski, 2000): If the model is not adjusted, it needs to be fairly accurate. This, however, is against the basic assumption that there is (considerable) uncertainty. If the uncertainty is only in the form of disturbances and not in the model parameters, it may be sufficient to use a fixed model. On the other hand, if the model parameters are 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* (Eaton & Rawlings, 1990; Ruppen et al., 1998; Noda, Chida, Hasebe, & Hashimoto, 2000): When model refinement is used, the need to start with an accurate model is alleviated, but the system must be sufficiently excited for estimating the uncertain parameters. Unfortunately, the optimal inputs often do not provide sufficient excitation. On the other hand, if sufficiently exciting inputs are provided for the sake of parameter identification, the resulting solution may not be optimal. This leads to a conflict between the objectives of parameter identification and optimization. This conflict has been studied in

the adaptive control literature under the label *dual control problem* (Roberts & Williams, 1981; Wittenmark, 1995).

4.2.3. Specificities of explicit run-to-run optimization

Run-to-run optimization is used 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. These measurements are most common in industrial practice (Bonvin, Srinivasan, & Ruppen, 2002).

Off-line measurements enable the set-up of a batch-to-batch or *run-to-run* optimization approach 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. Thus, 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. They vary essentially in the way the measurements are incorporated into the adaptation procedure.

- *Measurements for model refinement* (Filippi et al., 1986; Fotopoulos et al., 1994; Garcia, Cabassud, Le Lann, Rigal, & Casamatta, 1997): 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. There can also be a conflict between the objectives of parameter identification and optimization (Srinivasan & Bonvin, 2002).
- *Measurements instead of model prediction* (Zafiriou & Zhu, 1990; Dong, McAvoy, & Zafiriou, 1996; Chin, Lee, & Lee, 1999; Lee, Chin, Lee, & Lee, 1999): Any numerical procedure for dynamic optimization involves the following steps: (i) choice of initial inputs; (ii) calculation of the system states, the performance index J, and the constraints $\boldsymbol{S}$ and $\boldsymbol{T}$; and (iii) adaptation of the inputs towards the optimum (using for example gradient information). The steps (ii) and (iii) are then repeated until convergence. A particularity of run-to-run schemes is that the second step, which is usually performed using a model, can be replaced by measurements, i.e. J, $\boldsymbol{S}$, and $\boldsymbol{T}$ can be obtained from an experimental run. However, when a gradient-based algorithm is used for optimization, a model involving the state and adjoint variables is in general required to compute the gradient as in Eq. (6). Yet, an approach can be devised, where the states are measured from an experimental batch run and the adjoints are obtained from a process model. The model parameters $\boldsymbol{\theta}$ can be either fixed or adapted from run-to-run, while the disturbances $\boldsymbol{d}(t)$ are ignored in the computation of the adjoints.

4.2.4. Limitations of explicit optimization schemes

One of the well-cited disadvantages of explicit optimization schemes is the high computational burden involved in the on-line calculation of the optimal policy, especially for large-dimensional systems (Diwekar, 1995). Furthermore, there are implementation difficulties associated with a fixed model (possible poor prediction) and a refined model (lack of persistent excitation).

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 (Morari & Zafiriou, 1989). 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 (Aström & Wittenmark, 1995; Landau, Lozano, & Saad, 1998). The model parameters are estimated using input–output data (model refinement) and the controller parameters are updated accordingly. Instead of standard adaptive control, imagine now a control 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 is less efficient than standard adaptive control techniques. Yet, the model-based approaches for optimization under uncertainty presented in this subsection are analogous to an adaptive control scheme *without* direct feedback and, thus, suffer from the aforementioned drawbacks.

4.3. Implicit optimization

The underlying idea is to use measurements directly, i.e. without the on-line help of a process model, to guide the system towards the optimal solution. A feedback law of the type, $\boldsymbol{u}^k([t_l^k,\ t_f^k]) = \mathcal{K}(\boldsymbol{y}^j(i))$ is sought, that implicitly solves the optimization problem (21)–(25). Various possible options are discussed next.

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 for comparison with a reference.

- *Interpolation* (Krothapally, Bennett, Finney, & Palanki, 1999; Schenker & Agarwal, 2000): The optimal solution is worked out for different combinations of parameters and states as this is done for the solution of the Hamilton–Jacobi–Bellman equation in dynamic programming (Kirk, 1970; Srinivasan et al., 2002). An additional complication arises in the uncertain scenario since the solution has to be discretized along the parameter values as well. The solution is then stored, 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 best matches the behavior of the system is obtained, and the corresponding input strategy 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 requires either a computationally expensive look-up table or a closed-form feedback law that is analytically expensive or often impossible to obtain.
- *Reference tracking* (Gentric, Pla, Latifi, & Corriou, 1999; Ubrich et al., 1999): If a nominal process model is available, it can be used to compute the optimal solution numerically. Numerical optimization provides information not only regarding the optimal evolution of the inputs $\boldsymbol{u}^*(t)$ but also of the states $\boldsymbol{x}^*(t)$. In the absence of disturbances, open-loop application of the inputs $\boldsymbol{u}^*(t)$ or, equivalently, tracking *any* of the states $\boldsymbol{x}^*(t)$ will result in optimality. However, in the presence of uncertainty, tracking some combinations of the states can be better than open-loop application of the inputs. Hence, the idea to choose appropriate references and track them using feedback controllers. The main advantage of reference tracking is that feedback controllers are numerically inexpensive and their tuning is quite simple. However, the main bottleneck in incorporating reference tracking within the framework of optimization is the choice of optimal 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 references are not properly chosen, their tracking would be either infeasible or non-optimal. Thus, it is important to find signals, the tracking of which implies optimality. Though many studies in the literature have used 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 Section 5, this paper proposes to analyze the necessary conditions of optimality and track them appropriately. This will ensure optimality even in the presence of uncertainty.

4.3.2. Specificities of implicit run-to-run optimization

In implicit run-to-run optimization, the idea of using the measurements instead of model predictions for the purpose of optimization is taken to the extreme. An experimental run is used to determine the cost J and the constraints $\boldsymbol{S}$ and $\boldsymbol{T}$ (Clarke-Pringle & MacGregor, 1998). In addition, if a gradient-based optimization algorithm is used to update the decision variables, the gradient is also computed experimentally (contrarily to the model-based gradient discussed in Section 4.2.3). Every decision variable is perturbed, with an experimental run being performed for each perturbation in order to obtain the gradient. Hence, for each iteration of the optimization procedure, as many batch runs as there are scalar decision variables are necessary.

This procedure is termed evolutionary optimization in the literature (Box & Draper, 1987). 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 the constraints of the optimization problem. In such a case, the objective of the optimization algorithm is to take the system to the constraints. And, since the direction of update is known, neither additional runs nor a process model are necessary to obtain the gradient (Maarleveld & Rijnsdorp, 1970; Srinivasan, Primus, Bonvin, & Ricker, 2001; François, Srinivasan, & Bonvin, 2002).

5. Optimization via tracking of the necessary conditions of optimality

This section proposes an optimization scheme that is based on tracking appropriate references. This way, the optimal inputs are determined directly from process measurements and not from a (possibly inaccurate) model. The reference trajectories are chosen as the necessary conditions of optimality which remain valid in the presence of uncertainty. This particular choice of reference signals constitutes the novelty in the proposed methodology.

Another important difference between the proposed approach and the studies available in the literature is that, here, path and terminal objectives are treated on an equal footing. In addition, the fact that batch runs are typically repeated over time provides the possibility of combining on-line and run-to-run optimization.

5.1. Necessary conditions of optimality

The dynamic optimization problem (1)–(3) has two types of constraints: the path constraints limit the inputs and the states *during the batch*, while the terminal constraints limit the outcome of the batch *at final time*. This gives rise to two corresponding types of objectives, the path and terminal objectives. A characterization of the optimal solution can separate the path from the terminal objectives. For this, the optimal inputs are partitioned into: (i) the various arcs $\boldsymbol{\eta}(t)$ that cater to the path objectives; and (ii) the scalar input parameters $\boldsymbol{\pi}$ (e.g. switching times, final time) that affect the terminal objectives.

Certain parts of the optimal solution are determined by the constraints of the optimization problem, while others exploit the intrinsic compromises present in the system. The characterization proposed in Srinivasan et al. (2002) separates the constraint-seeking decision variables, $\bar{\boldsymbol{\eta}}(t)$ and $\bar{\boldsymbol{\pi}}$, from the sensitivity-seeking decision variables, $\tilde{\boldsymbol{\eta}}(t)$ and $\tilde{\boldsymbol{\pi}}$.

With this partitioning, the necessary conditions of optimality read:

$$\begin{array}{lll} & \text{Path} & \text{Terminal} \\ \text{Constraints} & \bar{\boldsymbol{S}}(\boldsymbol{x},\ \boldsymbol{\theta},\ \boldsymbol{u}) = \boldsymbol{0} & \bar{\boldsymbol{T}}(\boldsymbol{x}(t_f,\ \boldsymbol{\theta})) = \boldsymbol{0} \\ \text{Sensitivities} & \boldsymbol{\lambda}^T(\partial \boldsymbol{F}/\partial \tilde{\boldsymbol{\eta}}) = \boldsymbol{0} & \partial \phi/\partial \tilde{\boldsymbol{\pi}} = \boldsymbol{0} \end{array} \tag{33}$$

The constraint-seeking arcs $\bar{\boldsymbol{\eta}}(t)$ are determined from the active path constraints $\bar{\boldsymbol{S}} = \boldsymbol{0}$, and $\tilde{\boldsymbol{\eta}}(t)$ from the sensitivity conditions $\boldsymbol{\lambda}^T(\partial \boldsymbol{F}/\partial \tilde{\boldsymbol{\eta}}) = \boldsymbol{0}$. On the other hand, the active terminal constraints $\bar{\boldsymbol{T}} = \boldsymbol{0}$ determine the constraint-seeking parameters $\bar{\boldsymbol{\pi}}$, whilst the sensitivity-seeking parameters $\tilde{\boldsymbol{\pi}}$ are obtained from the sensitivity conditions $\partial \phi/\partial \tilde{\boldsymbol{\pi}} = \boldsymbol{0}$.

Due to uncertainty, the numerical values of the various arcs and switching times might differ considerably from those of the nominal solution. However, optimal operation corresponds to verifying all four parts of the necessary conditions of optimality (Eq. (33)), and this also in the presence of uncertainty.

5.2. Description of the NCO-tracking scheme

5.2.1. Assumptions for NCO-tracking

The core assumption is that the set of active constraints (both path and terminal) is known and does not change with uncertainty. This assumption implies that the structure of the optimal solution (the types and sequence of arcs that constitute the optimal solution) of the true system is known a priori.

The solution structure can be obtained in two ways: (i) educated guess by an experienced operator; or (ii) visual inspection of the solution obtained from numerical optimization of a simplified model. Quite often, experience dictates the qualitative shape of the inputs. Otherwise, a simplified (or tendency) model of the process can be used to compute a numerical solution from which the various arcs are identified.

Under the assumption that the types and sequence of arcs and the set of active terminal constraints remain unchanged, then satisfying the active constraints $\bar{\boldsymbol{S}} = \boldsymbol{0}$ and $\bar{\boldsymbol{T}} = \boldsymbol{0}$ and regulating the sensitivities $\boldsymbol{\lambda}^T\ (\partial \boldsymbol{F}/\partial \tilde{\boldsymbol{\eta}}) = \boldsymbol{0}$ and $\partial \phi/\partial \tilde{\boldsymbol{\pi}} = \boldsymbol{0}$ around zero leads to optimality.

5.2.2. Controller structure for NCO-tracking

The notations $\mathscr{P}$ and $\mathscr{T}$ are used to represent the path ($\bar{\boldsymbol{S}}$ and $\boldsymbol{\lambda}^T\ (\partial \boldsymbol{F}/\partial \tilde{\boldsymbol{\eta}})$) and terminal ($\bar{\boldsymbol{T}}$ and $\partial \phi/\partial \tilde{\boldsymbol{\pi}}$) quantities that need to be regulated around zero for optimality. The reference values for $\mathscr{P}$ and $\mathscr{T}$ are simply $\boldsymbol{0}$ as indicated by Eq. (33). Since the proposed scheme is set up to track the necessary conditions of optimality, it will be referred to as the NCO-tracking scheme.

The structure given in Fig. 2 is proposed for tracking the necessary condition of optimality. $\mathscr{P} = \boldsymbol{0}$ and $\mathscr{T} = \boldsymbol{0}$ are satisfied using path and terminal feedback controllers, respectively. The trajectory generator computes the current inputs $\boldsymbol{u}(t)$ from $\boldsymbol{\eta}(t)$ and $\boldsymbol{\pi}$ that are generated by the path and terminal controllers. The tracking aspects with respect to the four elements of the necessary conditions of optimality (Eq. (33)) are discussed next.

1) *Path constraints:* 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 constrained variable is directly available. On the other hand, if the constrained quantity cannot be measured directly, some type of inference (Joseph & Brosilow, 1978; Doyle, 1998) or state estimation is necessary. Thus, tracking the path constraints by adjusting $\bar{\boldsymbol{\eta}}(t)$ is often straightforward.
2) *Terminal constraints:* Off-line measurements of the constrained terminal quantities are typically available, as these are often related to quality specifications. Since the measurements are available only at the end of the batch, a batch-to-batch adaptation scheme can be used to adapt $\bar{\boldsymbol{\pi}}$ and drive the system towards the terminal constraints. Thus, the key difference between the path and terminal constraints is that the former allow intra-run (within-batch) adaptation, while the adaptation is only inter-run (run-to-run) in the latter.

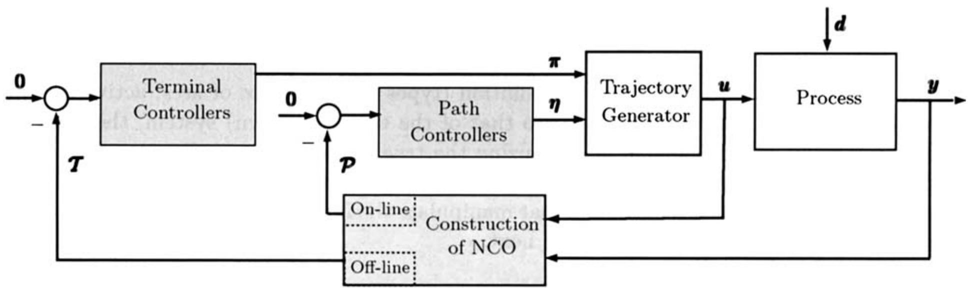

Fig. 2. Optimization via tracking of the necessary conditions of optimality.

3) *Path sensitivities:* Two cases have to be distinguished: (a) $H_{\mathbf{uu}} \neq \mathbf{0}$; and (b) $H_{\mathbf{uu}} = \mathbf{0}$. In the first case, the loss in performance due to non-satisfaction of $\boldsymbol{\lambda}^{\mathrm{T}}\ (\partial \boldsymbol{F}/\partial \tilde{\boldsymbol{\eta}})$ can be considerable. A neighboring extremal controller as described in Bryson (1999) can be used to force the first variation $\boldsymbol{\lambda}^{\mathrm{T}}\ (\partial \boldsymbol{F}/\partial \tilde{\boldsymbol{\eta}})$ to zero. In contrast, when $H_{\mathbf{uu}} = \mathbf{0}$, state controllability is lost along the optimal sensitivity-seeking trajectory and thus, the standard neighboring extremal controller cannot be used. However, when $H_{\mathbf{uu}} = \mathbf{0}$, the loss in performance is often negligible and $\tilde{\boldsymbol{\eta}}(t)$ can often simply be kept at their predetermined values.
4) *Terminal sensitivities:* For the computation of terminal sensitivites, multiple process runs are typically required. Runs with different values of the decision variables $\tilde{\boldsymbol{\pi}}$ can be performed and the differences are used to compute the sensitivities. With this approach, as many runs as there are decision variables are necessary to compute all the sensitivities. An alternative is to use a model of the process for calculating the sensitivities as in Zafiriou and Zhu (1990).

As discussed in Srinivasan et al. (2002), there is often much more to be gained by keeping the path and terminal constraints active than by forcing the sensitivity-seeking variables to zero. This means that it pays off to work close to the active constraints, i.e. where there is a lot to gain!

The above discussion indicates that optimization through tracking of the necessary conditions of optimality is easiest to implement when: (i) the path and terminal constraints can be directly measured; and (ii) the sensitivity-seeking arcs and parameters are either absent or can be kept at predetermined values without much effect on the cost.

5.3. Design aspects

If the selected structure of the optimal solution (types and sequence of arcs, active terminal constraints) is correct, i.e. it corresponds to that of the true (unknown) system, then the proposed tracking scheme will be capable of optimizing the true system. The path controllers are PI-type controllers that manipulate the various input arcs $\boldsymbol{\eta}(t)$ to regulate $\mathscr{P}$ around zero. The terminal controllers are also PI-type controllers that manipulate the input parameters $\boldsymbol{\pi}$ to regulate $\mathscr{T}$ around zero. Various design aspects are treated next.

- *Role of the model:* If the structure of the optimal inputs cannot be obtained via the educated guess of experienced personnel, then a process model is necessary to determine numerically the structure of the optimal inputs. Since the role of the model is only to provide the correct input structure, 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.
- *Difference in time scales—on-line vs. off-line measurements:* In general, there is a difference in time scales between the path and terminal controllers. The path controllers work within a batch using on-line measurements (the running index is the batch time t) (Benthack, 1997). The terminal controllers operate on a batch-to-batch basis using off-line measurements (the running index is the batch number k) (Srinivasan et al., 2001). Note that the switching instants, which are typically included in $\boldsymbol{\pi}$, are adapted on a run-to-run basis. When on-line measurements are not available, the path controllers are inactive. However, if off-line measurements of $\mathscr{P}$ are available, it is possible to use the path controllers in a batch-to-batch mode to generate $\boldsymbol{\eta}(t)$ and so force the system to be closer to the path constraints during the next batch (Moore, 1993). Furthermore, if $\mathscr{T}$ can be predicted from on-line measurements, it may be possible to use the terminal controllers within the batch to generate $\boldsymbol{\pi}$ (Yabuki & MacGregor, 1997). An alternative for meeting terminal constraints using on-line measurements consists of tracking some feasible trajectory, $y_{\mathrm{ref}}(t)$, whose main purpose is to guarantee the satisfaction of the terminal objectives at final time, i.e. $y_{\mathrm{ref}}(t_f) = \mathscr{T} = \mathbf{0}$ (Welz, Srinivasan, & Bonvin, 2002).

- *Centralized vs. decentralized controllers:* The path controllers can be considered as a multivariable controller. However, since $\tilde{\boldsymbol{\eta}}$ does not affect $\bar{\boldsymbol{S}}$, a first input–output pairing can be performed, i.e. $\bar{\boldsymbol{\eta}}$ is paired with $\bar{\boldsymbol{S}}$ and $\tilde{\boldsymbol{\eta}}$ with $\boldsymbol{\lambda}^{\mathrm{T}}$ $(\partial \boldsymbol{F}/\partial \tilde{\boldsymbol{\eta}})$. Still $\bar{\boldsymbol{\eta}}$, $\tilde{\boldsymbol{\eta}}$, $\bar{\boldsymbol{S}}$, and $\boldsymbol{\lambda}^{\mathrm{T}}$ $(\partial \boldsymbol{F}/\partial \tilde{\boldsymbol{\eta}})$ are vector quantities. A sensitivity analysis or relative gain considerations (Ogunnaike & Ray, 1994) can be used for proper input–output pairings or the design of a decoupling scheme. As a result, appropriate decentralized controllers can be designed. A similar procedure is used to set up the terminal controllers.
- *Disturbance rejection:* The presence of disturbances influences both $\boldsymbol{\eta}(t)$ and $\boldsymbol{\pi}$. The disturbances affecting $\boldsymbol{\eta}(t)$ within the batch are rejected by the path controllers. However, the effect of any disturbance affecting $\boldsymbol{\pi}$ within the batch cannot be rejected since the terminal controllers only work on a batch-to-batch basis. Disturbances that remain constant over several batches (e.g. raw material variations) can be rejected by the terminal controllers.
- *Backoffs from constraints:* In the presence of uncertainty that cannot be compensated by feedback, the use of conservative margins, called *backoffs* (see Section 3.4), is necessary to ensure feasibility. The presence of measurement errors also calls for backoffs. The probability density function of the states can be calculated from an estimate of the uncertainty. The backoffs are then chosen such that the spread of the states remains within the feasible region with a certain probability (Visser et al., 2000). Note that the backoffs typically vary with time.

  Due to the sensitivity reduction that is characteristic of feedback control, the conservatism resulting from the presence of uncertainty can be reduced considerably with the proposed tracking framework in comparison with standard robust optimization schemes. Furthermore, the controller parameters can be chosen so as to minimize the spread in the states resulting from the uncertainty. With reduced backoffs, the process can be driven closer to the active constraints, thereby leading to improved performance. Note that the use of feedback is particularly useful when the uncertainty tends to increase with time during a batch run.

6. Illustrative example

In this section, the various optimization strategies discussed above will be illustrated and compared through the simulated operation of a fed-batch bioreactor in the presence of uncertainty. Another comparative example dealing with the optimization of a batch distillation column can be found in Welz et al. (2002). This example, which represents a penicillin fermentation process, has the structure of Example 2 of the companion paper (Srinivasan et al., 2002) wherein the nominal solution has been characterized. The numerical values, however, are different here as they have been adapted from Bajpai and Reuss (1981).

- *Reactions:* $S \rightarrow^{X} X$, $S \rightarrow^{X} P$.
- *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-transfer limitation typically occurring at large biomass concentrations.

6.1. Nominal optimization

6.1.1. 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}, v: kinetic parameters; and Y_{x}, Y_{p}, yield coefficients.

6.1.2. Model equations

$$\dot{X} = \mu(S)X - \frac{u}{V}X \quad X(0) = X_{\mathrm{o}}, \tag{34}$$

$$\dot{S} = -\frac{\mu(S)X}{Y_{\mathrm{x}}} - \frac{vX}{Y_{\mathrm{p}}} + \frac{u}{V}(S_{\mathrm{in}} - S) \quad S(0) = S_{\mathrm{o}}, \tag{35}$$

$$\dot{P} = vX - \frac{u}{V}P \quad P(0) = P_{\mathrm{o}}, \tag{36}$$

$$\dot{V} = u \quad V(0) = V_{\mathrm{o}}, \tag{37}$$

with $\mu(S) = \mu_{\mathrm{m}} S/(K_{\mathrm{m}} + S + (S^2/K_{\mathrm{i}}))$ and the numerical values given in Table 1.

Table 1
Model parameters, operating bounds and initial conditions

μ_{m}	0.02	l/h
K_{m}	0.05	g/l
K_{I}	5	g/l
Y_{x}	0.5	g[X]/g[S]
Y_{p}	1.2	g[P]/g[S]
v	0.004	l/h
S_{in}	200	g/l
u_{min}	0	l/h
u_{max}	1	l/h
X_{max}	3.7	g/l
t_{f}	150	h
X_{o}	1	g/l
S_{o}	0.5	g/l
P_{o}	0	g/l
V_{o}	150	l

6.1.3. Optimization problem

$$\max_{u(t)} J = P(t_f), \tag{38}$$
$$\text{s.t.} \quad (34)\text{–}(37),$$
$$X(t) \le X_{max},$$
$$u_{min} \le u(t) \le u_{max}.$$

6.1.4. Optimal solution

The nominal optimal solution obtained numerically is given in Fig. 3. It consists of three arcs, u_{sens}, u_{min}, and u_{path} (Srinivasan et al., 2002):

- The initial condition for S is chosen so that $S_o = S^* = 0.5$ g/l. Due to this specific choice of initial condition, the sensitivity-seeking arc, u_{sens}, can be applied right from the start in order to increase X as quickly as possible.
- The input is then lowered to u_{min} in order to reach $S = S_e$ as quickly as possible. The switching time between u_{sens} and u_{min} 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 $u = u_{path}$. This guarantees $X = X_{max}$ and $S = S_e$ in that interval.

For the numerical values given in Table 1, the concentration of penicillin at final time is $P^* = 1.68$ g/1.

6.1.5. Analytical expressions for u_{sens} and u_{path}

The sensitivity-seeking arc corresponds to being on the surface $S = S^* = \sqrt{K_i K_m}$, which can be differentiated once to obtain the input:

$$u_{sens} = \frac{V}{S_{in} - S}\left(\frac{1}{Y_x}\mu(S)X + \frac{1}{Y_p}vX\right)\Bigg|_{S=S^*}. \tag{39}$$

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

$$u_{path} = \mu(S)V\,|_{X=X_{max}}. \tag{40}$$

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

$$\dot{S} = -\frac{1}{Y_x}\mu(S)X_{max} - \frac{1}{Y_p}vX_{max} + \mu(S)(S_{in} - S). \tag{41}$$

Note that these dynamics are unstable, with S_e being the corresponding equilibrium point. If the biomass constraint $X = X_{max}$ is entered with $S = S_e = 0.0037$ g/l then S will remain at S_e.

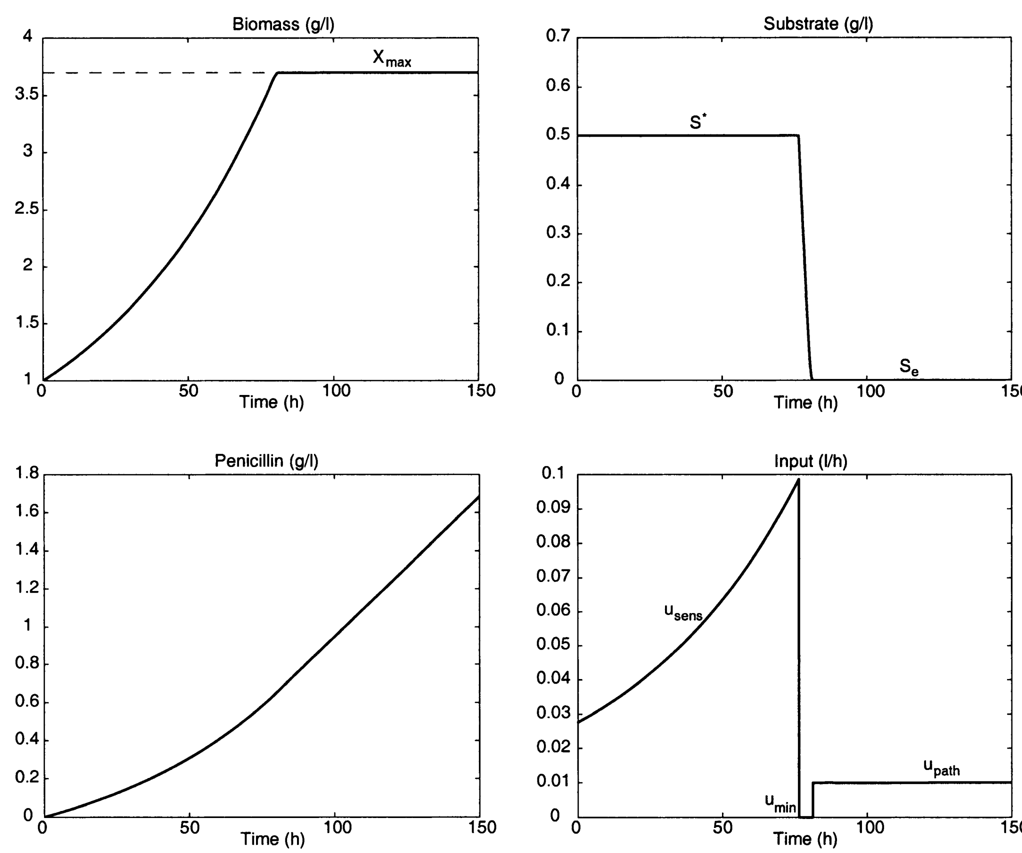

Fig. 3. Nominal optimal trajectories.

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 S_{in} is normally distributed with mean 200 g/l, and standard deviation 25 g/l. S_{in} is assumed to change every hour.

The maximum value of Y_x corresponds to the most unfavorable situation with respect to meeting the path constraint $X(t) \le X_{max}$. Thus, the value $Y_x^{max} = 0.5$, together with the average value $S_{in}^{av} = 200$ g/l is used to compute the conservative solution. Using the concept of backoff introduced in Eqs. (18)–(20), the robust optimization problem for a confidence level of 95% reads:

$$\max_{u(t)} \; J = \bar{P}(t_f), \tag{42}$$

$$\text{s.t.} \quad \dot{\bar{\boldsymbol{x}}} = \boldsymbol{F}(\bar{\boldsymbol{x}}, \bar{\boldsymbol{\theta}}, u), \quad \bar{\boldsymbol{x}}(0) = \boldsymbol{x}_o,$$

$$\bar{X}(t) - X_{max} + b_X \le 0,$$

$$u_{min} \le u(t) \le u_{max},$$

where $\boldsymbol{F}$ represents the model equations (34)–(37). The backoff b_X is calculated such that $\bar{X}(t) - X_{max} + b_X = 0 \Rightarrow P[X(t) \le X_{max}] = 0.95$. Assuming that the states are normally distributed, the propagation method, Eqs. (13) and (14), is used to compute the probability density function of $\boldsymbol{x}(t)$. The backoff $b_X = 0.19$ is calculated using the iterative procedure described in Section 3.4.

The conservative input is applied open-loop to various instantiations of the true reactor (various Y_x values and S_{in} realizations). The corresponding biomass and substrate profiles are shown in Fig. 4. The feasibility of the biomass constraint despite the perturbations in S_{in} is ensured with a probability of 95% by the introduction of backoff. Thus, the conservative input can be applied safely to the true process in an open-loop fashion. However, important oscillations of the substrate concentration result from the uncertainty, and even wash-out of the substrate may happen for low values of Y_x. The resulting loss in performance is important and will be analyzed later.

6.3. Measurement-based optimization

Two different measurement-based on-line optimization scenarios are considered, the model-based explicit optimization, and the model-free implicit optimization.

6.3.1. Explicit optimization

This strategy involves repeating the optimization at periodic intervals. A direct sequential optimization approach (CVP) is used with a piecewise-constant input parameterization using 40 elements (Srinivasan et al., 2002). The optimization routine is time-consuming and often converges to a local minimum, which requires restart and manual guidance to the global minimum.

Full-state measurement is assumed, and the current states act as the initial conditions for the dynamic system in the optimization problem. Also, with the assumption that most of the substrate is indeed consumed to produce biomass through the first reaction, the uncertain parameter Y_x is estimated from the on-line measurements of biomass and substrate concentrations as $Y_x = (XV - X_0V_0)/(S_{in}(V - V_0) + S_0V_0 - SV)$ (this results from biomass and substrate balancing with $\nu X = 0$).

6.3.2. Implicit optimization

The strategy based on tracking the necessary conditions of optimality is considered with the following features:

- Along the first interval, a PI-controller is used to keep $S = S^* = 0.5$ g/l, which corresponds to maintaining maximum biomass growth rate.
- In the second interval, the input is set to its lower bound $u_{min} = 0$.

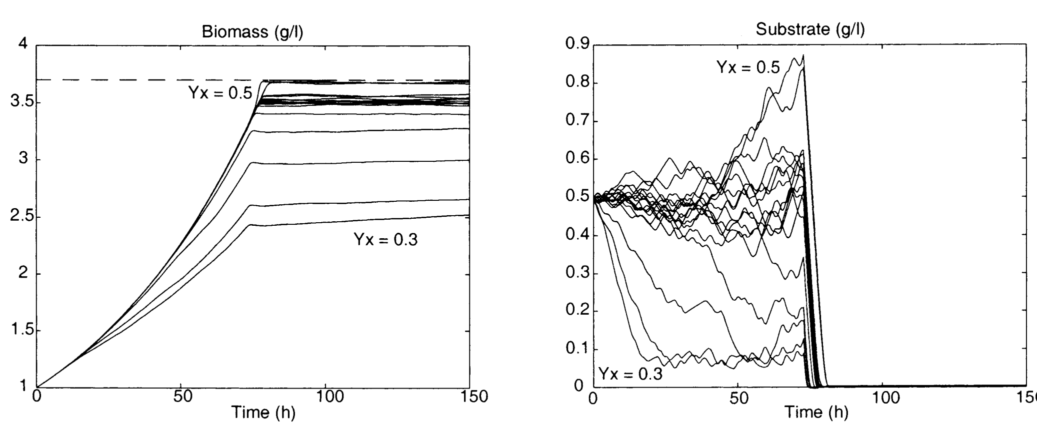

Fig. 4. Open-loop implementation of the robust input calculated with $Y_x = 0.5$ and $S_{in} = 200$ g/l: biomass and substrate profiles for different instantiations of the true reactor (different Y_x values and S_{in} realizations).

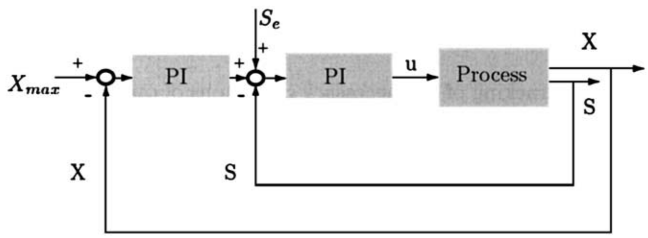

Fig. 5. Cascade structure for the control of biomass and stabilization of the substrate dynamics.

- In the third interval, a biomass controller is used to track the biomass constraint X_{max}. Also, for internal stability, S is maintained near the equilibrium value S_e. A cascade structure (Fig. 5) is used to stabilize the substrate dynamics and cope with the large difference in time scales between the biomass and substrate dynamics. The error in biomass concentration provides the reference for the equilibrium substrate concentration via the master PI-controller.

Although the determination of the switching instant between the first and second arcs is rather involved, a simple heuristic approach is taken here: the input is switched between u_{sens} and u_{min} when $X = 0.96X_{max}$. The input is then kept at u_{min} until $S = S_e$ is reached. If the choice of the first switching instant is not accurate, X will be different from X_{max} when S reaches S_e is reached. However, this error is driven to zero by the control structure of Fig. 5.

Simulation results using the NCO-tracking scheme are presented in Fig. 6 for different instantiations of the true reactor. The variations in the biomass and substrate profiles, and thus also in the backoff from the biomass constraint, are much smaller than in 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 (Bastin & Dochain, 1990).

6.4. Performance comparison

Table 2 illustrates the average performance of the various optimization strategies for different instantiations of the true bioreactor. For each optimization strategy and each value of Y_x, 50 different realizations of S_{in} are considered for computing an average cost.

The nominal cost is fairly independent of the value of Y_x and the variations in S_{in}. This can be explained as follows: according to Eqs. (34)–(37), Y_x and S_{in} affect directly only the dynamics of S, whose variations, however, can be compensated by adapting $u(t)$. Thus, after compensation, all the state evolutions, and also the

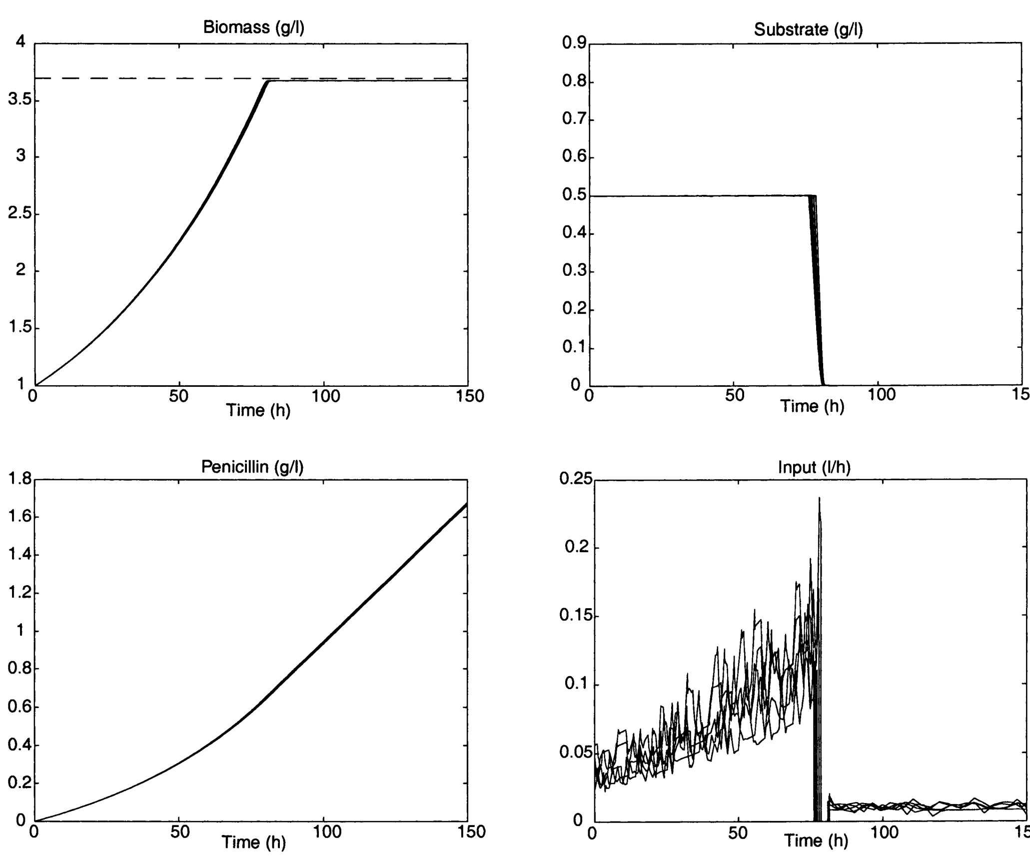

Fig. 6. Optimization via tracking of the necessary conditions of optimality for different Y_x values and S_{in} realizations.

Table 2
Performance comparison (product concentration in g/l) for different values of the uncertain plant parameter Y_x

Y_x	Nominal optimization	Robust optimization	Explicit optimization with n_o				NCO-tracking
			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

Nominal optimization → solution with Y_x known and no disturbance; robust optimization → solution with $Y_x = 0.5$ and backoff $b_x = 0.19$ to account for the variations in S_{in} ($\sigma = 25$ g/l); n_o → number of on-line optimizations; NCO-tracking → optimization based on tracking the necessary conditions of optimality.

objective function, are fairly close to each other. This is illustrated in Fig. 6, where variations of the input $u(t)$ are capable of keeping the state variables nearly unaffected by the uncertainty in Y_x and S_{in}.

This gives the impression that tracking any of the states over the entire interval can provide optimality. However, the evolution of P over the entire interval, and that of X during the first part, are uncontrollable around the optimal solution. Thus, tracking S in the first part and X in the final one is the only reasonable option, which is in fact the solution provided by the necessary conditions of optimality.

Table 2 compares the performance of the ideal situation (nominal case with no disturbance and Y_x known) with that of the robust and measurement-based optimization schemes. The loss in performance compared to the nominal situation has two main causes:

1) *The introduction of backoff in order to remain feasible despite variations in* S_{in}: The effect of the 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 necessary backoff is different. The backoff is largest for the robust optimization and is reduced progressively with increasing reoptimization frequency in the explicit optimization scheme. The smallest conservatism is obtained with the NCO-tracking scheme. Thus, the average performance increases from the robust optimization over the explicit optimization to the NCO-tracking scheme.
2) *Low substrate concentrations resulting from an incorrect value of* Y_x *in the model:* The model value of Y_x is 0.5, whereas the true value used in the simulations ranges from 0.3 to 0.5. Since the model will predict a larger biomass production than is actually happening, the computed feed rate of substrate will tend to be smaller, and with it also the substrate concentration. The effect of non-optimal S values ($S < S^*$) at the beginning of the batch is strongest in the case $Y_x = 0.3$, as can be seen in Fig. 4. The amount of biomass produced at the end of the first arc can be considerably lower than X_{max}. This causes a loss in performance of 24.5% in the case of robust optimization. This loss in performance can be reduced with measurement-based optimization schemes. 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 $S^* = 0.5$ g/l.

With the explicit repeated optimization scheme, it is observed that significant performance improvement can be 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 NCO-tracking scheme. This performance is only slightly lower than the nominal performance that is obtained without uncertainty. Hence, the NCO-tracking 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 can lead 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 settings. Hence, the goal of this work has been to analyze *measurement-based* optimization strategies capable of coping with uncertainty and making optimization more applicable to industry.

It is also possible to 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 has attempted to use feedback control for the sake of optimality. In this sense, the proposed approach has greater industrial potential.

Acknowledgements

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

References

Abel, O., Helbig, A., Marquardt, W., Zwick, H., & Daszkowski, T. (2000). Productivity optimization of an industrial semi-batch polymerization reactor under safety constraints. *Journal of Process Control 10* (4), 351–362.

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

Alamir, M., & Balloul I. (1999). Robust control of general constrained batch processes. In *European control conference*, number F805. Karlsruhe, Germany.

Aström, K. J., & Wittenmark, B. (1995). *Adaptive control* (2nd ed.. Reading, MA: Addison-Wesley.

Bajpai, R. K., & Reuss, M. (1981). Evaluation of feeding strategies in carbon-regulated secondary metabolite production through mathematical modelling. *Biotechnology and Bioengineering 13*, 717–738.

Bastin, G., & Dochain, D. (1990). *On line estimation and adaptive control of bioreactors*. Amsterdam: Elsevier.

Benthack, C. (1997). *Feedback-based optimization of a class of constrained nonlinear systems: application to a biofilter*. PhD thesis 1717. Lausanne, Switzerland: Swiss Federal Institute of Technology (EPFL).

Bonvin, D. (1998). Optimal operation of batch reactors—a personal view. *Journal of Process Control 8* (5–6), 355–368.

Bonvin, D., Srinivasan, B., & Ruppen, D. (2002). Dynamic optimization in the batch chemical industry. Proceedings of the CPC-VI Conference, American Institute of Chemical Engineers, Symposium Series, 326 (98), 255–273.

Box, G. E. P., & Draper, N. R. (1987). *Empirical model-building and response surfaces*. New York: John Wiley.

Bryson, A. E. (1999). *Dynamic optimization*. Menlo Park, CA: Addison-Wesley.

Chin, I. S., Lee, K. S., & Lee, J. H. (1999). A unified framework for combined quality and tracking control of batch processes. In *14th IFAC world congress* (Vol. N) (pp. 349–354). Beijing, PR China.

Chu, W. B. Z., & Constantinides, A. (1988). Modeling, optimization, and computer control of cephalosporin C fermentation process. *Biotechnology and Bioengineering 32*, 277–288.

Clarke-Pringle, T. L., & MacGregor, J. F. (1998). Optimization of molecular weight distribution using batch-to-batch adjustments. *Industrial and Engineering Chemistry Research 37*, 3660–3669.

Diwekar, U. M. (1995). *Batch distillation: simulation, optimal design and control*. Washington: Taylor and Francis.

Dong, D., McAvoy, T. J., & Zafiriou, E. (1996). Batch-to-batch optimization using neural networks. *Industrial and Engineering Chemistry Research 35*, 2269–2276.

Doyle, F. J. (1998). Nonlinear inferential control for process applications. *Journal of Process Control 8* (5–6), 339–353.

Eaton, J. W., & Rawlings, J. B. (1990). Feedback control of nonlinear processes using on-line optimization techniques. *Computers and Chemical Engineering 14*, 469–479.

Filippi, C., Greffe, J. L., Bordet, J., Villermaux, J., Barnay, J. L., Ponte, B., & Georgakis, C. (1986). Tendency modeling of semi-batch reactors for optimization and control. *Computers and Chemical Engineering 41*, 913–920.

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

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

François, G., Srinivasan, B., & Bonvin, D. (2002). Run-to-run optimization of batch emulsion polymerization. In *15th IFAC world congress*. Vol. I-Mo-M11. Barcelona, Spain.

Friedland, B. (1986). *Control system design—an introduction to state-space methods*. New York: McGraw-Hill.

Garcia, V., Cabassud, M., Le Lann, M. V., Rigal, L., & Casamatta, G. (1997). 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* (pp. 63–68). Budapest.

Gentric, C., Pla, F., Latifi, M. A., & Corriou, J. P. (1999). Optimization and nonlinear control of a batch emulsion polymerization. *The Chemical Engineering Journal 75* (1), 31–46.

Grossmann, I. E., & Sargent, R. W. H. (1978). Optimum design of chemical plants with uncertain parameters. *American Institute of Chemical Engineering Journal 24*, 1021–1028.

Gygax, R. (1988). Chemical reaction engineering for safety. *Computers and Chemical Engineering 43* (8), 1759–1771.

Halemane, K. P., & Grossmann, I. E. (1983). Optimal process design under uncertainty. *American Institute of Chemical Engineering Journal 29*, 425–433.

Jazwinski, A. H. (1970). *Stochastic processes and filtering theory*. New York: Academic Press.

Joseph, B., & Brosilow, C. (1978). Inferential control of processes. III. Construction of optimal and suboptimal dynamic estimators. *American Institute of Chemical Engineering Journal 24* (3), 500–508.

Kirk, D. E. (1970). *Optimal control theory: an introduction*. London: Prentice-Hall.

Krothapally, M., Bennett, B., Finney, W., & Palanki, S. (1999). Experimental implementation of an on-line optimization scheme to batch PMMA synthesis. *ISA Transactions 38*, 185–198.

Landau, I. D., Lozano, R., & Saad, M. M. (1998). *Adaptive control*. London: Springer-Verlag.

Lee, K. S., Chin, I. S., Lee, H. J., & Lee, J. H. (1999). Model predictive control technique combined with iterative learning for batch processes. *American Institute of Chemical Engineering Journal 45* (10), 2175–2187.

Ljung, L. (1987). *System identification: theory for the user*. Englewood Cliff, NJ: Prentice-Hall.

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

McLennan, F., & Kowalski, B., *Process analytical chemistry* (1995). London: Blackie Academic and Professional.

Meadows, E. S., & Rawlings, J. B. (1991). Model identification and control of a semibatch chemical reactor. In *American control conference* (pp. 249–255). Boston, MA.

Moore, K. L. (1993). *Iterative learning control for deterministic systems*. Advances in Industrial Control, London: Springer-Verlag.

Morari, M., & Zafiriou, E. (1989). *Robust process control*. London: Prentice-Hall.

Nichols, G. D. (1988). *On-line process analyzers*. New York: John Wiley.

Noda, M., Chida, T., Hasebe, S., & Hashimoto, I. (2000). On-line optimization system of pilot scale multi-effect batch distillation system. *Computers and Chemical Engineering 24*, 1577–1583.

Ogunnaike, B. A., & Ray, W. H. (1994). *Process dynamics, modeling and control*. New York: Oxford University Press.

Pai, C. C. D., & Hughes, R. R. (1987). Strategies for formulating and solving two-stage problems for process design under uncertainty. *Computers and Chemical Engineering 11*, 695–706.

Ponnuswamy, S. R., Shah, S. L., & Kiparissides, C. A. (1987). Computer optimal control of batch polymerization reactors. *Industrial and Engineering Chemistry Research 26*, 2229–2236.

Rawlings, J. B., Jerome, N. F., Hamer, J. W., & Bruemmer, T. M. (1989). End-point control in semi-batch chemical reactors. In *IFAC DYCORD+'89* (pp. 339–344). Maastricht, Netherlands.

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

Robertson, D. G., Lee, J. H., & Rawlings, J. B. (1996). A moving-horizon based approach for least squares estimation. *American Institute of Chemical Engineering Journal 42* (8), 2209–2224.

Ruppen, D., Benthack, C., & Bonvin, D. (1995). Optimization of batch reactor operation under parametric uncertainty—computational aspects. *Journal of Process Control 5* (4), 235–240.

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

Sage, A. P., & White, C. C. (1977). *Optimal systems control* (2nd ed.. Englewood Cliffs: Prentice-Hall.

Schenker, B., & Agarwal, M. (2000). On-line optimized feed switching in semi-batch reactors using semi-empirical dynamic models. *Control Engineering Practice 8* (12), 1393–1403.

Soroush, M., & Valluri, S. (1994). An approach to optimization and control of batch processes. In *American control conference* (pp. 490–494). Baltimore, MD.

Srinivasan, B., & Bonvin, D. (2002). Interplay between identification and optimization in run-to-run optimization schemes. In *American control conference* (pp. 2174–2179). Anchorage, Alaska.

Srinivasan, B., Palanki, S., & Bonvin, D. (2002). Dynamic optimization of batch processes: I. Characterization of the nominal solution. *Computers and Chemical Engineering* (X-ref: PII S00981354(02)001175).

Srinivasan, B., Primus, C. J., Bonvin, D., & Ricker, N. L. (2001). Run-to-run optimization via generalized constraint control. *Control Engineering Practice 9*, 911–919.

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

Terwiesch, P., Ravemark, D., Schenker, B., & Rippin, D. W. T. (1998). Semi-batch process optimization under uncertainty: theory and experiments. *Computers and Chemical Engineering 22* (1-2), 201–213.

Ubrich, O., Srinivasan, B., Stoessel, F., & Bonvin, D. (1999). Optimization of a semi-batch reaction system under safety constraints. In *European control conference* (pp. F306.1–F306.6). Karlsruhe, Germany.

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

Welz, C., Srinivasan, B., & Bonvin, D. (2002). Evaluation of measurement-based optimization schemes for batch distillation. In *15th IFAC world congress* (Vol. I-Mo-M11). Barcelona, Spain.

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

Yabuki, Y., & MacGregor, J. F. (1997). Product quality control in semi-batch reactors using mid-course correction policies. In *IFAC ADCHEM'97* (pp. 189–194). Banff, Canada.

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