Available online at www.sciencedirect.com
SCIENCE DIRECT®
Computers and Chemical Engineering 27 (2003) 1–26
Computers & Chemical Engineering
www.elsevier.com/locate/compchemeng

Dynamic optimization of batch processes I. Characterization of the nominal solution

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

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

Received 31 July 2000; received in revised form 22 April 2002; accepted 22 April 2002

Abstract

The optimization of batch processes has attracted attention in recent years because, in the face of growing competition, it is a natural choice for reducing production costs, improving product quality, meeting safety requirements and environmental regulations. This paper starts with a brief overview of the analytical and numerical tools that are available to analyze and compute the optimal solution. The originality of the overview lies in the classification of the various methods. The interpretation of the optimal solution represents the novel element of the paper: the optimal solution is interpreted in terms of constraints and compromises on the one hand, and in terms of path and terminal objectives on the other. This characterization is key to the utilization of measurements in an optimization framework, which will be the subject of the companion paper.

Keywords: Dynamic optimization; Optimal control; Numerical methods; Constraints; Sensitivities; Batch processes; Chemical reactors

1. Introduction

Batch and semi-batch processes are of considerable importance in the fine chemicals industry. A wide variety of specialty chemicals, pharmaceutical products, and certain types of polymers are manufactured in batch operations. Batch processes are typically used when the production volumes are low, when isolation is required for reasons of sterility or safety, and when the materials involved are difficult to handle. With the recent trend in building small flexible plants that are close to the markets, there has been a renewed interest in batch processing (Macchietto, 1998).

1.1. Characteristics of batch processes

In batch operations, all the reactants are charged in a tank initially and processed according to a pre-determined course of action during which no material is added or removed. In semi-batch operations, a reactant may be added with no product removal, or a product may be removed with no reactant addition, or a combination of both. From a process systems point of view, the key feature that differentiates continuous processes from batch and semi-batch processes is that continuous processes have a steady state, whereas batch and semi-batch processes do not (Bonvin, 1998). This paper considers batch and semi-batch processes in the same manner and, thus herein, the term 'batch processes' includes semi-batch processes as well.

Schematically, batch process operations involve the following main steps (Rippin, 1983; Allgor, Barrera, Barton, & Evans, 1996):

- *Elaboration of production recipes:* The chemist investigates the possible synthesis routes in the laboratory. Then, certain recipes are selected that provide the range of concentrations, flowrates or temperatures for the desired reactions or separations to take place and for the batch operation to be feasible. This development step is specific to the product being manufactured (Basu, 1998) and will not be addressed here.

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

0098-1354/02/$ - see front matter © 2002 Elsevier Science Ltd. All rights reserved.
PII: S0098-1354(02)00116-3

- *Production planning, resource allocation, and scheduling:* Once a recipe has been formulated, the next step is to make its operation profitable in the existing plant by allocating the required unit operations to a set of available equipments and by scheduling the individual operations to meet the demand for a set of products. The reader interested in planning and scheduling operations is referred to the following articles (Rippin, 1989; Giritligil, Cesur, & Kuryel, 1998; Ku & Karimi, 1990; Reklaitis, 1995).
- *Safe and efficient production:* This step consists of ensuring the performance of an individual unit or group of units by adjusting the process variables within the ranges provided by the recipes. Optimization is particularly important in order to meet safety (Gygax, 1988; Ubrich, Srinivasan, Stoessel, & Bonvin, 1999; Abel, Helbig, Marquardt, Zwick, & Daszkowski, 2000) and operational constraints (Rawlings, Jerome, Hamer, & Bruemmer, 1989; Ruppen, Bonvin, & Rippin, 1998). Due to the non-steady-state nature of batch processes, the process variables need to be adjusted with time. Hence, this step involves the rather difficult task of determining time-varying profiles through dynamic optimization.

1.2. Dynamic optimization in industry

In the face of increased competition, process optimization provides an unified framework for reducing production costs, meeting safety requirements and environmental regulations, improving product quality, reducing product variability, and ease of scale-up (Mehta, 1983; Bonvin, 1998). From an industrial perspective, the main processing objective is of economic nature and is stated in terms such as return, profitability or payback time of an investment (Lahteemaki, Jutila, & Paasila, 1979; Barrera & Evans, 1989; Friedrich & Perne, 1995).

Though the potential gains of optimization could be significant, there have been only a few attempts to optimize operations through mathematical modeling and optimization techniques. Instead, the recipes developed in the laboratory are implemented conservatively in production, and the operators use heuristics gained from experience to adjust the process periodically, which may lead to slight improvements from batch to batch (Wiederkehr, 1988). The main implications of current industrial practice with respect to optimization are presented in Bonvin, Srinivasan, and Ruppen (2001). The stumbling blocks for the use of mathematical modeling and optimization techniques in industry have been the lack of:

- *Reliable models:* Reliable models have been difficult or too costly to obtain in the fast changing environment of batch processing. Modern software tools such as Aspen Plus, PRO/II, or gPROMs have found wide application to model *continuous* chemical processes (Marquardt, 1996; Pantelides & Britt, 1994). The situation is somewhat different in the batch chemistry. Though batch-specific packages such as Batch Plus, BATCHFRAC, CHEMCAD, BatchCAD, or BaSYS are available, they are not generally applicable. Especially the two important unit operations, reaction and crystallization, still represent a considerable challenge to model at the industrial level.
- *Reliable measurements:* Traditionally, batch processes have been operated with very little instrumentation. The measurements that could possibly compensate model uncertainty have simply not been available. Nevertheless, there is a clear indication that recent advances in sensor technology are helping remove this handicap (McLennan & Kowalski, 1995).

In the authors' opinion, there are two additional reasons for the non-penetration of optimization techniques in the industrial environment:

- *Interpretability of the optimal solution:* Optimization is typically performed using a model of the process, with the optimization routine being considered as a black box. If the resulting optimal solution is not easy to interpret physically, it will be difficult to convince industry to use these optimal profiles.
- *Optimization framework:* The optimization literature is largely model-based, with only limited studies regarding the use of measurements. Due to the large amount of uncertainty (e.g. model mismatch, disturbances) prevailing in industrial settings, there is incentive to use measurements as a way to combat uncertainty. Thus, a framework that would use measurements rather than a model of the process for implementing the optimal solution is needed.

1.3. Goal of the papers

The goal of this series of two papers is twofold. The first objective is to provide a unified view of the methods available to solve dynamic optimization problems. The idea is not to provide a comprehensive survey with details, but rather to show the major directions in which the field has developed. This confers a significant tutorial value to these papers. The first paper deals with the analytical and numerical solution methods, while the second one treats various approaches for optimization under uncertainty. Thus, although the papers expose a fair amount of well-known material, the way this material is presented is clearly original.

The second objective is to investigate the use of measurements as a way to optimize uncertain batch processes. For this purpose, this series of papers

addresses the last two issues mentioned in Section 1.2. The first paper focuses on *interpreting* the various arcs that constitute the optimal solution in terms of the path and terminal objectives of the optimization problem, such as the cost, constraints and sensitivities. This will allow a sound physical interpretation of the optimal solution and will also be key in using measurements for the sake of optimality in uncertain batch processes. The companion paper (Srinivasan, Bonvin, Visser, & Palanki, 2002) addresses the issue of optimization under uncertainty, where a novel approach is presented that uses measurements to meet the necessary conditions of optimality in the presence of uncertainty.

1.4. Organization of the paper

The paper is organised as follows: various problem formulations for the optimization of batch processes are presented in Section 2. The main analytical and numerical solution methods are briefly presented and compared in Sections 3 and 4, respectively. Since these two sections introduce the necessary background material, they can be skipped by the reader familiar with the optimization literature and its terminology. The interpretation of the optimal solution is performed in Section 5 and illustrated through various examples in Section 6. Finally, conclusions are drawn in Section 7.

2. Problem formulations

In batch process operations, the process variables undergo significant changes during the duration of the batch. There is no steady state and thus no constant setpoints around which the key variables can be regulated. Hence, the major objective in batch operations is *not* to keep the system at some optimal constant setpoints, but rather to *optimize* an objective function that expresses the system performance. Optimizing an objective function corresponds to, for example, achieving a desired product quality at the most economical cost, or maximizing the product yield for a given batch time.

The optimization is performed in the presence of constraints. In addition to the dynamic system equations acting as constraints, there might be bounds on the inputs as well as state-dependent constraints. Input constraints are dictated by actuator limitations. For instance, non-negativity of flowrates is a common input constraint. State-dependent constraints typically result from safety and operability considerations such as limits on temperature and concentrations. Terminal constraints normally arise from selectivity or performance considerations. For instance, if multiple reactions occur in a batch reactor, it might be desirable to force the final concentrations of some species below given limits to facilitate or eliminate further downstream processing. Thus, batch optimization problems involve both dynamic and static constraints and fall under the class of *dynamic optimization* problems.

The mathematical formulation of the optimization problem will be stated first. The problem will then be reformulated using Pontryagin's Minimum Principle (PMP) and the principle of optimality of Hamilton–Jacobi–Bellman (HJB). The advantages of one formulation over another depend primarily on the numerical techniques used. Thus, a comparison of the different formulations will be postponed until the discussion of the numerical solution approaches in Section 4.4.

2.1. Direct formulation

Dynamic optimization problems were first posed for aerospace applications in the 1950s. These problems can be formulated mathematically as follows (Lee & Markus, 1967; Kirk, 1970; Bryson & Ho, 1975):

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

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

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

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; $\boldsymbol{S}$ the ζ-dimensional vector of path constraints (which include state constraints and input bounds); $\boldsymbol{T}$ the τ-dimensional vector of terminal constraints; $\boldsymbol{F}$, a smooth vector function; 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)).

The problem formulation (1)–(3) is quite general. Even when an integral cost needs to be considered, e.g. $J = \bar{\phi}(\boldsymbol{x}(t_f)) + \int_0^{t_f} L(\boldsymbol{x}, \boldsymbol{u})\, dt$, where L is a smooth scalar function representing the integral cost, the problem can be converted into the form of Eqs. (1)–(3) by the introduction of the additional state x_{cost}. With $\dot{x}_{cost} = L(\boldsymbol{x}, \boldsymbol{u})$, $x_{cost}(0) = 0$, the terminal cost $J = \bar{\phi}(\boldsymbol{x}(t_f)) + x_{cost}(t_f)$ can be obtained. Also, systems governed by differential-algebraic equations can be formulated in this framework by including the algebraic equations as equality path constraints in Eq. (3). However, the numerial solution can be considerably more complicated for higher index problems.

2.2. Pontryagin's formulation

Using PMP, the problem of optimizing the *scalar* cost functional J in Eqs. (1)–(3) can be reformulated as that of optimizing the Hamiltonian *function* $H(t)$ as follows (Pontryagin, Boltyanskil, Gamkrelidge, & Mishchenko, 1962; Bryson & Ho, 1975):

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

$$\text{s.t.} \quad \dot{\boldsymbol{x}} = \boldsymbol{F}(\boldsymbol{x}, \boldsymbol{u}), \quad \boldsymbol{x}(0) = \boldsymbol{x}_0, \tag{5}$$

$$\dot{\boldsymbol{\lambda}}^T = -\frac{\partial H}{\partial \boldsymbol{x}}, \quad \boldsymbol{\lambda}^T(t_f) = \left.\frac{\partial \phi}{\partial \boldsymbol{x}}\right|_{t_f} + \boldsymbol{\nu}^T \left.\left(\frac{\partial \boldsymbol{T}}{\partial \boldsymbol{x}}\right)\right|_{t_f}, \tag{6}$$

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

where $\boldsymbol{\lambda}(t) \neq 0$ is the n-dimensional vector of adjoint variables (Lagrange multipliers for the system equations), $\boldsymbol{\mu}(t) \geq 0$ the ζ-dimensional vector of Lagrange multipliers for the path constraints, and $\boldsymbol{\nu} \geq 0$ the τ-dimensional vector of Lagrange multipliers for the terminal constraints. The Lagrange multipliers $\boldsymbol{\mu}$ and $\boldsymbol{\nu}$ are non-zero when the corresponding constraints are active and zero otherwise so that $\boldsymbol{\mu}^T \boldsymbol{S}(\boldsymbol{x}, \boldsymbol{u}) = 0$ and $\boldsymbol{\nu}^T \boldsymbol{T}(\boldsymbol{x}(t_f)) = 0$ always. Also note that $\boldsymbol{\mu}^T \boldsymbol{S} = \Sigma_{j=1}^{\zeta} \mu_j S_j = 0$ implies that every term $\mu_j S_j$ of the summation has to be identically equal to zero.

The necessary conditions of optimality are $H_{\boldsymbol{u}} = (\partial H / \partial \boldsymbol{u}) = \boldsymbol{0}$, which implies:

$$\frac{\partial H(t)}{\partial \boldsymbol{u}} = \boldsymbol{\lambda}^T \frac{\partial \boldsymbol{F}}{\partial \boldsymbol{u}} + \boldsymbol{\mu}^T \frac{\partial \boldsymbol{S}}{\partial \boldsymbol{u}} = \boldsymbol{0}. \tag{8}$$

For a free terminal time, an additional condition, referred to as the transversality condition, needs to be satisfied (Kirk, 1970; Bryson & Ho, 1975):

$$H(t_f) = (\boldsymbol{\lambda}^T \boldsymbol{F} + \boldsymbol{\mu}^T \boldsymbol{S})|_{t_f} = 0. \tag{9}$$

Note that the boundary conditions for the state equations (5) and adjoint equations (6) are split, i.e. the *initial* conditions of the state equations and the *terminal* conditions of the adjoint equations are known. Thus, the PMP-formulation leads to a two-point boundary value problem (TPBVP).

2.3. HJB formulation

The HJB formulation uses the principle of optimality to transform the problem of optimizing the *scalar* cost functional J in Eqs. (1)–(3) into the resolution of a partial differential equation (Kirk, 1970; Bryson & Ho, 1975):

$$\frac{\partial V(\boldsymbol{x}, t)}{\partial t} + \min_{\boldsymbol{u}(t)} \left(\frac{\partial V(\boldsymbol{x}, t)}{\partial \boldsymbol{x}} \boldsymbol{F}(\boldsymbol{x}, \boldsymbol{u}) + \boldsymbol{\mu}^T \boldsymbol{S}(\boldsymbol{x}, \boldsymbol{u}) \right) = 0, \tag{10}$$

with the boundary conditions:

$$\left.\frac{\partial V(\boldsymbol{x}, t)}{\partial t}\right|_{t_f} = 0, \tag{11}$$

$$V(\boldsymbol{x}(t_f), t_f) = \phi(\boldsymbol{x}(t_f)) + \boldsymbol{\nu}^T \boldsymbol{T}(\boldsymbol{x}(t_f)), \tag{12}$$

where $V(\boldsymbol{x}, t)$ is the return function or, equivalently, the minimum cost if the system has the states $\boldsymbol{x}$ at time $t \leq t_f$. Eq. (11) is the transversality condition. The link between the PMP and HJB formulations is the fact that the adjoints are the sensitivities of the cost (return function) with respect to the states:

$$\boldsymbol{\lambda}^T = \frac{\partial V}{\partial \boldsymbol{x}}. \tag{13}$$

Thus, the term to be minimized in Eq. (10) is the Hamiltonian H and the partial differential equation (10) represents the dynamics of the adjoints, i.e. Eq. (6):

$$\dot{\boldsymbol{\lambda}}^T = \frac{\mathrm{d}}{\mathrm{d}t}\frac{\partial V}{\partial \boldsymbol{x}} = \frac{\partial}{\partial \boldsymbol{x}}\frac{\partial V}{\partial t} = -\frac{\partial H_{\min}}{\partial \boldsymbol{x}}, \tag{14}$$

where $H_{\min}$ is the minimum value of the Hamiltonian.

3. Analytical solution methods

The solution of the dynamic optimization problem (1)–(3) consists of one or several intervals. The inputs are continuous and differentiable within each interval. The time instants at which the inputs switch from one interval to another are called *switching times*. In this section, analytical expressions for the inputs in each of the intervals are obtained from the necessary conditions of optimality based on PMP. In particular, it is shown that analytical expressions for the inputs can also be obtained in terms of the system states without using the adjoints.

3.1. Adjoint-based computation of the optimal inputs

For the computation of the analytical expressions, the inputs are considered individually. Analytical expression for the optimal input u_i is derived, but the expression may depend on $u_j, j \neq i$. Thus, a set of coupled dynamic equations needs to be solved in order to determine the input vector $\boldsymbol{u}$. The necessary condition of optimality for input u_i is given by:

$$H_{u_i} = \frac{\partial H}{\partial u_i} = \boldsymbol{\lambda}^T \frac{\partial \boldsymbol{F}}{\partial u_i} + \boldsymbol{\mu}^T \frac{\partial \boldsymbol{S}}{\partial u_i} = \boldsymbol{\lambda}^T \boldsymbol{F}_{u_i} + \boldsymbol{\mu}^T \boldsymbol{S}_{u_i} = 0. \tag{15}$$

H_{u_i} has two parts, the system dependent part $\boldsymbol{\lambda}^T \boldsymbol{F}_{u_i}$ and the constraints dependent part $\boldsymbol{\mu}^T \boldsymbol{S}_{u_i}$. It will be shown next how the input u_i can be determined in a given interval from the necessary condition of optimality (15). For this, two solution scenarios have to be considered depending on the value of $\boldsymbol{\lambda}^T \boldsymbol{F}_{u_i}$.[1]

[1] In the literature on optimal control of control-affine systems, the terminology used to distinguish between the two different cases is non-singular vs. singular. This terminology is derived from the singularity of the Hessian matrix $H_{\boldsymbol{uu}}$. Instead, the discussion here focuses on whether or not the inputs are determined by the active path constraints.

- Active path constraints

If $\boldsymbol{\lambda}^{\mathrm{T}}\boldsymbol{F}_{u_i} \neq 0$ in a certain interval, then Eq. (15) requires $\boldsymbol{\mu} \neq \boldsymbol{0}$ in that interval. So, one of the path constraints must be active, and the input u_i can be inferred from the active constraint. For example, when only bounds on the input u_i are considered, i.e. $u_i - u_{i,\max} \leq 0$ and $u_{i,\min} - u_i \leq 0$, then since $\boldsymbol{\mu} \geq \boldsymbol{0}$:

$$u_i = \begin{cases} u_{i,\max} & \text{for } \boldsymbol{\lambda}^{\mathrm{T}}\boldsymbol{F}_{u_i} < 0 \\ ? & \text{for } \boldsymbol{\lambda}^{\mathrm{T}}\boldsymbol{F}_{u_i} = 0. \\ u_{i,\min} & \text{for } \boldsymbol{\lambda}^{\mathrm{T}}\boldsymbol{F}_{u_i} > 0 \end{cases} \tag{16}$$

- Solution inside the feasible region

For the case $\boldsymbol{\lambda}^{\mathrm{T}}\boldsymbol{F}_{u_i} = 0$, it may be possible to express the optimum value of u_i as a function of $\boldsymbol{x}$ and $\boldsymbol{\lambda}$, $u_i(\boldsymbol{x}, \boldsymbol{\lambda})$, directly from that condition and the solution will be inside the feasible region. The problem of Linear Quadratic Regulator (LQR) falls under this category.

However, it often happens that $\boldsymbol{\lambda}^{\mathrm{T}}\boldsymbol{F}_{u_i}$ is independent of u_i as, for example, in the case of control-affine systems with $\boldsymbol{F}(\boldsymbol{x}, \boldsymbol{u}) = \boldsymbol{f}(\boldsymbol{x}) + \boldsymbol{G}(\boldsymbol{x})\boldsymbol{u}$, for which $\boldsymbol{F}_{u_i} = \boldsymbol{G}_i(\boldsymbol{x})$ is independent of $\boldsymbol{u}$. If $\boldsymbol{\lambda}^{\mathrm{T}}\boldsymbol{F}_{u_i} = 0$ and u_i cannot be obtained directly from that condition, the following idea is used. Since $H_{u_i} = 0$ for all t, its time derivatives $(\mathrm{d}^l/\mathrm{d}t^l)H_{u_i} = 0, \ \forall\ l \geq 0$. Differentiating Eq. (15) once with respect to time leads to:

$$\frac{\mathrm{d}H_{u_i}}{\mathrm{d}t} = \dot{\boldsymbol{\lambda}}^{\mathrm{T}}\boldsymbol{F}_{u_i} + \boldsymbol{\lambda}^{\mathrm{T}}\left(\frac{\partial \boldsymbol{F}_{u_i}}{\partial \boldsymbol{x}}\ \dot{\boldsymbol{x}} + \frac{\partial \boldsymbol{F}_{u_i}}{\partial \boldsymbol{u}}\ \dot{\boldsymbol{u}}\right) + \sum_{j=1}^{\zeta}\left(\dot{\mu}_j \frac{\partial S_j}{\partial u_i} + \mu_j \frac{\mathrm{d}}{\mathrm{d}t}\frac{\partial S_j}{\partial u_i}\right) = 0. \tag{17}$$

The last term in Eq. (17) stems from the path constraints. Each individual term in the summation is equal to zero as shown next. From Eq. (15) and $\boldsymbol{\lambda}^{\mathrm{T}}\boldsymbol{F}_{u_i} = 0$, $\boldsymbol{\mu}^{\mathrm{T}}\boldsymbol{S}_{u_i} = 0$ in the interval, which leads to two possibilities for the jth constraint: (i) the constraint $S_j(\boldsymbol{x}, \boldsymbol{u})$ is not active and $\mu_j = 0$; also, $\dot{\mu}_j = 0$ since $\mu_j = 0$ over an interval and, thus, the two terms of the summation are zero; (ii) $S_j(\boldsymbol{x}, \boldsymbol{u})$ is active; this implies $\mu_j \neq 0$ but $\partial S_j/\partial u_i = 0$ to satisfy $\boldsymbol{\mu}^{\mathrm{T}}\boldsymbol{S}_{u_i} = 0$, since every term $\mu_j(\partial S_j/\partial u_i)$ has to be identically equal to zero. Also, $(\mathrm{d}/\mathrm{d}t)(\partial S_j/\partial u_i) = 0$ since $\partial S_j/\partial u_i = 0$ over an interval, and the two terms of the summation are zero. Thus, the last term in Eq. (17) can be dropped. Using Eqs. (5) and (6) for $\dot{\boldsymbol{x}}$ and $\dot{\boldsymbol{\lambda}}$ gives:

$$\frac{\mathrm{d}H_{u_i}}{\mathrm{d}t} = \boldsymbol{\lambda}^{\mathrm{T}}\left(\frac{\partial \boldsymbol{F}_{u_i}}{\partial \boldsymbol{x}}\ \boldsymbol{F} - \frac{\partial \boldsymbol{F}}{\partial \boldsymbol{x}}\ \boldsymbol{F}_{u_i} + \frac{\partial \boldsymbol{F}_{u_i}}{\partial \boldsymbol{u}}\ \dot{\boldsymbol{u}}\right) - \boldsymbol{\mu}^{\mathrm{T}}\frac{\partial \boldsymbol{S}}{\partial \boldsymbol{x}}\ \boldsymbol{F}_{u_i}$$

$$= \boldsymbol{\lambda}^{\mathrm{T}}\Delta \boldsymbol{F}_{u_i} - \boldsymbol{\mu}^{\mathrm{T}}\frac{\partial \boldsymbol{S}}{\partial \boldsymbol{x}}\ \boldsymbol{F}_{u_i} = 0, \tag{18}$$

where the operator Δ is defined as:

$$\Delta \boldsymbol{v} = \frac{\partial \boldsymbol{v}}{\partial \boldsymbol{x}}\ \boldsymbol{F} - \frac{\partial \boldsymbol{F}}{\partial \boldsymbol{x}}\ \boldsymbol{v} + \sum_{k=0}^{\infty}\frac{\partial \boldsymbol{v}}{\partial \boldsymbol{u}^{(k)}}\ \boldsymbol{u}^{(k+1)}, \tag{19}$$

with $\boldsymbol{u}^{(k)}$ representing the kth time differentiation of $\boldsymbol{u}$. A summation is introduced in Eq. (19) since, in general, $\boldsymbol{v}$ is not only a function of $\boldsymbol{u}$ but also of its time derivatives. The operator Δ represents the time differentiation of a vector function along the trajectories of the dynamic system and is studied in the systems literature using tools of Lie algebra (Isidori, 1989).

Continuing in a similar manner, it can be shown that the successive time derivatives of H_{u_i} are given by:

$$\frac{\mathrm{d}^l H_{u_i}}{\mathrm{d}t^l} = \boldsymbol{\lambda}^{\mathrm{T}}\Delta^l \boldsymbol{F}_{u_i} - \boldsymbol{\mu}^{\mathrm{T}}\frac{\partial \boldsymbol{S}}{\partial \boldsymbol{x}}\ \Delta^{l-1}\boldsymbol{F}_{u_i} = 0. \tag{20}$$

Note that H_{u_i} is differentiated further only when $\boldsymbol{\lambda}^{\mathrm{T}}\Delta^{l-1}\boldsymbol{F}_{u_i} = 0$. Also, $\Delta^2 \boldsymbol{v} = \Delta(\Delta \boldsymbol{v})$, etc. The time derivatives inherit the structure of H_{u_i} and have two parts as well, the system dependent part and the constraints dependent part. Time differentiation is repeated until either $\boldsymbol{\lambda}^{\mathrm{T}}\Delta^l \boldsymbol{F}_{u_i} \neq 0$ or u_i appears explicitly in $\boldsymbol{\lambda}^{\mathrm{T}}\Delta^l \boldsymbol{F}_{u_i}$. This gives rise to two intrinsically different solution scenarios that are generalizations of what happens when $\boldsymbol{\lambda}^{\mathrm{T}}\boldsymbol{F}_{u_i} \neq 0$ or u_i appears explicitly in $\boldsymbol{\lambda}^{\mathrm{T}}\boldsymbol{F}_{u_i}$.

- Active path constraints

Let ς_i be the first value of l for which $\boldsymbol{\lambda}^{\mathrm{T}}\Delta^l \boldsymbol{F}_{u_i} \neq 0$. Then, a non-zero $\boldsymbol{\mu}$ is required to satisfy Eq. (20). This implies that at least one of the path constraints is active. To compute the optimal input u_i, the active constraint needs to be differentiated ς_i times. This means that only those constraints that have relative degree $r_{ij} = \varsigma_i$ can be active. Recall that *the relative degree* r_{ij} of the active constraint $S_j(\boldsymbol{x}, \boldsymbol{u})$ with respect to u_i is the number of time differentiations of $S_j(\boldsymbol{x}, \boldsymbol{u})$ that are necessary for the input u_i to appear explicitly (Palanki, Kravaris, & Wang, 1993; Bryson & Ho, 1975). Though different choices of $\boldsymbol{\mu}$ are possible to satisfy Eq. (20), the non-negativity of $\boldsymbol{\mu}$ restricts this choice. Furthermore, since only one of the constraints will be active, i.e. the most restrictive of the possible constraints, $\boldsymbol{\mu}$ will indicate the constraint from which the input u_i can be determined.

- Solution inside the feasible region

Let the order of singularity[2], σ_i, be the first value of l for which the input u_i appears explicitly and independently in $\boldsymbol{\lambda}^{\mathrm{T}}\Delta^l \boldsymbol{F}_{u_i}$. Then, the optimal input u_i can be determined as a function of the states and adjoints,

[2] Some authors use the degree of singularity, s_i, which is the highest time derivative that is still independent of the input u_i. Thus, $s_i = \sigma_i - 1$ (Palanki, Kravaris, & Wang, 1993, 1994).

$u_i(\boldsymbol{x}, \boldsymbol{\lambda})$, from the conditions $\boldsymbol{\lambda}^{\mathrm{T}}\Delta^l \boldsymbol{F}_{u_i} = 0$, for $l = 0, 1, \ldots, \sigma_i$.

Let ρ_i be the dimension of state space that can be reached by manipulating u_i. This means that $(n-\rho_i)$ directions in $\boldsymbol{x}$ are not affected by the input u_i and, conversely, there exist $(n-\rho_i)$ directions in $\boldsymbol{\lambda}$ that do not affect u_i. Also, since the adjoints enter linearly in $\boldsymbol{\lambda}^{\mathrm{T}}\Delta^l \boldsymbol{F}_{u_i} = 0$, as many adjoint variables as there are conditions (i.e. $\sigma_i + 1$) can be eliminated. Thus, among the n adjoint variables, $(n-\rho_i)$ can be eliminated due to the aforementioned independence and $(\sigma_i + 1)$ from the optimality conditions. Thus, the optimal input u_i will depend on $n-(n-\rho_i)-(\sigma_i+1) = (\rho_i - \sigma_i - 1) = \xi_i$ adjoint variables.

The value of ξ_i indicates the number of degrees of freedom that are available in choosing the optimal input. The following classification can be made depending on the value of ξ_i:

- $\xi_i > 0$: The optimal input u_i depends on ξ_i adjoint variables, for the computation of which differential equations need to be solved. Thus, the feedback is *dynamic* in nature.
- $\xi_i = 0$: The optimal input is independent of the adjoint variables. This leads to a feedback that is *static* in nature.
- $-\infty < \xi_i < 0$: This corresponds to the system being constrained to a *surface*, with the relative degree of the surface with respect to u_i being $(-\xi_i)$.
- $\xi_i = -\infty$ $(\sigma_i = \infty)$: If $\rho_i = n$, the input u_i cannot be inside the feasible region (Benthack, 1997). If $\rho_i < n$, then, depending on the cost function, the optimal input u_i is either on the active path constraints or is non-unique (Baumann, 1998).

3.2. Adjoint-free computation of optimal inputs

As seen from Eqs. (15) and (18), the fist-order necessary conditions of optimality are functions of both the system states $\boldsymbol{x}$ and the adjoints $\boldsymbol{\lambda}$. The computation of the optimal inputs is made easier if the adjoint variables can be *eliminated* from the necessary conditions of optimality. Though the adjoints are required to determine the switching instants and the sequence of arcs, an adjoint-free computation of the optimal inputs is possible within the various intervals and is addressed next.

- Active path constraints

When the input u_i is computed from an active path constraint, this part of the optimal solution does not depend on the adjoint variables. Each path constraint $S_j(\boldsymbol{x}, \boldsymbol{u})$ is differentiated along the trajectories of Eq. (2) as illustrated here for the first differentiation:

$$\frac{\mathrm{d}S_j}{\mathrm{d}t} = \frac{\partial S_j}{\partial \boldsymbol{x}}\,\boldsymbol{F} + \frac{\partial S_j}{\partial \boldsymbol{u}}\,\dot{\boldsymbol{u}}. \tag{21}$$

Time differentiation of $S_j(\boldsymbol{x}, \boldsymbol{u})$ is continued until the input u_i appears in $(\mathrm{d}^{r_{ij}} S_j / \mathrm{d}t^{r_{ij}})$. $r_{ij} = \infty$ indicates that the input u_i does not influence the constraint S_j and, thus, u_i cannot be determined from S_j. However, when $r_{ij} < \infty$, the input u_i obtained from $(\mathrm{d}^{r_{ij}} S_j / \mathrm{d}t^{r_{ij}}) = 0$ represents a possible optimal input.

- Solution inside the feasible region

When the optimal solution is inside the feasible region (i.e. no constraint is active), the optimal solution does not depend on the adjoint variables if $\xi_i \leq 0$. To obtain the optimal input independently of $\boldsymbol{\lambda}$ even when $\xi_i > 0$, the following idea is used. Consider the matrix

$$\mathscr{M}_i = [\boldsymbol{F}_{u_i} : \Delta^1 \boldsymbol{F}_{u_i} : \cdots : \Delta^{\rho_i - 1} \boldsymbol{F}_{u_i} : \cdots]. \tag{22}$$

Instead of stopping the differentiation of H_{u_i} when u_i appears explicitly in $\boldsymbol{\lambda}^{\mathrm{T}}\Delta^{\sigma_i}\boldsymbol{F}_{u_i}$, it is continued until the structural rank of $\mathscr{M}_i$ is ρ_i. The input being inside the feasible region corresponds to $\Delta^l \boldsymbol{F}_{u_i} = \boldsymbol{0}$, $\forall\, l \geq 0$. This means that the rank of $\mathscr{M}_i$ is lower than its structural rank. Using this condition, all the adjoint variables can be eliminated at the cost of including the derivatives of u_i up to the order ξ_i. Note that the derivatives of u_i are well defined within the interval, though this may not be true at the switching times.

If $\rho_i = n$, the optimal input u_i is obtained from the condition $\det(\mathscr{M}_i) = 0$. Thus, the system of equations $\det(\mathscr{M}_i) = 0$ replaces the adjoint equations (6). If $\xi_i > 0$, this system of equations is differential in nature since it contains derivatives of u_i up to the order ξ_i. The initial conditions of u_i, $\dot{u}_i$, …, $u_i^{(\xi_i - 1)}$ form ξ_i additional decision variables. Thus, in summary, whether or not the computation is adjoint-free, a system of differential equations of order ξ_i needs to be integrated to compute the optimal input.

If $\rho_i < n$, then, by an appropriate transformation of the states, it can be arranged that only the first ρ_i states of the system are influenced by u_i. In this case, the determinant of the submatrix of $\mathscr{M}_i$ consisting of the first ρ_i rows can be used to compute the optimal input.

The four cases for the value of ξ_i discussed in the preceding subsection can be revisited in the context of adjoint-free computation. For $\xi_i > 0$, $\mathscr{M}_i$ loses rank for a specific combination of $\boldsymbol{x}$, u_i, $\dot{u}_i$, …, $u_i^{\xi_i}$, while for $\xi_i = 0$, the rank loss is for a combination of $\boldsymbol{x}$ and u_i only. For $-\infty < \xi_i < 0$, the rank of $\mathscr{M}_i$ depends only on $\boldsymbol{x}$ and, for $\xi_i = -\infty$, $\mathscr{M}_i$ does not lose rank at all.

The optimal input u_i being inside the feasible region corresponds to physical compromises and tradeoffs that are intrinsic to the system. The absence of intrinsic tradeoffs is represented by the condition $\xi_i = -\infty$ or $\sigma_i = \infty$ and is important for practical applications. This

guarantees that the optimal solution is always on path constraints. This condition is satisfied in controllable linear systems, feedback-linearizable systems, and flat systems, i.e. a large class that encompasses many practical systems (Palanki et al., 1993; Benthack, 1997).

3.3. Limitation of the analytical approach

It has been shown above that the optimal solution possesses the following properties:

- The inputs may be discontinuous; yet, in between discontinuities, the inputs are analytic.
- Two types of intervals are possible between switching instants depending on whether or not the solution is determined by active path constraints; analytical expressions for the inputs can be obtained for each type of intervals.

The main disadvantage of the analytical approach is that it involves symbolic computations that become arduous for high-order systems. So, a *purely* analytical approach cannot be used to determine the optimal solution for most practical problems, except for very simple cases (e.g. problems with $n = 1$ or 2). However, the analytical expressions developed in this section can help parameterize the inputs for computing the numerical solution, as will be described in Section 4.1.3. On the other hand, if the goal of the analysis is primarily to understand the arcs that constitute the optimal solution, it often suffices to work with a simplified (or tendency) model that represents the main physical compromises present in the system.

The adjoint-free approach has additional problems. It provides all possible types of arcs that *might occur* and not those that are *actually present* in the solution. Therefore, though the analysis indicates the possibility of having the solution inside the feasible region, it may happen that, for the optimization problem at hand, the solution is always determined by path constraints. Another disadvantage with the adjoint-free approach is that the *sequence* of intervals that form the optimal solution and the switching times between the various intervals need to be known a priori.

4. Numerical solution methods

Several numerical methods have been proposed in the literature to solve the class of problems described in Section 2. In this section, these methods are classified into three broad categories according to the underlying formulation:

1) Direct optimization methods, where the optimization (1)–(3) is performed directly.
2) PMP-based methods, where the differential–algebraic equations (5)–(9) are solved.
3) HJB-based methods, where the partial differential equation (10)–(12) is solved.

These methods are briefly described below.

4.1. Direct optimization methods

As seen in Section 3, except for some simple cases, a numerical approach is necessary to solve the optimization problem (1)–(3). Since the decision variables $\boldsymbol{u}(t)$ are *infinite dimensional*, the inputs need to be parameterized using a finite set of parameters in order to utilize numerical techniques. Depending on whether the dynamic equations (2) are integrated explicitly or implicitly, two different approaches have been reported in the literature, i.e. the sequential and simultaneous approaches, respectively.

4.1.1. Sequential approach

In this approach, the optimization is carried out in the space of the input variables only. For some parameterization of $\boldsymbol{u}(t)$, the differential equations (2) are integrated using standard integration algorithms and the objective function J is evaluated. This corresponds to a 'feasible' path approach since the differential equations are satisfied at each step of the optimization. A piecewise-constant or piecewise -polynomial approximation of the inputs is often utilized. The basic procedure is as follows:

1) Parameterize the inputs using a finite number of decision variables (typically piecewise polynomials). The vector of decision variables also includes t_f.
2) Choose an initial guess for the decision variables.
3) Integrate the system states to the final time and compute the performance index J and the constraints $\boldsymbol{S}$ and $\boldsymbol{T}$.
4) Use an optimization algorithm (such as steepest descent or Quasi–Newton methods (Gill, Murray, & Wright, 1981)) to update the values of the decision variables. Repeat Steps 3–4 until the objective function is minimized.

If a piecewise-constant approximation over equally-spaced time intervals is made for the inputs, the method is referred to as Control Vector Parameterization (CVP) in the literature (Ray, 1981; Edgar & Himmelblau, 1988; Teo, Goh, & Lim, 1989). This approach has been extended to differential–algebraic systems of index 1 in Vassiliadis, Sargent, and Pantelides (1994a,b). The CVP approach has been utilized in several chemical engineering applications, e.g. reactive distillation (Sargent & Sullivan, 1979; Sorensen, Macchietto, Stuart, & Skogestad, 1996), industrial batch process (Ishikawa,

Natori, Liberis, & Pantelides, 1997), and batch distillation systems (Pollard & Sargent, 1970; Mujtaba & Macchietto, 1997; Furlonge, Pantelides, & Sorensen, 1999).

While the CVP approach is straightforward to implement, it tends to be slow, especially in dealing with inequality path constraints (Bell, Limebeer, & Sargent, 1996). This is mainly due to the fact that this feasible path method requires repeated and expensive solution of the differential equations. Furthermore, the quality of the solution is strongly dependent on the parameterization of the control profile (Logsdon & Biegler, 1989).

4.1.2. Simultaneous approach

The most computationally-intensive part of the sequential approach is Step 3, where the system equations are integrated accurately, even when the decision variables are far from the optimal solution. In the simultaneous approach, an approximation of the system equations is introduced in order to avoid explicit integration for each input profile, thereby reducing the computational burden. The key characteristic of the simultaneous approach is the fact that the optimization is carried out in the full space of discretized inputs *and* states. Thus, in general, the differential equations are satisfied only at the *solution* of the optimization problem (Vassiliadis et al., 1994a). This is therefore called an 'infeasible path' approach. The basic procedure is as follows (Neuman & Sen, 1973; Tsang, Himmelblau, & Edgar, 1975):

1) Parameterize both the inputs and the states using a finite number of decision variables (typically piecewise polynomials). The vector of decision variables also includes t_f.
2) Discretize the differential equations (2), i.e. the differential equations are satisfied only at a finite number of time instants (typically via orthogonal collocation). These two steps transform the dynamic optimization problem (1)–(3) into a standard nonlinear program (NLP).
3) Choose an initial guess for the decision variables.
4) Iteratively solve for the optimal set of decision variables using an NLP code.

Since the above procedure typically leads to a large NLP, efficient numerical methods are necessary to solve this problem (Gill et al., 1981). With the development of Successive Quadratic Programming (SQP), reduced-space SQP, the interior-point approach and the conjugate gradient methods, the NLPs resulting from the simultaneous approach can be solved efficiently (Biegler, 1984; Renfro, Morshedi, & Asbjornsen, 1987; Cervantes & Biegler, 1998; Biegler, Cervantes, & Wächter, 2002). The role of finite elements in terms of node locations and breakpoints in order to account for control profile discontinuities is studied in (Cuthrell & Biegler, 1987, 1989; Logsdon & Biegler, 1989). The simultaneous approach has been utilized in several batch reactor applications (Renfro et al.; Eaton & Rawlings, 1990; Ruppen, Benthack, & Bonvin, 1995).

The use of simultaneous methods requires awareness of the tradeoff between approximation and optimization (Srinivasan, Myszkorowski, & Bonvin, 1995). It could turn out that a less accurate approximation of the integration gives a better cost. Thus, since the objective in Step 4 is merely the optimization of the cost, the solution obtained could correspond to an inadequate state approximation. Improvement of the integration accuracy requires either introducing accuracy as a constraint or increasing the number of collocation points. Especially when the system is stiff, a very fine grid, which translates into a large number of decision variables, is needed (Villadsen & Michelsen, 1978; Terwiesch, Agarwal, & Rippin, 1994).

The direct multiple shooting method (Bock & Platt, 1984) is a hybrid between the sequential and simultaneous methods discussed above. In this approach, the time interval $[0, t_f]$ is divided into P stages. Except for the first stage, the initial conditions of the various stages are considered as decision variables along with continuity constraints stating that the initial states of every stage should match the final ones of the preceding stage. This procedure is an 'infeasible' path method as in simultaneous approaches, while the integration is accurate as in sequential approaches. Extensions of the direct multiple shooting methods to differential–algebraic systems are described in Schulz, Bock, and Steinbach (1998).

4.1.3. Analytical parameterization approach

The piecewise-constant or -polynomial approximations discussed above require a large number of parameters for the solution to be fairly accurate. On the other hand, the most efficient parameterization in terms of the number of parameters, corresponds to the initial conditions of the adjoints $\boldsymbol{\lambda}(0)$, along with the discontinuities in the adjoint variables resulting from the presence of state constraints. It was shown in Section 3.1 that, for each interval, it is possible to obtain analytical expressions for the optimal inputs, i.e. $\boldsymbol{u}(\boldsymbol{x}, \boldsymbol{\lambda})$. Thus, the state and adjoint equations (5)–(6) read:

$$\dot{\boldsymbol{x}} = \boldsymbol{F}(\boldsymbol{x},\ \boldsymbol{u}(\boldsymbol{x},\ \boldsymbol{\lambda})) = \mathscr{F}(\boldsymbol{x},\ \boldsymbol{\lambda}), \quad \boldsymbol{x}(0) = \boldsymbol{x}_0, \tag{23}$$

$$\dot{\boldsymbol{\lambda}}^{\mathrm{T}} = -\frac{\partial H}{\partial \boldsymbol{x}}(\boldsymbol{x},\ \boldsymbol{\lambda}), \quad \boldsymbol{\lambda}^{\mathrm{T}}(t_f) = \left.\frac{\partial \phi}{\partial \boldsymbol{x}}\right|_{t_f} + \boldsymbol{\nu}^{\mathrm{T}} \left.\left(\frac{\partial \boldsymbol{T}}{\partial \boldsymbol{x}}\right)\right|_{t_f}. \tag{24}$$

Note that identical numerical results are obtained if the adjoint variables are scaled by a constant factor. Thus, though there are n initial conditions for the adjoints, one of them can be chosen arbitrarily (e.g.

equal to 1), while the remaining $(n-1)$ components of $\boldsymbol{\lambda}(0)$ become the decision variables. Once the initial conditions $\boldsymbol{\lambda}(0)$ and the possible jumps in $\boldsymbol{\lambda}$ are specified, Eqs. (23) and (24) can in principle be integrated to give $\boldsymbol{\lambda}(t)$. However, this parameterization suffers from numerical problems resulting from integrating the adjoint equations forward in time.

The analytical parameterization approach represents an alternative based on the analytical expressions for the optimal inputs that can be obtained using the adjoint-free approach of Section 3.2. The switching times and the initial conditions of the dynamic feedback are used to completely parameterize the inputs. The advantages of this approach is that it is numerically well conditioned and the parameterization is *exact* and in most cases *parsimonious*.

However, since the proposed parameterization treats each interval separately, the global picture is lost. Thus, the choice of the sequence of intervals needs to be handled separately. In general, a mixed-integer type of algorithm is necessary for this purpose. Typically, an initial sequence of intervals is guessed and the sequence determined iteratively upon checking the necessary conditions of optimality. The basic procedure is summarized below:

1) Choose an initial sequence of intervals.
2) Determine *numerically* the optimal switching times and, possibly, the initial conditions of the dynamic feedback using the sequential approach for the given sequence.
3) Compute the adjoint variables for the resulting optimal solution by integrating Eq. (6) backward in time, and check the necessary conditions of optimality.
4) If these conditions are not satisfied, choose a different sequence of intervals and repeat Steps 2–4 until the necessary conditions are verified.

In Step 3, the validity of the sequence of arcs can be checked using the necessary conditions of optimality since these conditions are satisfied *if and only if* the correct sequence is picked. Note that the problem of choosing a new sequence of arcs (Step 4) is still an open issue. While it is possible to search for all the possible sequences, which is computationally expensive, the physics of the problem often can guide this choice.

The aforementioned procedure is very effective when the solution is determined by constraints, which is the case for many batch processes. In contrast, when applied to problems that have a low-order of singularity (e.g. the linear quadratic problem, $\rho_i = n$, $\sigma_i = 0$), procedure involves integrating $\xi_i = \rho_i - \sigma_i - 1 = (n-1)$ differential equations and choosing $(n-1)$ initial conditions. In this case, the analytical parameterization amounts to choosing the $(n-1)$ initial conditions for the adjoints.

4.2. PMP-based methods

The necessary conditions of optimality (8) are key to the PMP-based methods. On the one hand, they can provide closed-form expressions for the optimal inputs as functions of the state and adjoint variables. On the other hand, the gradient information $\partial H/\partial \boldsymbol{u}$ available from Eq. (8) can be used to generate the search direction in gradient-based schemes.

4.2.1. Shooting method

In the shooting approach (Ray & Szekely, 1973; Bryson, 1999), the optimization problem is cast into that of solving a system of differential–algebraic equations. The optimal inputs are expressed analytically in terms of the states and the adjoints, $\boldsymbol{u}(\boldsymbol{x}, \boldsymbol{\lambda})$. The decision variables include the initial conditions $\boldsymbol{\lambda}(0)$ that are chosen in order to satisfy $\boldsymbol{\lambda}(t_f)$. The basic procedure is as follows:

1) Parameterize $\boldsymbol{\mu}$ using a finite number of variables. The vector of decision variables also includes $\boldsymbol{\lambda}(0)$, $\boldsymbol{v}$ and t_f.
2) Choose an initial guess for the decision variables.
3) Integrate Eqs. (23) and (24) forward in time using $\boldsymbol{x}(0)$, $\boldsymbol{\lambda}(0)$, and compute $\boldsymbol{\lambda}(t_f)$.
4) Check whether Eqs. (24), (7) and (9) are verified; for the terminal conditions $\boldsymbol{\lambda}(t_f)$, the values obtained by integration in Step 4 should match those specified in Eq. (24). Update the decision variables (using for example steepest descent or Quasi–Newton methods (Gill et al., 1981)) and repeat Steps 4–5 until convergence.

The shooting method (Bryson & Ho, 1975; Kirk, 1970), also referred to as boundary condition iteration (BCI) (Jaspan & Coull, 1972), has been used in several batch applications, e.g. free-radical polymerization (Hicks, Mohan, & Ray, 1969; Sacks, Lee, & Biesenberger, 1972), batch bulk polymerization (Chen & Jeng, 1978), batch methyl methacrylate polymerization (Thomas & Kiparissides, 1984), batch fermentation (Chu, 1987), and fed-batch fermentation (Parulekar & Lim, 1985; Lim, Tayeb, Modak, & Bonte, 1986).

There are several difficulties associated with the shooting method (Murthy, Gangiah, & Husain, 1980). Firstly, it can exhibit stability problems in integrating the adjoint equations forward in time. Secondly, unless a good initial guess for the adjoint variables is available (which is rarely the case since the adjoints represent sensitivities), it is computationally expensive to find the optimal solution. Furthermore, the method does not work when there are discontinuities in the adjoints,

which is often the case in the presence of state constraints. Additional degrees of freedom are necessary to handle these situations.

4.2.2. State and adjoint parameterization

Two approaches are discussed next where both the states and the adjoints are parameterized and the analytical expressions for the optimal inputs $\boldsymbol{u}(\boldsymbol{x}, \boldsymbol{\lambda})$ are used.

4.2.2.1. Discretization (NR). This approach uses parameterization and discretization of the states and adjoints (Goh & Teo, 1988). The basic procedure is as follows:

1) Parameterize $\boldsymbol{x}$, $\boldsymbol{\lambda}$ and $\boldsymbol{\mu}$ using a finite number of decision variables (typically piecewise polynomials). The vector of decision variables also includes $\boldsymbol{v}$ and t_f.
2) Discretize the differential equations (23)–(24) for a finite number of time instants (typically via orthogonal collocation). These two steps transform the set of nonlinear differential–algebraic equations (23)–(24) into a set of nonlinear algebraic equations.
3) Choose an initial guess for the decision variables.
4) Iteratively solve for the optimal set of decision variables using, for example, the Newton–Raphson (NR) algorithm (Schwarz, 1989).

4.2.2.2. Quasi-linearization (QL). In this approach, Eqs. (23) and (24) are solved using successive linearization (Bryson & Ho, 1975; Kirk, 1970; Lee, 1968). The basic procedure is as follows:

1) Parameterize $\boldsymbol{x}$, $\boldsymbol{\lambda}$, and $\boldsymbol{\mu}$ using a finite number of decision variables (typically piecewise polynomials). The vector of decision variables also includes $\boldsymbol{v}$ and t_f.
2) Choose an initial guess for the decision variables.
3) Linearize the differential equations (23)–(24) around the current guess. This transforms the set of nonlinear differential–algebraic equations (23)–(24) into a set of linear differential–algebraic equations.
4) Solve the set of linear differential–algebraic equations analytically by appropriate use of transition matrices.
5) Using the solution of Step 4 as the next guess for the decision variables, repeat Steps 3–5 until convergence.

The discretization and quasi-linearization methods work well if the solution is smooth and the unknown boundary conditions are not particularly sensitive to initialization errors. The methods inherit the problems of the simultaneous method regarding the tradeoff between approximation and optimization (Srinivasan et al., 1995). Also, as with the shooting method, a good initial guess for the decision variables is needed for these methods to work well.

4.2.3. Gradient method

Here, the necessary conditions of optimality (8) provides the gradient along which the decision variables can be updated. This approach resembles the sequential approach of the direct formulation except that the gradient is calculated using Eq. (8). The basic procedure is as follows:

1) Parameterize $\boldsymbol{u}$ and $\boldsymbol{\mu}$ using a finite number of variables. The vector of decision variables also includes $\boldsymbol{v}$ and t_f.
2) Choose an initial guess for the decision variables.
3) Integrate the state equations (5) from 0 to t_f.
4) Integrate the adjoint equations (6) backward in time from t_f to 0 and compute the gradient $\partial H/\partial \boldsymbol{u}$ using Eq. (8).
5) Use an optimization algorithm (such as steepest descent or Quasi–Newton methods (Gill et al., 1981)) to update the values of the decision variables. Repeat Steps 3–5 until H is minimized.

The main advantage of the gradient method lies in the fact that a good initial guess for the decision variables is beneficial but not critical to the convergence. This approach has been applied widely to chemical engineering optimization problems (Jaspan & Coull, 1972; Diwekar, 1995; Ramirez, 1997). In control vector iteration (CVI), which is a variant of the gradient method, the input parameterization is not incorporated explicitly in the algorithm (Ray, 1981). However, for any practical implementation of CVI, the inputs need to be parameterized.

4.3. A HJB-based method: dynamic programming

The dynamic programming approach, which utilizes the HJB formulation, is discussed next. The key idea behind dynamic programming is the principle of optimality, i.e. 'parts of an optimal trajectory are also optimal' (Bellman, 1957). This approach is equivalent to computing $V(\boldsymbol{x}, t)$ in Eq. (10) with discretization in both states and time. The minimization in Eq. (10) is performed using exhaustive search. To make the search feasible, the domain has to be restricted. Hence, the inputs are also discretized both in time and amplitude.

The time interval $[0, t_f]$ is divided into P stages, with $[t_{p-1}, t_p]$ being the time interval corresponding to the pth stage. When the terminal time is free, the duration of the stages are additional decision variables for the minimisation using exhaustive research (Bojkov & Luus, 1994). Considering the fact that $(\partial V/\partial t)\Delta t + (\partial V/$

$\partial \boldsymbol{x})(\mathrm{d}\boldsymbol{x}/\mathrm{d}t)\Delta t = V(\boldsymbol{x}(t+\Delta t),\, t+\Delta t) - V(\boldsymbol{x}(t),\, t)$, Eq. (10) can be integrated over the time interval $[t_{p-1},\, t_p]$. Then, the return function at time t_{p-1} can be written as:

$$V(x_{P-1},\ t_{P-1}) = \min_{(t_p - t_{p-1}),\boldsymbol{u}[t_{p-1},\ t_p]} \left(V(\boldsymbol{x}_{p,}\ t_p) + \int_{t_{p-1}}^{t_p} \boldsymbol{\mu}^{\mathrm{T}} \boldsymbol{S}\ \mathrm{d}t \right),\ \boldsymbol{x}(t_{p-1}) \quad (25)$$

$= \boldsymbol{x}_{P-1}$

$\boldsymbol{x}^{d}_{P-1}$ at time t_{P-1}

where $\boldsymbol{x}_p$ is the state at t_p obtained by integrating the system with inputs $\boldsymbol{u}$ and the initial condition $\boldsymbol{x}(t_{P-1}) = \boldsymbol{x}_{P-1}$ over the interval $[t_{p-1},\, t_p]$. Since the boundary condition of V is known at final time, Eq. (25) is solved iteratively for decreasing values of p.

A complication arises from the state discretization since $V(\boldsymbol{x}_p, t_p)$ will only be calculated for a set of discrete values. When integration is performed from a discretization point $\boldsymbol{x}^{\mathrm{d}}{}_{P-1}$ at time t_{P-1}, $\boldsymbol{x}_p$ will typically not correspond to a discretization point. Thus, the question is how to calculate the return function at $\boldsymbol{x}_p$. One option is to interpolate between the return functions at various discretization points at time t_p. An alternative, which will be used here, is to merely use the optimal control $\boldsymbol{u}([t_p, t_f])$ that corresponds to the grid point closest to $\boldsymbol{x}_p$ and integrate the system from t_p to t_f to get the return function. The basic procedure is as follows (Bellman, 1957; Kirk, 1970):

1) Choose the number of stages P.
2) Choose the number of x-grid points, N, and the number of allowable values for each input, M_i, $i = 1, 2, \ldots, m$.
3) Choose a region for each input, $\mathscr{R}_{ip}$, $i = 1, 2, \ldots, m$, and $p = 1, 2, \ldots, P$.
4) Start at the last time stage. For each x-grid point, integrate the state equations from t_{P-1} to t_P for all allowable values of the inputs and determine the values of the inputs that minimize the performance index.
5) Step back one stage (say Stage p). Integrate the state equations from t_{p-1} to t_p for each of the x-grid points and all the allowable values of the inputs. To continue integration from t_p to t_P, choose the optimal inputs from the earlier stages that correspond to the grid point closest to the resulting $\boldsymbol{x}_p$. Compare the values of the cost functions and, for each x-grid point at t_{p-1}, determine the optimal inputs for Stage p.
6) Repeat Step 5 until the initial time t_0 is reached.
7) Reduce the regions $\mathscr{R}_{ip}$ for the allowable input values by using the best input policy as the midpoint for the allowable input values at each stage. Repeat Steps 4–7 until a specified tolerance for the regions is reached.

This approach (Luus & Rosen, 1991; Luus, 1994; Bojkov & Luus, 1994) has been used for the optimization of numerous batch applications, e.g. fed-batch fermentors (Luus, 1992) and semi-batch reactors (Guntern, Keller, & Hungerbühler, 1998). Iterative dynamic programming is compared to the direct sequential approach for the dynamic optimization of a distillation column in Fikar, Latifi, Fournier, and Creff (1998).

The two key advantages of dynamic programming are: (i) it is one of the few methods available for computing the *global* minimum; and (ii) the number of iterations, and thereby the time needed for the optimization, can be estimated a priori (dependent mainly on the tolerance for the $\mathscr{R}_{ip}$ regions). In addition, dynamic programming provides a feedback policy that can be used for on-line implementation: if, due to mismatch in initial conditions, the real trajectory deviates from the predicted optimal one, the optimal inputs that correspond to the x-grid point closest to the real value at a given time instant can be used. The major disadvantage of dynamic programming is its computational complexity, though small-sized problems can be handled efficiently. However, in the presence of constraints, the computational complexity reduces since the constraints limit the search space.

4.4. Classification of numerical optimization schemes

Table 1 classifies the different numerical schemes available for solving dynamic optimization problems according to the underlying problem formulation and the level of parameterization. Typically, the problem is easiest to solve when both the states and the inputs are parameterized (first row in Table 1). When integration of the system equations is used, parameterization of the states can be avoided (second row). When, in addition, analytical expressions derived from the necessary conditions of optimality are used to represent the inputs, both the states and the inputs are continuous (third row). The two empty boxes in the table result from the absence of an analytical solution for the partial differential equation (10)–(12) of the HJB formulation.

The sequential and simultaneous *direct* optimization approaches are by far the methods of choice. Their only disadvantage is that the input parameterization is often chosen arbitrarily by the user. Note that the efficiency of the approach and the accuracy of the solution depend crucially on the way the inputs are parameterized. Though the analytical parameterization approach can be used to alleviate this difficulty, it becomes arduous for large size problems. On the other hand, the numerical methods based on *PMP* are often numerically ill conditioned. Though *dynamic programming* is computationally expensive, it is preferred in certain scenarios due to the fact that the time needed for optimization can be predetermined.

Table 1
Classification of numerical schemes for dynamic optimization

State and input handling	Problem formulation		
	Direct	PMP	HJB
States—parameterized Inputs—parameterized	Simultaneous approach (NLP)	State and adjoint parameterization (NR, QL)	Dynamic programming (DP)
States—continuous Inputs—parameterized	Sequential approach (CVP)	Gradient method (CVI)	–
States—continuous Inputs—continuous	Analytical parameterization approach	Shooting method (BCI)	–

5. Interpretation of the optimal solution

As seen in Section 3, the optimal inputs are either determined by the constraints of the problem or by compromises that are intrinsic to the system. In other words, there are certain time intervals for which the inputs are determined by path constraints, and other intervals where the inputs are inside the feasible region to take advantage of the compromises. This section will attempt to characterize, i.e. understand and interpret, the optimal solution.

A particularity of final-time dynamic optimization problems is the presence of terminal objectives in the form of a cost and constraints. The sequence of arcs and the switching times between them need to consider these terminal objectives. Here again, certain switching times are adapted to satisfy the terminal constraints, while others take advantage of compromises in the system.

The necessary conditions of optimality (7)–(9) can be rewritten in partitioned form as:

$$\begin{array}{lll} & \text{Path} & \text{Terminal} \\ \text{Constraints} & \boldsymbol{\mu}^{\mathrm{T}}\boldsymbol{S}(\boldsymbol{x},\ \boldsymbol{u}) = 0 & \boldsymbol{v}^{\mathrm{T}}\boldsymbol{T}(\boldsymbol{x}(t_{\mathrm{f}})) = 0 \\ \text{Sensitivities} & \boldsymbol{\lambda}^{\mathrm{T}}(\partial \boldsymbol{F}/\partial \boldsymbol{u}) + \boldsymbol{\mu}^{\mathrm{T}}(\partial \boldsymbol{S}/\partial \boldsymbol{u}) = \boldsymbol{0} & \boldsymbol{\lambda}^{\mathrm{T}}(t_{\mathrm{f}}) - (\partial\phi/\partial \boldsymbol{x})|_{t_{\mathrm{f}}} - \boldsymbol{v}^{\mathrm{T}}(\partial \boldsymbol{T}/\partial \boldsymbol{x})|_{t_{\mathrm{f}}} = \boldsymbol{0},\ \ H(t_{\mathrm{f}}) = 0 \end{array} \tag{26}$$

The following observations can be made:

- The necessary conditions of optimality have two parts: (i) the constraint part (first row of Eq. (26)); and (ii) the sensitivity part (second row of Eq. (26)).
- Both the constraint and sensitivity parts have two elements: (i) the path elements corresponding to quantities during the run (first column of Eq. (26)); and (ii) the terminal elements related to quantities at the end of the run (second column of Eq. (26)).

As a result, a characterization of the optimal solution will be proposed that: (i) treats the path and terminal objectives independently; and (ii) separates *constraint*- from *sensitivity-seeking* decision variables. The path objectives correspond to either being on path constraints or following inputs that force the path sensitivities to zero, while the terminal objectives correspond to either meeting terminal constraints or optimizing the terminal cost. The constraint-seeking decision variables are those that push the system to the (path and terminal) constraints of the problem, while sensitivity-seeking decision variables exploit the intrinsic compromises present in the system for optimizing the cost. The separation of constraint- and sensitivity-seeking variables has also been studied in the context of numerical optimization (Wolbert, Joulia, Koehret, & Biegler, 1994)

5.1. Separation of path and terminal objectives

The objective of this subsection is to partition the optimal inputs into: (i) time-dependent values or arcs, $\boldsymbol{\eta}(t)$, that cater to *path* objectives; and (ii) scalar values or parameters, $\boldsymbol{\pi}$, that typically consist of switching instants and handle *terminal* objectives. For this purpose, it is necessary to know the structure of the optimal solution, i.e. (i) the types of arcs; (ii) the sequence of arcs; and (iii) the active terminal constraints. These can be determined either via the educated guess of an experienced operator or by visual inspection of the solution obtained from numerical optimization. Each

interval is tagged according to the type it could represent. The analytical expressions for the inputs can be used for verification but are typically not needed here.

5.1.1. Meeting path objectives

Path objectives correspond to tracking the active path constraints and forcing the path sensitivities to zero. These objectives are achieved through adjustment of the inputs in the various arcs $\boldsymbol{\eta}(t)$ with the help of appropriate controllers, as will be discussed in the companion paper (Srinivasan et al., 2002). Also, among the switching instants, a few correspond to reaching the path constraints in minimum time. Thus, these switching instants are also considered as a part of $\boldsymbol{\eta}(t)$. The effect of any deviation in these switching instants will be corrected by the controllers that keep the corresponding path objectives active.

5.1.2. Meeting terminal objectives

Upon meeting the path objectives, the optimal inputs still have residual degrees of freedom that will be used to meet the terminal objectives, i.e. satisfying terminal constraints and optimizing the terminal cost. These input parameters $\boldsymbol{\pi}$ include certain switching times and additional decision variables (e.g. the initial conditions of the inputs as described in Section 3.2).

Upon meeting the path objectives, the optimization problem reduces to that of minimizing a terminal cost subject to terminal constraints *only*. Let the inputs be represented by $\boldsymbol{u}(\boldsymbol{\pi}, \boldsymbol{x}, t)$. Then, the optimization problem (1)–(3) can be rewritten as:

$$\min_{\boldsymbol{\pi}} \; J = \phi(\boldsymbol{x}(t_f)), \tag{27}$$

$$\text{s.t.} \quad \dot{\boldsymbol{x}} = \boldsymbol{F}(\boldsymbol{x}, \boldsymbol{u}(\boldsymbol{\pi}, \boldsymbol{x}, t)), \quad \boldsymbol{x}(0) = \boldsymbol{x}_0, \tag{28}$$

$$\boldsymbol{T}(\boldsymbol{x}(t_f)) \le \boldsymbol{0}. \tag{29}$$

The necessary conditions of optimality for Eqs. (27)–(29) are:

$$\boldsymbol{v}^{\mathrm{T}} \boldsymbol{T}(\boldsymbol{x}(t_f)) = 0 \quad \text{and} \quad \frac{\partial \phi}{\partial \boldsymbol{\pi}} + \boldsymbol{v}^{\mathrm{T}} \frac{\partial \boldsymbol{T}}{\partial \boldsymbol{\pi}} = \boldsymbol{0}. \tag{30}$$

Let $\bar{\tau}$ be the number of active terminal constraints. The number of decision variables arising from the aforementioned input parameterization, n_π, needs to satisfy $n_\pi \ge \bar{\tau}$ in order to be able to meet all the active terminal constraints. Note that n_π is finite.

5.2. Separation of constraint- and sensitivity-seeking decision variables

This subsection deals with the separation of the decision variables according to the nature of the objectives (*constraints* vs. *sensitivities*). This separation should be done for both $\boldsymbol{\eta}(t)$ and $\boldsymbol{\pi}$.

5.2.1. Separation of constraint- and sensitivity-seeking input directions $\boldsymbol{\eta}(t)$

In each interval, some of the path constraints may be active. If there are active path constraints, the inputs or combinations of inputs that push the system to the path constraints can be separated from those combinations that have no effect on meeting the path constraints. Let $\bar{\zeta}$ be the number of active path constraints in a given interval. Clearly, $\bar{\zeta} \le m$. In the single input case, and in the extreme cases $\bar{\zeta} = 0$ and $\bar{\zeta} m$, this problem of separation does not arise. In the other cases, the idea is to use a transformation, $\boldsymbol{\eta}(t)^{\mathrm{T}} \rightarrow [\bar{\boldsymbol{\eta}}(t)^{\mathrm{T}} \; \tilde{\boldsymbol{\eta}}(t)^{\mathrm{T}}]$, such that $\bar{\boldsymbol{\eta}}(t)$ is a $\bar{\zeta}$-dimensional vector that has a handle on meeting the path constraints and $\tilde{\boldsymbol{\eta}}(t)$ is a vector of dimension $(m - \bar{\zeta})$ that does not affect the path constraints, but the sensitivities instead. Thus, $\bar{\boldsymbol{\eta}}(t)$ are referred to as the constraint-seeking input directions, and $\tilde{\boldsymbol{\eta}}(t)$ as the sensitivity-seeking input directions.

Let $\bar{\boldsymbol{S}}(\boldsymbol{x}, \boldsymbol{u})$ denote the active constraints and $\bar{\boldsymbol{\mu}}$ the corresponding Lagrange multipliers. Let r_j be the relative degree of the constraint $\bar{S}_j(\boldsymbol{x}, \boldsymbol{u}) = 0$ with respect to the input that is determined from it. The directions $\bar{\boldsymbol{\eta}}(t)$ and $\tilde{\boldsymbol{\eta}}(t)$ can be computed using the matrix $\mathbf{G}_S = [\{(\partial/\partial \boldsymbol{u})(\mathrm{d}^{r_1} \bar{S}_1/\mathrm{d}t^{r_1})\} \; \{(\partial/\partial \boldsymbol{u})(\mathrm{d}^{r_2} \bar{S}_2/\mathrm{d}t^{r_2})\} \; \cdots]^T$. The singular value decomposition gives $\mathbf{G}_S = \mathbf{U}_S \boldsymbol{\Sigma}_S \mathbf{V}_S^{\mathrm{T}}$, where $\mathbf{U}_S$ has dimension $\bar{\zeta} \times \bar{\zeta}$, $\boldsymbol{\Sigma}_S$ has dimension $\bar{\zeta} \times m$ and $\mathbf{V}_S$ has dimension $m \times m$. The matrices $\mathbf{U}_S$, $\boldsymbol{\Sigma}_S$, and $\mathbf{V}_S$ can be partitioned into:

$$\mathbf{U}_S = [\bar{\mathbf{U}}_S \; \tilde{\mathbf{U}}_S], \quad \boldsymbol{\Sigma}_S = \begin{bmatrix} \bar{\boldsymbol{\Sigma}}_S & \mathbf{0} \\ \mathbf{0} & \mathbf{0} \end{bmatrix}, \quad \mathbf{V}_S = [\bar{\mathbf{V}}_S \; \tilde{\mathbf{V}}_S], \tag{31}$$

where $\bar{\mathbf{U}}_S$ and $\bar{\mathbf{V}}_S$ correspond to the first $\bar{\zeta}$ columns of their respective matrices and $\tilde{\mathbf{U}}_S$ and $\tilde{\mathbf{V}}_S$ to the remaining columns. $\bar{\boldsymbol{\Sigma}}_S$ is the $\bar{\zeta} \times \bar{\zeta}$ submatrix of $\boldsymbol{\Sigma}_S$. Due to the structure of $\boldsymbol{\Sigma}_S$, $\mathbf{G}_S = \bar{\mathbf{U}}_S \bar{\boldsymbol{\Sigma}}_S \bar{\mathbf{V}}_S^{\mathrm{T}}$. $\tilde{\mathbf{V}}_S$ is of dimension $m \times (m - \bar{\zeta})$ and corresponds to the input directions that do not affect the constraints. Thus, the constraint- and sensitivity-seeking directions are defined as: $\bar{\boldsymbol{\eta}}(t) = \bar{\mathbf{V}}_S^{\mathrm{T}} \boldsymbol{\eta}(t)$ and $\tilde{\boldsymbol{\eta}}(t) = \tilde{\mathbf{V}}_S^{\mathrm{T}} \boldsymbol{\eta}(t)$. Note that $\bar{\boldsymbol{\eta}}(t)$ is a combination of all inputs that have the same relative degree with respect to the active constraints $\bar{\boldsymbol{S}}$. The directions $\tilde{\boldsymbol{\eta}}(t)$ are orthogonal to the directions $\bar{\boldsymbol{\eta}}(t)$. Also, for the sensitivity-seeking input directions, this construction guarantees that the vector $(\partial/\partial \bar{\boldsymbol{\eta}})(\mathrm{d}^k \bar{S}_j/\mathrm{d}t^k) = \boldsymbol{0}$ for $k = 0, 1, \ldots, r_j$. The transformation $\boldsymbol{\eta}^{\mathrm{T}} \rightarrow [\bar{\boldsymbol{\eta}}^{\mathrm{T}} \; \tilde{\boldsymbol{\eta}}^{\mathrm{T}}]$ is, in general, state dependent and can be obtained analytically if piecewise analytical expressions for the optimal inputs are available (see Section 3). Otherwise, a numerical analysis is necessary to obtain this transformation.

With the proposed transformation, the necessary conditions of optimality for the path objectives are:

$$\bar{\boldsymbol{S}} = \mathbf{0}, \quad \frac{\partial H}{\partial \tilde{\boldsymbol{\eta}}} = \boldsymbol{\lambda}^{\mathrm{T}} \frac{\partial \boldsymbol{F}}{\partial \tilde{\boldsymbol{\eta}}} = \mathbf{0}, \qquad \frac{\partial H}{\partial \bar{\boldsymbol{\eta}}} = \boldsymbol{\lambda}^{\mathrm{T}} \frac{\partial \boldsymbol{F}}{\partial \bar{\boldsymbol{\eta}}} + \bar{\boldsymbol{\mu}}^{\mathrm{T}} \frac{\partial \bar{\boldsymbol{S}}}{\partial \bar{\boldsymbol{\eta}}} = \mathbf{0}. \tag{32}$$

Thus, the optimal values along the constraint-seeking directions are determined by the active path constraints $\bar{\boldsymbol{S}} = \mathbf{0}$, whilst the optimal values along the sensitivity-seeking directions are determined from the sensitivity conditions $\boldsymbol{\lambda}^{\mathrm{T}}(\partial \boldsymbol{F}/\partial \tilde{\boldsymbol{\eta}}) = \mathbf{0}$. The third condition in Eq. (32) determines the value of $\bar{\boldsymbol{\mu}}$. In fact, the advantage of separating the constraint-seeking from the sensitivity-seeking input directions is that the necessary conditions of optimality can be derived without the knowledge of the Lagrange multiplier $\bar{\boldsymbol{\mu}}$.

5.2.2. Separation of constraint- and sensitivity-seeking input parameters $\boldsymbol{\pi}$

In the input parameter vector $\boldsymbol{\pi}$, there are elements whose variations affect the active terminal constraints, $\bar{\boldsymbol{T}}$, and others that do not. The idea is then to separate the two using a transformation, $\boldsymbol{\pi}^{\mathrm{T}} \rightarrow [\bar{\boldsymbol{\pi}}^{\mathrm{T}}\ \tilde{\boldsymbol{\pi}}^{\mathrm{T}}]$, such that $\bar{\boldsymbol{\pi}}$ is a $\bar{\tau}$-dimensional vector and $\tilde{\boldsymbol{\pi}}$ is of dimension $(n_{\pi} - \bar{\tau})$. Similar to the classification of the input directions, $\bar{\boldsymbol{\pi}}$ are referred to as the constraint-seeking input parameters (with a handle on meeting terminal constraints) and $\tilde{\boldsymbol{\pi}}$ as the sensitivity-seeking input parameters (which are of no help in meeting terminal constraints but will affect the sensitivities).

Similar to the input directions, the constraint- and sensitivity-seeking input parameters can be obtained using the matrix $\mathbf{G}_T = \partial \bar{\boldsymbol{T}}/\partial \boldsymbol{\pi}$. The singular value decomposition gives $\mathbf{G}_T = \mathbf{U}_T \boldsymbol{\Sigma}_T \mathbf{V}_T^{\mathrm{T}}$, where $\mathbf{U}_T$ has dimension $\bar{\tau} \times \bar{\tau}$, $\boldsymbol{\Sigma}_T$ has dimension $\bar{\tau} \times n_{\pi}$ and $\mathbf{V}_T$ has dimension $n_{\pi} \times n_{\pi}$. The matrices $\mathbf{U}_T$, $\boldsymbol{\Sigma}_T$, and $\mathbf{V}_T$ can be partitioned into:

$$\mathbf{U}_T = [\bar{\mathbf{U}}_T\ \tilde{\mathbf{U}}_T], \quad \boldsymbol{\Sigma}_T = \begin{bmatrix} \bar{\Sigma}_T & \mathbf{0} \\ \mathbf{0} & \mathbf{0} \end{bmatrix}, \quad \mathbf{V}_T = [\bar{\mathbf{V}}_T,\ \tilde{\mathbf{V}}_T], \tag{33}$$

where $\bar{\mathbf{U}}_T$ and $\bar{\mathbf{V}}_T$ correspond to the first $\bar{\tau}$ columns of their respective matrices and $\tilde{\mathbf{U}}_T$ and $\tilde{\mathbf{V}}_T$ to the remaining columns. The constraint- and sensitivity-seeking parameters can be defined as: $\bar{\boldsymbol{\pi}} = \bar{\mathbf{V}}_T^{\mathrm{T}} \boldsymbol{\pi}$ and $\tilde{\boldsymbol{\pi}} = \tilde{\mathbf{V}}_T^{\mathrm{T}} \boldsymbol{\pi}$. This construction guarantees $\partial \bar{\boldsymbol{T}}/\partial \tilde{\boldsymbol{\pi}} = \mathbf{0}$. Since analytical expressions for $\partial \bar{\boldsymbol{T}}/\partial \boldsymbol{\pi}$ are not available in most cases, this transformation is computed numerically. Though this transformation is in general nonlinear, a linear approximation can always be found in the neighborhood of the optimum. This approach was used in François, Srinivasan, and Bonvin (2002) for the run-to-run optimization of batch emulsion polymerization.

Using this transformation, the necessary conditions of optimality (30) can be rewritten as:

$$\bar{\boldsymbol{T}} = \mathbf{0}, \quad \frac{\partial \phi}{\partial \tilde{\boldsymbol{\pi}}} = \mathbf{0}, \quad \frac{\partial \phi}{\partial \bar{\boldsymbol{\pi}}} + \bar{\boldsymbol{\nu}}^{\mathrm{T}} \frac{\partial \bar{\boldsymbol{T}}}{\partial \bar{\boldsymbol{\pi}}} = \mathbf{0}. \tag{34}$$

Thus, the active constraints $\bar{\boldsymbol{T}} = \mathbf{0}$ determine the optimal values of the constraint-seeking input parameters, whilst the optimal values of the sensitivity-seeking input parameters are determined from the sensitivity conditions $\partial \phi/\partial \tilde{\boldsymbol{\pi}} = 0$. The Lagrange multipliers $\bar{\boldsymbol{\nu}}$ are calculated from $(\partial \phi/\partial \bar{\boldsymbol{\pi}}) + \bar{\boldsymbol{\nu}}^{\mathrm{T}}(\partial \boldsymbol{T}/\partial \bar{\boldsymbol{\pi}}) = \mathbf{0}$.

5.3. Reasons for interpreting the optimal solution

The interpretation of the optimal solution described in this section has several advantages that will be addressed next.

5.3.1. Physical insight

The practitioner likes to be able to relate the various arcs forming the optimal solution to the physics of his problem, i.e. the cost to be optimized and the path and terminal constraints. This knowledge is key towards the acceptability of the resulting optimal solution in industry.

5.3.2. Numerical efficiency

The efficiency of numerical methods for solving dynamic optimization problems characterized by a discontinuous solution depends strongly on the parameterization of the inputs. Thus, any parametrization that is close to the physics of the problem will tend to be fairly parsimonious and adapted to the problem at hand. This advantage is most important for the class of problems where the solution is determined by the constraints, a category, that encompasses most batch processes.

5.3.3. Simplified necessary conditions of optimality

With the introduction of $\bar{\boldsymbol{S}}$, $\bar{\boldsymbol{T}}$, $\bar{\boldsymbol{\eta}}$, $\tilde{\boldsymbol{\eta}}$, $\bar{\boldsymbol{\pi}}$ and $\tilde{\boldsymbol{\pi}}$, the necessary conditions of optimality reduce to:

	Path	Terminal
Constraints	$\bar{\boldsymbol{S}}(\boldsymbol{x}, \boldsymbol{u}) = \mathbf{0}$	$\bar{\boldsymbol{T}}(\boldsymbol{x}(t_{\mathrm{f}})) = \mathbf{0}$
Sensitivities	$\boldsymbol{\lambda}^{\mathrm{T}}(\partial \boldsymbol{F}/\partial \tilde{\boldsymbol{\eta}}) = \mathbf{0}$	$\partial \phi/\partial \tilde{\boldsymbol{\pi}} = \mathbf{0}$

(35)

The optimal values along the constraint-seeking directions, $\bar{\boldsymbol{\eta}}^*(t)$, are determined by the active path constraints $\bar{\boldsymbol{S}} = \mathbf{0}$, whilst $\tilde{\boldsymbol{\eta}}^*(t)$ are determined from the sensitivity conditions $\boldsymbol{\lambda}^{\mathrm{T}}(\partial \boldsymbol{F}/\partial \tilde{\boldsymbol{\eta}}) = \mathbf{0}$. On the other hand, the active terminal constraints $\bar{\boldsymbol{T}} = \mathbf{0}$ determine the optimal values of the constraint-seeking parameters, $\bar{\boldsymbol{\pi}}^*$, whilst $\tilde{\boldsymbol{\pi}}^*$ are determined from the sensitivity conditions $\partial \phi/\partial \tilde{\boldsymbol{\pi}} = \mathbf{0}$. This idea can be used to incorporate measurements into the optimization framework so as to combat uncertainty, which will be the subject of the companion paper (Srinivasan et al., 2002).

5.3.4. Variations in cost

Though the necessary conditions of optimality have four parts as in Eq. (35), each part has a different effect on the cost. Often, active constraints have a much larger influence on the cost than sensitivities do. Thus, separating constraint- and sensitivity-seeking decision variables reveals where most of the optimization potential lies.

The Lagrange multipliers $\boldsymbol{\mu}$ and $\boldsymbol{\nu}$ capture the deviations in cost resulting from the path and terminal constraints not being active so that, to a first-order approximation, $\delta J = \int_0^{t_f} \boldsymbol{\mu}^{\mathrm{T}}\, \delta \boldsymbol{S}\, \mathrm{d}t + \boldsymbol{\nu}^{\mathrm{T}}\, \delta \boldsymbol{T}$. On the other hand, if the inputs are inside the feasible region, the first-order approximation of the cost deviation is zero, $\delta J = (H_{\boldsymbol{u}} S_{\boldsymbol{u}}) = \boldsymbol{0}$, since by definition $H_u = \boldsymbol{0}$. Thus, the loss in performance due to non-optimal inputs is often less important in a sensitivity-seeking arc than in a constraint-determined arc. Thus, when implementing an optimal control policy, care should be taken to keep the constraints active since this often corresponds to a large gain in performance.

The second-order approximation of the deviation in performance gives $\delta J = (1/2)\delta \boldsymbol{u}^{\mathrm{T}} \boldsymbol{H}_{\boldsymbol{uu}} \delta \boldsymbol{u}$. If $\boldsymbol{H}_{\boldsymbol{uu}} \neq \boldsymbol{0}$, the loss could still be significant. However, if $\boldsymbol{H}_{\boldsymbol{uu}} = \boldsymbol{0}$, i.e. for an order of singularity $\sigma > 0$, then small deviations of $\boldsymbol{u}$ from the optimal trajectory will result in negligibly small loss in cost. This negligible effect of input variations on the cost can also be attributed to the loss of state controllability.

6. Examples

This section presents the optimal solution for several qualitatively different examples. The emphasis will be on characterizing the optimal solution by determining those parts of the optimal solution that push the system towards constraints and those parts that seek to reduce the sensitivities. Also, a clear distinction will be made between path and terminal objectives. The reason for choosing four examples (instead of only one) is to illustrate the various features that an optimal solution might exhibit. These features are indicated in Table 2.

In every example, the following approach is used: (i) a numerical solution is first obtained using the direct sequential method and piecewise-constant parameterization of the input; (ii) the different arcs in the solution are interpreted in terms of satisfying path and terminal objectives; (iii) with the knowledge of the sequence of arcs, the analytical parameterization approach is used to get an exact solution. This last step is not always necessary, and may not even be appropriate for large problems. Nevertheless, the analytical expressions are provided for all examples here since they provide valuable insight into the solution.

In the sequel, the subscripts $(\cdot)_{\mathrm{des}}$, $(\cdot)_{\mathrm{min}}$, $(\cdot)_{\mathrm{max}}$, $(\cdot)_{\mathrm{o}}$, and $(\cdot)_{\mathrm{f}}$ represent desired, minimum, maximum, initial, and final values, respectively. u_{sens} will be used to represent a sensitivity-seeking input inside the feasible region, and u_{path} an input that keeps a path constraint active.

6.1. Isothermal semi-batch reactor with a safety constraint (Ubrich et al., 1999)

6.1.1. Description of the reaction system

- *Reaction:* $A + B \rightarrow C$.
- *Conditions:* Semi-batch, exothermic, isothermal.
- *Objective:* Minimize the time needed to produce a given amount of C.
- *Manipulated variable:* Feed rate of B.
- *Constraints:* Input bounds; constraint on the maximum temperature reached under cooling failure; constraint on the maximum volume.
- *Comments:* In the case of a cooling failure, the system becomes adiabatic. The best strategy is to immediately stop the feed. Yet, due to the presence of unreacted components in the reactor, the reaction goes on. Thus, chemical heat will be released, which causes an increase in temperature. The maximum attainable temperature under cooling failure is given by:

$$T_{\mathrm{cf}}(t) = T(t) + \min(c_A(t),\ c_B(t)) \frac{(-\Delta H)}{\rho c_{\mathrm{p}}}, \tag{36}$$

where the variables and parameters are described in Section 6.1.2, and the term $\min(c_A, c_B)$ serves to calculate the maximum extent of reaction that could

Table 2
Features present in the various examples

#	Example	Path constraints	Terminal constraints	Sensitivity-seeking arc	Number of inputs	Terminal time
1	Reactor with a safety constraint	Yes	Yes	No	1	Free
2	Bioreactor with inhibition and a biomass constraint	Yes	No	Yes	1	Fixed
3	Reactor with parallel reactions and selectivity constraints	No	Yes	Yes	1	Fixed
4	Non-isothermal reactor with series reactions and a heat removal constraint	Yes	Yes	Yes	2	Fixed

occur following the failure.Without any constraints, optimal operation would simply consist of adding all the available B at initial time (i.e. batch operation). However, because of the safety constraint, the feeding of B has to account for the possible cooling failure. Once the volume constraint is attained, the feed rate is set to zero.

6.1.2. Problem formulation

6.1.2.1. Variables and parameters. c_X, concentration of species X; n_X, number of moles of species X; V, reactor volume; u, feed rate of B; $c_{B\text{in}}$, inlet concentration of B; k, kinetic parameter; T, reactor temperature; T_{cf}, temperature under cooling failure; ΔH, reaction enthalpy; ρ, density; and c_{p}, heat capacity.

6.1.2.2. Model equations.

$$\dot{c}_A = -kc_A c_B - \frac{u}{V}\, c_A \quad c_A(0) = c_{A\text{o}}, \tag{37}$$

$$\dot{c}_B = -kc_A c_B + \frac{u}{V}(c_{B\text{in}} - c_B) \quad c_B(0) = c_{B\text{o}}, \tag{38}$$

$$\dot{V} = u \quad V(0) = V_{\text{o}}. \tag{39}$$

The concentration of C is given by:

$$c_C = \frac{c_{A\text{o}} V_{\text{o}} + c_{C\text{o}} V_{\text{o}} - c_A V}{V}. \tag{40}$$

The numerical values are given in Table 3.

6.1.2.3. Model reduction. The dynamic model (37)–(39) can be reduced since the three differential equations are linearly dependent, as shown next. The balance equations for various species and total mass read:

$$\dot{n}_A = -kc_A c_B V \quad n_A(0) = n_{A\text{o}}, \tag{41}$$

$$\dot{n}_B = -kc_A c_B V + c_{B\text{in}} u \quad n_B(0) = n_{B\text{o}}, \tag{42}$$

Table 3
Model parameters, operating bounds and initial conditions for Example 1

k	0.0482	l/mol h
T	70	°C
ΔH	−60 000	J/mol
ρ	900	g/l
c_p	4.2	J/gK
$c_{B\text{in}}$	2	mol/l
$u_{\min}$	0	l/h
$u_{\max}$	0.1	l/h
$T_{\max}$	80	°C
$V_{\max}$	1	l
$n_{C\text{des}}$	0.6	mol
$c_{A\text{o}}$	2	mol/l
$c_{B\text{o}}$	0.63	mol/l
V_{o}	0.7	l

$$\dot{V} = u, \quad V(0) = V_{\text{o}}. \tag{43}$$

Eq. (42) can be expressed in terms of Eqs. (41) and (43):

$$\dot{n}_B = \dot{n}_A + c_{B\text{in}}\dot{V} \Rightarrow \frac{\text{d}}{\text{d}t}(n_B - n_A - Vc_{B\text{in}}) = 0, \tag{44}$$

indicating that $I = n_B - n_A - Vc_{B\text{in}} = V(c_B - c_A - c_{B\text{in}})$ is a reaction invariant (Srinivasan, Amrhein, & Bonvin, 1998). Integration of Eq. (44) from 0 to t allows expressing c_B in terms of other states and initial conditions:

$$c_B = \frac{(c_{B\text{o}} - c_{A\text{o}} - c_{B\text{in}})V_{\text{o}} + (c_A + c_{B\text{in}})V}{V}. \tag{45}$$

6.1.2.4. Optimization problem.

$$\min_{t_{\text{f}}, u(t)} J = t_{\text{f}},$$

s.t. (36), (40), (41), (43), (45)

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

$$T_{\text{cf}}(t) \le T_{\max},$$

$$V(t_{\text{f}}) \le V_{\max},$$

$$n_C(t_{\text{f}}) \ge n_{C\text{des}}. \tag{46}$$

6.1.2.5. Specific choice of experimental conditions. Let the experimental conditions be such that the number of moles of B that can be added is less than the initial number of moles of A, then $c_B(t) \le c_A(t)$. Since isothermal conditions are chosen, the condition $T_{\text{cf}}(t) \le T_{\max}$ implies $c_B(t) \le c_{B\max}$, with $c_{B\max} = \rho c_{\text{p}}(T_{\max} - T)/(-\Delta H)$. Furthermore, the initial conditions correspond to having as much B as possible, i.e. $c_{B\text{o}} = c_{B\max} = 0.63$ mol/l.

6.1.3. Optimal solution

The optimal input and the corresponding evolution of the concentrations of A, B and C obtained numerically are given in Fig. 1. The optimal input consists of the two arcs u_{path} and $u_{\min}$:

- Since the initial conditions verify $c_{B\text{o}} = c_{B\max}$, u_{path} is applied to keep $c_B = c_{B\max}$, i.e. $T_{\text{cf}} = T_{\max}$.
- Once $V = V_{\max}$ is attained, the input is set to $u_{\min} = 0$.
- Once $n_C = n_{C\text{des}}$ is attained, the batch is stopped so as to minimize the final time.

For the numerical values provided in Table 3, the minimal time $J^* = t_{\text{f}}^* = 19.80$ h is obtained with the switching time $t_{\text{s}} = 11.44$ h.

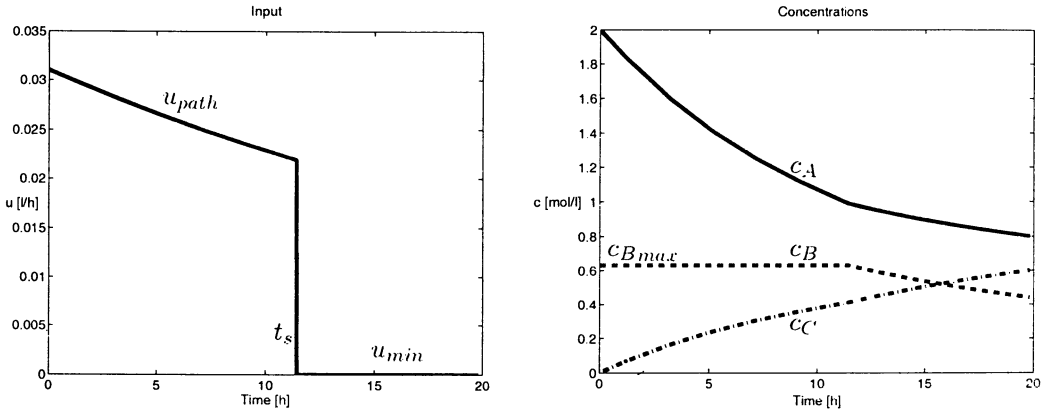

Fig. 1. Optimal input and evolution of the concentrations for Example 1.

6.1.3.1. Analytical expression for u_{path}. Since $c_B(t)$ has relative degree 1, the optimal input that keeps the path constraint $c_B = c_{B\max}$ active can be obtained by differentiating the path constraint once with respect to time:

$$u_{\text{path}} = \left(\frac{kc_A c_B V}{c_{B\text{in}} - c_B}\right)\Bigg|_{c_B = c_{B\max}}. \tag{47}$$

6.1.3.2. Effect of different experimental conditions.

1) If $c_{B\text{o}} < c_{B\max}$, the optimal input has an additional arc. Initially, the input is at the upper bound $u_{\max}$ in order to attain the path constraint as quickly as possible. Once T_{cf} reaches $T_{\max}$, the two arcs presented in Fig. 1 form the optimal solution.
2) If the number of moles of B that can be added is larger than the initial number of moles of A, the optimal input has an additional arc. Once $c_B(t) = c_A(t)$ is attained, the input switches to its maximum value since this no longer affects T_{cf}. Then, when the volume reaches $V = V_{\max}$, the input is set to $u_{\min} = 0$.

6.1.3.3. Effect of constraints.

1) Without the safety constraint, it would be optimal to operate in batch mode, where all the B is fed initially, leading to $t_{\text{f}}^* = 17.3$ h. Thus, the 'price' to pay for safety is a longer time (19.8 h) to attain the same conversion.
2) Without the volume constraint, the optimal solution would correspond to continue feeding B in such a way that the safety constraint is met. Since more B could be added this way, the final time would reduce to $t_{\text{f}}^* = 18.4$ h.

6.1.4. Interpretation of the optimal solution

6.1.4.1. Meeting path objectives. In both arcs, the input is determined by a constraint. In fact, the matrix $\mathcal{M} = [F_u \quad \Delta F_u]$ indicates that the optimal input cannot be inside the feasible region. Consider the dynamic model given by Eqs. (41) and (43), together with Eq. (45). Then,

$$\mathbf{F} = \begin{bmatrix} -kc_A c_B V \\ u \end{bmatrix}, \quad \mathbf{F}_u = \begin{bmatrix} 0 \\ 1 \end{bmatrix}, \quad \Delta\mathbf{F}_u \begin{bmatrix} kc_A(c_{B\text{in}} - c_B) \\ 0 \end{bmatrix}. \tag{48}$$

The matrix $\mathcal{M}$ has structural rank 2. Since $(c_{B\text{in}} - c_B)$ is always positive, $\mathcal{M}$ can only lose rank for the trivial case $c_A = 0$. Thus, the rank is independent of the evolution of the states and input ($\sigma = \infty$), and the optimal input is always determined by a path constraint.

6.1.4.2. Meeting terminal objectives. The switching time t_{s} between u_{path} and $u_{\min}$ and the terminal time t_{f} are adjusted to satisfy the terminal constraints $V(t_{\text{f}}) = V_{\max}$ and $n_C(t_{\text{f}}) = n_{C\text{des}}$. Thus, the two input parameters are constraint-seeking.

6.2. Fed-batch bioreactor with inhibition and a biomass constraint (Visser, Srinivasan, Palanki, & Bonvin, 2000)

6.2.1. Description of the reaction system

- *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.
- *Comments:* The specific growth rate $\mu(S)$ contains an inhibition term:

Table 4
Model parameters, operating bounds and initial conditions for Example 2

μ_m	0.53	l/h
K_m	1.2	g/l
K_i	22	g/l
Y_x	0.4	
Y_p	1	
v	0.5	l/h
S_{in}	20	g/l
u_{min}	0	l/h
u_{max}	1	l/h
X_{max}	3	g/l
t_f	8	h
X_o	1	g/l
S_o	0	g/l
P_o	0	g/l
V_o	2	l

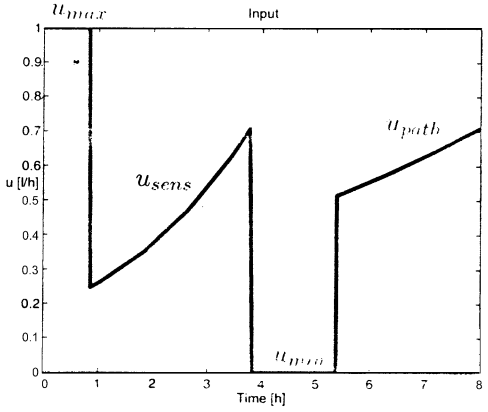

Fig. 2. Optimal input for Example 2.

$$\mu(S) = \frac{\mu_m S}{K_m + S + (S^2/K_i)}.$$

Owing to the presence of inhibition, it will be shown that the optimal substrate value corresponds to $d\mu/dS = 0$ (i.e. $S^* = \sqrt{K_m K_i}$). Without any constraints, optimality would consist of operating at $S = S^*$ so as to increase X, and thus P, as quickly as possible. However, there is a constraint on the biomass concentration, which is motivated by oxygen-transfer limitation typically occurring at large biomass concentrations. The interesting part is that the optimal input cannot switch immediately from u_{sens} (corresponding to $S = S^*$) to u_{path} (corresponding to $X = X_{max}$) since the internal dynamics are unstable. An additional arc is required to lower the substrate concentration to the equilibrium value S_e.

6.2.2. Problem formulation

6.2.2.1. 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.

6.2.2.2. Model equations.

$$\dot{X} = \mu(S)X - \frac{u}{V}X \quad X(0) = X_o, \tag{49}$$

$$\dot{S} = -\frac{\mu(S)X}{Y_x} - \frac{vX}{Y_p} + \frac{u}{V}(S_{in} - S) \quad S(0) = S_o, \tag{50}$$

$$\dot{P} = vX - \frac{u}{V}P \quad P(0) = P_o, \tag{51}$$

$$\dot{V} = u \quad V(0) = V_o, \tag{52}$$

with $\mu(S) = (\mu_m S)/(K_m + S + (S^2/K_i))$ and the numerical values given in Table 4.

6.2.2.3. Model reduction. As in Example 1, one state is redundant. The redundant state is first removed to make calculations simpler. With $x_1 = XV$, $x_2 = PV$, $x_3 = V$, the reaction dynamics can be described by:

$$\dot{x}_1 = \mu(S)x_1 \quad x_1(0) = X_o V_o, \tag{53}$$

$$\dot{x}_2 = vx_1 \quad x_2(0) = P_o V_o, \tag{54}$$

$$\dot{x}_3 = u \quad x_3(0) = V_o, \tag{55}$$

where the substrate concentration is obtained from a mass balance:

$$S = \frac{1}{x_3}\left(S_o V_o + S_{in}(x_3 - V_o) - \frac{1}{Y_x}(x_1 - X_o V_o) - \frac{1}{Y_p}(x_2 - P_o V_o)\right). \tag{56}$$

6.2.2.4. Optimization problem.

$$\max_{u(t)} J = P(t_f),$$

$$\text{s.t.} \quad (53) - (56)$$

$$X(t) \le X_{max},$$

$$u_{min} \le u(t) \le u_{max}. \tag{57}$$

6.2.3. Optimal input

The optimal input obtained numerically is given in Fig. 2. It consists of the four intervals u_{max}, u_{sens}, u_{min} and u_{path}:

- The input is initially at the upper bound, u_{max}, in order to increase S as quickly as possible.
- Once the optimal substrate concentration S^* is reached, u_{sens} is applied in order to keep S at S^* and thus increase X and P as quickly as possible. The input is inside the feasible region.
- The input is then lowered to u_{min} in order to reach the equilibrium value S_e as quickly as possible. 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 the biomass concentration reaches X_{max}, the input is set to u_{path} in order to maintain $X = X_{max}$ and $S = S_e$.

For the numerical values provided in Table 4, the maximum product concentration is $P^*(t_f) = 8.2$ g/l, and the three switching times are 0.862, 3.83, and 5.385 h.

6.2.3.1. Analytical expression for u_{path}. The path constraint corresponds to $X = X_{max}$. The input can be obtained by differentiating the path constraint once with respect to time:

$$u_{path} = \mu(S)V|_{X=X_{max}}. \tag{58}$$

When $u = u_{path}$ is applied at $X = X_{max}$, the substrate dynamics and its linear approximation are:

$$\dot{S} = -\frac{1}{Y_x}\,\mu(S)X_{max} - \frac{1}{Y_p}\,vX_{max} + \mu(S)(S_{in} - S), \tag{59}$$

$$\Delta\dot{S} = \frac{\partial\mu}{\partial S}\left(S_{in} - S - \frac{X_{max}}{Y_x}\right)\Delta S - \mu(S)\Delta S. \tag{60}$$

It can be verified numerically that the linear approximation of the substrate dynamics is unstable. Hence, to remain bounded, the biomass constraint has to be entered with the substrate value S_e that corresponds to the equilibrium point of the internal dynamics (Eq. (59)). This way, during the last arc, the substrate concentration will stay at $S = S_e$.

6.2.3.2. Analytical expression for u_{sens}. The analytical expression for u_{sens} can be calculated from the loss of rank of the matrix $\mathcal{M} = [\mathbf{F}_u, \Delta\mathbf{F}_u, \Delta^2\mathbf{F}_u]$:

$$\mathbf{F} = \begin{bmatrix}\mu(S)x_1\\ vx_1\\ u\end{bmatrix}, \quad \mathbf{F}_u = \begin{bmatrix}0\\0\\1\end{bmatrix},$$
$$\Delta\mathbf{F}_u = -\begin{bmatrix}(\partial\mu/\partial x_3)x_1\\0\\0\end{bmatrix}, \tag{61}$$

$$\Delta^2\mathbf{F}_u = \begin{bmatrix}(\partial\mu/\partial x_1)(\partial\mu/\partial x_3)x_1^2 - (\partial^2\mu/\partial x_1\partial x_3)x_1^2\mu - vx_1^2(\partial^2\mu/\partial x_2\partial x_3)\\ v(\partial\mu/\partial x_3)x_1\\0\end{bmatrix} - u\begin{bmatrix}(\partial^2\mu/\partial x_3^2)x_1\\0\\0\end{bmatrix}. \tag{62}$$

The matrix $\mathcal{M}$ has structural rank $\rho = 3$, but the rank depends on the states. The loss of rank can be analyzed using $\det(\mathcal{M}) = 0$, which occurs when:

$$vx_1^2\left(\frac{\partial\mu}{\partial x_3}\right)^2 = vx_1^2\left(\frac{\partial\mu}{\partial S}\frac{\partial S}{\partial x_3}\right)^2 = v\frac{x_1^2}{x_3^2}(S_{in} - S)^2\left(\frac{\partial\mu}{\partial S}\right)^2 = 0. \tag{63}$$

$S = S_{in}$, $x_1 = 0$, or $\partial\mu/\partial S = 0$ are solutions of $\det(\mathcal{M}) = 0$. Since $S = S_{in}$ and $x_1 = 0$ result in trivial solutions, rank drop occurs for $\partial\mu/\partial S = 0$, which corresponds to $S = S^* = \sqrt{K_m K_i}$. Though the input appears in $\Delta^2\mathbf{F}_u$, $\det(\mathcal{M})$ is independent of u since the vector that multiplies u in Eq. (62) is parallel to $\Delta\mathbf{F}_u$. Thus, an additional differentiation is required to obtain the input ($\sigma = 3$, $\xi = -1$) or, equivalently, the *surface* $S = S^*$ can be differentiated once to obtain the input:

$$u_{sens} = \frac{V}{S_{in} - S}\left(\frac{1}{Y_x}\,\mu(S)X + \frac{1}{Y_p}\,vX\right)\Bigg|_{S=S^*}. \tag{64}$$

6.2.4. Interpretation of the optimal solution

6.2.4.1. Meeting path constraints. In all intervals, except the second one, $u = \tilde{\eta}$. In the second interval, $u = u_{sens} = \tilde{\eta}$. It can be verified numerically that a small deviation of the input in this interval has very little influence on the cost.

6.2.4.2. Meeting terminal objectives. Though there are three switching times, there are all linked to achieving some intermediate goals (getting to the path constraints): the first switching corresponds to reaching $S = S^*$, while the second and third switchings are determined upon attaining $S = S_e$ and $X = X_{max}$. Thus, there is no degree of freedom left for meeting any terminal objective, which is logical since there is no terminal constraint!

6.3. Isothermal semi-batch reactor with parallel reactions and selectivity constraints (Ruppen et al., 1998; Srinivasan, Primus, Bonvin, & Ricker, 2001)

6.3.1. Description of the reaction system

- *Reactions:* $A + B \rightarrow C$, $2B \rightarrow D$.
- *Conditions:* Semi-batch, isothermal.
- *Objective:* Maximize the production of C at a given final time.

- *Manipulated variable:* Feed rate of B.
- *Constraints:* Input bounds; constraints on the maximum concentrations of B and D at final time.
- *Comments:* If the second (undesired) reaction were absent, it would be optimal to have as large a value of c_B as possible. The optimization potential is created by the presence of the second reaction, thereby giving rise to a possible compromise. However, this compromise is only present when there is a constraint on the final amount of D. Note that, in the absence of constraints, optimal operation would simply consist of adding all the available B at initial time (i.e. batch operation). Furthermore, since the amount of B present in the reactor at final time is limited, the feed rate of B is turned off towards the end of the batch.

6.3.2. Problem formulation

6.3.2.1. Variables and parameters. c_X, concentration of species X; V, reactor volume; u, feed rate of B; $c_{B\text{in}}$, inlet concentration of B; and k_1, k_2, kinetic parameters.

6.3.2.2. Model equations.

$$\dot{c}_A = -k_1 c_A c_B - \frac{u}{V} c_A \quad c_A(0) = c_{A\text{o}}, \tag{65}$$

$$\dot{c}_B = -k_1 c_A c_B - 2k_2 c_B^2 + \frac{u}{V}(c_{B\text{in}} - c_B) \quad c_B(0) = c_{B\text{o}}, \tag{66}$$

$$\dot{V} = u \quad V(0) = V_\text{o}, \tag{67}$$

with

$$c_C = \frac{1}{V}(c_{A\text{o}} V_\text{o} - c_A V), \tag{68}$$

$$c_D = \frac{1}{2V}((c_A + c_{B\text{in}} - c_B)V - (c_{A\text{o}} + c_{B\text{in}} - c_{B\text{o}})V_\text{o}). \tag{69}$$

The numerical values are given in Table 5.

Table 5
Model parameters, operating bounds and initial conditions for Example 3

k_1	0.053	l/mol min
k_2	0.128	l/mol min
$c_{B\text{in}}$	5	mol/l
$u_{\min}$	0	l/min
$u_{\max}$	0.001	l/min
$c_{B\text{f,max}}$	0.025	mol/l
$c_{D\text{f,max}}$	0.15	mol/l
$c_{A\text{o}}$	0.72	mol/l
$c_{B\text{o}}$	0.05	mol/l
V_o	1	l
t_f	250	min

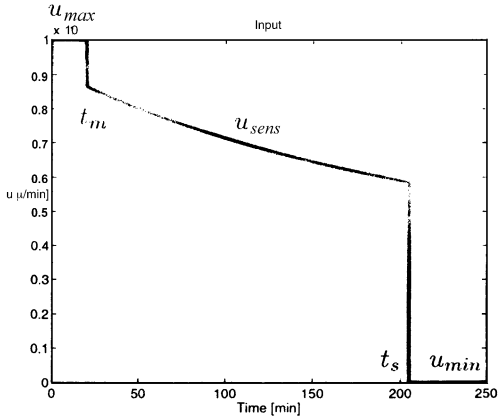

Fig. 3. Optimal input for Example 3.

6.3.2.3. Optimization problem.

$$\max_{u(t)} J = V(t_\text{f})\, c_C(t_\text{f}),$$

$$\text{s.t.} \quad (65) - (69)$$

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

$$c_B(t_\text{f}) \le c_{B\text{f,max}},$$

$$c_D(t_\text{f}) \le c_{D\text{f,max}}. \tag{70}$$

6.3.3. Optimal input

The optimal input obtained numerically is given in Fig. 3. It consists of the three intervals $u_{\max}$, u_{sens} and $u_{\min}$:

- The input is initially at the upper bound $u_{\max}$ to increase c_B and thus the rate of the desired reaction.
- The input switches to the sensitivity-seeking arc so that only a limited amount of D is produced.
- The input switches to $u_{\min}$ so that c_B can meet its constraint at final time.

For the numerical values provided in Table 5, the maximum number of moles of C is $n_C^*(t_\text{f}) = 0.43$ mol, and the two switching times are $t_\text{m} = 20.25$ and $t_\text{s} = 205$ h.

6.3.3.1. Analytical expression for u_{sens}. The analytical expression for u_{sens} can be determined using the matrix $\mathscr{M} = [\mathbf{F}_u, \Delta\mathbf{F}_u, \Delta^2\mathbf{F}_u]$:

$$\mathbf{F} = \begin{bmatrix} -k_1 c_A c_B \\ -k_1 c_A c_B - 2k_2 c_B^2 \\ 0 \end{bmatrix} + \frac{1}{V}\begin{bmatrix} -c_A \\ c_{B\text{in}} - c_B \\ V \end{bmatrix} u,$$

$$\mathbf{F}_u = \frac{1}{V}\begin{bmatrix} -c_A \\ c_{B\text{in}} - c_B \\ V \end{bmatrix},$$

$$\Delta\mathbf{F}_u = \frac{1}{V}\begin{bmatrix} k_1 c_A (c_{B\text{in}} - c_B) \\ k_1 c_A (c_{B\text{in}} - c_B) + 2k_2 c_B (2c_{B\text{in}} - c_B) \\ 0 \end{bmatrix},$$

$$\Delta^2\mathbf{F}_u = \frac{c_{B\text{in}}}{V}\begin{bmatrix} k_1^2 c_A^2 + 4k_1 k_2 c_A c_B \\ k_1^2 c_A^2 + 4k_1 k_2 c_A c_B + 8k_2^2 c_B^2 \\ 0 \end{bmatrix} - 2(c_{B\text{in}} - c_B) \times \frac{u}{V^2}\begin{bmatrix} k_1 c_A \\ k_1 c_A - 2k_2 (c_{B\text{in}} - c_B) \\ 0 \end{bmatrix}.$$

The matrix $\mathscr{M}$ has structural rank $\rho = 3$. However, since the rank depends on the states and input, it may be possible to reduce it by an appropriate combination of states and input, i.e. $u_{\text{sens}} = u(\boldsymbol{x})$. The combination of $\boldsymbol{x}$ and u for which the rank of $\mathscr{M}$ drops can be computed from $\det(\mathscr{M}) = 0$. The input appears in $\Delta^2\mathbf{F}_u$, which indicates that the order of singularity is $\sigma = 2$. Since $\xi = 0$, a static feedback for the optimal input can be computed from $\det(\mathscr{M}) = 0$:

$$u_{\text{sens}} = \frac{c_{B\text{in}} c_B V (k_1 c_A (2c_{B\text{in}} - c_B) + 4k_2 c_B c_{B\text{in}})}{2(c_{B\text{in}} - c_B)}. \tag{71}$$

6.3.4. Interpretation of the optimal solution

6.3.4.1. Meeting path objectives. In the first and third intervals, $u = \bar{\eta}$, while $u = u_{\text{sens}} = \tilde{\eta}$ in the second one. It can be verified numerically that a small deviation of the input in the second interval has very little influence on the cost.

6.3.4.2. Meeting terminal objectives. The two switching times t_{m} and t_{s} parameterize the solution completely. In turn, they are determined by the two active terminal constraints $c_B(t_{\text{f}}) = c_{B\text{f,max}}$ and $c_D(t_{\text{f}}) = c_{D\text{f,max}}$. Thus, the two input parameters in this example are constraint-seeking parameters.

Since the second interval is sensitivity-seeking, it is possible to approximate it by a constant, without significant loss in performance. Consider the case where $u_{\text{sens}}(t)$ is approximated by the scalar value u_{s}. Then, there are three parameters, (t_{m}, t_{s}, and u_{s}), to meet the two terminal constraints and optimize the cost. The gain matrix $\mathbf{G}_T$: $\boldsymbol{\pi} \to \boldsymbol{T}$ computed in the neighborhood of the optimal solution, with $\boldsymbol{\pi} = [t_{\text{m}}, t_{\text{s}}, u_{\text{s}}]^{\text{T}}$ and $\boldsymbol{T}(\boldsymbol{x}(t_{\text{f}})) = [c_{\text{D}}(t_{\text{f}}) - c_{D\text{f,max}}, c_B(t_{\text{f}}) - c_{B\text{f,max}}]^{\text{T}}$, is given by:

$$\mathbf{G}_T = \begin{bmatrix} 0.6 \times 10^{-3} & 0.5 \times 10^{-3} & 1.5 \times 10^{2} \\ 1.9 \times 10^{-5} & 1.1 \times 10^{-3} & 0.2 \times 10^{2} \end{bmatrix}.$$

Using singular value decomposition, it can be seen that $\tilde{\pi} = t_{\text{m}} + 4.7 \times 10^{-2}\, t_{\text{s}} - 4 \times 10^{-6}\, u_{\text{s}}$. Since the contributions from t_{s} and u_{s} are negligible, t_{m} essentially acts as the sensitivity-seeking parameter. It is interesting to note that t_{m}, which was a constraint-seeking parameter when u_{sens} was not approximated, becomes a sensitivity-seeking parameter after the approximation. This can be explained by the fact that the new input parameter u_{s} has a strong effect on the constraints (see the last column of $\mathbf{G}_T$) and becomes the dominant constraint-seeking parameter.

6.4. Non-isothermal semi-batch reactor with series reactions and a heat removal constraint

6.4.1. Description of the reaction system

- *Reactions:* $A + \text{B} \to C \to D$.
- *Conditions:* Semi-batch, exothermic, non-isothermal, operated in a jacketed reactor such that the reactor temperature can be adjusted quickly.
- *Objective:* Maximize the production of C at a given final time.
- *Manipulated variables:* Feed rate of B and reactor temperature.
- *Constraints:* Bounds on feed rate and reactor temperature; constraint on the maximum heat that can be removed by the cooling system; constraint on the maximum volume.
- *Comments:* The reactor temperature is assumed to be a manipulated variable though, in practice, either the flowrate or the temperature in the cooling jacket is manipulated. The heat balance equation for the reactor is: $\rho c_{\text{p}}(V\text{d}T/\text{d}t) = q_{\text{rx}} - q_{\text{in}} - q_{\text{ex}}$, where V is the reactor volume; T, the reactor temperature; ρ, the density; c_{p}, the heat capacity; q_{rx}, the rate of heat produced by the reactions; q_{in}, the rate of heat removal associated with the feed of B; and q_{ex}, the rate of heat removal through the cooling jacket. The inclusion of the heat balance equation would complicate the analytical expressions without fundamentally changing the types and sequence of arcs present in the solution. Thus, for simplicity, the heat balance equation will be neglected. However, to guarantee meeting the upper bound on q_{ex} even in the worst scenario, it is necessary to limit q_{rx} as follows: $q_{\text{rx}} \leq \max(q_{\text{ex}}) + \min(q_{\text{in}} + \rho c_{\text{p}}(V\text{d}T/\text{d}t)) \equiv q_{\text{rx,max}}$. Thus, an upper bound on the heat rate produced by the reactions, $q_{\text{rx}} \leq q_{\text{rx,max}}$, is imposed as a constraint.The consumption of the desired product C by the undesired reaction is reduced by lowering the temperature towards the end of the batch. The compromise between the production and consumption of C corresponds to a sensitivity-seeking temperature profile. As far as the feed rate is concerned, it is first determined by the heat removal constraint and then by the volume constraint.Without any constraints, optimal operation would consist of adding all the available B at initial time and following a temperature profile that expresses the compromise between the production and consumption of C.

6.4.2. Problem formulation

6.4.2.1. Variables and parameters. c_X, concentration of species X; T, reactor temperature; u, feed rate of B with inlet concentration $c_{B\text{in}}$; V, reactor volume; q_{rx}, heat production rate; $k_{1\text{o}}$, $k_{2\text{o}}$, pre-exponential factors; E_1, E_2, activation energies; R, gas constant; ΔH_1, ΔH_2, reaction enthalpies.

6.4.2.2. Model equations.

$$\dot{c}_A = -k_1 c_A c_B - \frac{u}{V} c_A \quad c_A(0) = c_{A\text{o}}, \tag{72}$$

$$\dot{c}_B = -k_1 c_A c_B + \frac{u}{V}(c_{B\text{in}} - c_B) \quad c_B(0) = c_{B\text{o}}, \tag{73}$$

$$\dot{c}_C = k_1 c_A c_B - k_2 c_C - \frac{u}{V} c_C \quad c_C(0) = c_{C\text{o}}, \tag{74}$$

$$\dot{V} = u \quad V(0) = V_{\text{o}}, \tag{75}$$

with $k_1 = k_{1\text{o}}\, \text{e}^{-E_1/RT}$, $k_2 = k_{2\text{o}}\, \text{e}^{-E_2/RT}$. Note that the input variable T acts on the system via the temperature-dependent coefficients k_1 and k_2. The numerical values used in this study are given in Table 6.

6.4.2.3. Model reduction. Since Eqs. (72)–(75) are linearly dependent, one of the states can be removed, which leads to:

$$\dot{x}_1 = -k_1 x_1 c_B \quad x_1(0) = V_{\text{o}} c_{A\text{o}}, \tag{76}$$

$$\dot{x}_2 = k_2(x_1 - x_2) \quad x_2(0) = V_{\text{o}}(c_{A\text{o}} + c_{C\text{o}}), \tag{77}$$

$$\dot{x}_3 = u \quad x_3(0) = V_{\text{o}}, \tag{78}$$

where $x_1 = Vc_A$, $x_2 = V(c_A + c_C)$, $x_3 = V$, and

Table 6
Model parameters, operating bounds and initial conditions for Example 4

$k_{1\text{o}}$	4	l/mol h
$k_{2\text{o}}$	800	l/h
E_1	6×10^3	J/mol
E_2	20×10^3	J/mol
R	8.31	J/mol K
ΔH_1	-3×10^4	J/mol
ΔH_2	-10^4	J/mol
$u_{\min}$	0	l/h
$u_{\max}$	1	l/h
$T_{\min}$	20	°C
$T_{\max}$	50	°C
$V_{\max}$	1.1	l
$q_{\text{rx,max}}$	1.5×10^5	J/h
$c_{A\text{o}}$	10	mol/l
$c_{B\text{o}}$	1.1685	mol/l
$c_{C\text{o}}$	0	mol/l
V_{o}	1	l
$c_{B\text{in}}$	20	mol/l
t_{f}	0.5	h

$$c_B = \frac{1}{x_3}(c_{B\text{in}} x_3 + x_1 + V_{\text{o}}(c_{B\text{o}} - c_{A\text{o}} - c_{B\text{in}})). \tag{79}$$

6.4.2.4. Optimization problem.

$$\max_{u(t),\ T(t)} J = c_C(t_{\text{f}}) V(t_{\text{f}}),$$

s.t. (76) − (79)

$$T_{\min} \le T(t) \le T_{\max},$$

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

$$(-\Delta H_1) k_1 c_A c_B V + (-\Delta H_2) k_2 c_C V \le q_{\text{rx,max}},$$

$$V(t_{\text{f}}) \le V_{\max}. \tag{80}$$

6.4.2.5. Specific choice of experimental conditions. Let the initial conditions be chosen such that as much B as possible is charged initially in the reactor while still meeting the heat removal constraint. Thus, $c_{B\text{o}}$ is chosen to verify $(-\Delta H_1) k_1\, c_{A\text{o}}\, c_{B\text{o}}\, V_{\text{o}} + (-\Delta H_2) k_2\, c_{C\text{o}}\, V_{\text{o}} = q_{\text{rx,max}}$.

6.4.3. Optimal inputs

The optimal inputs obtained numerically are given in Fig. 4. Each one consists of two arcs, u_{path} and $u_{\min}$ for the feed rate, and $T_{\max}$ and T_{sens} for the temperature:

- Since the initial condition $c_{B\text{o}}$ verifies $q_{\text{rx}}(0) = q_{\text{rx,max}}$, the feed rate input u_{path} is applied to keep that path constraint active.
- Once the volume constraint is attained, the feed rate is set to $u_{\min} = 0$.
- The temperature starts at its upper bound $T_{\max}$ to favor the desired reaction.
- Later, the temperature switches to T_{sens} to take advantage of the temperature-dependent compromise between the production and consumption of C.

When the temperature goes inside the feasible region, there is a discontinuity in the feed rate due to the coupling between the two inputs. Similarly, when the feed rate switches to zero to satisfy the volume constraint, there is a discontinuity in the rate of change of the temperature. For the numerical values provided in Table 6, the maximum number of moles of C is $n_C^*(t_{\text{f}}) = 2.02$ mol, and the two switching times are $t_T = 0.05$ and $t_u = 0.3165$ h.

6.4.3.1. Analytical expression for u_{path}. The arc u_{path} is obtained by differentiating the path constraint regarding the heat production rate once with respect to time:

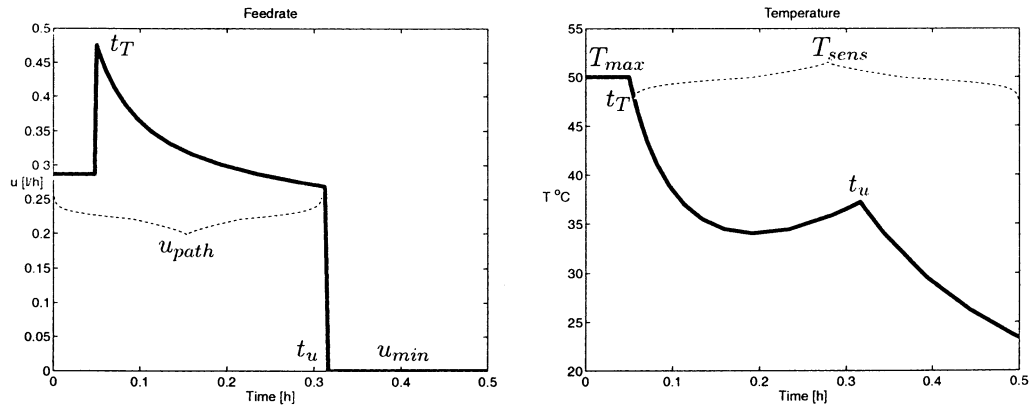

Fig. 4. Optimal feed rate and temperature profiles for Example 4.

$$u_{\text{path}} = V \times \frac{((-\Delta H_1)k_1^2 c_A c_B(c_A + c_B) - (-\Delta H_2)k_2(k_1 c_A c_B - k_2 c_C))}{(-\Delta H_1)k_1 c_A(c_{B\text{in}} - c_B)} - \frac{\dot{T}V}{RT^2}\frac{((-\Delta H_1)E_1 k_1 c_A c_B + (-\Delta H_2)E_2 k_2 c_C)}{(-\Delta H_1)k_1 c_A(c_{B\text{in}} - c_B)}. \quad (81)$$

6.4.3.2. Analytical expression for T_{sens}. T_{sens} is obtained from the combination of x, u, and T for which the rank of $\mathscr{M}_T = [\mathbf{F}_T \quad \Delta\mathbf{F}_T \quad \Delta^2\mathbf{F}_T]$ drops:

$$\mathbf{F} = \begin{bmatrix} -k_1 c_A c_B V \\ -k_2 c_C V \\ u \end{bmatrix}, \quad \mathbf{F}_T = \frac{V}{RT^2}\begin{bmatrix} E_1 k_1 c_A c_B \\ E_2 k_2 c_C \\ 0 \end{bmatrix},$$

$$\Delta\mathbf{F}_T = -\frac{V}{RT^2}\begin{bmatrix} 0 \\ k_1 k_2 c_A c_B(E_1 - E_2) \\ 0 \end{bmatrix} - \frac{\dot{T}V}{R^2T^4} \times \begin{bmatrix} E_1 k_1 c_A c_B(E_1 - 2RT) \\ E_2 k_2 c_C(E_2 - 2RT) \\ 0 \end{bmatrix} + \frac{E_1 u}{RT^2}\Delta\mathbf{F}_u.$$

The matrix $\mathscr{M}_T$ has structural rank $\rho_T = 2$ since the third element of all involved vector fields is zero. Intuitively, this is because the temperature cannot affect the volume. Even though the structural rank is 2, the rank depends on the states and inputs. The expression for T_{sens} can be computed from the determinant of the first two rows of $\mathbf{F}_T$, and $\Delta\mathbf{F}_T$. Since $\mathbf{F}_T$ is already a function of T, the order of singularity is $\sigma_T = 0$. Since $\xi_T = 1$, T_{sens} corresponds to a dynamic feedback:

$$\dot{T}_{\text{sens}} = -\frac{RT^2 k_1 c_A c_B}{E_2 c_C} - \frac{RT^2(c_{B\text{in}} - c_B)}{c_B(E_1 - E_2)}\frac{u}{V}. \quad (82)$$

The initial condition of T_{sens} as it enters the sensitivity-seeking arc is a decision variable, but it can be verified numerically that it is equal to $T_{\max}$. It is interesting to note that u_{path} depends on $\dot{T}$, and $\dot{T}_{\text{sens}}$ depends on u. Thus, if in a given interval u is determined by the path constraint and T is sensitivity-seeking, then Eqs. (81) and (82) have to be solved simultaneously.

6.4.4. Interpretation of the optimal solution

6.4.4.1. Meeting path objectives. The three arcs of this solution need to be addressed separately:

- In the first arc, both inputs are on path constraints, i.e. $\bar{\boldsymbol{\eta}} = \{u_{\text{path}},\ T_{\max}\}$, and $\tilde{\eta} = \{\}$.
- In the second arc, only the path constraint regarding the heat production rate is active, for which two inputs are available. The gain matrix $\mathbf{G}_S$: $[u,\ \dot{T}] \to q_{\text{rx,max}}$ is given by:
$$[(-\Delta H_1)k_1 c_A(c_{B\text{in}} - c_B) \quad \frac{(-\Delta H_1)E_1 k_1 c_A c_B V + (-\Delta H_2)E_2 k_2 c_C V}{RT^2}]$$
- So, the singular value decomposition of the gain matrix can be used to compute $\bar{\eta}$ and $\tilde{\eta}$ (see Section 5.2):
$$\bar{\eta} = u\,(-\Delta H_1)k_1 c_A(c_{B\text{in}} - c_B) + \dot{T}\frac{(-\Delta H_1)E_1 k_1 c_A c_B V + (-\Delta H_2)E_2 k_2 c_C V}{RT^2},$$
$$\tilde{\eta} = u\frac{(-\Delta H_1)E_1 k_1 c_A c_B V + (-\Delta H_2)E_2 k_2 c_C V}{RT^2} - \dot{T}\,(-\Delta H_1)k_1 c_A(c_{B\text{in}} - c_B).$$
- In the third arc, only the input bound for the feed rate is active. So, $\bar{\eta} = u_{\min}$, and $\tilde{\eta} = T_{\text{sens}}$.

6.4.4.2. Meeting terminal objectives. The two switching times t_T and t_u parameterize the solution completely. Since there is only one active terminal constraint, $V(t_f) = V_{\max}$, a combination of the two switching times is constraint-seeking. The gain matrix, in the neighborhood of the optimum, $\mathbf{G}_T$: $\boldsymbol{\pi} \to V(t_f) - V_{\max}$, with $\boldsymbol{\pi} = [t_T \quad t_u]^{\mathrm{T}}$, is given by $\mathbf{G}_T = [-0.365 \quad 0.268]$. The con-

straint-seeking parameter is $\bar{\pi} = 0.365t_T - 0.268t_u$, while the sensitivity-seeking parameter is given by $\tilde{\pi} = 0.268t_T + 0.365t_u$.

7. Conclusions

This paper first presented an overview of the analytical and numerical techniques that are available in the literature to solve dynamic optimization problems with a special emphasis on classification. For the analytical techniques, the classification was based on whether or not the adjoint variables are used. For the numerical techniques, the classification was based on the problem formulation and the type of parameterization used.

Most techniques proposed in the literature for the optimization of dynamic processes take a black-box approach. In contrast, this paper emphasizes the *solution structure* that arises from interplay between manipulated variables, cost and constraints. Certain intervals of the optimal solution are linked to path constraints, while others are shown to be inside the feasible region. Thus, the optimal inputs can be presented in terms of constraint- and sensitivity-seeking intervals. In addition, some of the switching times between the various intervals can also be interpreted as constraint- and sensitivity-seeking parameters that are selected for meeting terminal constraints or optimizing the terminal-time objective function. A series of examples are provided to illustrate these concepts.

The main application of the characterization presented in this paper is linked to the design of measurement-based optimization strategies. Handling uncertainty in the context of optimization is a very important issue, especially in the presence of constraints on quality and safety. Most optimization techniques are *model-based*, whilst accurate models of industrial batch processes are rarely available. With the advent of recent developments in sensor technology, *measurement-based* optimization strategies could be used to cope with uncertainty. These strategies, which constitute the focus of the companion paper (Srinivasan et al., 2002), are based on the characterization presented in this work.

Acknowledgements

The authors would like to thank the anonymous reviewer, whose helpful comments have been used to improve this paper, 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.

Allgor, R. J., Barrera, M. D., Barton, P. I., & Evans, L. B. (1996). Optimal batch process development. *Computers and Chemical Engineering 20* (6–7), 885–896.

Barrera, M. D., & Evans, L. B. (1989). Optimal design and operation of batch processes. *Chemical Engineering Communication 82*, 45–66.

Basu, P. K. (1998). Pharmaceutical process development is different! *Chemical Engineering Progress 94* (9), 75–82.

Baumann, T. (1998). *Infinite-order singularity in terminal-cost optimal control: application to robotic manipulators*. PhD thesis 1778. Lausanne, Switzerland: Swiss Federal Institute of Technology (EPFL).

Bell, M. L., Limebeer, D. J. N., & Sargent, R. W. H. (1996). Robust receding horizon optimal control. *Computers and Chemical Engineering 20*, S781–S786.

Bellman, R. E. (1957). *Dynamic programming*. Princeton University Press.

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).

Biegler, L. T. (1984). Solution of dynamic optimization problems by successive quadratic programming and orthogonal collocation. *Computers and Chemical Engineering 8*, 243–248.

Biegler, L. T., Cervantes, A. M., & Wächter, M. A. (2002). Advances in simultaneous strategies for dynamic process optimization. *Chemical Engineering Science 57*, 575–593.

Bock, H. G., & Platt, K. J. (1984). A multiple shooting algorithm for direct solution of optimal control problems. In *9th IFAC world congress* (pp. 242–247). Budapest.

Bojkov, B., & Luus, R. (1994). Time-optimal control by iterative dynamic programming. *Industrial and Engineering Chemistry Research 33*, 1486–1492.

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. (2001). Dynamic optimization in the batch chemical industry. Proceedings of the CPC-VI Conference. *American Institute of Chemical Engineers, Symposium Series 326* (98), 255–273.

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

Bryson, A. E., & Ho, Y. C. (1975). *Applied optimal control*. Washington DC: Hemisphere.

Cervantes, A., & Biegler, L. T. (1998). Large-scale DAE optimization using a simultaneous NLP formulation. *American Institute of Chemical Engineers Journal 44* (5), 1038–1050.

Chen, S. A., & Jeng, W. F. (1978). Minimum end time policies for batchwise radical chain polymerization. *Chemical Engineering Science 33*, 735.

Chu, W. B. (1987). *Modeling, optimization and computer control of the Cephalosporin C fermentation process.* PhD thesis. New Brunswick, New Jersey, USA: Rutgers, The State University of New Jersey.

Cuthrell, J. E., & Biegler, L. T. (1987). On the optimization of differential-algebraic process systems. *American Institute of Chemical Engineers Journal 33*, 1257–1270.

Cuthrell, J. E., & Biegler, L. T. (1989). Simultaneous optimization methods for batch reactor control profiles. *Computers and Chemical Engineering 13*, 49–62.

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

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

Edgar, T. F., & Himmelblau, D. M. (1988). *Optimization of chemical processes*. New York: McGraw-Hill.

Fikar, M., Latifi, M. A., Fournier, F., & Creff, Y. (1998). Control-vector parameterization versus iterative dynamic programming in dynamic optimization of a distillation column. *Computers and Chemical Engineering 22*, S625–S628.

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

Friedrich, M., & Perne, R. (1995). Design and control of batch reactors: an industrial viewpoint. *Computers and Chemical Engineering 19*, S357–S368.

Furlonge, H. I., Pantelides, C. C., & Sorensen, E. (1999). Optimal operation of multivessel batch distillation columns. *American Institute of Chemical Engineers Journal 45* (4), 781–800.

Gill, P. E., Murray, W., & Wright, M. H. (1981). *Practical optimization*. London: Academic Press.

Giritligil, S., Cesur, S., & Kuryel, B. (1998). Overview of planning and scheduling of batch process operations. In *IFAC DYCOPS-5* (pp. 461–466). Corfu, Greece.

Goh, C. J., & Teo, K. L. (1988). Control parameterization: a unified approach to optimal control problems with general constraints. *Automatica 24*, 3–18.

Guntern, C., Keller, A. H., & Hungerbühler, K. (1998). Economic optimisation of an industrial semi-batch reactor applying dynamic programming. *Industrial and Engineering Chemistry Research 37* (10), 4017–4022.

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

Hicks, J., Mohan, A., & Ray, W. H. (1969). The optimal control of polymerization reactors. *Canadian Journal of Chemical Engineering 47*, 590.

Ishikawa, T., Natori, Y., Liberis, L., & Pantelides, C. C. (1997). Modelling and optimisation of an industrial batch process for the production of dioctyl phthalate. *Computers and Chemical Engineering 21*, S1239–S1244.

Isidori, A. (1989). *Nonlinear control systems*. Berlin: Springer-Verlag.

Jaspan, R. K., & Coull, J. (1972). Trajectory optimization techniques in chemical reaction engineering. II. Comparison of the methods. *American Institute of Chemical Engineers Journal 18* (4), 867–869.

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

Ku, H. M., & Karimi, I. F. (1990). Scheduling in serial multiproduct batch processes with due-date penalties. *Industrial and Engineering Chemistry Research 29*, 580–590.

Lahteemaki, E., Jutila, E., & Paasila, M. (1979). Profitability as a criterion of batch process control design. *Computers and Chemical Engineering 3* (1–4), 197.

Lee, E. B., & Markus, L. (1967). *Foundations of optimal control theory*. New York: John Wiley.

Lee, E. S. (1968). *Quasilinearization and invariant imbedding with applications to chemical engineering and adaptive control*. New York: Academic Press.

Lim, C., Tayeb, Y. J., Modak, J. M., & Bonte, P. (1986). Computational algorithms for optimal feed rates for a class of fed-batch fermentation: numerical results for penicillin and cell mass production. *Biotechnology and Bioengineering 28*, 1408–1420.

Logsdon, J. S., & Biegler, L. T. (1989). Accurate solution of differential-algebraic optimization problems. *Industrial and Engineering Chemistry Research 28* (11), 1628–1639.

Luus, R. (1992). Optimization of fed-batch fermentors by iterative dynamic programming. *Biotechnology and Bioengineering 41*, 599–602.

Luus, R. (1994). Optimal control of batch reactors by iterative dynamic programming. *Journal of Process Control 4*, 218–226.

Luus, R., & Rosen, O. (1991). Application of dynamic programming to final state constrained optimal control problems. *Industrial and Engineering Chemistry Research 30*, 1525–1530.

Macchietto, S. (1998). Batch process engineering revisited: adding new spice to old recipes. In *IFAC DYCOPS-5*. Corfu, Greece.

Marquardt, W. (1996). Tends in computer-aided modeling. *Computers and Chemical Engineering 20*, 591–609.

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

Mehta, G. A. (1983). The benefits of batch process control. *Chemical Engineering Progress 79* (10), 47–52.

Mujtaba, M., & Macchietto, S. (1997). Efficient optimization of batch distillation with chemical reaction using polynomial curve fitting technique. *Industrial and Engineering Chemistry Research 36* (6), 2287–2295.

Murthy, B. S. N., Gangiah, K., & Husain, A. (1980). Performance of various methods in computing optimal policies. *The Chemical Engineering Journal 19*, 201–208.

Neuman, C. P., & Sen, A. (1973). A suboptimal control algorithm for constrained problems using cubic splines. *Automatica 9*, 601–613.

Palanki, S., Kravaris, C., & Wang, H. Y. (1993). Synthesis of state feedback laws for end-point optimization in batch processes. *Chemical Engineering Science 48* (1), 135–152.

Palanki, S., Kravaris, C., & Wang, H. Y. (1994). Optimal feedback control of batch reactors with a state inequality constraint and free terminal time. *Chemical Engineering Science 49* (1), 85.

Pantelides, C. C., & Britt, H. I. (1994). Multipurpose process modeling environments. In *FOCAPD'94* (pp. 128–141). Snowmass, CO.

Parulekar, S. J., & Lim, H. C. (1985). Modeling, optimization and control of semi-batch fermentation. *Advances in Biochemical Engineering/Biotechnology 32*, 207.

Pollard, G. P., & Sargent, R. W. H. (1970). Off line computation of optimum controls for a plate distillation column. *Automatica 6*, 59–76.

Pontryagin, L. S., Boltyanskil, V. G., Gamkrelidge, R. V., & Mishchenko, E. F. (1962). *The mathematical theory of optimal processes*. New York: Interscience.

Ramirez, W. F. (1997). *Application of optimal control to enhanced oil recovery*. The Netherlands: Elsevier.

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, The Netherlands.

Ray, W. H. (1981). *Advanced process control*. New York: McGraw-Hill.

Ray, W. H., & Szekely, J. (1973). *Process optimization*. New York: John Wiley.

Reklaitis, G. V. (1995). Scheduling approaches for batch process industries. *ISA Transactions 34*, 349–358.

Renfro, I. G., Morshedi, A. M., & Asbjornsen, O. A. (1987). Simultaneous optimization and solution of systems described by differential–algebraic equations. *Computers and Chemical Engineering 11* (5), 503–517.

Rippin, D. W. T. (1983). Design and operation of multiproduct and multipurpose batch chemical plants: an analysis of problem structure. *Computers and Chemical Engineering 7*, 463–481.

Rippin, D. W. T. (1989). Control of batch processes. In *IFAC DYCORD+'89* (pp. 115–125). Maastricht, The Netherlands.

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.

Sacks, M. E., Lee, S., & Biesenberger, J. A. (1972). Optimum policies for batch chain addition polymerizations. *Chemical Engineering Science 27*, 2281.

Sargent, R. W. H., & Sullivan, G. R. (1979). Development of feed change-over policies for refinery distillation units. *Industrial & Engineering Chemistry Process Design and Development 18*, 113–124.

Schulz, V. H., Bock, H. G., & Steinbach, M. C. (1998). Exploiting invariants in the numerical solution of multipoint boundary value problems for DAEs. *Society for Industrial and Applied Mathematics Journal on Scientific Computing 19*, 440–467.

Schwarz, H. R. (1989). *Numerical analysis—a comprehensive introduction*. New York: John Wiley.

Sorensen, E., Macchietto, S., Stuart, G., & Skogestad, S. (1996). Optimal control and on-line operation of reactive batch distillation. *Computers and Chemical Engineering 20* (12), 1491–1498.

Srinivasan, B., Amrhein, M., & Bonvin, D. (1998). Reaction and flow variants/invariants in chemical reaction systems with inlet and outlet streams. *American Institute of Chemical Engineers Journal 44* (8), 1858–1867.

Srinivasan, B., Bonvin, D., Visser, E., & Palanki, S. (2002). Dynamic optimization of batch processes: II. Role of measurements in handling uncertainty. *Computers and Chemical Engineering*. X-ref: S00981354(02)001163.

Srinivasan, B., Myszkorowski, P., & Bonvin, D. (1995). A multicriteria approach to dynamic optimization. In *American control conference* (pp. 1767–1771). Seattle, WA.

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.

Teo, K. L., Goh, C. J., & Lim, C. C. (1989). A computational method for a class of dynamical optimization problems in which the terminal-time is conditionally free. *IMA Journal of Mathematical Control and Information 6* (1), 81–95.

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

Thomas, A., & Kiparissides, C. (1984). Computation of near-optimal temperature and initiator policies for a batch polymerization reactor. *Canadian Journal of Chemical Engineering 62*, 284.

Tsang, T. H., Himmelblau, D. M., & Edgar, T. F. (1975). Optimal control via collocation and nonlinear programming. *International Journal of Control 21* (5), 763–768.

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.

Vassiliadis, V. S., Sargent, R. W. H., & Pantelides, C. C. (1994a). Solution of a class of multistage dynamic optimization problems. 1. Problems without path constraints. *Industrial and Engineering Chemistry Research 33* (9), 2111–2122.

Vassiliadis, V. S., Sargent, R. W. H., & Pantelides, C. C. (1994b). Solution of a class of multistage dynamic optimization problems. 2. Problems with path constraints. *Industrial and Engineering Chemistry Research 33* (9), 2123–2133.

Villadsen, J., & Michelsen, M. L. (1978). *Solution of differential equation models by polynomial approximation*. Englewood Cliffs: Prentice-Hall.

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.

Wiederkehr, H. (1988). Examples of process improvements in the fine chemicals industry. *Computers and Chemical Engineering 43*, 1783–1791.

Wolbert, D., Joulia, X, Koehret, B., & Biegler, L. T. (1994). Flowsheet optimization and optimal sensitivity analysis using exact derivatives. *Computers and Chemical Engineering 18* (11/12), 1083–1095.