Dynamic Optimization of Batch Processes: I. Characterization of the Nominal Solution

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

June 21, 2001

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. The main bottleneck in using optimization in industry is the presence of uncertainty. The most natural way to compensate for uncertainty, and thus to improve process operations, is through the use of measurements. This forms the subject of this series of two papers.

In this first part, the optimal input profiles are expressed in terms of arcs and switching times, of which some push the system to the constraints of the problem while the others exploit the intrinsic compromise present in the system for the purpose of optimality. Such a characterization improves considerably the interpretability of the solution, enhances the numerical efficiency, and acts as a necessary first step towards a measurement-based optimization framework.

Keywords: Dynamic optimization, Optimal control, Batch processes, Chemical reactors, Bioreactors.

1 Introduction

Batch and semi-batch processes are of considerable importance to 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 of consumption, there has been a renewed interest in batch processing [45].

1.1 Characteristics of Batch Processes

In batch operations, all the reactants are charged in a tank initially and processed according to a predetermined course of action during which no material is added or removed. In semi-batch operations, a reactant may be continuously added with no product removal, or a product may be removed continuously 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 are inherently time-varying in nature [11]. This paper considers batch and semi-batch processes in the same manner and, thus, the term 'batch processes' includes semi-batch processes as well.

Schematically, batch process operations involve the following main steps [66, 2]:

- *Elaboration of Production Recipes*: The chemist formulates the chemical synthesis in the laboratory. The recipes provide the range of concentrations, flowrates or temperatures for which the desired reaction and/or separation can take place and the batch operation is feasible. This development step is specific to the product being manufactured [4] and will not be addressed here.

- *Capacity Planning 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 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 [67, 27, 36, 64].

- *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 range provided by the recipes. The optimization of this step is particularly important in order to meet safety [30, 83, 1] and operational constraints [61, 69]. Due to the time-varying nature of batch processes, the optimal values of the process variables need to be adjusted with time. Hence, it involves the rather difficult task of determining time-varying profiles through dynamic optimization.

1.2 Dynamic Optimization in Industry

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 [37, 3, 24]. In the face of increased competition, process optimization is a natural choice for reducing production costs, meeting safety requirements and environmental regulations, improving product quality, reducing product variability, and ease of scale-up [48, 11].

Though the potential gains of optimization could be significant, there has been only a few attempts to optimize operations through mathematical modeling and optimization techniques. The recipes developed in the laboratory are implemented conservatively in production, and the operator uses heuristics gained from experience to adjust the process periodically, which might lead to slight improvements from batch to batch [88]. The implications of the current industrial situation regarding the choice of an appropriate optimization approach is presented in [12]. The stumbling blocks for

the use of mathematical modeling and optimization techniques in industrial practice 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 [46, 55]. The situation is somewhat different in 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, 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 for the uncertainty in the models have simply not been available. Nevertheless, there is a clear indication that recent advances in sensor technology are helping remove the first handicap mentioned above. Developments in chromatographic and spectroscopic sensors enable near on-line (i.e., with minimum time delay) estimation of chemical composition in reacting mixtures [52, 47]. As an example of new and promising developments, rugged NIR spectroscopic sensors can be used to monitor the performance in industrial chemical reactors. The physical measurements (absorbance or reflectance) extend over several hundred wavelengths and are rather specific with respect to chemical composition.

Even when it is assumed that reliable models and measurements are available, from the authors' perspective there exist two reasons for the non-penetration of optimization techniques in the industrial environment:

- *Interpretability of the Optimal Solution*: Optimization is typically performed using a process model and an optimization routine is considered as a black box. Since the resulting optimal solution may not be easy to interpret, it is difficult to convince industry to use such a profile.

- *Measurement-based Optimization Framework*: The theory of optimization available in the literature is model-based, and very little study has been devoted to incorporating uncertainty and measurements into the optimization framework. So, a framework which uses measurements rather than a model of the process for implementing the optimal solution would be very welcome.

This series of two papers addresses the last two issues mentioned above. The present paper deals with the nominal optimal solution for which a characterization is proposed that improves *interpretability*. The companion paper addresses the issue of optimization under uncertainty. It is shown therein that a simplified tendency model and a few available measurements can be used effectively in a so-called *measurement-based optimization* framework.

1.3 Organization of the Paper

The paper is written in a tutorial style. Various problem formulations for the dynamic optimization of batch processes are presented in Section 2. Analytical and numerical solution methods are discussed in Sections 3 and 4 respectively. The characterization of the optimal solution is performed in Section 5. Section 6 presents a series of examples to illustrate the theory, and 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 setpoint around which the process can be regulated. Hence, the major objective in batch operations is *not* to keep the system at an optimal constant setpoint but to *optimize* some objective function that expresses the system performance. Optimizing an objective function amounts 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 may 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 or 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 and the principle of optimality of Hamilton-Jacobi-Bellman. 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 Subsection 4.4.

2.1 Direct Formulation

Dynamic optimization problems were first posed for aerospace applications in the 1950s to minimize a combination of terminal and integral cost functions. These problems can be formulated mathematically as follows [38, 34, 14]:

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

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

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

where J is the scalar performance index to be minimized, x the n-vector of states with known initial conditions x_0, u the m-vector of inputs, S the ζ-vector of path constraints (which include state constraints and input bounds), T the τ-vector of terminal constraints, F a smooth vector function, ϕ a smooth scalar function representing the terminal cost, and t_f the final time.

The problem formulation (1)–(3) is quite general. Even when an integral cost needs to be considered, e.g., $J = \bar{\phi}(x(t_f)) + \int_0^{t_f} L(x,u)dt$, where L a smooth function representing the integral cost, it can be converted into the form (1)–(3) by the introduction of an additional state: $\dot{x}_{cost} = L(x,u),\ x_{cost}(0) = 0$, which results in the terminal cost $J = \bar{\phi}(x(t_f)) + x_{cost}(t_f)$. Let J^*

be the optimal cost of (1)–(3). It is interesting to note that the minimum time problem with the additional constraint $\phi(x(t_f)) \leq J^*$, i.e.,

$$\min_{t_f, u(t)} \quad t_f \tag{4}$$

$$s.t. \quad \dot{x} = F(x,u), \quad x(0) = x_0 \tag{5}$$

$$S(x,u) \leq 0, \quad T(x(t_f)) \leq 0 \tag{6}$$

$$\phi(x(t_f)) \leq J^* \tag{7}$$

will lead to exactly the same optimal inputs as (1)–(3), though the numerical conditioning of the two problems, (1)–(3) and (4)–(7), may differ considerably. The equivalence of solutions is verified using the necessary conditions of optimality (see next subsection). So, without loss of generality, the final time will be assumed fixed in this paper.

2.2 Pontryagin's Formulation

Using Pontryagin's Minimum Principle (PMP), the problem of optimizing the *scalar* cost functional J in (1)–(3) can be reformulated as that of optimizing the Hamiltonian *function* $H(t)$ as follows [58, 14]:

$$\min_{u(t),\mu(t),\nu} H = \lambda^T F(x,u) + \mu^T S(x,u) \tag{8}$$

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

$$\dot{\lambda}^T = -\frac{\partial H}{\partial x}, \quad \lambda^T(t_f) = \left.\frac{\partial \phi}{\partial x}\right|_{t_f} + \nu^T \left.\left(\frac{\partial T}{\partial x}\right)\right|_{t_f} \tag{10}$$

where $\lambda(t) \neq 0$ is the n-vector of adjoint states (Lagrange multipliers for the system equations), $\mu(t) \geq 0$ the ζ-vector of Lagrange multipliers for the path constraints, and $\nu \geq 0$ the τ-vector of Lagrange multipliers for the terminal constraints. The Lagrange multipliers μ and ν are nonzero when the corresponding constraints are active and zero otherwise so that $\mu^T S(x,u) = 0$ and $\nu^T T(x(t_f)) = 0$ always. The necessary condition of optimality is $H_u = \frac{\partial H}{\partial u} = 0$, which implies that x, u, λ, μ, and ν exist such that the following equalities hold:

$$\dot{x} = F(x,u), \quad x(0) = x_0 \tag{11}$$

$$\dot{\lambda}^T = -\frac{\partial H}{\partial x} = -\lambda^T \frac{\partial F}{\partial x} - \mu^T \frac{\partial S}{\partial x}, \quad \lambda^T(t_f) = \left.\frac{\partial \phi}{\partial x}\right|_{t_f} + \nu^T \left.\left(\frac{\partial T}{\partial x}\right)\right|_{t_f} \tag{12}$$

$$\mu^T S = 0, \qquad \nu^T T = 0 \tag{13}$$

$$\frac{\partial H}{\partial u} = \lambda^T \frac{\partial F}{\partial u} + \mu^T \frac{\partial S}{\partial u} = 0 \tag{14}$$

2.3 Hamilton–Jacobi–Bellman formulation

The Hamilton–Jacobi–Bellman (HJB) formulation transforms the problem of optimizing the *scalar* cost functional J in (1)–(3) into the resolution of a partial differential equation by utilizing the

principle of optimality [34, 14]:

$$\frac{\partial V(x,t)}{\partial t} + \min_{u(t),\mu(t),\nu} \left(\frac{\partial V(x,t)}{\partial x} F(x,u) + \mu^T S(x,u) \right) = 0 \tag{15}$$
$$V(x(t_f), t_f) = \phi(x(t_f)) + \nu^T \, T(x(t_f))$$

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

$$\lambda^T = \frac{\partial V}{\partial x} \tag{16}$$

The term to be minimized in (15) is the Hamiltonion H. Thus, the partial differential equation (15) represents the time evolution of the adjoints (10):

$$\dot{\lambda}^T = \frac{d}{dt}\frac{\partial V}{\partial x} = \frac{\partial}{\partial x}\frac{\partial V}{\partial t} = -\frac{\partial H_{min}}{\partial x} \tag{17}$$

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

3 Analytical Solution Methods

A complete analytical solution for the optimization problem (1)-(3) is not feasible except for very simple cases. However, analytical expressions can be computed for pieces of the optimal solution, which in turn can be used to: i) understand the types of arcs that can constitute the solution, and ii) enhance the efficiency of the numerical optimization algorithm.

In dynamic optimization problems, the optimal values of the manipulated variables are either determined by the constraints of the problem or by the compromises present in the system. So, there are certain intervals in time during which the inputs are determined by path constraints and other intervals where they are in the interior of the feasible region to take advantage of the compromise. The time at which the inputs switch from one interval to another is called a *switching time*. Thus, the optimal solution is seen to possess the following properties:

- The inputs may be discontinuous; yet in between discontinuities the inputs are analytic [14].
- 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.

3.1 Piecewise Analytical Expressions for the Optimal Inputs

Analytical expressions for the inputs can be obtained from the necessary condition of optimality based on PMP. The results presented here extend those provided in [53, 54, 59] for single-input control-affine systems.

Consider the necessary condition of optimality for input u_i:

$$H_{u_i} = \frac{\partial H}{\partial u_i} = \lambda^T \frac{\partial F}{\partial u_i} + \mu^T \frac{\partial S}{\partial u_i} = \lambda^T F_{u_i} + \mu^T S_{u_i} = 0 \tag{18}$$

H_{u_i} has two parts, the system dependent part $\lambda^T F_{u_i}$ and the constraints dependent part $\mu^T S_{u_i}$. When $\lambda^T F_{u_i} \neq 0$ in a certain interval, then equation (18) requires $\mu \neq 0$ in that interval. So, one of the path constraints is 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 $\mu \geq 0$:

$$u_i = \begin{cases} u_{i,max} & \text{for} \quad \lambda^T F_{u_i} < 0 \\ ? & \text{for} \quad \lambda^T F_{u_i} = 0 \\ u_{i,min} & \text{for} \quad \lambda^T F_{u_i} > 0 \end{cases} \tag{19}$$

For the case $\lambda^T F_{u_i} = 0$, it might be possible to obtain u_i as a function of x and λ from this condition. However, it may also happen that $\lambda^T F_{u_i}$ is independent of u_i as is the case in control-affine systems, i.e, $F(x,u) = f(x) + G(x)\,u$, for which $F_{u_i} = G_i(x)$ is independent of u.

3.1.1 Differentiation of H_{u_i}

When $\lambda^T F_{u_i} = 0$ and u_i cannot be obtained directly from this condition, the following idea is used. Since $H_{u_i} = 0$ for all t, its time derivatives $\frac{d^j}{dt^j} H_{u_i} = 0,\ \forall j \geq 0$. Differentiating (18) once with respect to time leads to:

$$\frac{dH_{u_i}}{dt} = \dot{\lambda}^T F_{u_i} + \lambda^T \left(\frac{\partial F_{u_i}}{\partial x}\ \dot{x} + \frac{\partial F_{u_i}}{\partial u}\ \dot{u} \right) + \sum_{j=1}^{\zeta} \left(\dot{\mu}_j \frac{\partial S_j}{\partial u_i} + \mu_j \frac{d}{dt} \frac{\partial S_j}{\partial u_i} \right) \tag{20}$$

The summation term in (20) stems from the path constraints and is equal to zero as shown next. From (18) and $\lambda^T F_{u_i} = 0$, $\mu^T S_{u_i} = 0$ during the interval which leads to two possibilities: (i) the constraint $S_j(x,u)$ is not active and $\mu_j = 0$; also, $\dot{\mu}_j = 0$ since $\mu_j = 0$ over an interval, and so the two terms of the summation are zero; (ii) $S_j(x,u)$ is active. This implies $\mu_j \neq 0$ but $\frac{\partial S_j}{\partial u_i} = 0$ to satisfy $\mu^T S_{u_i} = 0$. Also, $\frac{d}{dt}\frac{\partial S_j}{\partial u_i} = 0$ since $\frac{\partial S_j}{\partial u_i} = 0$ over an interval, and so the two terms of the summation are zero. Thus, the summation in (20) can be dropped. Using (11)-(12) for $\dot{x}$ and $\dot{\lambda}$ gives:

$$\begin{aligned} \frac{dH_{u_i}}{dt} &= \lambda^T \left(\frac{\partial F_{u_i}}{\partial x}\ F - \frac{\partial F}{\partial x} F_{u_i} + \frac{\partial F_{u_i}}{\partial u}\ \dot{u} \right) - \mu^T \frac{\partial S}{\partial x} F_{u_i} && (21) \\ &= \lambda^T \Delta F_{u_i} - \mu^T \frac{\partial S}{\partial x} F_{u_i} && (22) \end{aligned}$$

where the operator Δ is given by:

$$\Delta v = \frac{\partial v}{\partial x} F - \frac{\partial F}{\partial x} v + \sum_{k=0}^{\infty} \frac{\partial v}{\partial u^{(k)}} u^{(k+1)} \tag{23}$$

with $u^{(k)}$ representing the k^{th} time differentiation of u. The summation is introduced in (23) since, in general, v is not only a function of u but also of its time derivatives. The operator Δ represents the time differentiation of a vector function v along the trajectories of the dynamic system (2) and is studied in the systems literature using tools of Lie algebra.

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

$$\frac{d^{\,j} H_{u_i}}{dt^{\,j}} = \lambda^T \Delta^j F_{u_i} - \mu^T \frac{\partial S}{\partial x} \Delta^{j-1} F_{u_i} = 0 \tag{24}$$

Note that $\Delta^2 v = \Delta(\Delta v)$, 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 $\lambda^T \Delta^j F_{u_i} \neq 0$ or u_i appears explicitly in $\Delta^j F_{u_i}$. This gives rise to two intrinsically different solutions that are discussed below:

1. **Active path constraints:** If $\lambda^T \Delta^{\varsigma_i} F_{u_i} \neq 0$ after the $\varsigma_i{}^{\text{th}}$ time differentiation of H_{u_i}, a nonzero μ is required to satisfy (24). This implies that at least one of the path constraints is active. Though different choices of μ are possible to satisfy $\frac{d^{\varsigma_i}}{dt^{\varsigma_i}} H_{u_i} = 0$, the non-negativity of μ restricts this choice. Furthermore, since only one of the constraints will be active, i.e., the most restrictive of the possible constraints, μ will indicate the constraint from which the input u_i can be determined. To compute the optimal input u_i, the active constraint needs to be differentiated ς_i times. This means that only those constraints which have relative degree ς_i can be active. Recall that *the relative degree of* $S(x,u)$ is the number of time differentiations of $S(x,u)$ that are necessary for the input u_i to appear explicitly [53, 14].

2. **Solution in the interior of the feasible region:** If $\lambda^T \Delta^{\sigma_i} F_{u_i} = 0$ but u_i appears explicitly in $\Delta^{\sigma_i} F_{u_i}$, the input u_i can be determined as a function of the states and adjoints from the conditions $\lambda^T \Delta^j F_{u_i} = 0$, for $j = 0, 1, \cdots, \sigma_i$ [1].

   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 x are not affected by the input u_i and, conversely, there exist $(n - \rho_i)$ directions in λ that do not affect u_i. Also, since the adjoints enter linearly in $\lambda^T \Delta^j F_{u_i} = 0$, 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 above mentioned independence and $(\sigma_i + 1)$ from the optimality conditions. So, the optimal input u_i will depend on $n - (n - \rho_i) - (\sigma_i + 1) = (\rho_i - \sigma_i - 1)$ adjoint variables.

   The following classification can be made depending on the relative values of σ_i and ρ_i:

   - $\sigma_i < \rho_i - 1$: The optimal input u_i depends on $(\rho_i - \sigma_i - 1)$ adjoint variables, for the computation of which differential equations need to be solved. So, the feedback is *dynamic* in nature.
   - $\sigma_i = \rho_i - 1$: The optimal input is independent of the adjoint variables. This leads to a feedback which is *static* in nature.
   - $\rho_i - 1 < \sigma_i < \infty$: This corresponds to the system being on a *surface*, where the relative degree of the surface with respect to u_i is $(\sigma_i + 1 - \rho_i)$.

[1] Some authors use the order of singularity, σ_i, which is the number of time differentiations of H_{u_i} required for the input u_i to appear explicitly [35, 53] while others use the degree of singularity, s_i, which is the highest time derivative which is still independent of the input. Thus, $s_i = \sigma_i - 1$ [53, 54]

- $\sigma_i = \infty$: If $\rho_i = n$, the input u_i is determined by active path constraints [8]. 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 [5].

Though an analytical expression is provided for the optimal input u_i, the expression may depend on $u_j,\ j \neq i$. Thus, a set of coupled dynamic equations may have to be solved in order to determine the input vector u.

3.1.2 Adjoint-free Input Computation

When the input u_i is computed from an active constraint or when it is in the interior of the feasible region with $\sigma_i \geq (\rho_i - 1)$), the optimal solution does not depend on the adjoint variables. To obtain the optimal input independently of λ even when $\sigma_i < (\rho_i - 1)$, the following idea is used. Instead of stopping the differentiation of H_{u_i} when u_i appears explicitly in $\frac{d^{\sigma_i} H_{u_i}}{dt^{\sigma_i}}$, it is continued till the $(\rho_i - 1)^{\text{th}}$ differentiation. Thus, using the conditions $\Delta^j F_{u_i} = 0$, for $j = 0, 1, \cdots, \rho_i - 1$, all the adjoint variables can be eliminated. However, the optimality conditions will also depend on the derivatives of u_i up to the order $(\rho_i - \sigma_i - 1)$. So, a system of differential equations of order $(\rho_i - \sigma_i - 1)$ needs to be integrated to compute the optimal feedback, the initial conditions of which constitute additional decision variables.

3.1.3 Adjoint-free Analysis of the Types of Arcs

Though the knowledge of the adjoint variables is crucial to determine whether the solution is on active constraints or in the interior of the feasible region, the optimal inputs can be computed independently of the adjoints in each interval. This leads to the idea of analyzing the system equations and the path constraints without involving the cost or the adjoint equations, to decipher the types of arcs that can exist in the solution. However, this analysis is conservative in the sense that it provides all possible types of arcs that might occur and not those actually present. Therefore, though the analysis might indicate the possibility of having the solution in the interior of the feasible region, it might happen that, for the given optimization problem, the solution is always determined by the path constraints. Another disadvantage is that it does not provide the sequence of arcs.

1. **Active path constraints:** Each path constraint $S_j(x, u)$ is differentiated along the trajectories of (2) as illustrated below:

$$\frac{dS_j}{dt} = \frac{\partial S_j}{\partial x} F + \frac{\partial S_j}{\partial u} \dot{u} \tag{25}$$

   Time differentiation of $S_j(x, u)$ is continued until the input u_i appears in $\frac{d^{\varsigma_{ij}} S_j}{dt^{\varsigma_{ij}}}$. $\varsigma_{ij} = \infty$ indicates that the input u_i does not influence the constraint S_j, and thus, u_i cannot be be determined from S_j. In contrast, when $\varsigma_{ij} < \infty$, the input u_i obtained from $\frac{d^{\varsigma_{ij}} S_j}{dt^{\varsigma_{ij}}} = 0$ constitutes a possible optimal input.

2. **Solution in the interior of the feasible region:** To analyze the solution within the feasible region, F_{u_i} is differentiated along the trajectories of (2) as in (24). Consider the matrix

$$\mathcal{M}_i = \left[F_{u_i} \vdots \Delta^1 F_{u_i} \vdots \cdots \vdots \Delta^{\rho_i - 1} F_{u_i} \vdots \cdots \right] \tag{26}$$

The differentiation is not stopped when u_i appears explicitly in $\Delta^{\sigma_i} F_{u_i}$, but continued until the structural rank of $\mathcal{M}_i$ does not augment any further. In fact, ρ_i, which is the dimension of state space that can be reached by manipulating u_i, is the structural rank of $\mathcal{M}_i$.

The rank of $\mathcal{M}_i$ is a function of the states and the inputs. The optimal input u_i not being on active path constraints or, equivalently, being in the interior of the feasible region corresponds to the rank of $\mathcal{M}_i$ being lower than its structural rank. The four cases for the value of σ_i discussed above can be revisited in this context. When $\sigma_i < \rho_i - 1$, $\mathcal{M}_i$ loses rank for a specific combination of $x, u, \dot{u}, \cdots, u^{(\rho_i - \sigma_i - 1)}$, while for $\sigma_i = \rho_i - 1$, the rank loss is for a combination of x and u only. In the case when $\sigma_i > \rho_i - 1$, the rank of $\mathcal{M}_i$ depends only on x, and for $\sigma_i = \infty$, $\mathcal{M}_i$ does not lose rank at all.

If $\rho_i = n$, the optimal input u_i is obtained from the condition $\det(\mathcal{M}_i) = 0$. 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 such a case, the determinant of the submatrix of $\mathcal{M}_i$ consisting of the first ρ_i rows can be used to compute the optimal input.

The optimal input being in the interior of the feasible region corresponds to physical compromises and tradeoffs intrinsic to the system. The fact that there are no intrinsic tradeoffs is represented by the condition $\sigma_i = \infty$ and is important for practical applications. This guarantees that the optimal solution is always on the path constraints. These conditions are satisfied in controllable linear systems, feedback-linearizable systems, flat systems, and involutive-accessible systems, a category which encompasses many practical systems [53, 8].

3.1.4 Control-affine Systems

When the system is control affine, i.e., $F(x, u) = f(x) + G(x)\, u$, the inputs cannot be obtained directly from (18). Control-affine systems are quite common in batch chemical processes, where the manipulated inputs include flowrates of hot and cold fluids, and flowrates of reactants.

In the literature on optimal control of control-affine systems, the intervals where the inputs cannot be obtained directly from (18) are referred to as *singular* intervals [14]. In contrast, intervals where the inputs are computed from (18) are referred to as *nonsingular* intervals. When the path constraints are state-independent (typically, bounds on the inputs), a nonsingular interval implies that the optimal solution is determined by the input bounds, and a singular intervals implies that the optimal solution is inside the feasible region. This analogy does not hold in the presence of state-dependent path constraints (there exist singular intervals where the inputs are determined by path constraints) and when the dynamics do not possess the control-affine structure (there exist nonsingular intervals where the inputs are inside the feasible region). So, instead of using the singular/nonsingular terminology, which is widely used in the community dealing with control-affine systems, the discussion here focuses on whether or not the inputs are determined by the active path constraints.

3.1.5 Limitation of the Analytical Approach

The main disadvantage of the analytical approach is that it involves symbolic computations which become arduous for high-order systems. However, if the goal is mainly to understand the arcs that can constitute the optimal solution, it is often sufficient to use a simplified model of the system that represents the intrinsic physical compromises. This way, the amount of symbolic computation can be significantly reduced.

3.2 Input Parameterization

As discussed earlier, except for some simple examples, a numerical approach is necessary to solve the optimization problem (1)-(3). Since, the decision variables $u(t)$ are *infinite dimensional*, the inputs need to be parameterized using a finite set of parameters in order to utilize numerical techniques, . A piecewise constant or polynomial approximation of the inputs is often utilized. However, instead of relying on an approximation, it is possible to use the analytical expressions presented in the previous subsection.

Parameterization *via* initial values of adjoints: By solving (14) the optimal inputs u can be written as a function of x and λ, i.e., $u(x, \lambda)$. Then, the state and adjoint equations (11)-(13) read:

$$\dot{x} = F(x, u(x, \lambda)), \quad x(0) = x_0 \tag{27}$$

$$\dot{\lambda}^T = -\frac{\partial H}{\partial x}(x, \lambda), \quad \lambda^T(t_f) = \left.\frac{\partial \phi}{\partial x}\right|_{t_f} + \nu^T \left.\left(\frac{\partial T}{\partial x}\right)\right|_{t_f} \tag{28}$$

$$\mu^T S = 0, \qquad \nu^T T = 0 \tag{29}$$

So, once the initial conditions $\lambda(0)$ are specified, (27)-(28) can in principle be integrated to give $\lambda(t)$. Thus, the initial conditions $\lambda(0)$ completely characterize the optimal inputs, thereby providing an efficient parameterization. This parameterization, however, suffers from numerical problems resulting from integrating the adjoint equations forward in time and also from discontinuities in the adjoint variables in the presence of state constraints.

Parameterization *via* switching times: It was shown that, for each interval, it is possible to obtain analytical expressions for the optimal inputs that do not depend on the adjoint variables. So, if the sequence of intervals is known, the switching times and, possibly, the initial conditions for the dynamic feedback completely parameterize the inputs.

In comparison with piecewise constant or piecewise polynomial approximations, the parameterization proposed is *exact and parsimonious*. In comparison with choosing $\lambda(0)$, the adjoint system need not be integrated. This leads to numerical advantages and works well in the presence of state constraints. On the other hand, since the proposed parameterization treats every interval separately, the global picture is lost. So, the choice of the sequence of intervals needs to be handled separately.

This parameterization is quite appropriate and very effective when the solution is determined by the constraints, which is the case for many batch processes. In contrast, when applied to problems

like the linear quadratic problem that have a low-order of singularity ($\sigma_i = 0$), this parameterization boils down to choosing the initial conditions of the adjoints.

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 (11)–(14) are solved.
3. HJB-based methods, where the partial differential equation (15) is solved.

These methods are briefly described below.

4.1 Direct Optimization Methods

As seen in Section 3.2, the inputs need to be parameterized using a finite set of parameters. 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 a given $u(t)$, the differential equations (2) are integrated using standard integration algorithms to evaluate the objective function J. This corresponds to a "feasible" path approach since the differential equations are satisfied at each step of the optimization algorithm. The basic procedure is as follows:

1. Parameterize the inputs using a finite number of decision variables.
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 S and T.
4. Use an optimization algorithm (such as steepest descent [26]) to update the values of the decision variables. Repeat steps 3-4 until the objective function is minimized.

Typically, a piecewise constant approximation over equally spaced time intervals is made for the inputs and the method is referred to as Control Vector Parameterization (CVP) in the literature

[62, 22, 79]. This procedure has been utilized in various forms for solving dynamic optimization problems in chemical engineering applications [57, 71]. For instance in [84] and [85], a numerical algorithm is developed based on CVP to solve a class of multi-stage dynamic optimization problems that are modeled by a differential algebraic system of index 1. In this approach, CVP is coupled with a backward-difference formula for the integration. Inequality path constraints are handled through the combined application of discretization of the constraints at a finite number of points, and forcing an integral measure of their violation to zero. In [73], the CVP approach is utilized to compute optimal profiles of the operating variables for an industrial reactive distillation process. The approach of [85] is used in [32] for the optimization of an industrial batch process for the production of dioctyl phthalate, in [49] for the optimization of a single batch distillation unit, and in [25] for computing the optimal operating policies in multivessel batch distillation columns. CVP is compared to iterative dynamic programming for the dynamic optimization of a distillation column in [23].

While the CVP approach is straightforward to implement, it tends to be slow, especially in dealing with inequality path constraints [6]. 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 [41]. In that respect, the parsimonious parameterization presented in Section 3.2 can be very helpful.

Sequential approach based on parsimonious parameterization: The main stumbling block in using the parsimonious parameterization along with the sequential approach is the choice of the type and sequence of intervals. In general, a mixed-integer type of algorithm is necessary for this purpose. However, an initial sequence of intervals can be guessed and the sequence determined iteratively upon checking the necessary conditions. The basic procedure is summarized below:

1. Choose an initial sequence of intervals.
2. Determine *numerically* the switching times and, possibly, the initial conditions for the dynamic feedback using the sequential approach.
3. Compute the adjoint variables for the resulting optimal solution by integrating (10) 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.

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. So, in general, the differential equations are satisfied only at the *solution* of the optimization problem [84]. This is therefore called an "infeasible path" approach. The basic procedure is as follows:

1. Parameterize both inputs and states using a finite number of decision variables (typically piecewise polynomials).

2. Discretize the differential equation (2) for selected time instants, 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. Solve for the optimal set of decision variables by using an NLP code.

Since the above procedure typically leads to a large NLP, efficient numerical methods are necessary to solve this problem [26]. With the development of Successive Quadratic Programming (SQP) and MINOS, the NLPs resulting from the simultaneous approach can be solved efficiently [9]. Even high dimensional optimization problems have been solved with orthogonal collocation on finite elements and piecewise constant approximations of control profile [65]. The role of finite elements in terms of node locations and breakpoints that allow for control profile discontinuities is studied in [18, 19]. This leads to a formulation that enforces accurate solution of the differential equations and allows for a general description of the control profiles. The stability and error properties of implicit Runge-Kutta methods for solving DAEs are considered in [41]. The same authors also use these properties to enforce error constraints and method orders in a collocation-based NLP formulation of dynamic optimization problems. A numerical algorithm using a reduced space SQP algorithm is developed in [15].

The simultaneous approach, which was first used successfully in [51] and [82], has been demonstrated in several batch reactor applications. For instance, in [65], the optimal temperature profile to maximize the production of an intermediate product is computed using a global spline collocation. In [21], the simultaneous approach is utilized to calculate the sensitivity of the optimal solution to changes in model parameters. This framework is also used in [68] to develop an efficient technique for the optimization of batch processes under uncertainty.

The use simultaneous methods requires awareness of the tradeoff between approximation and optimization [76]. 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 [86, 80].

4.2 PMP-based Methods

The key to PMP-based methods is the necessary conditions of optimality (14). This, on the one hand, can provide a closed-form expression for the optimal inputs as a function of the state and adjoint variables. On the other hand, the gradient information $\frac{\partial H}{\partial u}$ available from (14) can be used to generate the search direction in gradient-based methods.

4.2.1 Shooting Method

In the shooting approach [63, 13], the optimization problem is cast into that of solving a system of differential-algebraic equations. This means that a search is done for x, u, λ, μ, and ν such that (27)–(29) are satisfied. Also, the state equations (27) and the adjoint equations (28) need to be solved *simultaneously*. However, the boundary conditions for the state and adjoint equations are split, i.e., the *initial* conditions of the state equations and the *terminal* conditions of the adjoint equations are known. Thus, the PMP approach leads to a two-point boundary value problem (TPBVP).

The shooting method treats a TPBVP as a multidimensional root finding problem, where $\lambda(0)$ are the roots to be found in order to satisfy $\lambda(t_f)$. The basic procedure is as follows:

1. Solve (14) analytically for $u(x, \lambda)$.
2. Parameterize $\mu(t)$ using a finite number of variables. The vector of decision variables also includes $\lambda(0)$ and ν.
3. Choose an initial guess for the decision variables.
4. Integrate (27) and (28) forward in time using $x(0)$, $\lambda(0)$, and $u(x, \lambda)$ obtained in Step 3, and compute $\lambda(t_f)$.
5. Check if equations (28) and (29) are verified; for the terminal conditions $\lambda(t_f)$, the values obtained by integration in Step 4 should match those specified in (28). Update the decision variables (using for example steepest descent [26]) and repeat Steps 4–5 until convergence.

The use and updating of the Lagrange multipliers μ and ν depend on the problem formulation and the type of optimization algorithm. If a constrained optimization routine is used, since the Lagrange multipliers are handled internally, μ and ν need not be considered explicitly as decision variables.

The shooting method [14, 34], also referred to as boundary condition iteration (BCI) [33], has been used in several batch applications. In polymerization, where the profiles of temperature and initiator concentration represent the inputs, this approach has been used for the optimization of free-radical polymerization [31, 70], batch bulk polymerization [16], and batch methyl methacrylate polymerization [81]. Optimal glucose feeding policy in a fed-batch fermentation for the production of penicillin is computed in [56, 40]. The optimal temperature and pH profiles for a batch cephalosporin C fermentation is calculated in [17].

There are several difficulties with this approach [50]. First, the method can have stability problems in integrating the adjoint equations forward in time. Furthermore, unless good initial guesses for the adjoint variables are available (which is rarely the case since the adjoints represent sensitivities), it is computationally expensive to find the optimal solution. The method does not work with discontinuities in the adjoints, which is typical in the presence of state constraints. Additional degrees of freedom are necessary to handle these situations.

4.2.2 Gradient Method

The necessary conditions of optimality (14) also provide the gradient along which the decision variables can be updated. The solution via the gradient method resembles the sequential approach of the direct formulation except that the gradient is calculated using (14). The basic procedure is as follows:

1. Parameterize u and μ using a finite number of variables. The vector of decision variables also includes ν.
2. Choose an initial guess for the decision variables.
3. Integrate the state equations (11) from 0 to t_f.
4. Integrate the adjoint equations (12) backward in time from t_f to 0 and compute the gradient $\frac{\partial H}{\partial u}$ using (14).
5. Use an optimization algorithm (such as steepest descent [26]) to update the values of the decision variables. Repeat Steps 3–5 until H is minimized.

This approach has been applied widely [33, 62, 20, 60]. The main advantage of the gradient method is that the initial guess of the decision variables is beneficial but not critical to the convergence. In contrast, the drawbacks are: (i) slow convergence close to the optimum, and (ii) large number of decision variables that may be necessary to parameterize the inputs.

Control vector iteration (CVI) follows the same basic procedure except that the input parameterization is not explicitly incorporated in the algorithm [62]. However, for any practical implementation of CVI, the inputs need to be parameterized.

4.2.3 State and Adjoint Parameterization

Two approaches will be discussed below where the states and adjoints are parameterized.

Discretization (NR)

This approach is similar to the simultaneous direct optimization method, where parameterization of the states and adjoints followed by appropriate discretization is used to avoid the problems encountered in the integration of the adjoints [28]. The basic procedure is as follows:

1. Parameterize x, u, λ, and μ using a finite number of decision variables (typically piecewise polynomials). The vector of decision variables also includes ν.
2. Discretize the differential equations (11)–(12) and the necessary conditions (14) for a finite number of time instants (typically *via* orthogonal collocation). These two steps transform the problem of solving a set of nonlinear differential-algebraic equations (11)–(14) into that of solving a set of nonlinear algebraic equations.
3. Choose an initial guess for the decision variables.

4. Solve for the optimal set of decision variables by using, for example, the Newton–Raphson (NR) algorithm [72] .

Quasi-linearization (QL)

In this approach, the analytical expression for the inputs provided by (14) is used. The two-point boundary value problem (27)–(29) is solved by successive linearization [14, 34, 39]. The basic procedure is as follows:

1. Parameterize x, λ, and μ using a finite number of decision variables (typically piecewise polynomials). The vector of decision variables also includes ν.
2. Choose an initial guess for the decision variables.
3. Linearize the differential equations (27)–(28) around the current guess. This transforms the problem of solving a set of nonlinear differential-algebraic equations (27)–(29) into that of solving 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, repeat Steps 3–5 until convergence.

The discretization and quasi-linearization methods work well if the solution is smooth and if 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 [76]. Also, as in the shooting method, good initial guesses are needed for these methods to work well.

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' [7]. This approach is equivalent to computing $V(x,t)$ in (15) with discretization performed in both states and time. The minimization in (15) is performed using exhaustive search. To make the search feasible, the domain of search has to be restricted. Hence, the inputs are also discretized both in time and amplitude.

Consider (15) integrated for an arbitrary small interval $[t, t+\Delta t]$:

$$\min_{u([t,t+\Delta t]),\mu([t,t+\Delta t]),\nu} \left(\frac{\partial V}{\partial t}\Delta t + \frac{\partial V}{\partial x}\frac{dx}{dt}\Delta t + \mu^T S\,\Delta t \right) = 0 \tag{30}$$

Since the first two terms of the minimization correspond to the difference $V(x(t+\Delta t), t+\Delta t) - V(x(t),t) = \frac{\partial V}{\partial t}\Delta t + \frac{\partial V}{\partial x}\frac{dx}{dt}\Delta t$, (30) can be written as:

$$\min_{u([t,t+\Delta t]),\mu([t,t+\Delta t]),\nu} \Big(V(x(t+\Delta t), t+\Delta t) - V(x(t),t) + \mu^T S\,\Delta t \Big) = 0 \tag{31}$$

The time interval $[0, t_f]$ is divided into P stages, with $[t_p, t_{p+1}]$ being the time interval corresponding to the $(p+1)^{th}$ stage. Integrating over the time interval $[t_p, t_{p+1}]$, the return function at time t_p can be written as:

$$V(x_p, t_p) = \min_{u([t_p,t_{p+1}]),\mu([t_p,t_{p+1}]),\nu} \left(V(x_{p+1}, t_{p+1}) + \int_{t_p}^{t_{p+1}} \mu^T S \, dt \right) \tag{32}$$

where x_{p+1} is the state at t_{p+1} obtained by integrating the system with inputs u and the initial condition $x(t_p) = x_p$ over the interval $[t_p, t_{p+1}]$ Since the boundary condition of V is known at final time, (32) is solved iteratively for decreasing values of p.

A complication arises from the state discretization since $V(x_{p+1}, t_{p+1})$ will only be calculated for a set of discrete values. When integration is performed from a discretization point x_p^d at time t_p, x_{p+1} will typically not correspond to a discretization point. Thus, the question is how to calculate the return function at x_{p+1}. One option is to interpolate between the return functions at various discretization points at time t_{p+1}. Another one, which will be used here, is to merely use the optimal control $u([t_{p+1}, t_f])$ that corresponds to the grid point closest to x_{p+1} and integrate the system from t_{p+1} to t_f to get the return function. The basic procedure is as follows [7, 34]:

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, $i = 1, 2, \cdots, m$.
3. Choose a region for each input, $\mathcal{R}_{ip}$, $i = 1, 2, \cdots, m$, and $p = 1, 2, \cdots, P$.
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 with all the allowable values of the inputs. To continue integration from t_p to t_f, choose the optimal inputs from the earlier stages that correspond to the grid point closest to the resulting 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 $\mathcal{R}_{ip}$ for the allowable input values by using the best input policy as the midpoint for the allowable input values at each stage. Repeat Steps 3–7 until a specified tolerance for the $\mathcal{R}_{ip}$ is reached.

This approach has been used in numerous batch applications [44, 42, 43, 10, 29]. The two key advantages of this method are: (i) dynamic programming is one of the few methods available to compute the *global* minimum, and (ii) the number of iterations, and thereby the time needed for the optimization, can be estimated *a priori* (specified only by the tolerance for the $\mathcal{R}_{ip}$). In addition, the approach provides a feedback policy which 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 Comparison of Numerical Optimization Schemes

Numerical Solution	Problem Formulation		
	Direct	PMP	HJB
States - continuous Inputs - continuous	-	Shooting method (BCI)	-
States - continuous Inputs - parameterized	Sequential approach (CVP)	Gradient method (CVI)	-
States - parameterized Inputs - parameterized	Simultaneous approach (NLP)	State and adjoint parameterization (NR, QL)	Dynamic programming (DP)

Table 1: Classification of numerical optimization schemes

Table 1 classifies the different numerical schemes detailed in the preceding subsections for solving dynamic optimization problems. Parameterization of the inputs and/or states is, in general, needed to render the problem numerically tractable. However, explicit integration of system equations can be used to avoid parameterization of the states. Also, the PMP formulation can provide analytical expressions for the optimal inputs, thereby avoiding the necessity of input parameterization. The three places in the table where there is no entry correspond to infinite dimensional problems that cannot be solved numerically.

Direct optimization methods are by far the best since PMP-based methods are often ill-conditioned and dynamic programming can be computationally expensive. The main disadvantage of direct optimization methods is that the input parameterization is often chosen arbitrarily by the user and only represents an approximation to the optimal solution. A large number of parameters may be required to accurately represent the optimal solution, which increases the number of decision variables. Thus, the efficiency of direct optimization methods depends crucially on the way the inputs are parameterized.

On the other hand, the necessary conditions of PMP provide piecewise analytical expressions for the optimal inputs. Thus, PMP can complement the direct optimization approach by providing a parsimonious parameterization of the inputs. Though not applicable to all problems, the sequential approach using parsimonious parameterization can be used for a large variety of batch processes. In many batch optimization problems, it can be shown *a priori* that the solution is on the constraints [78]. Then, there is no necessity to search for the optimal inputs in the interior of the feasible region, thereby enhancing the numerical efficiency of the optimization procedure.

5 Characterization of the Optimal Solution

In the present context, characterization means understanding or interpretation of the solution. A characterization of the optimal solution is proposed that: i) treats the instantaneous objectives and terminal objectives independently, and ii) separates *constraint-seeking variables* from *compromise-seeking variables*. The instantaneous objectives correspond to either being on the path constraints or following inputs which are inside the feasible region, while the terminal objectives correspond to either being on the terminal constraint or optimizing the terminal cost. The constraint-seeking variables are those that push the system to the (path and terminal) constraints of the problem, while compromise-seeking variables exploit the intrinsic compromises present in the system for optimizing the cost.

5.1 Separation of Instantaneous and Terminal Objectives

Structure of the optimal solution: The structure of the optimal solution consists of: i) the type of intervals (set of active path constraints), ii) sequence of intervals and the iii) active terminal constraints. Using this information, the optimal inputs can be dissected into: i) the values of the inputs in the various arcs, $\eta(t)$, and ii) the minimal parameterization, π, that typically consists of the switching instants. As will be shown later, $\eta(t)$ caters to instantaneous objectives, while π handles terminal objectives.

Determination of the structure of the optimal solution: The structure of the optimal solution can be obtained in three ways:

- educated guess by an experienced operator,
- piecewise analytical expressions for the optimal inputs,
- inspection of the solution obtained from numerical optimization.

In the first case, the operator provides the sequence of optimal operations based on intuition and experience. In the second case, an analysis of the system equations, problem formulation, and constraints indicates the types of intervals that can be present in the optimal solution. The sequence of intervals, however, is not trivial. It can be determined by trial-and-error, using necessary conditions of optimality to assess the validity of successive trials. In the third case, visual inspection of the numerical solution can be used to identify the various intervals present in the solution. Each interval has to be tagged according to the type it could represent. The analytical expressions for the inputs can be used for verification.

Minimal input parameterization: Let (i) the type and sequence of intervals be known, and (ii) the active path constraints be kept active, for example by tracking them using appropriate controllers. Then, the inputs can be parameterized using the switching times and, possibly, additional decision variables corresponding to the inputs that are not determined by the active path constraints.

Among the switching instants, a few correspond to reaching the path constraints in minimum time. These switching instants do not contribute to the terminal objectives since the controllers that keep

the corresponding path constraints active will correct the effect of any deviation in these switching instants. Removing these switching instants leads to a minimal input parameterization, π, that is necessary to handle the terminal objectives (meet the terminal constraints and optimize the terminal cost).

Using this parameterization, the optimization reduces to minimizing a terminal cost subject to terminal constraints *only*. Let the inputs be represented by $u(\pi, x, t)$. Then, the optimization problem (1)–(3) can be rewritten as:

$$\begin{aligned} \min_{\pi} \; J &= \phi(x(t_f)) && (33)\\ s.t. \quad & \dot{x} = F(x, u(\pi, x, t)), \quad x(0) = x_0 && (34)\\ & T(x(t_f)) \leq 0 && (35) \end{aligned}$$

The necessary conditions of optimality for (33)-(35) are:

$$\nu^T \, T(x(t_f)) = 0 \quad \text{and} \quad \frac{\partial \phi}{\partial \pi} + \nu^T \frac{\partial T}{\partial \pi} = 0 \tag{36}$$

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

5.2 Separation of Constraint-seeking and Compromise-seeking Variables

In any optimization problem, the optimal values of the decision variables are either determined by the constraints of the problem or by certain compromises present in the system. It would be useful to find a transformation that separates the two effects. Such a separation should also be done along the time axis when the decision variables are functions of time, as is the case here with $u(t)$.

Separation of constraint-seeking and compromise-seeking input directions: In each interval, some of the path constraints (or possibly none) can 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 which have no effect on meeting the path constraints.

Let $\bar{\zeta}$ be the number of active path constraints in a given interval. Clearly, $\bar{\zeta} \leq m$. 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 $\eta(t)^T \rightarrow [\bar{\eta}(t)^T \;\; \tilde{\eta}(t)^T]$ such that $\bar{\eta}(t)$ is a $\bar{\zeta}$-dimensional vector that has a handle on meeting the path constraints and $\tilde{\eta}(t)$ is a vector of dimension $(m - \bar{\zeta})$ that do not affect the path constraints. Thus, $\bar{\eta}(t)$ are referred to as the constraint-seeking input directions and $\tilde{\eta}(t)$ as the compromise-seeking input directions. The optimal values along the constraint-seeking directions, $\bar{\eta}^*(t)$, are determined by the active path constraints, whilst $\tilde{\eta}^*(t)$ are determined from optimality conditions.

Let r_j be the relative degree of the constraint $S_j(x, u) = 0$. Then, the constraint-seeking direction corresponding to the constraint $S_j(x, u) = 0$ is given by $\bar{\eta}_j(t) = \left(\frac{\partial}{\partial u} \frac{d^{r_j} S_j}{dt^{r_j}} \right) u$. The directions $\tilde{\eta}(t)$

are chosen orthogonal to the directions $\bar{\eta}$ computed from all active path constraints. So, for the compromise-seeking input directions, this construction guarantees that the vector $\frac{\partial}{\partial \tilde{\eta}} \frac{d^k S_j}{dt^k} = 0$ for $k = 0, 1, \cdots, r_j$. The transformation $\eta^T \rightarrow [\bar{\eta}^T \ \tilde{\eta}^T]$ is, in general, state dependent and can be obtained analytically if piecewise analytical expressions for the optimal inputs (see Section 3) are available. Otherwise, a numerical sensitivity analysis is necessary to obtain this transformation.

If, in the derivation of the analytical expressions for the inputs, an input u_i is computed from the active constraint $S_j(x, u) = 0$, and the other inputs are inside their respective feasible regions, this does not necessarily imply $\bar{\eta}(t) = u_i(t)$ and $\tilde{\eta} = \{u_k(t), k \neq i\}$. In fact, $\bar{\eta}_j(t)$ is a combination of all inputs that have the same relative degree with respect to $S_j(x, u)$ as the input u_i.

Separation of constraint-seeking and compromise-seeking parameters: In the parameter vector π, there are elements whose variations affect the active terminal constraints and others that do not. The idea is then to separate the two using a transformation $\pi^T \rightarrow [\bar{\pi}^T \ \tilde{\pi}^T]$ such that $\bar{\pi}$ is a $\bar{\tau}$-dimensional vector and $\tilde{\pi}$ is of dimension $(n_\pi - \bar{\tau})$ with $\frac{\partial T}{\partial \tilde{\pi}} = 0$. Then, the necessary conditions of optimality (36) can be rewritten as:

$$\begin{aligned} &\nu^T T(x(t_f)) = 0, \quad \text{and} \\ &\frac{\partial \phi}{\partial \bar{\pi}} + \nu^T \frac{\partial T}{\partial \bar{\pi}} = 0, \quad \frac{\partial \phi}{\partial \tilde{\pi}} = 0 \end{aligned} \tag{37}$$

The active constraints in $T(x(t_f)) = 0$ determine the optimal values of the decision variables $\bar{\pi}^*$, whilst $\tilde{\pi}^*$ are determined from the optimality conditions $\frac{\partial \phi}{\partial \tilde{\pi}} = 0$. Thus, $\bar{\pi}$ are referred to as the constraint-seeking parameters (which have a handle on meeting terminal constraints) and $\tilde{\pi}$ as the compromise-seeking parameters (which are of no help in meeting terminal constraints). Note the similarity with the classification of the input directions. The Lagrange multipliers ν are calculated from $\frac{\partial \phi}{\partial \bar{\pi}} + \nu^T \frac{\partial T}{\partial \bar{\pi}} = 0$.

If the constraint $T_j(x(t_f))$ is active, then the constraint-seeking parameter corresponding to this terminal constraint is $\bar{\pi}_j = \frac{\partial T_j}{\partial \pi} \pi$. $\tilde{\pi}$ is chosen orthogonal to the components of $\bar{\pi}$ computed from all active terminal constraints. This construction guarantees $\frac{\partial T}{\partial \tilde{\pi}} = 0$. Since analytical expressions for $\frac{\partial T}{\partial \pi}$ 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.

5.3 Implications of the Characterization

The knowledge of the structure of the optimal solution can help enhance the numerical efficiency of the optimization algorithm. This issue was addressed in Section 4.4. On the other hand, it helps in an efficient implementation of the optimal solution. The separation of constraint-seeking and compromise-seeking variables is equivalent to a sensitivity analysis which reveals where most of the optimization potential lies.

If the inputs are in the interior of the feasible region, then to a first-order approximation, the deviation in cost is $\delta J = H_u \delta u$. Since by definition $H_u = 0$, small deviation of u from the optimal

trajectory will cause a negligibly small loss in cost. On the other hand, the Lagrange multipliers μ and ν capture the deviation in cost resulting from the path and terminal constraints not being active. So, to a first-order approximation, $\delta J = \int_0^{t_f} \mu^T \delta S \; dt + \nu^T \delta T$. Thus, in general, there is *little* to be gained/lost along the compromise-seeking variables, while it pays off to keep the path and terminal constraints active.

Thus, at the implementation level, care should be taken to keep the constraints active. This idea can be used to integrate measurements in the optimization framework so as to combat uncertainty, which will be the subject of the companion paper [75].

6 Examples

This section presents various examples for which the optimal solution will be characterized. Since this solution will not necessarily be optimal in the presence of uncertainty, this issue will be covered in the companion paper [75]. Four semi-batch reactors are chosen to illustrate various features of optimal solutions. A classification of the examples is provided in Table 2. The first (rather simple) example is worked out in more details, while only the features of the optimal solution are presented for the others.

#	Example	Path constraints	Terminal constraints	Compromise -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	Nonisothermal reactor with series reaction and a heat removal constraint	Yes	Yes	Yes	2	Fixed

Table 2: Features present in the various examples

In the sequel, the subscripts $(\cdot)_{des}$, $(\cdot)_{min}$, $(\cdot)_{max}$, $(\cdot)_o$, and $(\cdot)_f$ represent desired, minimum, maximum, initial, and final values, respectively. u_{comp} will be used to represent a compromise-seeking input in the interior of the feasible region, and u_{path} an input that will keep a path constraint active.

6.1 Isothermal Semi-batch Reactor with a Safety Constraint [83]

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:

$$T_{cf}(t) = T(t) + \min(c_A(t), c_B(t))\frac{(-\Delta H)}{\rho\, c_p} \tag{38}$$

where the parameters are described in the next subsection, and the term $\min(c_A, c_B)$ serves to calculate the maximum extent of reaction that could 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 feedrate is set to zero.

6.1.2 Problem Formulation

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_{Bin}: 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.

Model equations:

$$\dot{c_A} = -k\, c_A\, c_B - \frac{u}{V}c_A \qquad c_A(0) = c_{Ao} \tag{39}$$

$$\dot{c_B} = -k\, c_A\, c_B + \frac{u}{V}(c_{Bin} - c_B) \qquad c_B(0) = c_{Bo} \tag{40}$$

$$\dot{V} = u \qquad V(0) = V_o \tag{41}$$

The concentration of C is given by

$$c_C = \frac{c_{Ao}V_o + c_{Co}V_o - c_A V}{V}. \tag{42}$$

The numerical values are given in Table 3.

k	0.0482	$\frac{\text{l}}{\text{mol h}}$
T	70	°C
ΔH	−60000	$\frac{\text{J}}{\text{mol}}$
ρ	900	$\frac{\text{g}}{\text{l}}$
c_p	4.2	$\frac{\text{J}}{\text{gK}}$
c_{Bin}	2	$\frac{\text{mol}}{\text{l}}$

u_{min}	0	$\frac{\text{l}}{\text{h}}$
u_{max}	0.1	$\frac{\text{l}}{\text{h}}$
T_{max}	80	°C
V_{max}	1	l
$n_{C\,des}$	0.6	mol

c_{Ao}	2	$\frac{\text{mol}}{\text{l}}$
c_{Bo}	0.63	$\frac{\text{mol}}{\text{l}}$
V_o	0.7	l

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

Optimization problem:

$$\begin{aligned} \min_{u(t),t_f} \quad & J = t_f \\ s.t. \quad & (39)-(42) \\ & T_{cf}(t) \leq T_{max} \\ & V(t_f) \leq V_{max} \\ & n_C(t_f) \geq n_{C\,des} \\ & u_{min} \leq u \leq u_{max} \end{aligned} \tag{43}$$

6.1.3 Characterization

Model Reduction:

The dynamic model (39)-(41) can be reduced since the three differential equations are linearly independent, as is shown next. The balance equations for various species and total mass read:

$$\dot{n_A} = -k\, c_A\, c_B\, V \qquad n_A(0) = n_{Ao} \tag{44}$$

$$\dot{n_B} = -k\, c_A\, c_B\, V + c_{Bin} u \qquad n_B(0) = n_{Bo} \tag{45}$$

$$\dot{V} = u, \qquad V(0) = V_o \tag{46}$$

(45) can be expressed in terms of (44) and (46):

$$\dot{n_B} = \dot{n_A} + c_{Bin}\dot{V} \quad \Rightarrow \quad \frac{d}{dt}(n_B - n_A - V\, c_{Bin}) = 0 \tag{47}$$

indicating that $I = n_B - n_A - V\, c_{Bin} = V(c_B - c_A - c_{Bin})$ is a reaction invariant [74]. Integration of (47) from 0 to t allows expressing c_B in terms of other states and initial conditions:

$$c_B = \frac{(c_{Bo} - c_{Ao} - c_{Bin})V_o + (c_A + c_{Bin})V}{V} \tag{48}$$

Types of arcs:
Consider the dynamic model given by (44) and (46), together with (48). The types of arcs that can be present are determined using the matrix $\mathcal{M} = [F_u \quad \Delta F_u]$.

$$F = \begin{bmatrix} -k\, c_A c_B V \\ 0 \end{bmatrix} + \begin{bmatrix} 0 \\ 1 \end{bmatrix} u, \qquad F_u = \begin{bmatrix} 0 \\ 1 \end{bmatrix}, \qquad \Delta F_u = \begin{bmatrix} k\, c_A(c_{Bin} - c_B) \\ 0 \end{bmatrix} \tag{49}$$

The matrix $\mathcal{M}$ has structural rank 2. Since $(c_{Bin} - c_B)$ 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 state and inputs ($\sigma = \infty$), and the optimal input is always on path constraints.

Thus, the possible options for the optimal solution are the input bounds and the path constraint that corresponds to meeting the safety constraint $T_{cf} = T_{max}$: (i) $u = u_{min}$, (ii) $u = u_{max}$, and (iii) $u = u_{path}$.

Specific choice of experimental conditions:
The number and sequence of arcs actually present in the optimal solution depend on the experimental conditions. Let the experimental conditions be chosen such that the number of moles of B that can be added is less than the initial number of moles of A, then $c_B(t) \leq c_A(t)$. Since isothermal conditions are chosen, the condition $T_{cf}(t) \leq T_{max}$ implies $c_B(t) \leq c_{Bmax}$ where $c_{Bmax} = \frac{\rho\, c_p(T_{max}-T)}{(-\Delta H)}$. Furthermore, the initial condition is chosen to have as much B as possible, i.e., $c_{Bo} = c_{Bmax} = 0.63\frac{\text{mol}}{\text{l}}$.

Sequence of arcs (Figure 1):

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

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

$$u_{path} = \left(\frac{k\, c_A\, c_B\, V}{c_{Bin} - c_B}\right)\Big|_{c_B = c_{Bmax}} \tag{50}$$

Minimal input parameterization:
The switching time t_s between u_{path} and u_{min} and the terminal time t_f are adjusted numerically to satisfy the terminal constraints $V(t_f) = V_{max}$ and $n_C(t_f) = n_{Cdes}$. Thus, the two parameters in this example are *constraint-seeking* parameters. For the numerical values provided in Table 3, the cost is $J = t_f = 19.80\ h$ and the switching time is $t_s = 11.44\ h$.

The optimal input and the corresponding evolution of the concentrations of A, B, and C are given in Figure 1. Notice that $c_B = c_{Bmax} = 0.63\frac{\text{mol}}{\text{l}}$ in the first interval, which corresponds to $T_{cf} = T_{max}$.

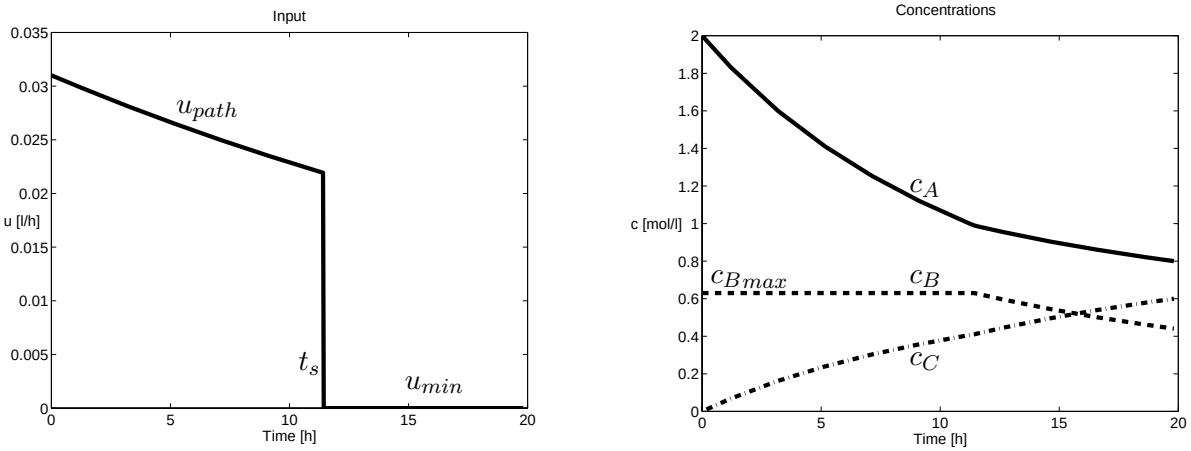

Figure 1: Optimal input and evolution of the concentrations for Example 1

6.1.4 Discussion

Wrong sequence of arcs:
Instead of the correct arc sequence, Sequence I = u_{path} and u_{min}, let one of the following sequences be chosen: Sequence II = u_{min}, u_{max}, and u_{path} or Sequence III = u_{min}, u_{max}, and u_{min}. Note that any sequence that has u_{min} after u_{path} is a superset of Sequence I and will lead to the same solution as in Figure 1 by reducing the other interval(s) to zero. Numerical optimization shows that the required amount of C cannot be attained with Sequence II, while a solution different from that in Figure 1 can be obtained with Sequence III. However, the cost with Sequence III is $t_f = 30.80$ h, which is much larger than $t_f = 19.80$ h obtained with Sequence I.

Using the necessary conditions of optimality, it can be argued that Sequences II and III are non-optimal. Consider the system equations (44), (46), and (48) along with the adjoint equations:

$$\begin{array}{ll} \dot{n}_A = -\frac{k\, n_A\, n_B}{V}, & \dot{\lambda}_{n_A} = \frac{k\,(n_A+n_B)}{V}\lambda_{n_A} - \frac{1}{V}\mu_{cf} \\ \dot{V} = u, & \dot{\lambda}_V = \frac{k\, n_A\,(c_{Bin}-c_B)}{V}\lambda_{n_A} - \frac{(c_{Bin}-c_B)}{V}\mu_{cf} \end{array} \tag{51}$$

with μ_{cf} being the Lagrange multiplier related to the cooling failure constraint. The terminal conditions for the adjoints are $\lambda_{n_A}(t_f) = 1$ and $\lambda_V(t_f) = \nu > 0$. The constraint on which the solution lies is determined by the sign of $\lambda^T F_u(t) = \lambda_V(t)$; $\lambda_V(t) > 0$ for $u = u_{min}$, $\lambda_V(t) < 0$ for $u = u_{max}$, and $\lambda_V(t) = 0$ for $u = u_{path}$.

The adjoint equations have two different structures depending on whether or not the path constraint is active. When the path constraint is not active $\mu_{cf} = 0$ and the adjoint equations read: $\dot{\lambda}_{n_A} = \frac{k\,(n_A+n_B)}{V}\lambda_{n_A}$ and $\dot{\lambda}_V = \frac{k\, n_A\,(c_{Bin}-c_B)}{V}\lambda_{n_A}$. If the cooling failure constraint is active, then $\mu_{cf} = k\, n_A \lambda_{n_A}$ to satisfy $\lambda_V = \dot{\lambda}_V = 0$. The adjoint equations become $\dot{\lambda}_{n_A} = \frac{k n_B}{V}\lambda_{n_A}$ and $\lambda_V = 0$. Since $n_A > 0$, $n_B > 0$, $V > 0$, $c_{Bin} > c_B$, the following conclusions can be drawn from (51): $\lambda_{n_A}(t) > 0$ and $\dot{\lambda}_V(t) \geq 0$, $\forall t$.

For Sequence II to be optimal, $\lambda_V(t)$ has to be first positive, then negative, and finally zero. For Sequence III to be optimal, $\lambda_V(t)$ has to change from positive to negative and then again positive. Both these are contradictions to the fact that λ_V is monotonic.

Nonparsimonious parameterization:
Since u_{min} and the analytical expression for u_{path} are known, the optimal input is parameterized by two parameters, the switching time between u_{path} and u_{min} and the terminal time. In contrast, if the input is parameterized as being piecewise constant on equally spaced intervals (which is the usual choice), then approximately 20 parameters are needed to determine the solution with the same accuracy (Table 4).

Number of Parameters	2	4	6	8	10	20	∞
Cost	20.23	19.93	19.87	19.84	19.82	19.80	19.80

Table 4: Cost as a function of the number of parameters in a piecewise-constant parameterization

Effect of different experimental conditions:

1. If $c_{Bo} < c_{Bmax}$, 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 Figure 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 $u_{min} = 0$.

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 $t_f = 17.3$ h. So, 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 $t_f = 18.4$ h.

6.2 Fed-batch Bioreactor with Inhibition and Biomass Constraint [87]

- *Reactions:* $S \rightarrow X \quad S \xrightarrow{X} P$.
- *Conditions:* Fed-batch, isothermal.
- *Objective:* Maximize the concentration of product P at a given final time.
- *Manipulated variable:* Feed rate of S.
- *Constraints:* Input bounds, constraint on the maximum biomass concentration.
- *Comments:* The specific growth rate $\mu(S)$ contains an inhibition term:

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

Owing to the presence of inhibition, it will be shown that the optimal substrate value corresponds to $\frac{d\mu}{dS} = 0$ (i.e., $S^* = \sqrt{K_m K_i}$). Without any constraints, optimality consists 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 limitation typically occurring at large biomass concentrations. The interesting part is that the optimal input cannot switch immediately from u_{comp} (corresponding to S^*) to u_{path} since the internal dynamics are unstable. An additional arc is required to lower the substrate concentration to an equilibrium value.

6.2.1 Problem formulation

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

Model equations:

$$\dot{X} = \mu(S)\, X - \frac{u}{V}\, X \qquad X(0) = X_o \tag{52}$$

$$\dot{S} = -\frac{\mu(S)\, X}{Y_x} - \frac{\nu\, X}{Y_p} + \frac{u}{V}\,(S_{in} - S) \qquad S(0) = S_o \tag{53}$$

$$\dot{P} = \nu\, X - \frac{u}{V}\, P \qquad P(0) = P_o \tag{54}$$

$$\dot{V} = u \qquad V(0) = V_o \tag{55}$$

with $\mu(S) = \frac{\mu_m\, S}{K_m + S + \frac{S^2}{K_i}}$ and the numerical values given in Table 5.

μ_m	0.53	$\frac{1}{\mathrm{h}}$
K_m	1.2	$\frac{\mathrm{g}}{\mathrm{l}}$
K_i	22	$\frac{\mathrm{g}}{\mathrm{l}}$
Y_x	0.4	
Y_p	1	
ν	0.5	$\frac{1}{\mathrm{h}}$
S_{in}	20	$\frac{\mathrm{g}}{\mathrm{l}}$

u_{min}	0	$\frac{1}{\mathrm{h}}$
u_{max}	1	$\frac{1}{\mathrm{h}}$
X_{max}	3	$\frac{\mathrm{g}}{\mathrm{l}}$
t_f	8	h

X_o	1	$\frac{\mathrm{g}}{\mathrm{l}}$
S_o	0	$\frac{\mathrm{g}}{\mathrm{l}}$
P_o	0	$\frac{\mathrm{g}}{\mathrm{l}}$
V_o	2	l

Table 5: Model parameters, operating bounds and initial conditions for Example 2

Optimization problem:

$$\begin{aligned} \max_{u(t)} \quad & J = P(t_f) \\ s.t. \quad & (52) - (55) \\ & X(t) \le X_{max} \\ & u_{min} \le u \le u_{max} \end{aligned} \tag{56}$$

6.2.2 Characterization

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

$$\dot{x_1} = \mu(S)\, x_1 \qquad x_1(0) = X_o\, V_o \tag{57}$$

$$\dot{x_2} = \nu\, x_1 \qquad x_2(0) = P_o\, V_o \tag{58}$$

$$\dot{x_3} = u \qquad x_3(0) = V_o \tag{59}$$

where the substrate concentration is obtained from the mass balance:

$$S = \frac{1}{x_3}\left(S_o\, V_o + S_{in}(x_3 - V_0) - \frac{1}{Y_x}(x_1 - X_o V_o) - \frac{1}{Y_p}(x_2 - P_o V_o)\right) \tag{60}$$

Types of arcs:

$$F = \begin{bmatrix} \mu(S)\ x_1 \\ \nu\ x_1 \\ 0 \end{bmatrix} + \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix} u, \qquad F_u = \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix}, \qquad \Delta F_u = -\begin{bmatrix} \frac{\partial \mu}{\partial x_3}\ x_1 \\ 0 \\ 0 \end{bmatrix} \tag{61}$$

$$\Delta^2 F_u = \begin{bmatrix} \frac{\partial \mu}{\partial x_1}\ \frac{\partial \mu}{\partial x_3}\ x_1^2 - \frac{\partial^2 \mu}{\partial x_1\ \partial x_3}\ x_1^2\ \mu - \nu\ x_1^2 \frac{\partial^2 \mu}{\partial x_2\ \partial x_3} \\ \nu \frac{\partial \mu}{\partial x_3}\ x_1 \\ 0 \end{bmatrix} - u \begin{bmatrix} \frac{\partial^2 \mu}{\partial x_3^2}\ x_1 \\ 0 \\ 0 \end{bmatrix} \tag{62}$$

The matrix $\mathcal{M} = [F_u \quad \Delta F_u \quad \Delta^2 F_u]$ has structural rank 3, but the rank depends on the states. The loss of rank can be analyzed using $\det(\mathcal{M}) = 0$, which occurs when:

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

$S = S_{in}$, $x_1 = 0$, or $\frac{\partial \mu}{\partial S} = 0$ are solutions to $\det(\mathcal{M}) = 0$. Since $S = S_{in}$ and $x_1 = 0$ result in trivial solutions, rank drop occurs for $\frac{\partial \mu}{\partial S} = 0$, which corresponds to $S = S^* = \sqrt{K_i\ K_m}$. Thus, it is possible to have a compromise-seeking input in the interior of the feasible region.

The possible options for the optimal solution are the input bounds, a compromise-seeking input, and the path constraint: (i) $u = u_{min}$, (ii) $u = u_{max}$, (iii) $u = u_{comp}$, and (iv) $u = u_{path}$.

Sequence of arcs (Figure 2):

- The input is initially at the upper bound, u_{max}, in order to increase S as quickly as possible.
- Once $S = S^*$, the input is given by u_{comp} in order to increase X, and thus P, as quickly as possible.
- The input is then lowered to u_{min} in order to reach $S = S_e$. The switching time between the second and the third interval should be so chosen that the conditions $X = X_{max}$ and $S = S_e$ occur at the same time instant.
- When $X = X_{max}$, the input is set to u_{path}.

Analytical expression for the inputs:
Though the input appears in $\Delta^2 F_u$, $det(\mathcal{M})$ is independent of u since the vector that multiplies u in (62) is parallel to ΔF_u. Thus, an additional differentiation is required to obtain the input (order of singularity, $\sigma = 3$) or, equivalently, the *surface* $S = S^*$ can be differentiated to obtain the input:

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

The path constraint corresponds to $X = X_{max}$. The input can be obtained by differentiating the path constraint once ($\varsigma = 1$):

$$u_{path} = \mu(S) V|_{X=X_{max}} \tag{65}$$

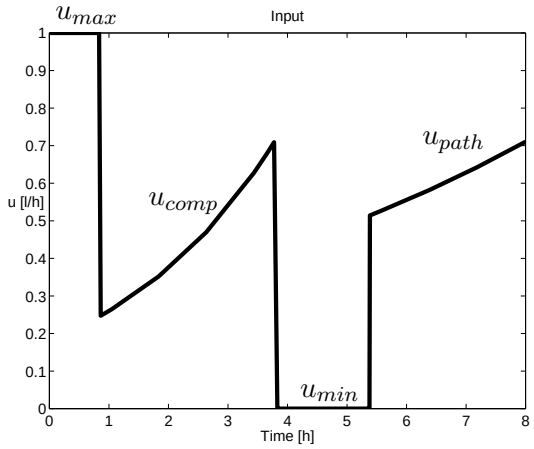

Figure 2: Optimal input for Example 2

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

$$\dot{S} = -\frac{1}{Y_x}\,\mu(S)\,X_{max} - \frac{1}{Y_p}\,\nu\,X_{max} + \frac{\mu(S)}{V}\,(S_{in} - S) \tag{66}$$

$$\Delta\dot{S} = \frac{\partial\mu}{\partial S}\left(\frac{S_{in} - S}{V} - \frac{X_{max}}{Y_x}\right)\Delta S - \frac{\mu(S)}{V}\Delta S \tag{67}$$

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 (66).

Since there is a single input, there is no need for separation of the input directions. The input can be either constraint-seeking or compromise-seeking depending on the interval. In fact, $u = \bar{\eta}$ in all intervals except the second one where $u = u_{comp} = \tilde{\eta}$. It can be verified numerically that a small deviation of the input in this interval has very little influence on the cost.

Minimal input parameterization:
Though there are three switching times, all of them represent achieving some intermediate goals, such as getting to the path constraints. The first switching corresponds to reaching $S = S^*$. The second and third switchings correspond to attaining $S = S_e$ and $X = X_{max}$. So, if there are controllers to force $S = S^*$ during the second interval and to have $S = S_e$ and $X = X_{max}$ during the last interval, then there is no degree of freedom left to meet any terminal objective. This is logical since there is no terminal constraint. Thus, the production of P is maximized by the type and sequence of arcs shown in Figure 2.

6.3 Isothermal Semi-batch Reactor with Parallel Reactions and Selectivity Constraints [69, 77]

6.3.1 Description of the Reaction System

- *Reactions:* $A + B \to C, \quad 2B \to D$.
- *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 if there is a constraint on the final amount of D. 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. 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).

6.3.2 Problem Formulation

Variables and parameters: c_X: Concentrations of species X, V: Reactor volume, u: Feed rate of B, c_{Bin}: Inlet concentration of B, and k_1, k_2: Kinetic parameters.

Model equations:

$$\dot{c_A} = -k_1\, c_A\, c_B - \frac{u}{V} c_A \qquad c_A(0) = c_{Ao} \tag{68}$$

$$\dot{c_B} = -k_1\, c_A\, c_B - 2\, k_2\, c_B^2 + \frac{u}{V}(c_{Bin} - c_B) \qquad c_B(0) = c_{Bo} \tag{69}$$

$$\dot{V} = u \qquad V(0) = V_o \tag{70}$$

with

$$c_C = \frac{1}{V}(c_A\, V - c_{Ao}\, V_o) \tag{71}$$

$$c_D = \frac{1}{2\,V}((c_A + c_{Bin} - c_B)\, V - (c_{Ao} + c_{Bin} - c_{Bo})\, V_o) \tag{72}$$

The numerical values are given in Table 6.

k_1	0.053	$\frac{\text{l}}{\text{mol min}}$
k_2	0.128	$\frac{\text{l}}{\text{mol min}}$
c_{Bin}	5	$\frac{\text{mol}}{\text{l}}$

u_{min}	0	$\frac{\text{l}}{\text{min}}$
u_{max}	0.001	$\frac{\text{l}}{\text{min}}$
$c_{Bf,max}$	0.025	$\frac{\text{mol}}{\text{l}}$
$c_{Df,max}$	0.15	$\frac{\text{mol}}{\text{l}}$

c_{Ao}	0.72	$\frac{\text{mol}}{\text{l}}$
c_{Bo}	0.05	$\frac{\text{mol}}{\text{l}}$
V_o	1	l
t_f	250	min

Table 6: Model parameters, operating bounds and initial conditions for Example 3

Optimization problem:

$$\begin{aligned} \max_{u(t)} \quad & J = V(t_f)\, c_C(t_f) \\ s.t. \quad & (68) - (70) \\ & c_B(t_f) \le c_{Bf,max} \\ & c_D(t_f) \le c_{Df,max} \\ & u_{min} \le u \le u_{max} \end{aligned} \tag{73}$$

6.3.3 Characterization

Types of arcs:
The types of arcs that can be present are determined using the matrix $\mathcal{M}$.

$$
\begin{aligned}
F &= \begin{bmatrix} -k_1\, c_A\, c_B \\ -k_1\, c_A\, c_B - 2\, k_2\, c_B^2 \\ 0 \end{bmatrix} + \frac{1}{V} \begin{bmatrix} -c_A \\ c_{Bin} - c_B \\ V \end{bmatrix} u, \qquad F_u = \frac{1}{V} \begin{bmatrix} -c_A \\ c_{Bin} - c_B \\ V \end{bmatrix} \\
\Delta F_u &= \frac{1}{V} \begin{bmatrix} k_1\, c_A\, (c_{Bin} - c_B) \\ k_1\, c_A\, (c_{Bin} - c_B) + 2\, k_2\, c_B\, (2\, c_{Bin} - c_B) \\ 0 \end{bmatrix} \\
\Delta^2 F_u &= \frac{c_{Bin}}{V} \begin{bmatrix} k_1^2 c_A^2 + 4\, k_1 k_2 c_A c_B \\ k_1^2 c_A^2 + 4\, k_1 k_2 c_A c_B + 8\, k_2^2 c_B^2 \\ 0 \end{bmatrix} - 2\,(c_{Bin} - c_B) \frac{u}{V^2} \begin{bmatrix} k_1 c_A \\ k_1 c_A - 2\, k_2 (c_{Bin} - c_B) \\ 0 \end{bmatrix}
\end{aligned}
$$

The matrix $\mathcal{M} = [F_u \quad \Delta F_u \quad \Delta^2 F_u]$ has structural rank 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., $u_{comp} = u(x)$. Hence, the options for the optimal solution are: (i) the minimum feedrate u_{min}, (ii) the maximum feedrate u_{max}, and (iii) a compromise-seeking feedrate $u = u_{comp}$.

Sequence of arcs (Figure 3):

- 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 compromise-seeking arc u_{comp} 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.

Analytical expression for the inputs:
The combination of x and u for which the rank of $\mathcal{M}$ drops can be computed from $\det(\mathcal{M}) = 0$. The input appears in $\Delta^2 F_u$, which indicates that the order of singularity is $\sigma = 2$. Since $\sigma + 1 = \text{rank}(\mathcal{M})$, a static feedback for the optimal input can be computed from $\det(\mathcal{M}) = 0$:

$$
u_{comp} = \frac{c_{Bin} c_B V\, (k_1 c_A (2\, c_{Bin} - c_B) + 4\, k_2 c_B c_{Bin})}{2(c_{Bin} - c_B)} \tag{74}
$$

The input can be either constraint-seeking or compromise-seeking depending on the interval. In fact, $u = \bar{\eta}$ in the first and third intervals, and $u = u_{comp} = \tilde{\eta}$ in the second one. It can be verified numerically that a small deviation of the input in this interval has very little influence on the cost.

Minimal input parameterization:
The two switching times (t_m and t_s – See Figure 3) parameterize the solution completely. In turn, they are determined by the two active terminal constraints $c_B(t_f) = c_{Bf,max}$ and $c_D(t_f) = c_{Df,max}$. Thus, the two parameters in this example are constraint-seeking parameters. Furthermore, it is possible to approximate $u_{comp}(t)$ by a piecewise constant or piecewise linear profile rather than

using the expression (74). Consider the case of a piecewise constant approximation of $u_{comp}(t)$ with the scalar value u_s. Thus, there are three parameters, (t_m, t_s, and u_s), to meet the two terminal constraints and optimize the cost.

Separation of constraint-seeking and compromise-seeking parameters:
To determine the constraint-seeking and compromise-seeking parameters, the gain matrix $\mathcal{G} : \pi \rightarrow T$, computed in the neighborhood of the optimal solution, with $\pi = [t_m, u_s, t_s]^T$ and $T(x(t_f)) = [c_D(t_f) - c_{Df,max}, c_B(t_f) - c_{Bf,max}]^T$, can be used:

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

The compromise-seeking parameter $\tilde{\pi}$ is given by the null space of $\mathcal{G}$ and corresponds to $\tilde{\pi} = t_m - 4 \times 10^{-6} u_s + 4.7 \times 10^{-2} t_s$. Since the contributions from u_s and t_s are negligible, t_m essentially acts as the compromise-seeking parameter. It is interesting to note that t_m, which was a constraint-seeking parameter when u_{comp} was not approximated, becomes a compromise-seeking parameter after the approximation.

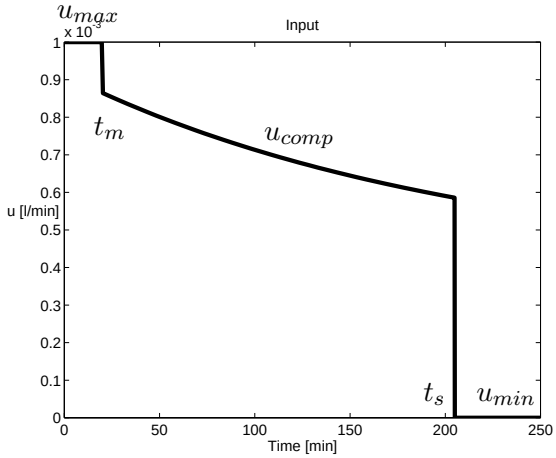

Figure 3: Optimal input for Example 3

6.4 Nonisothermal Semi-batch Reactor with Series Reactions and a Heat Removal Constraint

6.4.1 Description of the Reaction System

- *Reactions:* $A + B \rightarrow C \rightarrow D$.
- *Conditions:* Semi-batch, exothermic, nonisothermal, 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:* Feedrate of B and reactor temperature.
- *Constraints:* Bounds on feedrate 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: $\rho\, c_p\, \frac{d(V\,T)}{dt} = q_{rx} - q_{in} - q_{ex}$, where V is the 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 complicates the analytical expressions without fundamentally changing the type and sequence of arcs present in the solution. So, for simplicity, the heat balance equation is neglected. However, to guarantee meeting the upper bound on q_{ex} even in the worst case, it is necessary to limit q_{rx} as follows: $q_{rx} \leq \max(q_{ex}) + \min\left(q_{in} + \rho\, c_p\, \frac{d(V\,T)}{dt}\right) \equiv q_{rx,max}$. Thus, an upper bound on the heat rate produced by the reactions, $q_{rx} \leq q_{rx,max}$, is imposed as a constraint.

The optimal solution tries to reduce the temperature towards the end so that the undesired reaction, which utilizes the desired product C, is slowed down. The compromise between the production and consumption of C corresponds to a compromise-seeking temperature profile. As far as the feedrate 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

Variables and parameters: c_X: Concentration of species X, T: Reactor temperature, u: Feedrate of B with inlet concentration $c_{B_{in}}$, V: Reactor volume, q_{rx}: Heat production rate, k_{1o}, k_{2o}: Pre-exponential factors, E_1, E_2: Activation energies, R: Gas constant, $\Delta H_1, \Delta H_2$: Reaction enthalpies.

Model equations:

$$\dot{c_A} = -k_1\, c_A\, c_B - \frac{u}{V} c_A \qquad c_A(0) = c_{Ao} \tag{75}$$

$$\dot{c_B} = -k_1\, c_A\, c_B + \frac{u}{V}(c_{B_{in}} - c_B) \qquad c_B(0) = c_{Bo} \tag{76}$$

$$\dot{c_C} = k_1\, c_A\, c_B - k_2\, c_C - \frac{u}{V} c_C \qquad c_C(0) = c_{Co} \tag{77}$$

$$\dot{V} = u \qquad V(0) = V_o \tag{78}$$

with $k_1 = k_{1o}\, e^{\frac{-E_1}{RT}}$, $k_2 = k_{2o}\, e^{\frac{-E_2}{RT}}$. The numerical values are given in Table 7.

k_{1o}	4	$\frac{\text{l}}{\text{mol h}}$
k_{2o}	800	$\frac{\text{l}}{\text{h}}$
E_1	6×10^3	$\frac{\text{J}}{\text{mol}}$
E_2	20×10^3	$\frac{\text{J}}{\text{mol}}$
R	8.31	$\frac{\text{J}}{\text{molK}}$
ΔH_1	-3×10^4	$\frac{\text{J}}{\text{mol}}$
ΔH_2	-10^4	$\frac{\text{J}}{\text{mol}}$

u_{min}	0	$\frac{\text{l}}{\text{h}}$
u_{max}	1	$\frac{\text{l}}{\text{h}}$
T_{min}	20	°C
T_{max}	50	°C
V_{max}	1.1	l
$q_{rx,max}$	1.5×10^5	$\frac{\text{J}}{\text{h}}$

c_{Ao}	10	$\frac{\text{mol}}{\text{l}}$
c_{Bo}	1.1685	$\frac{\text{mol}}{\text{l}}$
c_{Co}	0	$\frac{\text{mol}}{\text{l}}$
V_o	1	l
c_{Bin}	20	$\frac{\text{mol}}{\text{l}}$
t_f	0.5	h

Table 7: Model parameters, operating bounds and initial conditions for Example 4

Optimization problem:

$$\begin{aligned} \max_{u(t),T(t)} \quad & J = c_C(t_f)V(t_f) \\ s.t. \quad & (75)-(78) \\ & T_{min} \le T(t) \le T_{max} \\ & u_{min} \le u(t) \le u_{max} \\ & (-\Delta H_1)\, k_1\, c_A\, c_B\, V + (-\Delta H_2)\, k_2\, c_C\, V \le q_{rx,max} \\ & V(t_f) \le V_{max} \end{aligned} \tag{79}$$

6.4.3 Characterization

Model reduction: Since the equations (75)-(78) are linear dependent, one of the state can be removed, which leads to:

$$\dot{x_1} = -k_1\, x_1\, c_B \qquad x_1(0) = V_o\, c_{A_o} \tag{80}$$

$$\dot{x_2} = k_2\, (x_1 - x_2) \qquad x_2(0) = V_o\, (c_{A_o} + c_{C_o}) \tag{81}$$

$$\dot{x_3} = u \qquad x_3(0) = V_o \tag{82}$$

where $x_1 = V\, c_A,\ x_2 = V\, (c_A + c_C),\ x_3 = V$, and $c_B = \frac{1}{x_3}(c_{B_{in}} x_3 + x_1 + V_o\, (c_{B_o} - c_{A_o} - c_{B_{in}}))$.

Types of arcs:
The types of arcs that can be present is determined using the matrices $\mathcal{M}_u$ and $\mathcal{M}_T$ for the two inputs.

$$F = \begin{bmatrix} -k_1\, c_A\, c_B\, V \\ -k_2\, c_C\, V \\ 0 \end{bmatrix} + \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix} u, \qquad F_u = \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix}, \qquad F_T = \frac{V}{R\,T^2} \begin{bmatrix} E_1\, k_1\, c_A\, c_B \\ E_2\, k_2\, c_C \\ 0 \end{bmatrix}$$

$$\Delta F_u = \begin{bmatrix} k_1\, c_A\, (c_{B_{in}} - c_B) \\ 0 \\ 0 \end{bmatrix}, \qquad \Delta^2 F_u = \begin{bmatrix} k_1^2\, c_A^2\, c_{B_{in}} \\ k_1\, k_2\, c_A\, (c_{B_{in}} - c_B) \\ 0 \end{bmatrix} - \frac{2u}{V}\Delta F_u + \frac{E_1\, \dot{T}}{R\,T^2}\Delta F_u$$

$$\Delta F_T = -\frac{V}{R\,T^2} \begin{bmatrix} 0 \\ k_1\, k_2\, c_A\, c_B\, (E_1 - E_2) \\ 0 \end{bmatrix} - \frac{\dot{T}\, V}{R^2\, T^4} \begin{bmatrix} E_1\, k_1\, c_A\, c_B\, (E_1 - 2\,R\,T) \\ E_2\, k_2\, c_C\, (E_2 - 2\,R\,T) \\ 0 \end{bmatrix} + \frac{E_1\, u}{R\,T^2}\Delta F_u$$

It can be verified that the matrix $\mathcal{M}_u = [F_u \quad \Delta F_u \quad \Delta^2 F_u]$ has structural rank 3. Furthermore, the rank is independent of the states and inputs. Thus, the input u is always determined by the input bounds and the path constraint: (i) $u = u_{min}$, (ii) $u = u_{max}$, and (iii) $u = u_{path}$.

The matrix $\mathcal{M}_T = [F_T \quad \Delta F_T \quad \Delta^2 F_T]$ has structural rank 2 since the third element of all concerned vector fields is zero. Intuitively, this is because the temperature cannot affect the volume. Even though the structural rank is 2, the rank depends of the states and inputs. So, the temperature can correspond to a compromise-seeking input and be in the interior of the feasible region. Hence, the possible arcs for the optimal temperature are: (i) $T = T_{min}$, (ii) $T = T_{max}$, and (iii) $T = T_{comp}$.

Specific choice of experimental conditions:
The number and sequence of arcs actually present in the optimal solution depend on the 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_{Bo} is chosen to verify $(-\Delta H_1)\ k_1\ c_{Ao}\ c_{Bo}\ V_o + (-\Delta H_2)\ k_2\ c_{Co}\ V_o = q_{rx,max}$.

Sequence of arcs:

- Since the initial condition c_{Bo} verifies $q_{rx}(0) = q_{rx,max}$, the feedrate input u_{path} is applied to keep the path constraint active.
- Once the constraint on the maximum volume is attained, the feedrate 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 $T_{comp}(t)$ to take advantage of the temperature-dependent compromise between the production and consumption of C.

Analytical expression for the inputs:
u_{path} is obtained by differentiating the path constraint regarding the heat production rate ($\varsigma_u = 1$):

$$\begin{aligned} u_{path} \;=\; & V\frac{\Big((-\Delta H_1)\ k_1^2\ c_A\ c_B\ (c_A + c_B) - (-\Delta H_2)\ k_2\ (k_1\ c_A\ c_B - k_2\ c_C)\Big)}{(-\Delta H_1)\ k_1\ c_A\ (c_{Bin} - c_B)} - \\ & \frac{\dot{T}\ V}{R\ T^2}\frac{\Big((-\Delta H_1)\ E_1\ k_1\ c_A\ c_B + (-\Delta H_2)\ E_2\ k_2\ c_C\Big)}{(-\Delta H_1)\ k_1\ c_A\ (c_{Bin} - c_B)} \end{aligned} \tag{83}$$

T_{comp} is obtained from the combination of x, u, and T for which the rank of $\mathcal{M}_T$ drops. This can be computed from the determinant of the first two rows of F_T and ΔF_T. Since F_T is already a function of T, the order of singularity is $\sigma_T = 0$. Since $\sigma_T + 1 < \text{rank}(\mathcal{M}_T) = 2$, T_{comp} corresponds to a dynamic feedback:

$$\dot{T}_{comp} = -\frac{R\ T^2\ k_1\ c_A\ c_B}{E_2\ c_C} - \frac{R\ T^2\ (c_{Bin} - c_B)}{c_B\ (E_1 - E_2)}\ \frac{u}{V} \tag{84}$$

The initial condition of T_{comp} as it enters the compromise-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 $\dot{T}_{comp}$ depends on u. So, if in a given interval u is determined by the path constraint and T is compromise-seeking, then the two equations (83) and (84) have to be solved simultaneously.

The optimal inputs for the feedrate and temperature are depicted in Figure 4. When the temperature switches to the interior of the feasible region, there is a discontinuity in the feedrate due to the coupling between the two inputs (see equation (83)). Similarly, when the feedrate switches to zero due to the volume constraint, there is a difference in the rate of change of temperature (see equation (84)).

Separation of constraint-seeking and compromise-seeking input directions:
There are three arcs in this solution and each of them needs to be addressed separately.

- In the first arc, both inputs are on path constraints, i.e, $\bar{\eta} = \{u, T\}$, and $\tilde{\eta} = \{\}$.

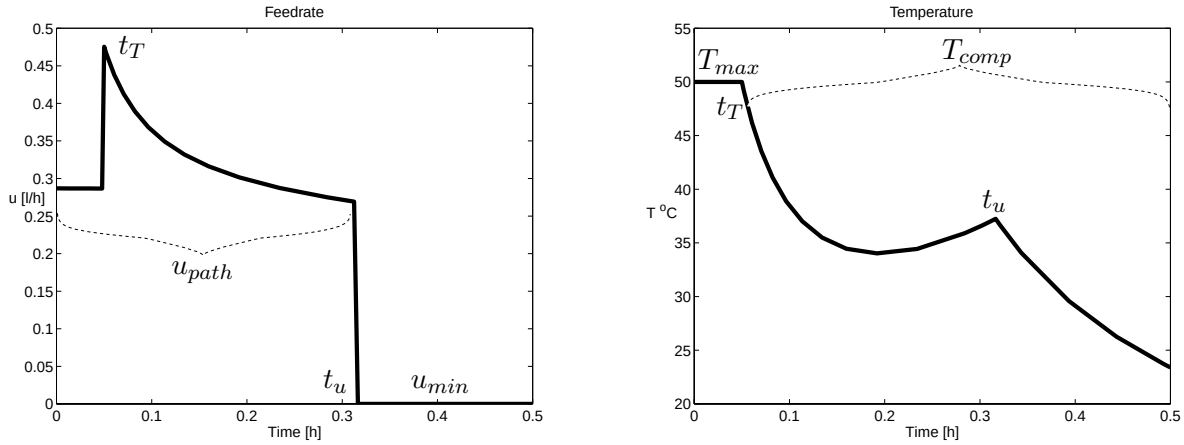

Figure 4: Optimal feedrate and temperature profiles for Example 4

- In the second arc, only the path constraint regarding the heat production rate is active, for which two inputs are available. Any choice of u and $\dot{T}$ which satisfies (83) will keep the path constraint active. So, (83) is differentiated with respect to u and $\dot{T}$ to compute the constraint-seeking direction $\bar{\eta}$ (see Section 5.2). The compromise-seeking direction $\tilde{\eta}$ is constructed orthogonal to the constraint-seeking direction.

$$\begin{aligned} \bar{\eta} &= u\,\frac{(-\Delta H_1)\,k_1\,c_A\,(c_{Bin}-c_B)}{V} + \dot{T}\,\frac{(-\Delta H_1)\,E_1\,k_1\,c_A\,c_B + (-\Delta H_2)\,E_2\,k_2\,c_C}{R\,T^2} \\ \tilde{\eta} &= u\,\frac{(-\Delta H_1)\,E_1\,k_1\,c_A\,c_B + (-\Delta H_2)\,E_2\,k_2\,c_C}{R\,T^2} - \dot{T}\,\frac{(-\Delta H_1)\,k_1\,c_A\,(c_{Bin}-c_B)}{V} \end{aligned}$$

- In the third arc, only the input bound for the feedrate is active. So, $\bar{\eta} = u$, and $\tilde{\eta} = T$.

Minimal input parameterization:
The two switching times (t_T and t_u – see Figure 4) parameterize the solution completely. Since there is only one active terminal constraint, $V(t_f) = V_{max}$, one combination is constraint-seeking.

Separation of constraint-seeking and compromise-seeking parameters:
The gain matrix, in the neighborhood of the optimum, $\mathcal{G} : \pi \to T$, with $\pi = [t_T\ t_u]^T$ and $T(x(t_f)) = V(t_f) - V_{max}$ is given by $\mathcal{G} = \begin{bmatrix} 0.3 & -0.4 \end{bmatrix}$. The compromise-seeking parameter $\tilde{\pi}$ is given by the null space of $\mathcal{G}$ and corresponds to $\tilde{\pi} = t_u - 1.33\,t_T$.

7 Conclusions

Most techniques proposed in the literature for the optimization of dynamic processes take a numerical approach. In contrast, this paper emphasizes the solution structure that arises from an interplay between manipulated variables, cost and constraints. A series of examples are provided to illustrate the theoretical concepts.

One of the advantages of understanding the structure of the optimal solution is the enhancement

of numerical efficiency. However, this advantage is important only for the class of problems where the solution is determined by the constraints, a category which contains most batch processes.

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*, while perfect models of industrial batch processes are rarely available. On the other hand, frequent process measurements which have been made possible by recent developments in sensor technology are now available in many industrial setups. Hence, *measurement-based* optimization strategies are necessary to cope with uncertainty. These strategies, which constitute the focus of the companion paper [75], are based on the characterization presented in this work.

References

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

[2] R. J. Allgor, M. D. Barrera, P. I. Barton, and L. B. Evans. Optimal batch process development. *Comp. Chem. Eng.*, 20(6–7):885–896, 1996.

[3] M. D. Barrera and L. B. Evans. Optimal design and operation of batch processes. *Chem. Eng. Comm.*, 82:45–66, 1989.

[4] P. K. Basu. Pharmaceutical process development is different! *Chem. Eng. Progress*, 94(9):75–82, 1998.

[5] T. Baumann. *Infinite-order Singularity in Terminal-Cost Optimal Control: Application to Robotic Manipulators*. PhD thesis 1778, Swiss Federal Institute of Technology (EPFL), Lausanne, Switzerland, 1998.

[6] M. L Bell, D. J. N. Limebeer, and R. W. H. Sargent. Robust receding horizon optimal control. *Comp. Chem. Eng.*, 20:S781–S786, 1996.

[7] R. E. Bellman. *Dynamic Programming*. Princeton University Press, 1957.

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

[9] L. T. Biegler. Solution of dynamic optimization problems by successive quadratic programming and orthogonal collocation. *Comp. Chem. Eng.*, 8:243–248, 1984.

[10] B. Bojkov and R. Luus. Time-optimal control by iterative dynamic programming. *Ind. Eng. Chem. Res.*, 33:1486–1492, 1994.

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

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

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

[14] A. E. Bryson and Y. C. Ho. *Applied Optimal Control*. Hemisphere, Washington DC, 1975.

[15] A. Cervantes and L.T. Biegler. Large-scale DAE optimization using a simultaneous NLP formulation. *AIChE J.*, 44(5):1038–1050, 1998.

[16] S. A. Chen and W. F. Jeng. Minimum end time policies for batchwise radical chain polymerization. *Chem. Engng. Sci.*, 33:735, 1978.

[17] W. B. Chu. *Modeling, Optimization and Computer Control of the Cephalosporin C Fermentation Process*. PhD thesis, Rutgers, The State University of New Jersey, New Brunswick, New Jersey, USA, 1987.

[18] J. E. Cuthrell and L. T. Biegler. On the optimization of differential-algebraic process systems. *AIChE J.*, 33:1257–1270, 1987.

[19] J. E. Cuthrell and L. T. Biegler. Simultaneous optimization methods for batch reactor control profiles. *Comp. Chem. Eng.*, 13:49–62, 1989.

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

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

[22] T. F. Edgar and D. M. Himmelblau. *Optimization of Chemical Processes*. McGraw-Hill, New York, 1988.

[23] M. Fikar, M. A. Latifi, F. Fournier, and Y. Creff. Control-vector parameterization versus iterative dynamic programming in dynamic optimization of a distillation column. *Comp. Chem. Eng.*, 22:S625–628, 1998.

[24] M. Friedrich and R. Perne. Design and control of batch reactors: An industrial viewpoint. *Comp. Chem. Eng.*, 19:S357–S368, 1995.

[25] H. I. Furlonge, C. C. Pantelides, and E. Sorensen. Optimal operation of multivessel batch distillation columns. *AIChE J.*, 45(4):781–800, 1999.

[26] P. E. Gill, W. Murray, and M. H. Wright. *Practical Optimization*. Academic Press, London, 1981.

[27] S. Giritligil, S. Cesur, and B. Kuryel. Overview of planning and scheduling of batch process operations. In *IFAC DYCOPS-5*, pages 461–466, Corfu, Greece, 1998.

[28] C. J. Goh and K. L. Teo. Control parameterization: A unified approach to optimal control problems with general constraints. *Automatica*, 24:3–18, 1988.

[29] C. Guntern, A. H. Keller, and K. Hungerbühler. Economic optimization of an industrial semi-batch reactor applying dynamic programming. *Ind. Eng. Chem. Res.*, 37(10):4017–22, 1998.

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

[31] J. Hicks, A. Mohan, and W. H. Ray. The optimal control of polymerization reactors. *Can. J. of Chem. Engng.*, 47:590, 1969.

[32] T. Ishikawa, Y. Natori, L. Liberis, and C. C. Pantelides. Modelling and optimisation of an industrial batch process for the production of dioctyl phthalate. *Comp. Chem. Eng.*, 21:S1239–S1244, 1997.

[33] R. K. Jaspan and J. Coull. Trajectory optimization techniques in chemical reaction engineering. II. Comparison of the methods. *AIChE J.*, 18(4):867–869, 1972.

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

[35] A. J. Krener. The higher order maximal principle and its application to singular extremals. *SIAM J. Control Optimization*, 15(2):256, 1977.

[36] H. M. Ku and I. F. Karimi. Scheduling in serial multiproduct batch processes with due-date penalties. *Ind. Eng. Chem. Res.*, 29:580–590, 1990.

[37] E. Lahteemaki, E. Jutila, and M. Paasila. Profitability as a criterion of batch process control design. *Comp. Chem. Eng.*, 3(1-4):197, 1979.

[38] E. B. Lee and L. Markus. *Foundations of Optimal Control Theory*. John Wiley, New York, 1967.

[39] E. S. Lee. *Quasilinearization and Invariant Imbedding with Applications to Chemical Engineering and Adaptive Control*. Academic Press, New York, 1968.

[40] H. C. Lim, Y. J. Tayeb, J. M. Modak, and P. Bonte. Computational algorithms for optimal feed rates for a class of fed-batch fermentation: Numerical results for penicillin and cell mass production. *Biotechnol. Bioengng.*, 28:1408–1420, 1986.

[41] J. S. Logsdon and L. T. Biegler. Accurate solution of differential-algebraic optimization problems. *Ind. Eng. Chem. Res.*, 28(11):1628–1639, 1989.

[42] R. Luus. Optimization of fed-batch fermentors by iterative dynamic programming. *Biotechnol. Bioengng.*, 41:599–602, 1992.

[43] R. Luus. Optimal control of batch reactors by iterative dynamic programming. *J. Process Contr.*, 4:218–226, 1994.

[44] R. Luus and O. Rosen. Application of dynamic programming to final state constrained optimal control problems. *Ind. Eng. Chem. Res.*, 30:1525–1530, 1991.

[45] S. Macchietto. Batch process engineering revisited: Adding new spice to old recipes. In *IFAC DYCOPS-5*, Corfu, Greece, 1998.

[46] W. Marquardt. Trends in computer-aided modeling. *Comp. Chem. Eng.*, 20:591–609, 1996.

[47] F. McLennan and B. Kowalski, editors. *Process Analytical Chemistry*. Blackie Academic and Professional, London, 1995.

[48] G. A. Mehta. The benefits of batch process control. *Chem. Eng. Progress*, 79(10):47–52, 1983.

[49] I. M. Mujtaba and S. Macchietto. Efficient optimization of batch distillation with chemical reaction using polynomial curve fitting technique. *Ind. Eng. Chem. Res.*, 36(6):2287–2295, 1997.

[50] B. S. N. Murthy, K. Gangiah, and A. Husain. Performance of various methods in computing optimal policies. *Chem. Eng. Journal.*, 19:201–208, 1980.

[51] C. P. Neuman and A. Sen. A suboptimal control algorithm for constrained problems using cubic splines. *Automatica*, 9:601–613, 1973.

[52] G. D. Nichols. *On-line Process Analyzers*. John Wiley, New York, 1988.

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

[54] S. Palanki, C. Kravaris, and H. Y. Wang. Optimal feedback control of batch reactors with a state inequality constraint and free terminal time. *Chem. Engng. Sci.*, 49(1):85, 1994.

[55] C. C. Pantelides and H. I. Britt. Multipurpose process modeling environments. In *FOCAPD'94*, pages 128–141, Snowmass, CO, 1994.

[56] S. J. Parulekar and H. C. Lim. Modeling, optimization and control of semi-batch fermentation. *Adv. Biochem. Engng. Biotechnol.*, 32:207, 1985.

[57] G. P. Pollard and R. W. H. Sargent. Off line computation of optimum controls for a plate distillation column. *Automatica*, 6:59–76, 1970.

[58] L. S. Pontryagin, V. G. Boltyanskil, R. V. Gamkrelidge, and E. F. Mishchenko. *The Mathematical Theory of Optimal Processes*. Interscience, New York, 1962.

[59] S. Rahman and S. Palanki. State feedback synthesis for on-line optimization in the presence of measurable disturbances. *AIChE J.*, 42:2869–2882, 1996.

[60] W. F. Ramirez. *Application of Optimal Control to Enhanced Oil Recovery*. Elsevier, The Netherlands, 1997.

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

[62] W. H. Ray. *Advanced Process Control*. McGraw-Hill, New York, 1981.

[63] W. H. Ray and J. Szekely. *Process Optimization*. John Wiley, New York, 1973.

[64] G. V. Reklaitis. Scheduling approaches for batch process industries. *ISA Trans.*, 34:349–358, 1995.

[65] J. G. Renfro, A. M. Morshedi, and O. A. Asbjornsen. Simultaneous optimization and solution of systems described by differential/algebraic equations. *Comp. Chem. Eng.*, 11(5):503–517, 1987.

[66] D. W. T. Rippin. Design and operation of multiproduct and multipurpose batch chemical plants: An analysis of problem structure. *Comp. Chem. Eng.*, 7:463–481, 1983.

[67] D. W. T. Rippin. Control of batch processes. In *IFAC DYCORD+'89*, pages 115–125, Maastricht, Netherlands, 1989.

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

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

[70] M. E. Sacks, S. Lee, and J. A. Biesenberger. Optimum policies for batch chain addition polymerizations. *Chem. Engng. Sci.*, 27:2281, 1972.

[71] R. W. H. Sargent and G. R. Sullivan. Development of feed change-over policies for refinery distillation units. *Ind. Engng. Chem. Process Des. Dev.*, 18:113–124, 1979.

[72] H. R. Schwarz. *Numerical Analysis - A Comprehensive Introduction*. John Wiley, New York, 1989.

[73] E. Sorensen, S. Macchietto, G. Stuart, and S. Skogestad. Optimal control and on-line operation of reactive batch distillation. *Comp. Chem. Eng.*, 20(12):1491–1498, 1996.

[74] B. Srinivasan, M. Amrhein, and D. Bonvin. Reaction and flow variants/invariants in chemical reaction systems with inlet and outlet streams. *AIChE J.*, 44(8):1858–1867, 1998.

[75] B. Srinivasan, D. Bonvin, E. Visser, and S. Palanki. Dynamic optimization of batch processes: II. Handling uncertainty using measurements. *Comp. Chem. Eng.*, page Submitted, 2001.

[76] B. Srinivasan, P. Myszkorowski, and D. Bonvin. A multi-criteria approach to dynamic optimization. In *American Control Conference*, pages 1767–1771, Seattle, WA, 1995.

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

[78] B. Srinivasan, O. Ubrich, D. Bonvin, and F. Stoessel. Optimal feedrate policy for systems with two reactions. In *IFAC DYCOPS-6*, pages 455–460, Cheju Island, Korea, 2001.

[79] K. L. Teo, C. J. Goh, and C. C. Lim. 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, 1989.

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

[81] I. M. Thomas and C. Kiparissides. Computation of near-optimal temperature and initiator policies for a batch polymerization reactor. *Can. J. of Chem. Engng.*, 62:284, 1984.

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

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

[84] V. S. Vassiliadis, R. W. H. Sargent, and C. C. Pantelides. Solution of a class of multistage dynamic optimization problems. 1. Problems without path constraints. *Ind. Eng. Chem. Res.*, 33(9):2111–2122, 1994.

[85] V. S. Vassiliadis, R. W. H. Sargent, and C. C. Pantelides. Solution of a class of multistage dynamic optimization problems. 2. Problems with path constraints. *Ind. Eng. Chem. Res.*, 33(9):2123–2133, 1994.

[86] J. Villadsen and M. L. Michelsen. *Solution of Differential Equation Models by Polynomial Approximation*. Prentice-Hall, Englewood Cliffs, 1978.

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

[88] H. Wiederkehr. Examples of process improvements in the fine chemicals industry. *Comp. Chem. Eng.*, 43:1783–1791, 1988.