Lehrstuhl für
Prozesstechnik
Prof. Dr.-Ing. W. Marquardt
RWTH Aachen

Technical Report LPT–2001–05

Flatness and Higher Order Differential Model Representations in Dynamic Optimization

J. Oldenburg, W. Marquardt

February 2001

Published in:
Computers and Chemical Engineering 26 (2002), No. 3, 385–400.

Enquiries should be addressed to:

Lehrstuhl für Prozesstechnik
RWTH Aachen
Templergraben 55
D–52056 Aachen

Tel.: +49 / 241 / 80 94668
Fax: +49 / 241 / 80 92326
E–Mail: secretary@lfpt.rwth–aachen.de

Flatness and higher order differential model representations in dynamic optimization

September 24, 2001

J. Oldenburg, W. Marquardt*
Lehrstuhl für Prozesstechnik, RWTH Aachen, Germany

Abstract

A novel class of methods for solving path and end point constrained dynamic optimization problems is proposed. These methods aim at improving the performance of dynamic optimization algorithms employed in on-line applications where the required solution time is a major concern. The presented approaches are all based on the reformulation of the dynamic model constraints into a higher order differential model representation in which state variable derivatives are eliminated. Based upon this representation, the complete state information can be accessed analytically through explicit equations implying that numerical integration as required by sequential optimization techniques is thus avoided. Since these equations depend on only relatively few variables containing the entire dynamic system behavior advantages over simultaneous optimization strategies can be expected as well. Three different dynamic optimization problem formulations involving higher order differential model representations are discussed, of which the first requires the dynamic system to be differentially flat. The remaining two, however, do not depend on the flatness property. By means of a set of examples, the three problem formulations are illustrated and classified according to their potential to improve the efficiency in solving dynamic optimization problems.

Keywords: Dynamic optimization, Differential flatness, Higher order differential model representations

1 Introduction

Dynamic optimization of chemical process systems is gradually maturing. It can be applied to a wide range of off-line and on-line problems including the design of operational strategies for batch or continuous processes during transient phases (Abel et al., 2000) as well as real-time monitoring and control employing receding horizon strategies (Allgöwer et al., 1999). In particular for real-time applications, computational efficiency and robustness are still important issues. Established direct approaches based on either control vector parameterization (Vassiliadis et al., 1994a; Vassiliadis et al., 1994b) or on control and state vector

*Corresponding author. E-mail address: marquardt@lfpt.rwth–aachen.de parameterization (Biegler, 2000; Leineweber, 1999) rely on state space formulations of differential-algebraic process models, approximate them by a nonlinear program (NLP) and exploit its structure to the extent possible with considerable success. These methods are also termed sequential or simultaneous, respectively.

State space models are, however, not the only possibility to represent nonlinear systems. Motivated by recent results on nonlinear control of flat systems (Fliess & Glad, 1993; Rothfuß et al., 1997) and nonlinear input-output representations (van der Schaft, 1989), we study dynamic optimization based on higher order differential representations where the derivatives of the state can be eliminated from the constraints and explicit equations for the state variables can be generated by symbolic preprocessing. Consequently, numerical integration of model and sensitivity equations is avoided which forms the major computational effort in methods based on control vector parameterization. Further, appropriate discretization strategies may lead to significantly less decision variables as compared to simultaneous approaches based on control and state variable discretization. A related method has recently been suggested by Agrawal & Faiz (1998) for dynamic optimization problems with exactly feedback linearizable nonlinear dynamics. Path and end point constrained dynamic optimization for the class of flat dynamic systems has been considered independently by Mahadevan et al. (2000), Faiz et al. (2000), Kansal et al. (2000), and by Oldenburg & Marquardt (2000). Steinbach (1997) discussed optimization of inverse dynamic systems, an approach which is also based on symbolic elimination of the dynamic model constraints.

This paper attempts to explore the potential of dynamic optimization based on higher order differential system representations in a general sense. A theoretical background for the reformulation of state space models is outlined in Section 3 for the general optimization problem stated in Section 2. Section 3 furthermore shows how these models are integrated into different modified dynamic optimization problem formulations according to the class of dynamic system considered, i.e. flat or non-flat dynamic systems. Suitable parameterization and discretization strategies are presented and related to the established techniques throughout Section 4. Section 5 provides the discussion of three different example optimization problems in order to illustrate and analyze the reformulation approaches presented. Furthermore, Section 5 assesses the potential of each reformulation approach to improve efficiency in dynamic optimization. Finally, Section 6 gives a perspective on the application of the suggested approach and points to further research issues.

2 Dynamic optimization problem

We consider the following dynamic optimization problem

$$\min_{\boldsymbol{x}_0,\, \boldsymbol{u}(t)} \mathcal{J}(\boldsymbol{x}(t_f), \boldsymbol{u}(t_f)) \tag{1}$$

$$\begin{aligned} \text{s.t.} \quad \dot{\boldsymbol{x}}(t) &= \boldsymbol{f}(\boldsymbol{x}(t), \boldsymbol{u}(t)),\ t \in [t_0, t_f]\ , & (2)\\ \boldsymbol{0} &\leq \boldsymbol{c}(\boldsymbol{x}(t), \boldsymbol{u}(t)),\ t \in [t_0, t_f]\ , & (3)\\ \boldsymbol{0} &\leq \boldsymbol{c}_f(\boldsymbol{x}(t_f), \boldsymbol{u}(t_f))\ , & (4)\\ \boldsymbol{0} &= \boldsymbol{x}(t_0) - \boldsymbol{x}_0\ , & (5) \end{aligned}$$

where $\boldsymbol{x} \in \mathbb{R}^n$ are the state variables, $\boldsymbol{u} \in \mathbb{R}^m$ are the control variables and $\boldsymbol{x}_0 \in \mathbb{R}^n$ denotes the initial state. The degrees of freedom $\boldsymbol{u}(t)$ and $\boldsymbol{x}_0$ are chosen to minimize an (economical) objective function $\mathcal{J} : \mathbb{R}^n \times \mathbb{R}^m \to \mathbb{R}$ subject to path and endpoint constraints in $\boldsymbol{x}$, $\boldsymbol{u}$. Without loss of generality, the objective function $\mathcal{J}$ is assumed to be of Mayer type. $\boldsymbol{f} : \mathbb{R}^n \times \mathbb{R}^m \to \mathbb{R}^n$ contains the right hand sides of the nonlinear dynamic system, and $\boldsymbol{c} : \mathbb{R}^n \times \mathbb{R}^m \to \mathbb{R}^{n_p}$, $\boldsymbol{c}_f : \mathbb{R}^n \times \mathbb{R}^m \to \mathbb{R}^{n_f}$ comprise constraints to be enforced either during the duration of the process or at the final time. Note, that differential-algebraic equation (DAE) systems are considered here only to the extent that they can be converted to (2) by symbolic manipulation. This is always possible even for high-index problems, if the algebraic equations are linear in the algebraic variables (Unger et al., 1995; Kumar & Daoutidis, 1995).

3 Model reformulation

Instead of solving (1)–(5) directly by means of established optimization techniques, we reformulate the dynamic model constraints (2). Motivated by recent results on differential flatness (Fliess & Glad, 1993; Rothfuß et al., 1997) we start with q auxiliary variables y_i and relate them by some suitably chosen functions h_i to the states $\boldsymbol{x}$:

$$y_i = h_i(\boldsymbol{x}) := \phi_{i,0},\ i = 1,..,q\ . \tag{6}$$

Each equation (6) is differentiated $\kappa_i \geq 1$ times:

$$\begin{aligned} \dot{y}_i &= \frac{dh_i(\boldsymbol{x})}{dt} = \frac{\partial h_i(\boldsymbol{x})}{\partial \boldsymbol{x}} f(\boldsymbol{x}, \boldsymbol{u}) := \phi_{i,1}\ , \\ &\vdots \\ y_i^{(\kappa_i)} &= \frac{d^{(\kappa_i)} h_i(\boldsymbol{x})}{dt^{(\kappa_i)}} = \ldots := \phi_{i,\kappa_i}\ . \end{aligned} \tag{7}$$

In total, we generate this way $q + \kappa$ auxiliary equations with $\kappa = \sum_{i=1}^{q} \kappa_i$. The functions $y_i^{(k)} = \phi_{i,k}$, $k = 0,..,\kappa_i$ depend on the state variables $\boldsymbol{x}$, but not on their time derivatives which are eliminated by (2) after each differentiation. In general, these functions depend not only on the controls u_j, but also on their time derivatives of higher order. With ℓ_j denoting the highest order derivative of u_j occurring in any of the functions $\phi_{i,k}$, and $\ell = \sum_{j=1}^{m} \ell_j$ we can form:

$$\begin{aligned} \boldsymbol{u}_j &= (u_j, \dot{u}_j, .., u_j^{(\ell_j)})^T,\ j = 1,..,m,\ \boldsymbol{u}_j \in \mathbb{R}^{\ell_j+1}\ , \\ \boldsymbol{U} &= (\boldsymbol{u}_1^T, \boldsymbol{u}_2^T, .., \boldsymbol{u}_m^T)^T,\ \boldsymbol{U} \in \mathbb{R}^{m+\ell}\ , \\ \boldsymbol{y}_i &= (y_i, \dot{y}_i, .., y_i^{(\kappa_i)})^T,\ i = 1,..,q,\ \boldsymbol{y}_i \in \mathbb{R}^{\kappa_i+1}\ , \\ \boldsymbol{Y} &= (\boldsymbol{y}_1^T, \boldsymbol{y}_2^T, .., \boldsymbol{y}_q^T)^T,\ \boldsymbol{Y} \in \mathbb{R}^{q+\kappa}\ , \\ \boldsymbol{\phi}_i &= (\phi_{i,0}, \phi_{i,1}, .., \phi_{i,\kappa_i})^T,\ i = 1,..,q,\ \phi_i \in \mathbb{R}^{\kappa_i+1}\ , \\ \boldsymbol{\Phi} &= (\boldsymbol{\phi}_1^T, \boldsymbol{\phi}_2^T, .., \boldsymbol{\phi}_q^T)^T,\ \boldsymbol{\Phi} \in \mathbb{R}^{q+\kappa}\ . \end{aligned}$$

Hence, equations (6) and (7) can be compactly written as

$$\boldsymbol{Y} - \boldsymbol{\Phi}(\boldsymbol{x}, \boldsymbol{U}) = \boldsymbol{0}\ , \tag{8}$$

a set of $\kappa + q$ equations which are redundant to the original equations (2) from which they have been generated. We now want to check, whether (8) can replace the original model (2) in the sense that any solution of (2),(6) is identical to a solution of (8) for given $\boldsymbol{U}$. Obviously, the equivalence conditions will yield information on the yet undetermined choice of the number of auxiliary variables q, the functions $h_i(\boldsymbol{x})$, and the number of differentiations κ_i.

3.1 Flat dynamic systems

There are special systems (2) which are called *differentially flat*, if there exist auxiliary variables y_i and functions $h_i(\boldsymbol{x})$[1], such that the states $\boldsymbol{x}$ and controls $\boldsymbol{u}$ can be expressed (at least locally) as functions of $\boldsymbol{Y}$ only (Fliess & Glad, 1993; Rothfuß et al., 1997). These functions follow from equations (8) which are then independent of the time derivatives of $\boldsymbol{u}$ in the case considered:

$$\boldsymbol{x} = \boldsymbol{g}_1(\boldsymbol{Y}),\ \boldsymbol{u} = \boldsymbol{g}_2(\boldsymbol{Y})\ . \tag{9}$$

The variables $\boldsymbol{y}_i$ are sometimes called "flat coordinates" or "linearizing outputs" of the dynamic system (2). The number of these variables is directly related to the number of inputs, i.e. $q = m$ must always hold. Flat systems include the presumably more special subclass of nonlinear systems affine in the control variables, e.g. $\boldsymbol{f}(\boldsymbol{x},\boldsymbol{u}) = \boldsymbol{f}_1(\boldsymbol{x}) + \sum_{j=1}^{m} \boldsymbol{f}_{2,j}(\boldsymbol{x})u_j$, which are *exactly state space linearizable*[2] (Isidori, 1995). Hence, for flat systems, equations (9) comprise a completely equivalent representation of equations (2),(6) and thus the state and control variables can be substituted in (1)–(5) to result in

$$\min_{\boldsymbol{Y}(t)} \mathcal{J}(\boldsymbol{g}_1(\boldsymbol{Y}(t_f)), \boldsymbol{g}_2(\boldsymbol{Y}(t_f))) \tag{10}$$

$$\begin{aligned}
\text{s.t.}\quad \boldsymbol{0} &\leq \boldsymbol{c}(\boldsymbol{g}_1(\boldsymbol{Y}(t)), \boldsymbol{g}_2(\boldsymbol{Y}(t))),\ t \in [t_0, t_f]\ ,\\
\boldsymbol{0} &\leq \boldsymbol{c}_f(\boldsymbol{g}_1(\boldsymbol{Y}(t_f)), \boldsymbol{g}_2(\boldsymbol{Y}(t_f)))\ ,\\
\boldsymbol{0} &= \boldsymbol{x}_0 - \boldsymbol{g}_1(\boldsymbol{Y}(t_0))\ ,
\end{aligned}$$

a reformulated optimization problem with (functional) algebraic instead of differential equation constraints. Thereby, the dynamic model constraints (2) are entirely eliminated from the optimization problem (1)–(5). With the solution of problem (10), $\boldsymbol{u}$ and $\boldsymbol{x}$ can be computed from (9). The m auxiliary variables and their κ derivatives concatenated in $\boldsymbol{Y}$ can be interpreted as degrees of freedom of the reformulated problem. The quantities in $\boldsymbol{y}_i$ are, however, not all independent. The existing relations between the elements of $\boldsymbol{y}_i$ will be explored below during the parameterization of the continuous problem.

Though a very compact formulation of the optimization problem has been obtained for flat systems, the approach is not completely satisfactory, since the class of differentially flat systems is known to be limited. In particular, large-scale industrial chemical process models with a large number of states and only few inputs are usually not flat. Kansal et al. (2000) addressed this problem by nonlinear time-scaling (Guay, 1999). This approach generalizes the class of exactly feedback linearizable single-input systems and thus also that of flat systems with a single input variable.

[1] Note, that these functions could also depend on $\boldsymbol{U}$.

[2] Note, that state space linearizability is sufficient for flatness but not necessary (Rothfuß, 1997).

3.2 Non-flat dynamic systems

An extension of the approach presented to general (multi-input) dynamic systems is, however, also possible in a different and rather intuitive way. This extension relies on the fact that the "tendency" of a dynamic system to be differentially flat is increasing with an increasing number of input variables (Rothfuß, 1997).

3.2.1 Defect elimination method

Usually, the number of input variables is determined by the process that is under consideration and each input variable refers to a certain physical entity. Thus, the introduction of any new input variable to the model will certainly alter its structure and thereby its dynamical behavior. In particular, this fact can be exploited to turn a non-flat system into a flat system by manipulating its input structure. This observation led to the idea of augmenting selected equations of the system by fictitious input variables $\boldsymbol{w}$ according to

$$\dot{\boldsymbol{x}} = \boldsymbol{f}(\boldsymbol{x}, \boldsymbol{u}) + \boldsymbol{W}\boldsymbol{w} \ . \tag{11}$$

To determine the number of fictitious input variables to be introduced we can utilize the definition of the "defect" of a nonlinear dynamic system (Fliess et al., 1995). The defect δ, a non-negative integer, is a measure for the distance of a dynamic system to differential flatness[3]. This measure is calculated by comparing the number of flat output functions required for a flat model representation as in (9) with the number of actual independent input variables. By definition, a flat system has no system defect since $\delta = q - m = 0$. Hence, the number of fictitious input variables to be introduced has to be equal to the system defect δ. Therefore, $\boldsymbol{W}$ is a $n \times \delta$ matrix with rank δ and $\boldsymbol{w}$ a $\delta \times 1$ vector. $\boldsymbol{W}$ is an indicator matrix with a single one per row and zero otherwise. Thus, matrix $\boldsymbol{W}$ contains δ entries with the value one. Zero-valued elements are placed in rows that correspond to equations of the original dynamic system which contain input variables $\boldsymbol{u}$. Hence, with a specified $\boldsymbol{W}$ the augmented dynamic system can be analyzed for differential flatness. Obviously, the determination of $\boldsymbol{W}$ will in general be an iterative procedure and, moreover, $\boldsymbol{W}$ will not be unique.

However, an augmented system with fictitious input variables can only be of any use if its dynamical behavior is equivalent to that of the original system (2). The only way to assure this equivalence condition is to manipulate the new input variable values. Fortunately, the natural formulation of a dynamic optimization problem admits to influence their values by simply imposing equality constraints forcing their values to be zero along the entire trajectory.

Assuming that the manipulated system is differentially flat, the fictitious input variables $\boldsymbol{w}$ can be expressed as functions of the flat outputs and their derivatives as in equations (9):

$$\boldsymbol{w}(t) = \boldsymbol{g}_3(\boldsymbol{Y}(t)), \ t \in [t_o, t_f] \ . \tag{12}$$

Hence, the optimization problem formulation

$$\min_{\boldsymbol{Y}(t)} \mathcal{J}(\boldsymbol{g}_1(\boldsymbol{Y}(t_f)), \boldsymbol{g}_2(\boldsymbol{Y}(t_f))) \tag{13}$$

[3] For linear dynamic systems δ is equal to the dimension of the uncontrollable subspace.

$$\begin{array}{rrcl} \text{s.t.} & \mathbf{0} & \leq & \boldsymbol{c}(\boldsymbol{g}_1(\boldsymbol{Y}(t)), \boldsymbol{g}_2(\boldsymbol{Y}(t))),\ t \in [t_0, t_f] \ , \\ & \mathbf{0} & \leq & \boldsymbol{c}_f(\boldsymbol{g}_1(\boldsymbol{Y}(t_f)), \boldsymbol{g}_2(\boldsymbol{Y}(t_f))) \ , \\ & \mathbf{0} & = & \boldsymbol{g}_3(\boldsymbol{Y}(t)),\ t \in [t_0, t_f] \ , \\ & \mathbf{0} & = & \boldsymbol{x}_0 - \boldsymbol{g}_1(\boldsymbol{Y}(t_0)) \ . \end{array}$$

resembles that for flat dynamic systems besides the fact that a set of additional nonlinear equality path constraints is imposed.

Although optimization problem formulations for both flat and non-flat dynamic systems were derived a further approach for general dynamic systems is presented. This approach makes also use of a higher order model representation but does not rely on any specific system property.

3.2.2 Method for higher order differential input-output model representations

The second equation in (9) comprises an input-output representation of a nonlinear flat system which is completely equivalent to the state space representation (2),(6). Input-output representations are, however, also known for general dynamic systems. Van der Schaft (1989) shows that any dynamic system (2),(6) satisfying mild regularity conditions can be reformulated into

$$\begin{array}{rcl} \dot{\boldsymbol{x}}_1 - \boldsymbol{f}_1(\boldsymbol{x}_1, \boldsymbol{x}_2, \boldsymbol{u}) & = & \mathbf{0} \ , \qquad (14) \\ \tilde{\boldsymbol{\Phi}}(\boldsymbol{x}_2, \boldsymbol{U}, \boldsymbol{Y}) & = & \mathbf{0} \ , \qquad (15) \end{array}$$

where $\boldsymbol{x} = (\boldsymbol{x}_1^T, \boldsymbol{x}_2^T)^T$ with $\boldsymbol{x}_1 \in \mathbb{R}^{n'}, \boldsymbol{x}_2 \in \mathbb{R}^{n-n'}$ and functions $\tilde{\boldsymbol{\Phi}} : \mathbb{R}^{n-n'} \times \mathbb{R}^{m+\ell} \times \mathbb{R}^{q+\kappa} \to \mathbb{R}^{n-n'+q}$. Further it is shown, that (15) can always be split into two sets of $n - n'$ and q equations. The first set can be solved (at least locally) for $\boldsymbol{x}_2$ to be then inserted into the second set to result in an input-output representation equivalent to the state space representation (2),(6). It is worth mentioning, that equations (15) are related to (8), since they are also generated by higher order differentiation of (6) and nonlinear combinations of the intermediate derivatives. For a more detailed discussion of the theoretical background and of the corresponding symbolical algorithm we refer to van der Schaft (1989).

Van der Schaft (1989) finally shows, that the states $\boldsymbol{x}_1$ correspond to the unobservable part of (2),(6). In order to illustrate equations (14),(15) and to show the unobservability in (14) we consider the following simple linear example (Rothfuß, 1997):

$$\begin{array}{rcl} \dot{x}_1 & = & -x_1 \ , \qquad (16) \\ \dot{x}_2 & = & x_2 + u \ , \qquad (17) \\ y & = & x_2 \ . \qquad (18) \end{array}$$

This system can be trivially transformed into the following two equations:

$$\begin{array}{rcl} \dot{x}_1 & = & -x_1 \ , \qquad (19) \\ 0 & = & -\dot{y} + y + u \ . \qquad (20) \end{array}$$

Equation (19) corresponds to equation (14) and forms the unobservable part of the system.

If the auxiliary variables y_i and the functions h_i are chosen such that the dynamic system (2),(6) is observable, the differential equations (14) do not occur. Instead of $\boldsymbol{x}_2$, $\boldsymbol{x}$ appears then in (15) which can be solved again at least locally:

$$\boldsymbol{x} = \boldsymbol{g}(\boldsymbol{Y},\boldsymbol{U}) \ . \tag{21}$$

Equation (21) can then be used to form the higher order input-output representation of the dynamic system according to:

$$\boldsymbol{0} = \boldsymbol{\psi}(\boldsymbol{Y},\boldsymbol{U}) \ , \ \boldsymbol{\psi} \in \mathbb{R}^q \ . \tag{22}$$

Since we are free to choose number and type of auxiliary variables y_i and to define functions h_i, we can always fulfill the observability condition. In the (usually not recommendable) extreme case, we could trivially choose $\boldsymbol{y} = \boldsymbol{x}$. Obviously, choosing $\boldsymbol{y} = \boldsymbol{x}$ would not lead to a reformulation and to any potential computational advantage. Hence, with a proper choice of auxiliary variables we can transform the dynamic system (2),(6) into explicit equations for $\boldsymbol{x}$. Then, the original optimization problem (1)–(5) can be reformulated into

$$\min_{\boldsymbol{Y}(t),\,\boldsymbol{U}(t)} \mathcal{J}(\boldsymbol{g}(\boldsymbol{Y}(t_f),\boldsymbol{U}(t_f)),\boldsymbol{u}(t_f)) \tag{23}$$

$$\begin{array}{rrcl} \text{s.t.} & \boldsymbol{0} & = & \boldsymbol{\psi}(\boldsymbol{Y}(t),\boldsymbol{U}(t)), \ t \in [t_0,t_f] \ , \\ & \boldsymbol{0} & \leq & \boldsymbol{c}(\boldsymbol{g}(\boldsymbol{Y}(t),\boldsymbol{U}(t)),\boldsymbol{u}(t)), \ t \in [t_0,t_f] \ , \\ & \boldsymbol{0} & \leq & \boldsymbol{c}_f(\boldsymbol{g}(\boldsymbol{Y}(t_f),\boldsymbol{U}(t_f)),\boldsymbol{u}(t_f)) \ , \\ & \boldsymbol{0} & = & \boldsymbol{x}_0 - \boldsymbol{g}(\boldsymbol{Y}(t_0),\boldsymbol{U}(t_0)) \ . \end{array}$$

where the model constraints (2) are substituted by the equivalent input-output representation in (22).

3.3 Structural differences of the reformulated problems

Although all three reformulation approaches (10),(13) and (23) have in common that state variable derivatives are eliminated and explicit equations for the system state can be stated there is a major structural difference between the flat (cf. problem (10)) and non-flat cases (cf. problems (13),(23)).

Within the method for flat systems the state and input variables are uniquely determined once the flat output functions are specified and their derivatives are calculated. Since the flat output functions $\boldsymbol{y}_i$ contained in $\boldsymbol{Y}$ themselves are not interrelated they can be chosen independently and thereby represent "pure" degrees of freedom of the dynamic system. In fact, when problem (10) is assumed to contain no path constraints and the final time constraints are equalities it can be regarded as an interpolation problem in which optimal trajectories $\boldsymbol{Y}(t)$ between fixed initial and final time points are to be determined.

On the other hand, the approaches for non-flat dynamic systems require additional nonlinear functional equations that interrelate either the auxiliary variables $\boldsymbol{y}_i$ through equation (12) in problem (13) or relate the auxiliary variables

$\boldsymbol{Y}$ to the inputs $\boldsymbol{U}$ through the input-output model representation (22) within problem (23). Hence, the latter approach may be interpreted as a boundary value problem (BVP) in $\boldsymbol{Y}$ for which an optimal input $\boldsymbol{U}$ is to be determined.

The consequences of these properties will be explored below during the discretization of the optimization problems. For reasons of convenience, we introduce abbreviations for the different methods. In particular, we use FS in case flat systems are considered and DE (Defect Elimination) or HO (Higher Order) for the two alternative treatments of non-flat systems.

4 Discretization

In order to determine optimal trajectories for the degrees of freedom the reformulated continuous time optimization problems (10),(13), and (23) have to be approximated by nonlinear programming problems (NLP) by parameterizing the degrees of freedom.

4.1 Problems (10) and (13)

In order to approximate $\boldsymbol{Y}(t)$ we parameterize the highest order derivative of each auxiliary variable y_i by the expansion

$$\tilde{y}_i^{(\kappa_i)}(t) \quad = \quad \sum_{j=1}^{K} \alpha_{i,j} \phi_j(t) \ , \tag{24}$$

where $\phi_j(t)$ are local basis functions represented by piecewise polynomials of order K defined on N_y finite elements with $t \in [t_k, t_{k+1}]$. Furthermore, we exploit the relation between the elements of $\boldsymbol{Y}(t)$ and deduce all lower order derivatives and the auxiliary variables themselves by successive integration:

$$\tilde{y}_i^{(j-1)}(t) \quad = \quad \int \tilde{y}_i^{(j)} dt \ + \gamma_{i,j}^{y} \ , \ 1 \leq j \leq \kappa_i \ . \tag{25}$$

This way, we obtain piecewise continuous polynomials for all derivatives up to order $\kappa_i - 1$ on $t \in [t_k, t_{k+1}]$, if continuity conditions for equations (25) are stated at the element boundaries. The parameters for the discretization of $\boldsymbol{Y}(t)$ include the expansion coefficients $\alpha_{i,j}$ and the constants of integration $\gamma_{i,j}^{y}$ collected in the vector $\boldsymbol{a}$:

$$\boldsymbol{a} \quad = [\boldsymbol{a}_1^T, .., \boldsymbol{a}_q^T]^T \ , \tag{26}$$

$$\boldsymbol{a}_i \quad = [\alpha_{i,1}^1, .., \alpha_{i,K}^{N_y}, \gamma_{i,1}^{y,1}, .., \gamma_{i,\kappa_i}^{y,N_y}]^T \ , \ i = 1, ..., q \ . \tag{27}$$

Hence, for the FS method we have the nonlinear program with the decision variables $\boldsymbol{a}$:

$$\min_{\boldsymbol{a}} \mathcal{J}(\boldsymbol{g}_1(\tilde{\boldsymbol{Y}}_f(\boldsymbol{a})), \boldsymbol{g}_2(\tilde{\boldsymbol{Y}}_f(\boldsymbol{a}))) \tag{28}$$

$$\begin{aligned} \text{s.t.} \quad \boldsymbol{0} \quad &\leq \quad \boldsymbol{c}(\boldsymbol{g}_1(\tilde{\boldsymbol{Y}}_i(\boldsymbol{a})), \boldsymbol{g}_2(\tilde{\boldsymbol{Y}}_i(\boldsymbol{a}))) \ , \ i = 1, ..., KN_y \ , \\ \boldsymbol{0} \quad &\leq \quad \boldsymbol{c}_f(\boldsymbol{g}_1(\tilde{\boldsymbol{Y}}_f(\boldsymbol{a})), \boldsymbol{g}_2(\tilde{\boldsymbol{Y}}_f(\boldsymbol{a}))) \ , \\ \boldsymbol{0} \quad &= \quad \boldsymbol{x}_0 - \boldsymbol{g}_1(\tilde{\boldsymbol{Y}}_0(\boldsymbol{a})) \ . \end{aligned}$$

Note, that $\boldsymbol{Y}_i$ stands for $\boldsymbol{Y}(t_i)$. A similar formulation is obtained for the DE method (cf. problem (13)). In this case we have to additionally incorporate the set of equality constraints

$$\boldsymbol{0} = \boldsymbol{g}_3(\tilde{\boldsymbol{Y}}_i(\boldsymbol{a})) \ , \ i = 1, ..., KN_y \ , \tag{29}$$

into the optimization problem (28). Equation (29) is obtained by applying an appropriate method to discretize the (time dependent) functional equation (12). We here apply collocation on finite elements, a well-known discretization method which is equivalent to implicit Runge-Kutta (IRK) techniques. For more details, we refer to Ascher et al. (1995). Note, that a proper choice of the basis functions in equation (24) also leads to the same type of discretization in (12).

In particular, if the collocation points within each finite element are chosen to be the roots of the Jacobi polynomial this method is referred to as Radau-IRK. The simplest version of this IRK method is the one-stage Radau-IRK where the only collocation point is located at the end of each finite element. Commonly, this method is referred to as the implicit Euler method. Besides one-stage Radau-IRK, we here also apply its two-stage and three-stage variants. It is important to note, that the order of the IRK method is directly related to the order K of the polynomial in the expansion (24). According to Ascher et al. (1995), the order of the polynomial expansion in (24) should be $K \geq \kappa$, where κ denotes the highest order derivative occurring in equation (12). For finite element lengths sufficiently small this leads to an order of consistency of $2K$ for collocation at Gaussian points and $2K - 1$ at Radau points. Despite its lower order of consistency the Radau scheme may have advantages when applied for direct optimization techniques since it is asymptotically stable (L-stable) also for large element lengths (Deuflhard & Bornemann, 1994; Bausa, 2000).

Finally, the optimal solution $\boldsymbol{a}^*$ of (28) is used to obtain the optimal $\boldsymbol{x}^*$ and $\boldsymbol{u}^*$ from (9).

4.2 Problem (23)

When applying the HO method (cf. problem (23)) we have to parameterize the highest order derivative of the inputs in addition to that of the auxiliary variables. Again, we specify trial functions for the highest order derivative of each input variable on N_u finite elements with $t \in [t_k, t_{k+1}]$ similarly to (24),

$$\tilde{u}_i^{(\ell_i)}(t) \quad = \quad \sum_{j=1}^{K_u} \beta_{i,j} \phi_j(t) \ , \tag{30}$$

and deduce lower order derivatives of the input trial functions by successive integration:

$$\tilde{u}_i^{(j-1)} \quad = \quad \int \tilde{u}_i^{(j)} dt \ + \gamma_{i,j}^u \ , \ 1 \leq j \leq \ell_i \ . \tag{31}$$

The parameters are concatenated in the vector $\boldsymbol{b}$:

$$\boldsymbol{b} \quad = [\boldsymbol{b}_1^T, .., \boldsymbol{b}_m^T]^T \ , \tag{32}$$

$$\boldsymbol{b}_i \quad = [\beta_{i,1}^1, .., \beta_{i,K_u}^{N_u}, \gamma_{i,1}^{u,1}, .., \gamma_{i,\ell_i}^{u,N_u}]^T \ , \ i = 1, ..., m \ . \tag{33}$$

As a result, we obtain the nonlinear program

$$\min_{\boldsymbol{a},\,\boldsymbol{b}} \mathcal{J}(\boldsymbol{g}(\tilde{\boldsymbol{Y}}_f(\boldsymbol{a}),\tilde{\boldsymbol{U}}_f(\boldsymbol{b})),\tilde{\boldsymbol{u}}_f(\boldsymbol{b})) \tag{34}$$

$$\begin{array}{lrcl} \text{s.t.} & \boldsymbol{0} & = & \boldsymbol{\psi}(\tilde{\boldsymbol{Y}}_i(\boldsymbol{a}),\tilde{\boldsymbol{U}}_i(\boldsymbol{b}))\ ,\ i=1,...,KN_y\ , \\ & \boldsymbol{0} & \leq & \boldsymbol{c}(\boldsymbol{g}(\tilde{\boldsymbol{Y}}_i(\boldsymbol{a}),\tilde{\boldsymbol{U}}_i(\boldsymbol{b})),\tilde{\boldsymbol{u}}_i(\boldsymbol{b}))\ ,\ i=1,...,KN_y\ , \\ & \boldsymbol{0} & \leq & \boldsymbol{c}_f(\boldsymbol{g}(\tilde{\boldsymbol{Y}}_f(\boldsymbol{a}),\tilde{\boldsymbol{U}}_f(\boldsymbol{b})),\tilde{\boldsymbol{u}}_f(\boldsymbol{b}))\ , \\ & \boldsymbol{0} & = & \boldsymbol{x}_0-\boldsymbol{g}(\tilde{\boldsymbol{Y}}_0(\boldsymbol{a}),\tilde{\boldsymbol{U}}_0(\boldsymbol{b}))\ , \end{array}$$

for which again collocation on finite elements based on Radau points is applied. The optimal $\boldsymbol{u}^*$ is determined from the optimal $\boldsymbol{a}^*$, $\boldsymbol{b}^*$ by (30),(31). Subsequently, the optimal $\boldsymbol{x}^*$ may be computed from (21).

4.3 Complexity considerations

Finally, we compare the complexity of the NLP of the suggested approaches with that of established direct optimization methods. This task is, however, not trivial since the complexity of the NLP depends on a number of factors which are difficult to rank. Furthermore, since the complexity of the NLP is a measure that is directly related to the solution technique applied we prefer to explore it below in conjunction with the example problems. But, one of the complexity measures, the number of parameters required in the NLP, can be directly compared as follows. Assuming for simplicity $N_y = N_u = N$, $K = K_u$, a number of $K \cdot (q+m) \cdot N + N \cdot \sum_{i=1}^{q} \kappa_i + N \cdot \sum_{j=1}^{m} \ell_j$ parameters is required for the HO method and $K \cdot q \cdot N + N \cdot \sum_{i=1}^{q} \kappa_i$ parameters for the FS and DE methods. These numbers compare to $K \cdot m \cdot N + n$ and $K \cdot (n+m) \cdot N + n \cdot N$ for the sequential and simultaneous methods, respectively. Here, K denotes the order of discretization (Ascher et al., 1995) for higher order differential equation systems as well as for first order ODEs.

The number of parameters required in the NLP is always significantly lower for the FS method as compared to the simultaneous method. It is, however, important to note, that the simultaneous formulation is inherently sparse whereas equations (9) employed to represent the states and inputs in the FS method lead to denser Jacobians and thus some loss of efficiency has to be expected despite the reduced number of parameters. The computational effort of the FS method can be expected much lower as compared to the sequential method, because with a comparably low number of parameters there is no need to numerically solve sensitivity differential equations in each iteration of the NLP solution algorithm. The same holds for the DE method besides the fact that an additional set of δ equality constraints is enforced. In case the HO method (cf. problem (23)) is applied, the number of parameters for the method depends on q, κ and ℓ. If a small number of auxiliary variables q is sufficient to ensure observability, a large $\kappa + \ell$, the total number of differentiated quantities, must be expected. Thus, since the contribution of κ, ℓ to the total number of parameters is small, the number of parameters required for the suggested approach is smaller as compared to the simultaneous method. If, on the other hand, $q \to n$, $\kappa \to q$, $\ell \to 0$, the complexity of the parameterization of the HO method approaches that of a simultaneous method. The potential saving of parameters in the HO approach without a loss of accuracy is also confirmed by the higher order collocation theory (Ascher et al., 1995). However, the local error estimates of the HO and DE methods strongly depend on smoothness properties of problems (13), (23), respectively, a fact which will become obvious in the subsequent section.

5 Illustrative examples

Within this section, the three different reformulation approaches (10),(13) and (23) are illustrated and analyzed by means of a set of example problems. We start with an example of a fermentation process representing the case of flat dynamic systems (cf. problem (10)). Subsequently, the defect elimination method of a non-flat system (cf. problem (13)) is illustrated with a semi-batch reactor and a simple batch distillation column. Finally, this section is concluded with an example of a non-flat system which is optimized using a higher order differential input-output model representation (cf. problem (23)).

5.1 Penicillin fermentation

Fermentation processes can be employed for the synthesis of the valuable antibiotic Penicillin G. In particular, bio-reactors in a fed batch operation mode have been widely used for that purpose. We here regard a fed batch process for the production of Penicillin which is modeled as follows:

$$\begin{aligned}
\frac{dx_1}{dt} &= \mu(x_2)x_1 \ , && (35)\\
\frac{dx_2}{dt} &= -\frac{1}{p_1}\mu(x_2)x_1 - \frac{1}{p_5}\mu_{pen}x_1 - m_s x_1 + p_2 u_1 \ , && (36)\\
\frac{dx_3}{dt} &= -\frac{1}{y_{pp}}x_1 + p_6 u_2 \ , && (37)\\
\frac{dx_4}{dt} &= \mu_{pen}x_1 - p_7 x_4 \ , && (38)\\
\text{with} \quad \mu(x_2) &= \frac{\mu_{max}x_2}{K_s V + x_2} \ .
\end{aligned}$$

For further details about the process including model constants, parameters and information about a laboratory scale plant we refer to (Rothfuß, 1997). The objective of the dynamic optimization is to drive the growth rate $\mu(x_2)$ and the concentration of the precursor $c_3 = x_3/V$ as quickly as possible to prespecified constant values and to keep their trajectories as close as possible to these values throughout the entire batch. The optimization problem is formulated subject to the dynamic model and path constraints as:

$$\min_{u_1(t),\, u_2(t)} \sum_{i=1}^{N} k_1(\mu(t_i) - 0.02)^2 + k_2(c_3(t_i) - 0.3)^2 \qquad (39)$$

$$\begin{aligned}
\text{s.t.} \quad & \textit{Equations}\ (35)\ \textit{to}\ (38)\ ,\\
& k_1 = 20.0\ ,\ k_2 = 2.0\ ,\\
& 0.1\ \leq c_3(t) \leq\ 0.4\ g/l\ ,\\
& 0.0\ \leq \mu(t) \leq\ 0.06\ 1/h\ ,
\end{aligned}$$

$$
\begin{aligned}
0.0 &\le u_1(t) \le 1.0\, l/h\ ,\\
0.0 &\le u_2(t) \le 1.0\, l/h\ ,\\
&\forall\, t\ \in\ [0, t_f];\ t_f = 30\, h\ .
\end{aligned}
$$

The initial conditions of the four differential states are $\boldsymbol{x}_0 = [1.5\,kg, 2\,kg, 25\,g, 1.6\,kg]^T$. The degrees of freedom are the two substrate feed rates glucose, u_1 and precursor u_2. An analysis of the fermenter model equations (35)–(38) reveals that the nonlinear system is differentially flat. By identifying two flat output functions, $y_1 = x_4$ ($\kappa_1 = 3$) and $y_2 = x_3$ ($\kappa_2 = 1$) the model can be expressed in terms of explicit equations for the system state and input (Rothfuß et al., 1997):

$$x_4 = y_1\ , \tag{40}$$

$$x_3 = y_2\ , \tag{41}$$

$$x_2 = g_{1,1}(y_1, \dot{y}_1, \ddot{y}_1)\ , \tag{42}$$

$$x_1 = g_{1,2}(y_1, \dot{y}_1)\ , \tag{43}$$

$$u_1 = g_{2,1}(y_1, \dot{y}_1, \ddot{y}_1, y_1^{(3)})\ , \tag{44}$$

$$u_2 = g_{2,2}(y_1, \dot{y}_1, \dot{y}_2)\ . \tag{45}$$

These equations simultaneously serve as a proof for differential flatness of (35)–(38), since the condition $q = m$ holds. Now, the highest order derivatives of the flat output functions are parameterized by piecewise constant trial functions on $N = 50$ equidistant time intervals according to (24). The arising NLP is then solved using SNOPT (Gill et al., 1998), a solver capable to exploit the sparsity of the Jacobian of the constraints. The Jacobian is determined efficiently by applying an automatic differentiation method (Bischof et al., 1996). The optimal profiles are presented in Figures (1) and (2). The optimization results are compared to results obtained with our own implementation of a collocation-based simultaneous approach (SIM) incorporating SNOPT (Gill et al., 1998) and with a sequential strategy (SEQ) implemented in DYNOPT (Abel et al., 1999). An optimality tolerance of 10^{-6} was employed for all cases. Two facts are most conspicuous when comparing the optimization results. Firstly, the proposed method outperforms the established techniques regarding the required computing time. Secondly, even though the total computing time is shorter for the new method it requires a larger number of SQP major iterations, a fact that might seem surprising at first sight. Moreover, the average computing time required for solving one QP subproblem is shorter when applying the FS method. To find a reason for the fast QP solutions and the slow SQP convergence rate, we take a closer look at the different dynamic optimization techniques in close relation to the model representation that was employed.

Table 1: Optimization results – Penicillin fermentation

Method	Objective	NLP par.	SQP ‖	ø time	QP ‖	ø time	***time*** [sec]
FS	$6.7 \cdot 10^{-6}$	300	101	0.185	984	0.019	**18.7**
SIM	$6.4 \cdot 10^{-6}$	500	86	0.36	1333	0.023	**31.0**
SEQ	$1.4 \cdot 10^{-6}$	100	30	3.11	512	0.182	**93.3**

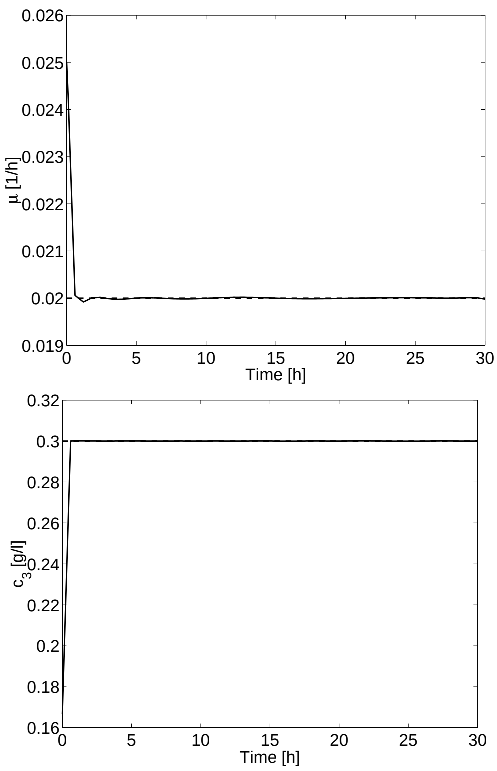

Figure 1: Rate of growth μ, concentration of the precursor c_3.

At first, the fast QP solution of the FS method is explained by the fact that all model constraints were eliminated by the problem reformulation. Furthermore, function evaluation is rather cheap since state and input information is available analytically. Thus, also exact gradient information is accessed at low cost since automatic differentiation methods can be directly applied. As a result, the combination of a cheap function and gradient evaluation and a compact QP formulation leads to very quick major iterations within an SQP framework.

The QP problem solved in the SEQ approach displays an equivalent structure. In fact, it is even more advantageous than the QP of the FS approach, since the objective function and path or end point constraints are in general less complex in the original problem formulation (1)–(5) without any symbolic preprocessing. However, function and gradient evaluations come at a high price within the sequential strategy due to the necessity of numerical state and sensitivity integration which, in contrast, is avoided by the FS method.

The SIM method requires a series of comparably complex QP subproblems to be solved within the SQP algorithm since a large set of discretized dynamic model equality constraints is enforced. At the same time, the number of decision variables is significantly higher as compared to the other approaches. Despite the fact that function and gradient evaluation is rather cheap, we end up with QP subproblems that require more computing time than the FS approach.

Secondly, the slow convergence rate of the SQP method when employed in the

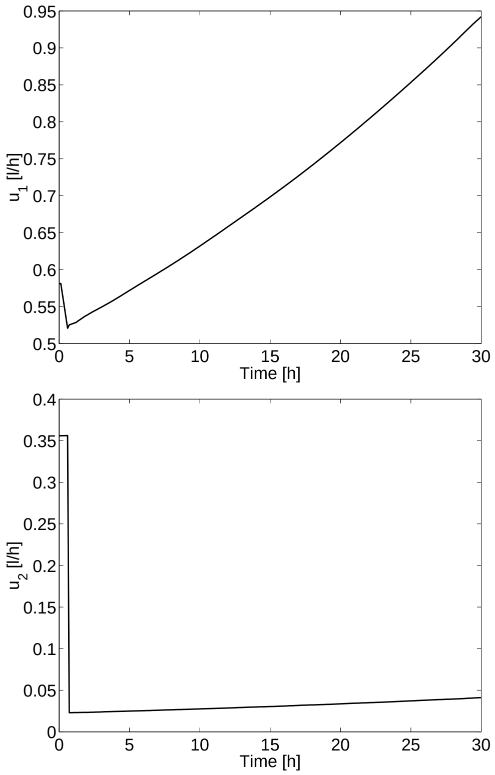

Figure 2: Feed rates u_1, u_2.

FS approach is a consequence of an inadequate search direction $\boldsymbol{p}_k$ determined in the QP subproblems. To find a reason for the "bad" direction itself, we have to recall that $\boldsymbol{p}_k$ is determined by minimizing a quadratic approximation of a (modified) Lagrangian subject to a set of linearized constraints around an iterate $\boldsymbol{x}_k$ (for more details, see Gill et al. (1998)). Or in other words, a local minimization is responsible for the global progress of the nonlinear optimization algorithm. Qualitatively speaking, a local minimization is expected to provide a "good" search direction $\boldsymbol{p}_k$ if the original problem is only moderately nonlinear since in this case, the QP subproblems properly reflect the nonlinear problem globally[4]. This fact becomes even more relevant when recalling that second order derivatives of the Lagrangian are not calculated exactly but approximated by BFGS quasi-Newton updates where curvature information is accumulated on the basis of first order derivatives.

A flat model representation contains the dynamic system information in a very compact way which inevitably leads to rather complex and highly nonlinear functions $\boldsymbol{g}_1$ and $\boldsymbol{g}_2$. Moreover, only few flat output functions and their time derivatives hold the entire dynamic system behavior. Hence, the corresponding QP subproblems involving the discretized functions $\boldsymbol{g}_1, \boldsymbol{g}_2$ will in general be less reliable in good search directions $\boldsymbol{p}_k$ and hence in providing fast global conver-

[4] An extreme case would be a quadratic objective subject to linear constraints which is solved within one step.

gence. The nonlinearity of the discretized problem (10) is further increased in the presence of inequality constraints that are active at least temporarily over the course of the optimization since the flat output functions $\boldsymbol{y}$ are assumed to be continuously differentiable. Thus, the Hessian of problem (10) has more curvature than the one of the sequential or simultaneous formulations. As a consequence, the accumulation of adequate Hessian information takes additional time and iterations.

5.2 Non-flat dynamic systems

5.2.1 Semi-batch reactor

In order to illustrate the DE method (cf. problem (13)) we treat the productivity optimization of a stirred semi-batch reactor in which a strongly exothermic reaction $A \to B \to C$ takes place. B is the desired product while C denotes an undesired byproduct. Before any reactant A is fed, the reactor is filled with a solvent. Assuming both (first order) reactions taking place in the liquid phase of constant density ρ_r and heat capacity c_{p_r} the reactor model is

$$
\begin{aligned}
\frac{dn_a}{dt} &= F_a - k_1 e^{\left(\frac{-E_1}{RT_r}\right)} n_a \ , && (46)\\
\frac{dn_b}{dt} &= k_1 e^{\left(\frac{-E_1}{RT_r}\right)} n_a - k_2 e^{\left(\frac{-E_2}{RT_r}\right)} n_b \ , && (47)\\
\frac{dn_c}{dt} &= k_2 e^{\left(\frac{-E_2}{RT_r}\right)} n_b \ , && (48)\\
\frac{dT_r}{dt} &= \frac{F_a M}{V_r \rho_r}(T_f - T_r) - k_1 e^{\left(\frac{-E_1}{RT_r}\right)} \frac{c_a}{\rho_r c_{p_r}} \Delta H_1 && (49)\\
&- \ k_2 e^{\left(\frac{-E_2}{RT_r}\right)} \frac{c_b}{\rho_r c_{p_r}} \Delta H_2 + \frac{\alpha_k A_k}{\rho_r V_{r,0} c_{p_r}} (T_w - T_r) \ , &&\\
V_r &= \frac{nM}{\rho_r} + V_r^0 , n = \sum n_i , c_i = \frac{n_i}{V_r} , i = a, b, c \, . && (50)
\end{aligned}
$$

It should be noted that (50) can be substituted into (46)–(49) to result in a system of type (2). The model parameters can be taken from Table 2. The optimal operational strategy should maximize the product concentration c_B at

Table 2: Model parameters – Semi-batch reactor

Parameter	Value	Parameter	Value
k_1	$15.0\,1/s$	k_2	$85.0\,1/s$
E_1	$30000\,kJ/kmol$	E_2	$40000\,kJ/kmol$
M	$50.0\,kg/kmol$	ρ_r	$1000\,kJ/m^3$
c_{p_r}	$3.9\,kJ/kg/K$	T_f	$300\,K$
ΔH_1	$-40000\,kJ/kmol$	ΔH_2	$-50000\,kJ/kmol$
α_k	$0.5\,kJ/s/m^2/K$	A_k	$8.0\,m^2$
c_{p_w}	$3.1\,kJ/kg/K$	$V_{r,0}$	$1.0\,m^3$
V_k	$0.4\,m^3$	ρ_w	$700\,kg/m^3$

the endpoint t_f. The only degree of freedom u of the process is the reactant feed flow $F_a(t)$. The cooling water in the reactor jacket (ρ_w, c_{p_w}) is held at a constant temperature $T_w = 300\,K$. At the final batch time, the integral amount of added feed is fixed to a predefined value and the reactant concentration $c_a(t_f)$ is restricted by an upper bound. The path constraint covers the case of a potential total loss of cooling capacity by bounding the adiabatic end temperature T_{ad} (Stoessel, 1995):

$$T_{ad} = T_r + \frac{V_r[-(\Delta H_1 + \Delta H_2)c_a - \Delta H_2 c_b]}{\rho_r c_{p_r} V_r + \rho_w c_{p_w} V_k} \tag{51}$$

Thus, the following dynamic optimization problem can be formulated:

$$\min_{F_a(t)} \; -c_B(t_f) \tag{52}$$

$$\begin{aligned} \text{s.t.} \quad & \textit{Equations } (46) \textit{ to } (50) \\ & 0\,K \leq T_{ad}(t) \leq \; 453K \\ & n(t_f) \; = 20\,kmol \;, \\ & 0 \leq c_a(t_f) \leq 1.7\,kmol/m^3 \;, \\ & 0 \leq F_a(t) \leq \; 180\,kmol/h \;, \\ & \quad \forall\, t \;\in\; [0, t_f];\; t_f = 6\,h \;. \end{aligned}$$

The initial state is fixed according to $n_i(t_0) = 0\,kmol,\, i = a, b, c;\, T_r(t_0) = 300\,K$. The process model (46)–(50) is *not* differentially flat. In fact, the system has defect 1 since two flat output functions would be required for a model transformation. We modify the enthalpy balance of the process model by adding a fictitious input variable w to (49). This is equivalent to introducing a new input variable $\bar{T}_w$ by augmenting the cooling temperature T_w by $\bar{w}$ according to:

$$\bar{T}_w = T_w + \bar{w},\; \bar{w} = \frac{\rho_r V_{r,0} c_{p_r}}{\alpha_k A_k} w \;. \tag{53}$$

Obviously, $w = \bar{w} = 0$ fixes the cooling temperature to its predefined value.

Thus, explicit equations (9) can be found for $\boldsymbol{x} = [n_a, n_b, n_c, T_r]^T$, $u = F_a$, and w with two suitable flat output functions which are determined by physical insight (Rothfuß et al., 1997). The molar holdup $n_c = y_1$ ($\kappa_1 = 3$) and the reactor temperature $T_r = y_2$ ($\kappa_2 = 2$) are chosen as flat outputs. Each of them characterizes one "subsystem" of the model. They relate to two subsystems comprising the material and enthalpy balances of the reactor, respectively. Thereby, we obtain the following set of equations of type (9):

$$\begin{aligned} n_c &= y_1 \;, && (54) \\ n_b &= g_{11}(\dot{y}_1, y_2) \;, && (55) \\ n_a &= g_{12}(\dot{y}_1, \ddot{y}_1, y_2, \dot{y}_2) \;, && (56) \\ T_r &= y_2 \;, && (57) \\ F_a &= g_{21}(\dot{y}_1, \ddot{y}_1, y_1^{(3)}, y_2, \dot{y}_2, \ddot{y}_2) \;, && (58) \\ w &= g_{22}(\dot{y}_1, \ddot{y}_1, y_1^{(3)}, y_2, \dot{y}_2, \ddot{y}_2) \;. && (59) \end{aligned}$$

Note, that w in equation (59) represents the fictitious input variable which is forced to zero by imposing an equality path constraint of type (12) at each collocation point t_i in $[t_0, t_f]$. The optimization problem (52) is solved using equations (54)–(59) by employing three-stage Radau-IRK discretization in order to satisfy $K \geq max(\kappa_1, \kappa_2) = 3$. Thus, the highest order derivatives of the output variables are approximated by piecewise polynomials of order three according to equation (24) while the time horizon is divided into 40 equidistant finite elements. Figures 3,4 illustrate the optimal profiles of the states and inputs while Table 3 summarizes the optimization results and reveals a comparison to the established techniques, i.e. our own implementation of a simultaneous solver (SIM) employing three-stage Radau-IRK discretization and SNOPT as well as DYNOPT with piecewise constant control approximation representing the sequential method (SEQ). As in the previous example, the DE approach requires the largest number of major iterations while the average computing time needed for solving one QP subproblem is still relatively short. However, the DE approach reveals a lower total computational efficiency relative to the SIM and SEQ method as the FS approach discussed in the previous section. This fact can be explained by the comparably large number of (linearized) QP constraints.

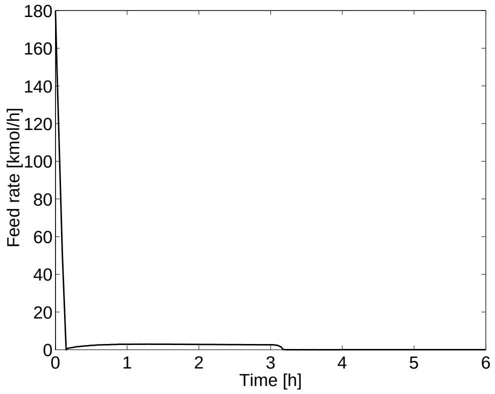

Figure 3: Feed rate F_a.

Table 3: Optimization results – Semi-batch reactor

Method	Objective	NLP par.	SQP ∥ ø time		QP ∥ ø time		***time*** [sec]
DE	−7.4176	440	13	1.165	717	0.0211	**15.1**
SIM	−7.4114	680	8	1.825	771	0.0189	**14.6**
SEQ	−7.4116	40	11	1.136	125	0.1	**12.5**

Generalization of these results is, however, rather difficult since the computing time underlies significant variations depending on chosen input parameters of the SQP and integration method. Depending on the example treated, the DE method may help to reduce computing time while it will be always inferior to the FS approach discussed in the previous section.

It is, however, important to note, that the potential of the suggested method

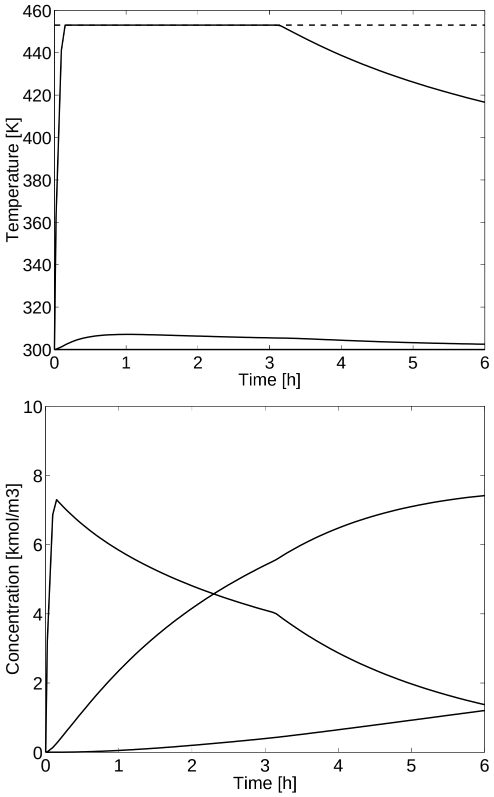

Figure 4: Reactor temperature T_r, adiabatic end temperature T_{ad} (top) and concentrations c_a, c_b, c_c (bottom).

also relies on characteristics of the dynamic optimization problem to be solved rather than the isolated dynamic process model itself. This hypothesis shall be explained by applying the DE method to a batch distillation example problem described next.

5.2.2 Batch distillation column

Within this section we consider the optimization of a single-stage ideal binary batch distillation column. For the distillation model we assume a constant relative volatility $\alpha = 1.5$ for both components throughout the column and a theoretical tray, i.e. the vapor leaving the tray is in equilibrium with the liquid on the tray. Further simplifying assumptions are a negligible vapor holdup and a constant vapor flow rate of $V = 50\ kmol/h$. A Francis weir formula is used to relate the liquid holdup on a tray to the liquid flow L rate leaving the tray. The batch column consists of a still pot charged with the feed mixture, one ideal tray, a total condenser, and a reflux drum. A reflux valve splits the stream leaving the reflux drum into a distillate and a reflux stream.

The process is modeled by the following set of differential equations:

$$\begin{aligned}
\frac{dx_{B,1}}{dt} &= \frac{1}{M_B}(L_{in,B}(x_{S,1}-x_{B,1}) - V(y_{B,1}-x_{B,1}))\ , && (60)\\
\frac{dM_B}{dt} &= L_{in,B} - V\ , && (61)\\
L_{in,B} &= L_{out,S} = k_D(M_S - M_{S,w})^{\frac{3}{2}}\ , && (62)\\
\frac{dx_{S,1}}{dt} &= \frac{1}{M_S}(L_{in,S}(x_{C,1}-x_{S,1}) && \\
&\quad -V(y_{S,1}-x_{S,1}) + V(y_{B,1}-x_{S,1}))\ , && (63)\\
\frac{dM_S}{dt} &= L_{in,S} - L_{out,S}\ , && (64)\\
L_{in,S} &= L_{Re,C} = k_D(M_C - M_{C,w})^{\frac{3}{2}}\ , && (65)\\
\frac{dx_{C,1}}{dt} &= \frac{1}{M_C}(V(y_{S,1}-x_{C,1}))\ , && (66)\\
\frac{dM_C}{dt} &= V - L_{Re,C} - L_{Dist,C}\ , && (67)\\
\alpha &= \frac{y_{j,1}(1-x_{j,1})}{(1-y_{j,1})x_{j,1}},\ j = B,S,C,\ R = \frac{L_{Re,C}}{L_{Dist,C}}\ . && (68)
\end{aligned}$$

The model parameters required for the weir formulae are specified as $M_{S,w} = 1.0\ kmol$, $M_{C,w} = 3.0\ kmol$ and $k_D = 1581,139\ (kmol^{\frac{3}{2}}s)^{-1}$. We search for the optimal reflux policy to obtain the highest purity possible in 1 $kmol$ distillate D and a fixed batch time of 1 $hour$ for given initial states. The amount of distillate and its purity at the final time is evaluated by discretization of equations (69),(70):

$$\begin{aligned}
x_{D,1}(t_f) &= \frac{1}{D}\int_{t_0}^{t_f} L_{Dist,C}\, x_{C,1}\, dt\ , && (69)\\
D &= \int_{t_0}^{t_f} L_{Dist,C}\, dt\ . && (70)
\end{aligned}$$

Alternatively, $x_{D,1}(t_f)$ and D could also be determined explicitly by numerical integration of (69),(70).

Hence, with fixed initial values for the molar fractions $x_{j,1}(t_0) = 0.5\ ,\ j = B,S,C$, and holdups $M_{B,1}(t_0) = 10.0\,kmol$, $M_{S,1}(t_0) = 1.09875\,kmol$, $M_{C,1}(t_0) = 3.09875\,kmol$, the optimization problem can be stated as:

$$\min_{R(t)}\ -x_{D,1}(t_f) \qquad (71)$$

$$\begin{aligned}
\text{s.t.}\quad & \mathit{Equations}\ (60)\ \mathit{to}\ (70)\ ,\\
& 1.0\ kmol \le D(t_f)\ ,\\
& 30.0 \le R(t) \le 120.0\ ,\\
& \quad \forall\, t\ \in\ [0,t_f];\ t_f = 1\ h\ .
\end{aligned}$$

An analysis of the original dynamic system (60)–(68) reveals that two flat output functions are required to obtain a flat model representation (9). Hence, with $R(t)$

being the only input variable the system defect is 1. When adding a fictitious input variable w to the right hand side of the material balance of the condenser,

$$\frac{dx_{C,1}}{dt} = \frac{1}{M_C}(V(y_{S,1} - x_{C,1})) + w \ , \tag{72}$$

the system can be converted into a flat representation (9),(12) with equation (72) replacing the original balance (67). Then, the corresponding flat output functions are the molar holdup $y_1 = M_B$ ($\kappa_1 = 3$) and the mole fraction $y_2 = x_{B,1}$ ($\kappa_2 = 3$) in the still pot of the column. Thus, the number of model variables is reduced by 4 using the two flat output functions to represent the nonlinear system dynamics. The dynamic model can be expressed in terms of the following explicit equations for the states and inputs:

$$\begin{aligned}
x_{B,1} &= y_1 \ , &(73)\\
M_B &= y_2 \ , &(74)\\
x_{S,1} &= g_{1,1}(y_1, \dot{y}_1, y_2, \dot{y}_2) \ , &(75)\\
M_S &= g_{1,2}(\dot{y}_2) \ , &(76)\\
x_{C,1} &= g_{1,3}(y_1, \dot{y}_1, \ddot{y}_1, y_2, \dot{y}_2, \ddot{y}_2) \ , &(77)\\
M_C &= g_{1,4}(\dot{y}_2, \ddot{y}_2) \ , &(78)\\
R &= g_{2,1}(\dot{y}_2, \ddot{y}_2, y_2^{(3)}) \ , &(79)\\
w &= g_{3,1}(y_1, \dot{y}_1, \ddot{y}_1, y_1^{(3)}, \dot{y}_2, \ddot{y}_2, y_2^{(3)}) \ . &(80)
\end{aligned}$$

Note, that w represents the fictitious input variable which will be forced to zero by an additional equality path constraint. The time-continuous problem

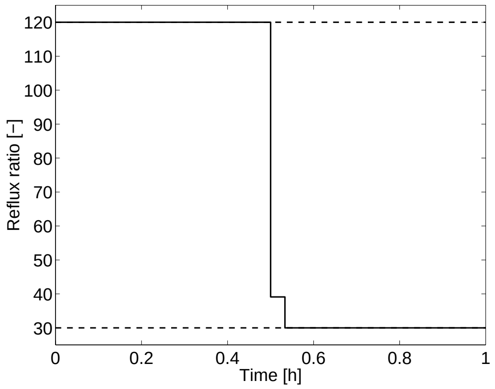

Figure 5: Optimal reflux ratio $R(t)$.

is discretized by employing a three-stage Radau-IRK scheme in order to satisfy $K \geq max(\kappa_1, \kappa_2) = 3$. Hence, the highest order derivatives of the output variables are approximated by piecewise polynomials of order three according to equation (24). The time horizon is divided into 30 equidistant elements. However, by directly applying the DE method to solve problem (71) using equations (73)–(80) we failed to obtain an optimal solution that (at least approximately) matches the "true" analytical optimal solution which is a bang-bang profile in

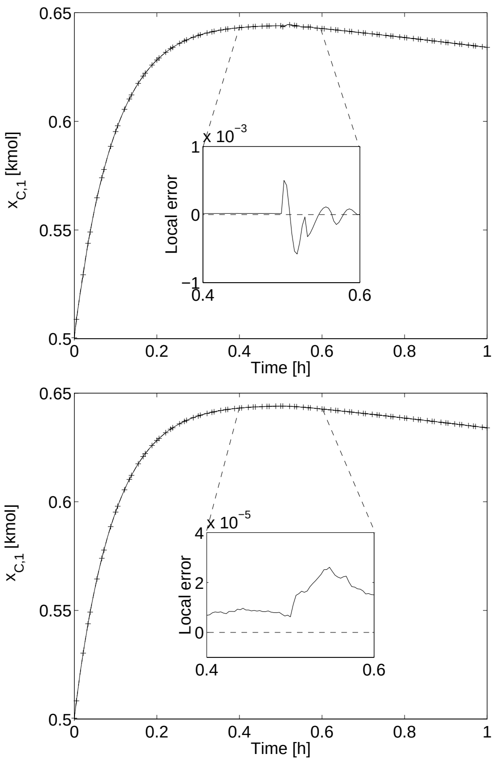

Figure 6: Condenser mole fraction for higher (top), first order ODE (bottom) and corresponding local integration error, + := collocation point.

the reflux ratio $R(t)$. An optimal control profile approximated by piecewise constant functions on 30 time elements which was determined with DYNOPT is depicted in Figure 5. Instead, with the DE method we determine a control profile that oscillates between the lower and upper control bound around $t = 0.5\ h$ leading to a slightly lower objective value than the one obtained with DYNOPT. To find a reason for these oscillations we applied the three-stage Radau-IRK method to the higher order ODE system (79),(80) and also to the original ODE (60)–(68) with a given control profile R taken from Figure 5. Figure 6 shows the corresponding integration errors in the condenser mole fraction profiles for the two ODE systems. Both where compared to a BDF method (SpeedUp, 1995) applied to the first order ODE system (60)–(68)(dashed lines). In the vicinity of the discontinuity, the solution of the higher order ODE reveals a local integration error which is significantly higher when compared to the original ODE system. This integration error leads to a decreased objective value since the integral of $x_{c,1}$ over $t \in [t_0, t_f]$ is slightly higher (see Figure 6). Thus, an oscillating control profile further decreases the objective function value at the expense of a higher integration error.

Having accounted for the oscillating control profile obtained with the DE approach we now examine the cause for the comparably high integration error itself. By means of Figure 6 it is easily identified that the first order ODE model

(60)–(68) is solved with higher accuracy around the point of discontinuity in R when compared to the solution of the corresponding higher order system although the same IRK-method on the same mesh was employed. At first sight, one may be surprised by the worse numerical integration result for the higher order ODE (79),(80). In order to investigate this behavior, equation (79) is solved for $y_2^{(3)}$, the highest order derivative of y_2. The complex and highly nonlinear right hand side of the ODE contains mixed nonlinear functions of R and the lower order derivatives $\ddot{y}_2, \dot{y}_2$:

$$y_2^{(3)} = f(\ddot{y}_2, \dot{y}_2, R) \; . \tag{81}$$

Ascher et al. (1995) show that the local error estimates of the method strongly depend on smoothness properties of the higher order ODE. Since (81) is nonlinear, we have to analyze the linearized ODE coefficients

$$c_i = \frac{\partial f(\ddot{y}_2, \dot{y}_2, R)}{\partial y_2^{(i)}} \; , \; i = 0, 1, 2 \; , \tag{82}$$

for the smoothness condition. Evaluation of (82) yields that the coefficients c_1, c_2 themselves depend on $\ddot{y}_2, \dot{y}_2$ and R. Recalling that R contains a jump discontinuity, it is obvious that the coefficients c_1, c_2 are as well discontinuous and thereby cause a dropping order of consistency (for more details, see Ascher et al. (1995)).

These results clearly indicate that higher order collocation methods with an error controlled adaptive mesh selection, such as proposed e.g. by Ascher et al. (1995) or Binder et al. (2000), should be applied to reliably solve dynamic optimization problems involving control discontinuities. A fine mesh will be required locally whenever steep gradients or discontinuities occur in the control variables. Furthermore, the analysis of the higher order ODE (79),(80) reveals that symbolic transformation of a first order ODE may change its dynamic system behavior to a significant extent, e.g. through the introduction of discontinuities into the coefficients (82) of the (linearized) high order ODE. Note, that the corresponding first order representation of (79),(80), which can be obtained by back-transformation, has eigenvalues that differ significantly from those of the original system (60)–(68). Clearly, this also affects the stiffness of the dynamic system. Hence, the high effort for a stable numerical solution of these ODE systems may often not be compensated by the reduced variable space achieved by symbolic transformation of the original dynamic system. The examples[5] treated in Sections 5.2.1 and 5.2.2 indicate that the numerical performance of the DE method will generally depend on combined properties of the reformulated model and the dynamic optimization problem.

5.2.3 Batch reactor

The final example problem treats the productivity optimization of a simple batch reactor (Logsdon & Biegler, 1992) which is modeled as:

$$\frac{dx_1}{dt} = -(u + \frac{u^2}{2})x_1 \; , \tag{83}$$

$$\frac{dx_2}{dt} = ux_1 \; , \tag{84}$$

[5] Note, that both examples are singular dynamic optimization problems.

where x_1 and x_2 denote the mole fractions of raw material and product, respectively. The (dimensionless) reactor temperature u represents the free input variable which is used to maximize the amount of product within a specified batch time of 1 *hour*. The dynamic optimization problem for the batch reactor is stated as:

$$\begin{aligned} &\min_{u(t)} \ -x_2(t_f) \qquad (85)\\ &\text{s.t.} \quad \textit{Equations}\ (83),(84)\ ,\\ &0.5 \leq u(t) \leq 5.0\ ,\\ &x_1(t_0) = 1.0\ ,\\ &x_2(t_0) = 0.0\ ,\\ &\qquad \forall\, t\ \in\ [0,t_f];\ t_f = 1\, h\ . \end{aligned}$$

The dynamic system (83),(84) is *not* differentially flat. This example points out nicely that nonlinear controllable single input systems do not necessarily have to be differentially flat while, in contrast, the reverse holds. Nevertheless, by introducing an auxiliary variable $y = x_2$ we can convert the first order ODE into a second order ODE. This way, we are at least able to apply the HO approach (cf. problem (23)) for non-flat systems as described in Section 3.3. The second order representation includes input and auxiliary variables and their time derivatives instead of the original state and input variables. The transformed system takes the following form:

$$\ddot{y} + (u + \frac{u^2}{2} - \frac{\dot{u}}{u})\dot{y} = 0\ . \qquad (86)$$

Equation (86) is referred to as the input-output representation of the model (van der Schaft, 1989). For convenience we refer to the auxiliary variables as the system output. Again, the highest order derivatives of both input and output variables are approximated by piecewise polynomials. Equation (86) is then discretized by applying a two-stage Radau-IRK method on 50 time elements. The input variable u is approximated by means of piecewise linear (continuous) polynomials on 25 finite elements according to equation (30). The discretized problem is optimized again using SNOPT (Gill et al., 1998). The results are presented in Figure 7. Almost coinciding solutions were also obtained with the simultaneous strategy using collocation on finite elements and with the sequential strategy as in the previous sections. The same approximation order (two-stage Radau-IRK) as for the HO method was used for the simultaneous method for the state discretization while the controls were approximated by piecewise linear trial functions on 25 equidistant time intervals. An optimality tolerance of 10^{-6} was used. We here focus on a comparison of the HO method with the simultaneous approach. For the sake of completeness, the results obtained with the sequential approach are, however, also included in Table 4. The results in Table 4 show that the HO method is not superior to the simultaneous approach with regard to the total computing time. As in the other examples, the SQP convergence rate of the HO method is substantially smaller than that of the SIM and SEQ approaches, while the computing time required to solve one QP subproblem is still relatively short. Obviously, in this case the overall performance of the SQP algorithm again suffers from search directions $\boldsymbol{p}_k$ that lead to poor convergence properties. At the same time, the reduction in computing time for the QP solutions is not sufficient to outweigh this drawback.

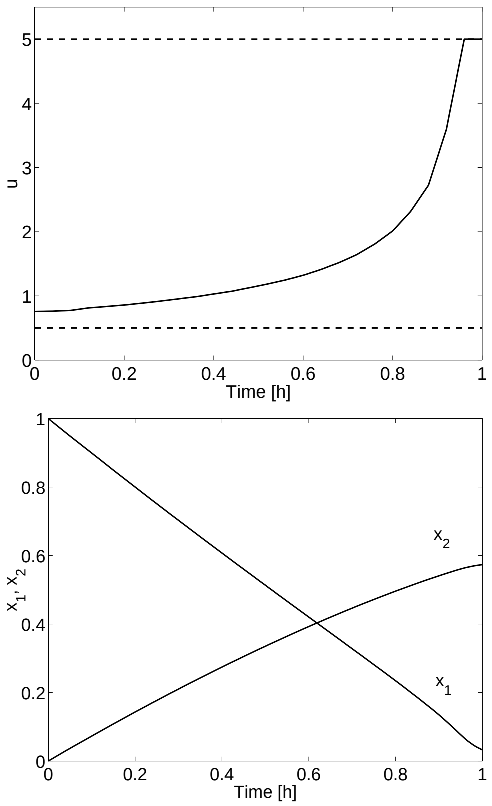

Figure 7: Dimensionless reactor temperature $u(t)$, mole fractions of reactant x_1 and product x_2.

6 Conclusions

In this contribution, three alternative approaches to dynamic optimization based on higher order dynamic model representations are introduced and analyzed rigorously. It is shown, that for differentially flat systems a reformulated optimization problem (cf. problem (10)) can be applied very efficiently. The required solution time can be reduced significantly when compared to sequential methods which is mainly due to the fact that numerical state and sensitivity integration is avoided. Moreover, the new method can also have advantages over the simultaneous approach since the number of optimization variables is reduced to a considerable extent.

We further discuss a natural extension of the optimization approach for the limited class of flat systems to treat non-flat systems. This approach is shown to produce optimal results while its efficiency is not superior to the established solution techniques anymore for the examples presented. The problems occurring in the defect elimination method trace back to the fact that stable numerical solution of higher ODEs incorporating switching control variables is rather difficult. Hence, optimization problems with bang-bang solutions cannot be handled efficiently with numerical algorithms without automatic grid adaptation, which

Table 4: Optimization results – Batch reactor

Method	Objective	NLP par.	SQP ‖ ø time		QP ‖ ø time		$\boldsymbol{time}$ [sec]
HO	– 0.5735	250	106	0.091	359	0.0267	**9.6**
SIM	– 0.5735	350	64	0.139	417	0.0213	**8.9**
SEQ	– 0.5735	26	68	0.162	94	0.117	**11.0**

is a rather strong restriction of the method. Although extrapolating the results obtained for general dynamic optimization problems is difficult, its potential is shown to be strongly dependent on the particular dynamic optimization problem considered.

Finally, a reformulation for a dynamic system into a higher order differential input-output representation is presented, which is intended to reduce the number of optimization variables of simultaneous methods. Again, the efficiency of the method is difficult to rate in a general sense. However, the results obtained in this paper clearly indicate that the DE and HO method require the integration of sophisticated numerical discretization methods with error controlled adaptive mesh selection.

Concluding, experience with a number of example problems indicates that the most promising approach among the methods presented in this paper will be the one for flat dynamic systems. Its main area of application should be the on-line optimization and control of chemical or bio-reactors since their nonlinear process models are rather often differentially flat. Despite the potential of reducing computational time the approach has the favorable property of being always feasible with respect to all dynamic model constraints like the sequential approaches are. This property is especially important for real-time applications since intermediate solutions may be implemented in case of hard time limits even though the optimization algorithm has not yet converged. Hence, a robust and efficient implementation of the FS method for on-line applications would be desirable. The identification of flat output functions and the model reformulation required to treat the illustrative example problems were found to be rather tedious and time-consuming. Future research activities should therefore focus on systematic methods to be able to reliably identify flatness of a dynamic system, to determine the corresponding flat output functions, and to find suitable fictitious inputs to eliminate the system defect.

Acknowledgement

This research is partly supported by the European Commission under grant G1RD-CT-1999-00146 in the INCOOP project. Fruitful discussions with Adel Mhamdi and Arnold Reusken are gratefully acknowledged.

References

Abel, O., Helbig, A., & Marquardt, W. (2000). Productivity optimization of an industrial semi-batch polymerization reactor under safety constraints, *J. Proc. Cont.*, *10*, 351–362.

Abel, O., Helbig, A., Oldenburg, J., & Marquardt, W. (1999). *DYNOPT User Manual, Release 2.4*, Lehrstuhl für Prozesstechnik, RWTH Aachen, Aachen, Germany.

Agrawal, S., & Faiz, N. (1998). Optimization of a class of nonlinear dynamic systems: New efficient method without Lagrange mulitpliers, *Journal of Optimization Theory and Applications*, *97*(1), 11–28.

Allgöwer, F., Badgwell, T., Qin, J., Rawlings, J., & Wright, S. (1999). Nonlinear predictive control and moving horizon estimation, *in* P. Frank (ed.), *Advances in Control*, Springer Verlag, pp. 391–449.

Ascher, U., Mattheij, R., & Russell, R. (1995). *Numerical Solution of Boundary Value Problems for Ordinary Differential Equations*, Classics in Applied Mathematics, SIAM, Philadelphia.

Bausa, J. (2000). *Dynamische Optimierung energie- und verfahrenstechnischer Prozesse*, VDI-Verlag, Düsseldorf. In German.

Biegler, L. (2000). Efficient solution of dynamic optimization and NMPC problems, *in* F. Allgöwer, & A. Zheng (eds), *Nonlinear Model Predictive Control*, Birkhäuser Verlag, Basel, pp. 219–243.

Binder, T., Blank, L., Dahmen, W., & Marquardt, W. (2000). Grid refinement in multiscale dynamic optimization, *Proc. European Symposium on Computer Aided Process Engineering 10, Florence, Italy, Mai 2000, Ed. S. Pierucci*, pp. 31–37.

Bischof, C., Khademi, P., Mauer, A., & Carle, A. (1996). Adifor2.0: Automatic differentiation of Fortran77 programs, *IEEE Computational Science & Engineering*, pp. 18–32.

Deuflhard, P., & Bornemann, F. (1994). *Numerische Mathematik II*, de Gruyter Lehrbuch, Berlin. In German.

Faiz, N., Agrawal, S., & Murray, R. (2000). Trajectory planning of differentially flat systems with dynamics and inequalites, *submitted to AIAA Journal of Guidance, Control*, .

Fliess, M., & Glad, S. (1993). An algebraic approach to linear and nonlinear control, *in* H. Trentelman, & J. Willems (eds), *Essays on Control: Perspectives in the Theory and its Applications*, Vol. 14, Birkhäuser Verlag, Basel, pp. 223–267.

Fliess, M., Levine, J., Martin, P., & Rouchon, P. (1995). Flatness and defect of nonlinear systems - introductory, theory and examples, *Int. J. of Control*, *71*(5), 745–765.

Gill, P., Murray, W., & Saunders, M. (1998). SNOPT: An SQP algorithm for large-scale constrained optimization, *Technical report*, Stanford University, Stanford, USA.

Guay, M. (1999). An algorithm for orbital feedback linearization of single-input control affine systems, *Systems & Control Letters*, *38*, 271–281.

Isidori, A. (1995). *Nonlinear Control Systems: An Introduction*, Springer-Verlag, Berlin.

Kansal, S., Forbes, J., & Guay, M. (2000). Constrained optimization of nonlinear, dynamic chemical processes – a normalized form approach, *Proc. IFAC-Symposium "Advanced Control of Chemical Processes", ADCHEM 2000*, pp. 791–796.

Kumar, A., & Daoutidis, P. (1995). Feedback control of nonlinear differential-algebraic-equation systems, *AIChE J.*, *41*(3), 619–636.

Leineweber, D. (1999). *Efficient reduced SQP methods for the optimization of chemical processes described by large sparse DAE models*, Reihe 3: Nr. 613, VDI-Verlag, Düsseldorf. In German.

Logsdon, J., & Biegler, L. (1992). Decomposition strategies for large-scale dynamic optimization problems, *Chem. Eng. Sci.*, *47*(4), 851–864.

Mahadevan, R., Agrawal, S., & Doyle, F. (2000). A flatness based approach to optimization of fed-batch bioreactors, *Proc. IFAC-Symposium "Advanced Control of Chemical Processes", ADCHEM 2000*, pp. 111–116.

Oldenburg, J., & Marquardt, W. (2000). Dynamic optimization based on higher order differential model representations, *Proc. IFAC-Symposium "Advanced Control of Chemical Processes", ADCHEM 2000*, pp. 809–814.

Rothfuß, R. (1997). *Anwendung der flachheitsbasierten Analyse und Regelung nichtlinearer Mehrgrößensysteme*, Reihe 8: Nr. 664, VDI-Verlag, Düsseldorf. In German.

Rothfuß, R., Rudolph, J., & Zeitz, M. (1997). Flatness: A new approach to control of nonlinear systems, *at-Automatisierungstechnik*, *45*(11), 517–525. In German.

SpeedUp (1995). *SpeedUp User Manual (Release 5.5D)*, Aspen Technology, Inc.

Steinbach, M. (1997). Optimal motion design using inverse dynamics, *Internal Report, Preprint SC 97-12, ZIB, Berlin*, .

Stoessel, F. (1995). Design thermally safe semibatch reactors, *Chem. Eng. Progr.*, pp. 46–53.

Unger, J., Kröner, A., & Marquardt, W. (1995). Structural analysis of differential-algebraic equation systems - theory and applications, *Comput. Chem. Eng.*, *19*(8), 867–882.

van der Schaft, A. (1989). Representing a nonlinear state space system as a set of higher-order differential equations, *Systems & Control Letters*, *12*, 151–161.

Vassiliadis, V., Sargent, R., & Pantelides, C. (1994a). Solution of a class of multistage dynamic optimization problems. 1. Problems without path constraints, *Ind. Eng. Chem. Res.*, *33*(9), 2111–2122.

Vassiliadis, V., Sargent, R., & Pantelides, C. (1994b). Solution of a class of multistage dynamic optimization problems. 2. Problems with path constraints, *Ind. Eng. Chem. Res.*, *33*(9), 2123–2133.