Dynamic Optimization with Complementarity Constraints: Smoothing for Direct Shooting

Adrian Caspari[a], Lukas Lüken[a], Pascal Schäfer[a], Yannic Vaupel[a], Adel Mhamdi[a], Lorenz T. Biegler[d], Alexander Mitsos[b,a,c,*]

[a]Process Systems Engineering (AVT.SVT), RWTH Aachen University, 52074 Aachen, Germany

[b]JARA-CSD, 52056 Aachen, Germany

[c]Energy Systems Engineering (IEK-10), Forschungszentrum Jülich, 52425 Jülich, Germany

[d]Carnegie Mellon University, Department of Chemical Engineering, Pittsburgh, PA 15213, USA

Abstract: We consider optimization of differential-algebraic equations (DAEs) with complementarity constraints (CCs) of algebraic state pairs. Formulating the CCs as smoothed nonlinear complementarity problem (NCP) functions leads to a smooth DAE, allowing for the solution in direct shooting. We provide sufficient conditions for well-posedness. Thus, we can prove that with the smoothing parameter going to zero, the solution of the optimization problem with smoothed DAE converges to the solution of the original optimization problem. Four case studies demonstrate the applicability and performance of our approach: (i) optimal loading of an overflow weir buffer tank, (ii) batch vaporization setpoint tracking, (iii) operation of a tank cascade, and (iv) optimal start-up of a rectification column. The numerical results suggest that the presented approach scales favorably: the computational time for solution of the tank cascade problem scales not worse than quadratically with the number of tanks and does not scale with the control grid.

Keywords: MPCCs with DAE, direct single-shooting, well-posedness analysis, optimization of smoothed DAE

1 Introduction

Discrete events occur in many relevant technical systems that are otherwise governed by continuous dynamics. Examples include disappearance of phases in phase equilibria, controller saturation, relief valves, overflow weirs, or flow inversion. Such systems are typically modeled as hybrid discrete-continuous differential algebraic equation systems (DAEs). Many of these systems can be instead modeled as nonsmooth DAEs, i.e., DAEs with continuous but nondifferentiable functions, cf. [1]. We focus on the optimization of nonsmooth DAEs, which we formulate as mathematical programs with complementarity constraints

*A. Mitsos, AVT Process Systems Engineering, RWTH Aachen University, 52074 Aachen, Germany
E-mail: amitsos@alum.mit.edu

(MPCC). While the simulation of nonsmooth DAEs can be performed using established methods and software [2, 3, 4], the optimization of these systems is more challenging.

The formulation of the optimization problems with a nonsmooth DAE requires methods for nonsmooth DAE integration, sensitivity analysis, and optimization methods [5, 1]. An alternative is the formulation of the optimization problem as a mixed-integer dynamic optimization problem (MIDO). Avraam et al. [6] applied full-discretization, whereas Allgor and Barton [7] applied direct single-shooting for the solution of a MIDO, and Oldenburg et al. [8] formulated a mixed-logic dynamic optimization (MLDO) problem and solved it with direct single-shooting. Although applied to relevant problems, the formulation of state-dependent discrete events as MIDO or MLDO may lead to a very high number of integer or Boolean variables and thereby to large-scale MIDOs or MLDOs which are computationally expensive to solve. Kraemer and Marquardt [9] showed that the local solution of large-scale discrete-continuous optimization problems reformulated and solved as MPCCs performed better than the formulation and local solution as mixed-integer nonlinear programs. Following this observation, we formulate the optimization of systems with discrete events occurring in time-dependent states as MPCC. Therein, complementarity constraints (CCs) are used to model nonsmooth discrete events. Smoothing the CCs finally yields a smooth DAE which we can then optimize using standard integration and optimization methods.

We consider dynamic optimization problems with N_{CC} CCs involving algebraic variable pairs on a finite time horizon $\mathcal{T} = [t_0, t_f]$

$$\min_{\boldsymbol{x},\boldsymbol{y},\boldsymbol{u}} \ \Phi(\boldsymbol{x}(t_f)) \tag{1a}$$

$$\text{s.t.} \quad \boldsymbol{M}\dot{\boldsymbol{x}}(t) = \boldsymbol{f}(\boldsymbol{x}(t), \boldsymbol{y}(t), \boldsymbol{u}(t)),\ \forall t \in \mathcal{T} \tag{1b}$$

$$\boldsymbol{0} = \boldsymbol{g}(\boldsymbol{x}(t), \boldsymbol{y}(t), \boldsymbol{u}(t)), \forall t \in \mathcal{T} \tag{1c}$$

$$\boldsymbol{0} = \boldsymbol{h}(\boldsymbol{x}(t_0), \boldsymbol{y}(t_0)) \tag{1d}$$

$$\boldsymbol{0} \geqslant \boldsymbol{c}(\boldsymbol{x}(t), \boldsymbol{y}(t), \boldsymbol{u}(t)),\ t \in \mathcal{T} \tag{1e}$$

$$y_{i_k}(t) \perp y_{i'_k}(t),\ \forall t \in \mathcal{T}, k \in \{1, ..., N_{CC}\} \tag{1f}$$

$$0 \leqslant y_{i_k}(t), y_{i'_k}(t),\ \forall t \in \mathcal{T}, k \in \{1, ..., N_{CC}\} \tag{1g}$$

with $\mathcal{X} := \mathbb{R}^{N_x} \times \mathbb{R}^{N_y} \times \mathbb{R}^{N_u}$, $t_0 \in \mathbb{R}$ the initial time, $t_f \in \mathbb{R}$ the final time, $\boldsymbol{u} : \mathcal{T} \to \mathbb{R}^{N_u}$ the control variables, $\boldsymbol{x} : \mathcal{T} \to \mathbb{R}^{N_x}$ and $\boldsymbol{y} : \mathcal{T} \to \mathbb{R}^{N_y}$ the differential and algebraic state variables, respectively, $\boldsymbol{f} : \mathcal{X} \to \mathbb{R}^{N_x}$ and $\boldsymbol{g} : \mathcal{X} \to \mathbb{R}^{N_y - N_{CC}}$ define the right-hand-side of a DAE with the constant and non-singular mass matrix $\boldsymbol{M} \in \mathbb{R}^{N_x \times N_x}$, while $\boldsymbol{h} : \mathcal{X} \to \mathbb{R}^{N_x}$ indicates the initial conditions and $\boldsymbol{c} : \mathcal{X} \to \mathbb{R}^{N_c}$ the constraints. y_{i_k} and $y_{i'_k}$ are those pairs of algebraic variables appearing in the CCs (1f)-(1g) with

$i_k, i'_k \in \{1, \ldots, N_y\}, i_k \neq i'_k$. The Mayer-type objective function to be minimized is $\Phi : \mathbb{R}^{N_x} \to \mathbb{R}$. For notational simplicity, we focus in the following on a single CC, i.e. $N_{CC} = 1$, and the corresponding sensitivity system. The theory also applies to multiple CCs and in fact the numerical examples provided have multiple. We use the following definition and assumptions.

Definition 1 *We define $\hat{\boldsymbol{y}} \in \mathbb{R}^{n_y - 2}$ as the vector of algebraic states without those states appearing in the CC, i.e., $\hat{\boldsymbol{y}} = (y_j)_{j \in \{1,\ldots,N_y\} \setminus \{i,i'\}}$.*

Assumption 1 *The functions $\boldsymbol{f}$ and $\boldsymbol{g}$ of the DAE (1b)-(1c) are smooth , i.e., they belong to the class of continuously differentiable functions $\mathcal{C}^1$.*

Assumption 2 $\left.\frac{\partial \boldsymbol{g}}{\partial[\hat{\boldsymbol{y}}, y_i]}\right|_{\boldsymbol{x}(\boldsymbol{u}(t),t),\boldsymbol{y}(\boldsymbol{u}(t),t),\boldsymbol{u}(t)}$ *and* $\left.\frac{\partial \boldsymbol{g}}{\partial[\hat{\boldsymbol{y}}, y_{i'}]}\right|_{\boldsymbol{x}(\boldsymbol{u}(t),t),\boldsymbol{y}(\boldsymbol{u}(t),t),\boldsymbol{u}(t)}$ *have full rank for all $t \in \mathcal{T}$.*

Assumption 2 states that the variables y_i, $y_{i'}$ both participate in the algebraic equations non-trivially. It is necessary and typically also sufficient for the solution of $\boldsymbol{g}$ with respect to either $[\hat{\boldsymbol{y}}, y_i]$ or $[\hat{\boldsymbol{y}}, y_{i'}]$.

Assumption 2 would be violated if the DAE (1b)-(1c) is high-index when $[\hat{\boldsymbol{y}}, y_i]$ or $[\hat{\boldsymbol{y}}, y_{i'}]$ are seen as the algebraic variables. The assumption is not very restrictive, as systems with higher differential indices can be transformed to index 1 systems.

Optimization problems subject to DAEs can be solved, inter alia, by full discretization [10] and direct sequential methods [11, 12]. In full discretization, state and control variable profiles are discretized resulting in a large-scale nonlinear program (NLP) solved with a standard NLP solver. All constraints are exposed to the NLP solver. Consequently, in the case of MPCCs, the NLP solver has to handle the CCs. Handling the CCs (1f)-(1g) is, however, challenging for NLP solvers, since such constraints violate the linear independence constraint qualification (LICQ) and Mangasarian Fromowitz constraint qualification (MFCQ) at all feasible points, cf. [13, 14, 15, 16]. Consequently, the multipliers of MPCCs are nonunique and unbounded or do not even exist [17], the KKT conditions are no longer necessary for a local minimum of (1) [16, 17]. As a consequence, the NLP is inherently ill-posed and the MPCC (1) has to be reformulated to a well-posed NLP, e.g., to a relaxed NLP [14, 18]. Some NLP solvers perform some of these reformulations during their iterations, e.g., by dropping dependent constraints or relaxing constraints in the subproblems at a given iteration, cf. [19, 20, 21, 15, 22]. For other NLP solvers, which cannot solve these problems directly, the MPCC can be reformulated using regularization [16], nonlinear complementary problem (NCP) functions [23, 24, 25, 26, 27, 28], or penalty formulations, cf. [22, 18, 20, 29, 21, 13], thus satisfying the constraint qualifications which allows the solution as NLP.

Several authors have solved MPCCs of the form (1) using a suitable reformulation and full discretization. Raghunathan and Biegler [30] used a modified interior-point method [21] for the optimal operation

of a distillation column. The authors used regularized CCs directly coupled to the barrier parameters of an interior-point algorithm. They later extended the work to dynamic optimization [31]. They used CCs for the appearance and disappearance of vapor-liquid equilibrium (VLE) and a weir overflow relation in a rectification column. They solved the dynamic optimization problem using full discretization and their previously developed method [21] to solve a series of regularized MPCCs. Baumrucker et al. [13] summarized several regularization and penalty formulations for MPCCs and concluded that the use of penalty formulations is beneficial in combination with an active set NLP solver. They further showed that penalty formulations are advantageous over the NCP formulation. In a succeeding work, they used full discretization with direct transcription for the optimization of hybrid dynamic systems with continuous state profiles over time [32].

Using full discretization for the solution of dynamic optimization problems with CCs requires the use of variable step-size discretization to accurately locate the switching points. Baumrucker and Biegler [32, 16] showed that a fixed step-size discretization leads to inaccurate state profiles and therefore to a nonsmooth dependency of the optimal solution on the initial values of the degrees of freedom of the optimization. However, the necessity of variable step-size introduces additional degrees of freedom, many of them without significant influence on the objective, and nonconvexities to the optimization, substantially increasing the computational effort required [32, 16].

These issues motivate the consideration of sequential methods for the solution of (1). Therein, the control variables are parameterized and passed as inputs to a DAE integrator, solving the DAE over the defined time horizon. The DAE integrator provides function values and gradient information to a NLP solver. The CCs are equality (1f) and inequality constraints (1g). The former can be given to the DAE integrator, whereas the latter has to be given to the NLP solver, since DAE integrators cannot directly handle it. As an alternative, CCs (1f)-(1g) can be equivalently formulated using NCP functions, which can be fully handled by the DAE integrator. The NCP functions, e.g., [28], are equality path constraints that are equivalent to the CCs (1f)-(1g). They may be exposed to the integrator and solved directly together with the DAE or can be given to the NLP solver as equality constraints [33]. As in full discretization, the latter requires the exact location of the switching points of the CCs by an adequate discretization. This favors the direct treatment of the NCP functions by the integrator. The use of variable step-size integration methods enables an efficient solution of the DAE together with the CCs. Thus, iterative optimization with changing step-size or with additional degrees of freedom and nonconvexities due to variable step-size, cf. [32], is not required. This makes the use of NCP functions promising for the solution of (1) in direct sequential methods.

In contrast to full discretization, the application of direct sequential methods for the solution of dynamic optimization problems with CCs including state variables has not yet been considered and will thus be investigated in this work. We study the use of direct single-shooting [11] allowing for the efficient solution of problems with time-variant algebraic states. Existing works focused on optimization problems with discrete time-invariant control variables. Stein et al. [27] studied different smooth reformulations for the solution of hybrid optimization problems and used direct single-shooting to optimize the feed stage and operation of a rectification column. However, Guo and Allison [34] considered CCs where the participating variables are time-invariant parameters. In that sense, this is simlar to [27]. Single-shooting is a standard approach for the solution of dynamic optimization problems, see e.g., [11, 16].

NCP functions lead to nonsmooth DAEs, which require special treatment for DAE integration and sensitivity analysis [1, 35]. Thus, we smoothen the NCP functions to allow for the use of standard NLP solvers and integrators. We provide a condition for (1b) and (1c), i.e., the smooth part of the DAE, for which the nonsmooth DAE and the smoothed DAE are both well-posed. Note that it is difficult in general to show well-posedness of a nonsmooth DAE and also of a smoothed DAE [1]. We thus analyze the well-posedness for an illustrative example. The analysis of the nonsmooth DAE is related to the work of Pang et al. [36, 37] on differential variational inequalities (DVIs). They showed that DVIs are conceptually equivalent to the problem class we consider in this work. While they studied the problem class through the analysis of DVIs, they mentioned that the analysis of DVIs as nonsmooth DAEs could presumably benefit from the theory existing for nonsmooth DAEs. In contrast, we study the problem class by using the theory of nonsmooth DAEs.

The remainder of the work is structured as follows. Section 2 starts by describing the solution approach we use to solve (1) using direct single-shooting and presents conditions for the well-posedness of the nonsmooth DAE with the Fischer-Burmeister equation, based on the structure of the DAE without a NCP function. Subsection 2.1.2 provides conditions for a smoothed DAE to be well-posed. We then propose a heuristic method to bypass suboptimal stationary points by adjusting the smoothing factor of the NCP function in Subsection 2.1.3. Finalizing Section 2, we discuss alternative approaches for the solution of (1). We present four case studies to illustrate the proposed approach, its performance, and the application to a large-scale process system in Section 3, and give conclusions in Section 4.

2 Solution Strategy using Direct Shooting

We solve the dynamic optimization problem (1) using direct single-shooting [11]; i.e., we transform (1) and then solve it as a NLP. The strategy is however not resticted to single-shooting and can also be

applied for other direct shooting approaches, e.g., direct multiple-shooting [12]. If the MPCC solution is strongly stationary, then an MPCC can be related to stationarity of an equivalent NLP and is, thus, the key assumption enabling the solution of the MPCC through NLP reformulations [22, 29, 14, 16, 17]. CCs (1f) - (1g) can be treated either by (I) the DAE integrator only, (II) the NLP solver, (III) or combination of both the integrator and the NLP solver. The following subsection describes the solution approach implemented (alternative I). Afterwards, we summarize alternative treatments (alternatives II and III) of the CCs for the solution of (1) using direct shooting.

2.1 DAEs with Nonlinear Complementarity Problem Functions

An NCP function, e.g., [23, 24, 25, 26, 27, 28, 38], is equivalent to the CCs (1f)-(1g) and can fully be treated by the integrator. In this approach, (1f)-(1g) are substituted by an equality using a NCP function. With full discretization, the use of NCP functions for CCs is disadvantageous [13]. On the other hand, with direct shooting, using an NCP function solved by the integrator benefits due to adjustable time steps in the integrator, in contrast to full discretization. Hence, we use NCP functions, which are solved by the DAE integrator, as substitute for (1f)-(1g). We use this alternative for the solution of (1) with direct single-shooting.

2.1.1 Analysis of well-posedness of DAEs with non-smooth NCP functions

NCP functions are nonsmooth and their use results in a nonsmooth DAE. In the following, we consider the Fischer-Burmeister NCP function [28] and show under which condition the resulting nonsmooth DAE including the Fischer-Burmeister NCP function is well-posed. A problem is well-posed, if existence, uniqueness, continuation and continuous/Lipschitz parametric dependence of the solution are satisfied, cf. [1]. Several works focus on the analysis and solution of nonsmooth DAEs, e.g., [39, 40, 41, 42]. We follow the line of argumentation by Stechlinski et al. [1] and restrict our analysis to a DAE with one CC. However, the analysis applies also for multiple CCs. The well-posedness analysis of the nonsmooth DAE uses Clarke's Generalized Jacobian [43]. Section 2.1.2 presents the well-posedness analysis of the corresponding smoothed DAE.

The Fischer-Burmeister function [28] is defined by

$$\varphi^{\mathrm{fb}}(y_i(t), y_{i'}(t)) := y_i(t) + y_{i'}(t) - \sqrt{(y_i(t))^2 + (y_{i'}(t))^2}. \tag{2}$$

A subset of the Clarke Generalized Jacobian of (2) has been derived by Fischer [28], who used (2) for

a Newton-type optimization method to replace the complementarity conditions of the necessary optimization conditions of an NLP. Using (2) to replace (1f)-(1g) results in the following nonsmooth DAE as equivalent to (1b)-(1c),(1f)-(1g):

$$\boldsymbol{M}\dot{\boldsymbol{x}}(t) = \boldsymbol{f}(\boldsymbol{x}(t), \boldsymbol{y}(t), \boldsymbol{u}(t)),\ \forall t \in \mathcal{T} \tag{3a}$$

$$\boldsymbol{0} = \boldsymbol{g}(\boldsymbol{x}(t), \boldsymbol{y}(t), \boldsymbol{u}(t)), \forall t \in \mathcal{T} \tag{3b}$$

$$0 = \varphi^{\text{fb}}(y_i(t), y_{i'}(t)),\ \forall t \in \mathcal{T} \tag{3c}$$

To establish non-singularity of the Jacobian we utilize a technical condition which in essence states that the derivatives of the algebraic constraints (3b) are linearly independent from the derivatives of the complementary constraint (3c) formulated using the Fischer-Burmeister function. This has to hold for all points for which the derivatives exist and also in the limit towards the singular point [0,0].

Assumption 3 *(i) For* $[y_i, y_{i'}] \neq [0,0]$*, the rows of the Jacobian* $\frac{\partial \boldsymbol{g}}{\partial \boldsymbol{y}}(\boldsymbol{x}(\boldsymbol{u}(t),t), \boldsymbol{y}(\boldsymbol{u}(t),t), \boldsymbol{u}(t))$ *are linearly independent of the derivative of the Fischer-Burmeister function* (2) $\frac{\partial \varphi^{fb}}{\partial \boldsymbol{y}}(\boldsymbol{y}(\boldsymbol{u}(t),t))$ *for all* $t \in \mathcal{T}$*. (ii) The rows of the Jacobian* $\frac{\partial \boldsymbol{g}}{\partial \boldsymbol{y}}(\boldsymbol{x}(\boldsymbol{u}(t),t), \boldsymbol{y}(\boldsymbol{u}(t),t), \boldsymbol{u}(t))$ *for* $[y_i, y_{i'}] = [0,0]$ *are linearly independent of* $\boldsymbol{v} : v_k = 0,\ \forall k \in \{1,...,n_y\}\backslash\{i,i'\}, v_i = 1 - r\cos(\phi), v_{i'} = 1 - r\sin(\phi)$ *with* $0 \leqslant \phi < 2\pi, 0 < r \leqslant 1$ *for all* $t \in \mathcal{T}$*.*

Assumption 3 is not easy to validate in general. For some systems, it can be shown to be satisfied based on the algebraic equations $\boldsymbol{g}$, either analytically or numerically. For other systems it can be shown a-posteriori along the solution trajectory in order to show the regularity of the solution. A guarantee that Assumption 3 is satisfied could be introduced by characterizing and finding all regions in the state space where Assumptions 3 is not satisfied and preventing the DAE states to lie inside these regions. We first state the following lemma and use it in the subsequent theorem. It shows the non-singularity of the Clarke Generalized Jacobian [43], which is necessary for the well-posedness of the nonsmooth DAE.

Lemma 1 *Consider the nonsmooth DAE* (3)*. Under the Assumptions 1-3, Clarke's Generalized Derivative of the algebraic equations* $[\boldsymbol{g}, \varphi]$ *with respect to* $\boldsymbol{y}$ *at* $\boldsymbol{x}(\boldsymbol{u}(t),t), \boldsymbol{y}(\boldsymbol{u}(t),t), \boldsymbol{u}(t)$ *is non-singular.*

Proof The Fischer-Burmeister function (2) is piecewise-continuously differentiable ($\mathbb{PC}^1$) as it is continuously differentiable everywhere except at the singleton $\{[0,0]\}$. $\boldsymbol{g}$ is smooth by Assumption 1. Thus, we have to show that the Clarke Generalized Jacobian is non-singular at the non-differentiable point [0,0].

The Fischer-Burmeister function (2) is Lipschitz continuous [28]. Local Lipschitz continuity is required for the definition of the Clarke Generalized Jacobian [43]. The Jacobian of (3c) with respect to y_i and

$y_{i'}$ for $[y_i, y_{i'}] \neq [0,0]$ is

$$\left.\frac{\partial \varphi^{\text{fb}}}{\partial y_i, y_{i'}}\right|_{y_i, y_{i'}} = \left[1 - \frac{y_i}{\sqrt{(y_i)^2 + (y_{i'})^2}}, 1 - \frac{y_{i'}}{\sqrt{(y_i)^2 + (y_{i'})^2}}\right]. \tag{4}$$

By using polar coordinates $y_i = r\cos(\phi)$ and $y_{i'} = r\sin(\phi)$, $r > 0$, i.e., $[y_i, y_{i'}] \neq [0,0]$, the derivative can be written in the following form

$$\left.\frac{\partial \varphi^{\text{fb}}}{\partial y_i, y_{i'}}\right|_{y_i = r\cos(\phi), y_{i'} = r\sin(\phi)} = [1 - \cos(\phi), 1 - \sin(\phi)]$$

Note that for $0 \leqslant \phi < 2\pi$ this defines the circle with center point [1,1] and radius 1. The Clarke Generalized Jacobian $\partial\varphi^{\text{fb}}$ is the convex hull of the limiting Jacobians at the non-differentiable point [43] and can thus be written for (3c) as

$$\partial\varphi^{\text{fb}}(0,0) = \operatorname{conv}\left(\left\{\lim_{[y_i, y_{i'}] \to [0,0]} \left(\left.\frac{\partial \varphi^{\text{fb}}}{\partial y_i, y_{i'}}\right|_{y_i, y_{i'}}\right) : [y_i, y_{i'}] \in \mathbb{R}^2 \backslash \{[0,0]\}\right\}\right).$$

Using polar coordinates, $\lim_{[y_i, y_{i'}] \to [0,0]}$, $[y_i, y_{i'}] \neq [0,0]$ is equivalent to $\lim_{r \to 0}$, $r > 0$. We thus have

$$\partial\varphi^{\text{fb}}(0,0) = \operatorname{conv}\left(\left\{\lim_{r \to 0} \left([1 - \cos(\phi), 1 - \sin(\phi)]\right) : 0 \leqslant \phi < 2\pi, r > 0\right\}\right),$$

which converges only if ϕ converges. Note that the Clarke Generalized Jacobian is independent of the radius r. It only depends on the angle ϕ duo to convergence in polar coordinates. Thus, the Clarke Generalized Jacobian of (3c) at the non-differentiable point [0,0] is given by

$$\partial\varphi^{\text{fb}}(0,0) = \left\{\left[1 - \rho\cos(\phi), 1 - \rho\sin(\phi)\right] : 0 \leqslant \phi \leqslant 2\pi, 0 \leqslant \rho \leqslant 1\right\},$$

i.e., the disk with center point [1,1] and radius 1. Note that $[0,0] \notin \partial\varphi^{\text{fb}}(0,0)$. The Clarke Generalized Jacobian of $[\boldsymbol{g}, \varphi]$ with respect to the algebraic variables at $[y_i, y_{i'}] = [0,0]$ then reads

$$\begin{aligned}&\partial[\boldsymbol{g}, \varphi^{\text{fb}}]^T(\boldsymbol{x}(\boldsymbol{u}(t), t), \boldsymbol{y}(\boldsymbol{u}(t), t), \boldsymbol{u}(t)) = \\ &\left\{\begin{bmatrix} \frac{\partial \boldsymbol{g}}{\partial \hat{\boldsymbol{y}}}, & \frac{\partial \boldsymbol{g}}{\partial y_i}, & \frac{\partial \boldsymbol{g}}{\partial y_{i'}} \\ \boldsymbol{0}, & 1 - \rho \cdot \cos(\phi), & 1 - \rho \cdot \sin(\phi) \end{bmatrix} : 0 \leqslant \phi < 2\pi, 0 \leqslant \rho \leqslant 1\right\}, \text{if } [y_i, y_{i'}] = [0,0]\end{aligned} \tag{5}$$

The partial derivatives in (5) are evaluated at $\boldsymbol{x}(\boldsymbol{u}(t), t), \boldsymbol{y}(\boldsymbol{u}(t), t), \boldsymbol{u}(t)$. By Assumption 2, $\frac{\partial \boldsymbol{g}}{\partial \hat{\boldsymbol{y}}, y_i}$ and $\frac{\partial \boldsymbol{g}}{\partial \hat{\boldsymbol{y}}, y_{i'}}$ have full rank. Note that $\left[\frac{\partial \boldsymbol{g}}{\partial \hat{\boldsymbol{y}}}, \frac{\partial \boldsymbol{g}}{\partial y_i}, \frac{\partial \boldsymbol{g}}{\partial y_{i'}}\right] = \frac{\partial \boldsymbol{g}}{\partial \boldsymbol{y}}$. Due to Assumption 3, the last row of the matrices

in (5) cannot be expressed as a linear combination of the rows of $\frac{\partial \boldsymbol{g}}{\partial \boldsymbol{y}}$. Further $\frac{\partial [\boldsymbol{g},\boldsymbol{\varphi}]}{\partial \boldsymbol{y}}$ is non-singular for $[y_i, y_{i'}] \neq 0$ due to Assumption 3. Consequently, the matrix in (5) is non-singular for all values of $0 \leqslant \phi < 2\pi, 0 \leqslant \rho \leqslant 1$ for all $t \in \mathcal{T}$. □

We state the following theorem about the well-posedness of the nonsmooth DAE (3).

Theorem 1 *Under the Assumptions 1-3, the nonsmooth DAE (3) is well-posed.*

Proof The nonsmooth DAE (3) is well-posed if the Fischer-Burmeister function (3c) is $\mathbb{PC}^1$ in the control variables and state variables, and the solution is regular, i.e., the DAE (3) has a generalized differential index of 1 [1]. Due to Assumption 1 and the nature of (3c), all functions of (3) are $\mathbb{PC}^1$. (3) has generalized differential index of 1, if the projection of the Clarke Jacobian of the algebraic equations of (3) is non-singular, [1]. Non-singularity of the Clarke Jacobian of the algebraic equations is satisfied by the Assumptions 1-3 due to Lemma 1. □

Hence, we showed that the nonsmooth DAE is well-posed. Although this holds for the Fischer-Burmeister function only, it establishes for all NCP functions, due to the equivalence of the NCP functions and the CC (1f) and (1g). E.g., we provide the analysis of a nonsmooth DAE with the max NCP function of [38] in the supplementary material, which is the first NCP function formulation for the solution of NLPs with nonlinear complementarity constraints. We also illustrate the well-posedness for an example in the supplementary material.

2.1.2 Well-Posedness of DAE with smoothed NCP Function

We now consider a smoothed NCP function, which leads to a smooth DAE and, thus, allows the application of standard integrators with smooth sensitivity analysis for the optimization. In contrast, the optimization with a nonsmooth NCP function requires a special sensitivity analysis within direct shooting, as done in [35]. We use the smoothed Fischer-Burmeister function

$$\varphi^{\text{sfb}}(y_i(t), y_{i'}(t)) := y_i(t) + y_{i'}(t) - \sqrt{(y_i(t))^2 + (y_{i'}(t))^2 + \varepsilon} \tag{6}$$

with $\varepsilon > 0$. The resulting equation is equivalent to the regularized CC

$$y_i(t) \cdot y_{i'}(t) = \varepsilon/2, \quad y_i(t) \geqslant 0, \quad y_{i'}(t) \geqslant 0.$$

Using the smoothed Fischer-Burmeister function (6) results in following smoothed DAE

$$\boldsymbol{M}\dot{\boldsymbol{x}}(t) = \boldsymbol{f}(\boldsymbol{x}(t), \boldsymbol{y}(t), \boldsymbol{u}(t)),\ \forall t \in \mathcal{T} \tag{7a}$$

$$\boldsymbol{0} = \boldsymbol{g}(\boldsymbol{x}(t), \boldsymbol{y}(t), \boldsymbol{u}(t)), \forall t \in \mathcal{T} \tag{7b}$$

$$0 = \varphi^{\text{sfb}}(y_i(t), y_{i'}(t)),\ \forall t \in \mathcal{T} \tag{7c}$$

with appropriate initial equations. Using (6) offers slower convergence to the solution of optimization problem for decreasing ε as compared to other regularization formulations [18]. However, other regularization formulations would require the use of an appropriate control variable discretization to locate the NCP switching points, since they can not directly be solved by the DAE integrator. A comparison of different formulations in the context of sequential dynamic optimization is left for future work. We use the following assumption, which is the counterpart to Assumption 3 for the smoothed DAE (7).

Assumption 4 *The rows of the Jacobian* $\frac{\partial \boldsymbol{g}}{\partial \boldsymbol{y}}(\boldsymbol{x}(\boldsymbol{u}(t),t), \boldsymbol{y}(\boldsymbol{u}(t),t), \boldsymbol{u}(t))$ *are linearly independent of the derivative of the smoothed Fischer-Burmeister function* (6) $\frac{\partial \varphi^{sfb}}{\partial \boldsymbol{y}}(\boldsymbol{y}(\boldsymbol{u}(t),t))$ *for all* $t \in \mathcal{T}$.

The following theorem states the well-posedness of the smoothed DAE (7) based on the results in Subsection 2.1.1.

Theorem 2 *Consider the DAE* (7). *If Assumptions 2-4 hold, the smoothed DAE* (7) *is well-posed.*

Proof (6) is differentiable for all $y_i, y_{i'}$. The Jacobian of the algebraic equations in (7) is

$$\partial[\boldsymbol{g}, \varphi^{\text{sfb}}]^T(\boldsymbol{x}(\boldsymbol{u}(t),t), \boldsymbol{y}(\boldsymbol{u}(t),t), \boldsymbol{u}(t)) = \begin{bmatrix} \frac{\partial \boldsymbol{g}}{\partial \hat{\boldsymbol{y}}}, & \frac{\partial \boldsymbol{g}}{\partial y_i}, & \frac{\partial \boldsymbol{g}}{\partial y_{i'}} \\ \boldsymbol{0}, & 1 - \frac{y_i(t)}{\sqrt{(y_i(t))^2 + (y_{i'}(t))^2 + \varepsilon}}, & 1 - \frac{y_{i'}(t)}{\sqrt{(y_i(t))^2 + (y_{i'}(t))^2 + \varepsilon}} \end{bmatrix} \tag{8}$$

The partial derivatives in (8) are evaluated at $\boldsymbol{x}(\boldsymbol{u}(t),t), \boldsymbol{y}(\boldsymbol{u}(t),t), \boldsymbol{u}(t)$. The Jacobian (8) is non-singular due to the Assumptions and the smoothed DAE (7) is well-posed. □

Thus, Assumptions 1-4 are sufficient for both the well-posedness of the nonsmooth DAE and the smoothed DAE. Due to the well-posedness of the nonsmooth DAE and the smoothed DAE, the solution of the latter converges to the solution of the first. Ralph and Wright [18] showed that the solution of MPCCs with the smoothed Fischer-Burmeister function substituting the complementarity constraints admits a convergence rate to the solution of the MPCCs of $\mathcal{O}(\varepsilon^{1/4})$. A detailed convergence analysis of

our approach is out of the scope of this work. It would require the convergence analysis for the solution of the smoothed DAE and the nonsmooth DAE.

2.1.3 Avoiding Suboptimal Stationary Solutions

Although the resulting smooth DAE with the smoothed NCP function can be solved using standard integrators and optimizers for smooth systems in direct single-shooting, the parameter ε in (6) has to be chosen carefully. An insensible choice for the parameter ε may lead to suboptimal stationary points. A too small value for the parameter may impede the NLP solver to move away from the initial guess provided by the user. However, a sufficiently small value for ε is required for the solution to converge to the solution of the nonsmooth problem, cf. [18]. Indeed, most NLP solvers converge to C-stationary solutions, which are suboptimal local solutions since there may be a descent direction [17]. We give an illustrative example in Section 3.2, where the NLP solver would converge in the first iteration to a suboptimal local solution, since the sensitivity of the objective with respect to the degrees of freedom of the NLP is small enough that the necessary optimality conditions are satisfied up to a given tolerance. To overcome this issue, we propose a heuristic approach that increases the smoothing parameter ε in (6). By also increasing the sensitivity with respect to the decision variables, convergence to the initial stationary point, whether a suboptimal local solution or a C-stationary point, is avoided. We illustrate this approach with an example in the supplementary material.

The presence of CCs implies nonconvexities in the optimization problem and typically the presence of suboptimal local minima to which NLP solvers may converge. To overcome poor stationary solutions, Kraemer and Marquardt [9] proposed an a posteriori reassignment of discrete decision variables. Therein, they enforce a different switching branch by reassigning binary variables, which we do by manipulating the smoothing parameter ε. In the presence of several discrete decision variables, as in problems of the form (1), the reassignment of a single discrete variable may not significantly affect the solution or a long post-processing procedure would be required implying long computational times. In addition, the post processing as proposed by [9] requires reassigning the variables appearing in the CCs, which would require reinitialization of the DAE such that the solution is consistent with the inverted variables of the CCs. In the numerical case studies in Section 3, we show that the adjustment of the smoothing parameter of a smoothed NCP function already helps to overcome poor local solutions. By adjusting the smoothing parameter, the sensitivity of the NCP function variables can be modified without the need of solving many post-processing NLPs. Similar to the a posteriori procedure of [9], a sequence of optimization problems has to be solved consequently. The optimization problem is repeatedly solved with decreasing smoothing

parameter starting with a large value for the parameter ε. Thus, we iteratively reduce the parameter ε until it reaches the tolerance of the DAE integrator. Alternatively, the adjustment of the parameter ε could be directly coupled to the NLP solver, as for instance in [21]. A more rigorous approach would require to evaluate the convergence of the solution of the regularized DAE to the solution of the nonsmooth DAE with decreasing ϵ and stop the decrease when the convergence tolerance is below the integration tolerance. This would require, e.g., the generalized implicit function theorem as in [41].

2.2 Solution Alternatives

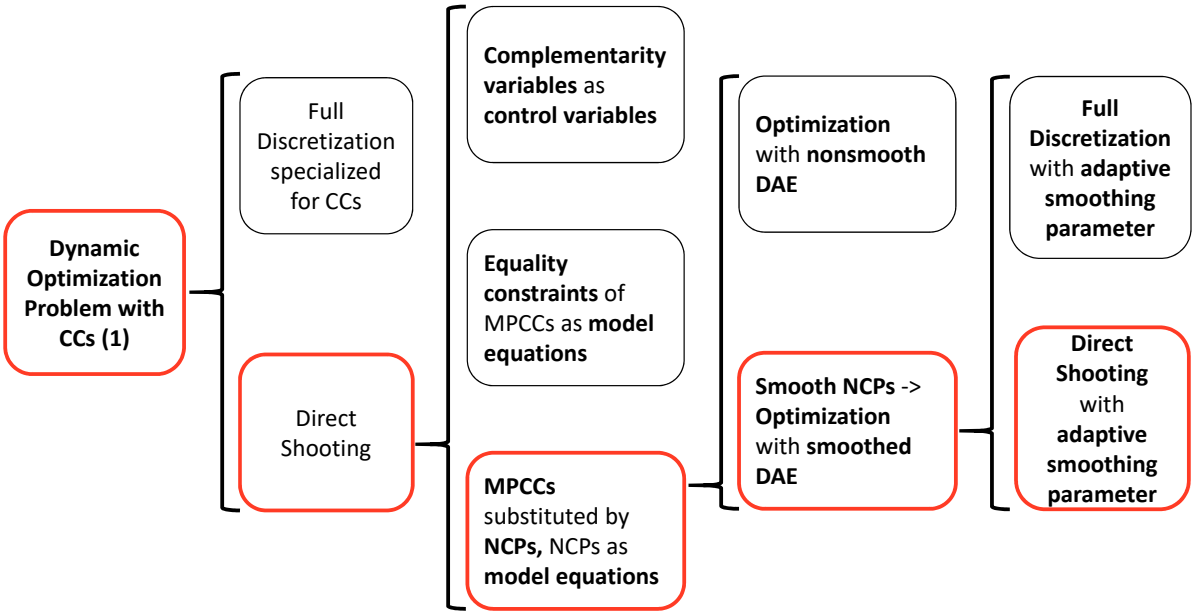


Fig. 1: Illustration of possible solution approaches. Selected approaches highlighted with bold, red frames.

We now discuss the solution alternatives II and III focusing on their respective advantages and disadvantages.

2.2.1 Alternative II: Complementarity Variables as Control Variables

When the CCs (1f) - (1g) are treated by the NLP solver, either $y_i(t)$ or $y_{i'}(t)$ would become a (additional) control variable; otherwise the solution of the DAE would not be unique due to the missing equation. (1f)-(1g) thus have to be satisfied by the NLP solver, resulting in several disadvantages. First, the active time points of the inequality path constraints (1g) have to be located exactly, cf. [16]. In addition, the

CCs at the NLP level violates constraint qualifications which in turn poses difficulties to the NLP solver, as described in the introduction.

2.2.2 Alternative III: Complementarity Equality Constraints as Part of the DAEs

The equality constraint (1f) can directly be solved by the integrator as an algebraic equation while the NLP solver is then given the constraints (1g). Although the equality constraint (1f) is satisfied by the DAE integrator, the active time points of the inequality path constraints (1g) have to be located exactly, as described in Subsection 2.2.1. Otherwise, the solution may be inaccurate [32, 16]. Although the location of the active points could be achieved by suitable adaptation strategies, e.g., [44, 45, 46], the discretization of algebraic states would likely lead to a very fine grid and additionally, in case of many CCs, to a large set of inputs to the DAE, so that the solution of the optimization problem becomes computationally more expensive. In addition, the constraints (1f)-(1g) lead either to the violation of the MFCQ and LICQ for any feasible point at the NLP level or an ill-posed sensitivity system at the integrator level.

Let $\boldsymbol{u}$ be parameterized by $\boldsymbol{p}$. The resulting sensitivities are $\boldsymbol{s}^x = \frac{\partial \boldsymbol{x}}{\partial \boldsymbol{p}}$, $\boldsymbol{s}^{\hat{y}} = \frac{\partial \hat{\boldsymbol{y}}}{\partial \boldsymbol{p}}$, $\boldsymbol{s}^{y_i} = \frac{\partial y_i}{\partial \boldsymbol{p}}$, $\boldsymbol{s}^{y_{i'}} = \frac{\partial y_{i'}}{\partial \boldsymbol{p}}$, and $\boldsymbol{s}^u = \frac{\partial \boldsymbol{u}}{\partial \boldsymbol{p}}$. The sensitivity system of (1b)-(1c) and the complementarity equality constraint (1f) is given by the following linear DAE, cf. e.g. [47]

$$\begin{aligned}
\boldsymbol{M}\dot{\boldsymbol{s}}^x(t) = {} & \left.\frac{\partial \boldsymbol{f}}{\partial \boldsymbol{x}}\right|_{\boldsymbol{x}(t),\boldsymbol{y}(t),\boldsymbol{u}(t)} \boldsymbol{s}^x(t) + \left.\frac{\partial \boldsymbol{f}}{\partial \hat{\boldsymbol{y}}}\right|_{\boldsymbol{x}(t),\boldsymbol{y}(t),\boldsymbol{u}(t)} \boldsymbol{s}^{\hat{y}}(t) + \left.\frac{\partial \boldsymbol{f}}{\partial y_i}\right|_{\boldsymbol{x}(t),\boldsymbol{y}(t),\boldsymbol{u}(t)} \boldsymbol{s}^{y_i}(t) + \\
& \left.\frac{\partial \boldsymbol{f}}{\partial y_{i'}}\right|_{\boldsymbol{x}(t),\boldsymbol{y}(t),\boldsymbol{u}(t)} \boldsymbol{s}^{y_{i'}}(t) + \left.\frac{\partial \boldsymbol{f}}{\partial \boldsymbol{u}}\right|_{\boldsymbol{x}(t),\boldsymbol{y}(t),\boldsymbol{u}(t)} \boldsymbol{s}^u(t),\ \forall t \in \mathcal{T} \qquad (9a) \\
\boldsymbol{0} = {} & \left.\frac{\partial \boldsymbol{g}}{\partial \boldsymbol{x}}\right|_{\boldsymbol{x}(t),\boldsymbol{y}(t),\boldsymbol{u}(t)} \boldsymbol{s}^x(t) + \left.\frac{\partial \boldsymbol{g}}{\partial \hat{\boldsymbol{y}}}\right|_{\boldsymbol{x}(t),\boldsymbol{y}(t),\boldsymbol{u}(t)} \boldsymbol{s}^{\hat{y}}(t) + \left.\frac{\partial \boldsymbol{g}}{\partial y_i}\right|_{\boldsymbol{x}(t),\boldsymbol{y}(t),\boldsymbol{u}(t)} \boldsymbol{s}^{y_i}(t) \qquad (9b) \\
& + \left.\frac{\partial \boldsymbol{g}}{\partial y_{i'}}\right|_{\boldsymbol{x}(t),\boldsymbol{y}(t),\boldsymbol{u}(t)} \boldsymbol{s}^{y_{i'}}(t) + \left.\frac{\partial \boldsymbol{g}}{\partial \boldsymbol{u}}\right|_{\boldsymbol{x}(t),\boldsymbol{y}(t),\boldsymbol{u}(t)} \boldsymbol{s}^u(t),\ \forall t \in \mathcal{T} \\
0 = {} & y_i(t)s^{y_{i'}}(t) + y_{i'}(t)s^{y_i}(t),\ \forall t \in \mathcal{T}, \qquad (9c)
\end{aligned}$$

with appropriate initial equations.

There are two scenarios possible: If $y_i = y_{i'} = 0$, then (9) does not define all sensitivities. Rather, (9c) is trivially satisfied and just one variable would be defined by the algebraic equations (9b). Consequently, (9) is not well-posed, i.e., there is no unique solution of (9). The ill-posedness of the sensitivity systems, in turn, causes non-uniqueness of the Lagrange multipliers at the NLP level, which is equivalent to violation of the LICQ.

On the other hand, if $y_i = 0 \wedge y_{i'} \neq 0$, then by (9c) $s^{y_i} = 0$. Thus, $s^{y_{i'}}$ can be calculated by (9b) and

(9) is well-posed, given Assumption 2. However, LICQ and MFCQ are violated since the gradient of the active inequality constraint ($y_i = 0$) with respect to the degrees of freedom of the NLP is zero ($s^{y_i} = 0$). The case $y_i \neq 0 \wedge y_{i'} = 0$ is analogous.

Consequently, alternative III is problematic.

Note that we could also use a NCP function to be solved by the NLP solver in the alternatives II and III. However, this would not overcome the issues discussed earlier. Further performed preliminary testing with alternatives II and III revealed substantial numerical difficulties. Hence, we use NCP functions, which are solved by the DAE integrator, as substitute for (1f)-(1g) (alternative I). We use this alternative for the solution of (1) with direct single-shooting. Fig. 1 illustrates the possible solution approaches for the solution of (1) and the approach we use. A detailed comparison of the solution alternatives I - III would be interesting, however, is out of the scope of this work.

3 Illustrative Examples

We investigate four numerical case-studies. All cases use the smoothed Fischer-Burmeister function (6), treated directly in the integrator by using direct single-shooting. The optimization problems are solved in the optimization framework DyOS [48] using direct single-shooting with wavelet adaptation [44]. The initial grid for the control variable profiles consists of a single interval. We use the NLP solver SNOPT [49] and the DAE integrator NIXE [50]. The models are implemented in Modelica and exported into DyOS as Functional Mockup Unit. The DAE integration tolerances are 10^{-8} and NLP feasibility and optimality tolerances are 10^{-6}. All computations are performed on a Microsoft Windows 7 desktop computer with an Intel(R) Core(TM) i3-6100 CPU running at 3.70 GHz and 8 GB RAM.

3.1 Overflow Weir Buffer Tank

Our first case-study addresses the optimal loading of an overflow weir buffer. The buffer tank has one feed and one outlet stream, which becomes active when the tank holdup exceeds a threshold (Fig. 2). A model of the buffer tank is provided in the supplementary material. There, we show that Assumptions 2-3 are satisfied and the nonsmooth DAE of the buffer tank is well-posed. Indeed, the determinant of the projection of the Clarke Jacobian of the algebraic equations with respect to the algebraic variables (provided in the supplementary material) is always nonzero, which further proves that it is nonsingular. Thus, the nonsmooth and smoothed versions of the buffer tank example are well-posed.

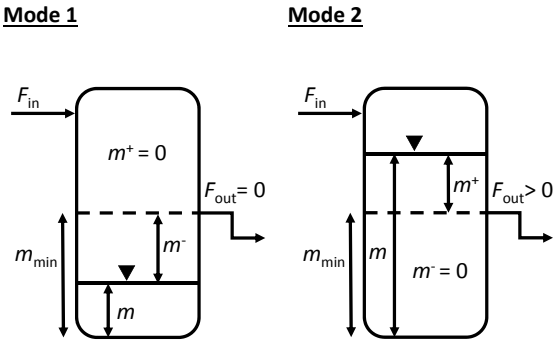
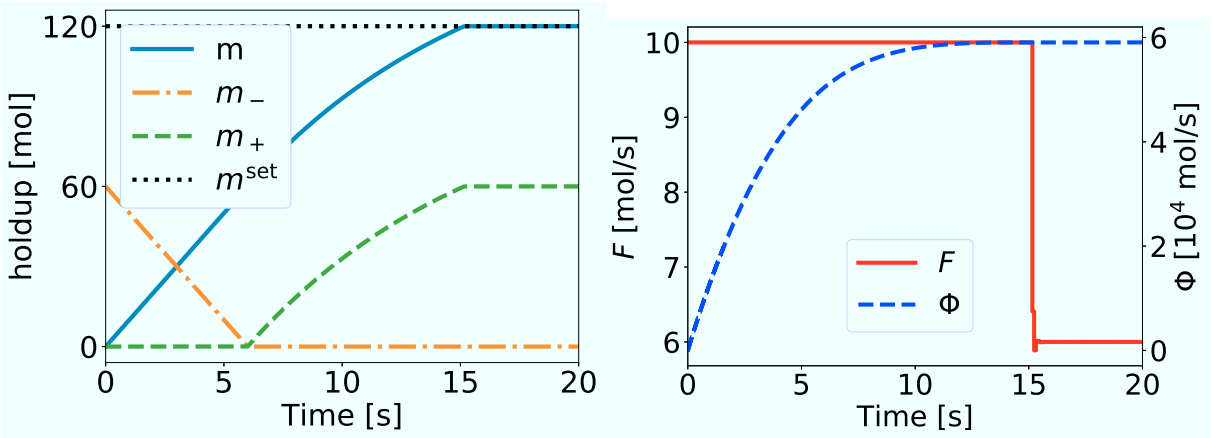


Fig. 2: Buffer tank with overflow weir. Mode 1, holdup less than minimum holdup. Mode 2, holdup higher than minimum holdup. The outlet stream F_{out} becomes active when the tank holdup m exceeds the minimum holdup m_{min}. m_- and m_+ denote the negative and positive deviations of m from m_{min}.

3.1.1 Dynamic Optimization



(a) (b)

Fig. 3: Optimization results of overflow weir buffer tank problem. (a) tank holdup. (b) feed flowrate and objective function.

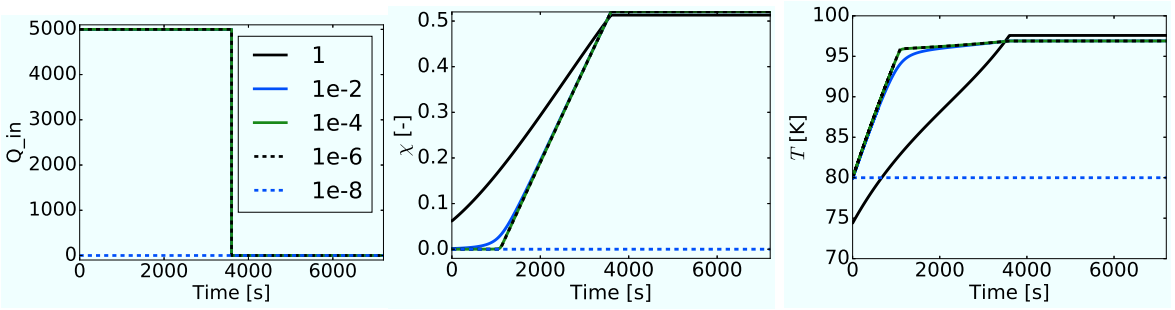
The optimization problem formulation is given in the supplementary material. We want to control the feed stream F, so that the difference of the holdup m and a specified holdup setpoint $m^{\text{set}} = 120$ mol is minimized, i.e.

$$\min_F \int_0^{t_f} \left(m(t) - m^{\text{set}}\right)^2 dt$$

The minimum holdup activating the overflow is $m_{\text{min}} = 60$ mol. For the optimal solution, we expect F to be at the upper bound until the desired buffer tank level is reached and then set equal to the outlet. The DAE describing the buffer tank has 1 differential eqatuion and 3 algebraic equations. The only constraints for the NLP are the bounds on F.

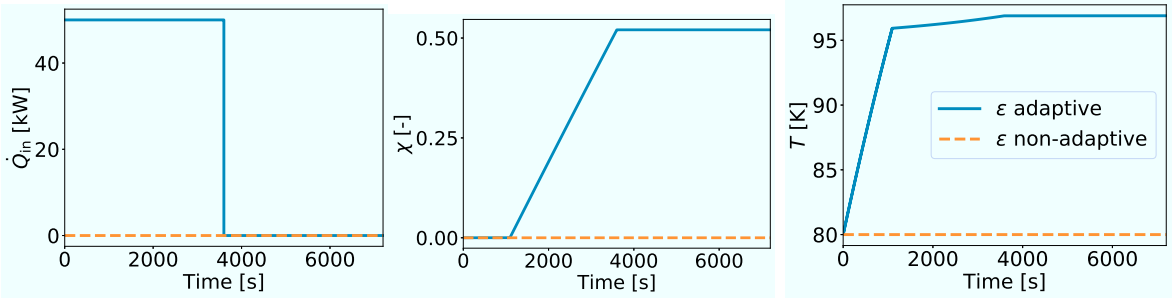
The results of the dynamic optimization are shown in Fig. 3. The tank holdup reaches the desired setpoint at about 15 s (Fig. 3a). The holdup starts at 0 mol and increases linearly until it reaches 60 mol. From then on, the holdup increases sub-linearly, due to the tank outlet flow. The profiles for m^- and m^+ reflect the holdup profile. The control variable profile, i.e., the tank feed rate is at the upper bound until the tank holdup reaches its setpoint (Fig. 3b). From then on it stays at 6 mol/s to keep the tank holdup at the desired setpoint. The solution approach successfully locates the switching points for the discrete events, since the results of the dynamic optimization are as expected. We can thus conclude the validity of the approach. A detailed validation of the control variable profile would require the solution of the optimization problems with the original nonsmooth DAE, as, e.g., in [35], or the formulation and solution as a corresponding DVI, which is, however, out of the scopt of this work.

3.2 Optimal Batch Vaporization



(a) (b) (c)

Fig. 4: Optimization results of the batch vaporization problem for different constant values for ε, ie., without adaptation of ε. (a) Heat stream control variable. (b) Vapor fraction. (c) Temperature.



(a) (b) (c)

Fig. 5: Optimization results of the batch vaporization problem. (a) Heat stream control variable. (b) Vapor fraction. (c) Temperature.

This case study addresses an optimal batch vaporization case-study using the relaxed VLE formulation as presented by Raghunathan and Biegler [31], but using a nonideal thermodynamic mixture.

Nevertheless, we can use the relaxed phase equilibrium equations for the VLE. Note the limitations of the formulation for liquid-liquid equilibrium [51]. The nonsmooth DAE with the relaxed VLE formulation and the Fischer-Burmeister function is not generally well-posed, as shown in the supplementary material. However, generalized differential index 1 is a local characteristic along a solution trajectory, and therefore the system is not generally required to be of index 1 for well-posedness of the DAE [41]. Thus, we use the smoothed Fischer-Burmeister function for optimization and hope the smoothed DAE to be well-posed along the solution trajectory. This case study illustrates the existence of undesired stationary points caused by the NCP function, which makes SNOPT to converge immediately at the first iteration to the initial values. The aim of the optimization is to control the vapor fraction χ to be at $\chi^{\text{set}} = 0.5$ by adjusting the heat transfer rate $\dot{Q}$ accordingly, and minimize

$$\int_0^{t_f} \left(\chi(t) - \chi^{\text{set}}\right)^2 dt.$$

The problem definition is provided in the supplementary material. The complete DAE of the boiler consists of one differential and 39 algebraic equations. The only constraints for the NLP are the bounds on Q_{in}.

The initial condition for the enthalpy corresponds to subcooled liquid. First, we present optimization results with different constant values of ε, i.e., without adaptive ε. Each optimization uses $\dot{Q}(t) = 0$ as initial guess. Afterwards, we compare the results with and without the adaptive strategy for the smoothing parameter ε.

Fig. 4 shows the optimization results with constant values of ε. Both the control and state variable profiles change with changing values of ε. In addition, the control variable profile stays constant at the initial guess profile for $\varepsilon = 10^{-8}$. This is a suboptimal stationary solution and a motivation for the adaptation strategy for ε as described in Section 2.1.3. If ε is increased, the control variable profile changes to the intuitively optimal solution (Fig. 4a). The state variables differ for different values of ε, although the control variable profiles are identical. The initial value for the temperature (Fig. 4c) changes with changing values of ε in order to meet the initial condition for the enthalpy.

In addition, we performed the optimizations with different DAE integrators and NLP solvers, using IDAS [52] and S-LIMEX [53] as additional integrators and IPOPT [54] as additional NLP solver. The results obtained using SNOPT and the three different DAE integrators are identical, as presented above. However, the resuts are different if IPOPT is used as NLP solver. For $\varepsilon = 10^{-8}$, IPOPT converges to the optimal solution and does not furnish the initial guess as solution. Thus, the adaptive strategy for ε would not be required in this case. However, this cannot be known a priori and may be very problem specific.

Thus, the adaptive strategy is usefull as a heuristic approach and adds a certain degree of assurance to prevent the NLP solver from converging to a suboptimal stationary solution.

We show two optimization results obtained with SNOPT to demonstrate the usefulness of the adaptive strategy for ε as described in Section 2.1.3: one optimization performed using the fixed value for $\varepsilon = 10^{-8}$ and one optimization performed repeatedly using decreasing values for ε starting with $\varepsilon = 1$ until $\varepsilon = 10^{-8}$ with an increment factor of 10. The heat transfer rate $\dot{Q}$ is initialized with piecewise constant control profiles consisting of one single element with the value of $\dot{Q}(t) = 0$. Each optimization is initialized with the solution of the previous optimization. The sensitivity of the objective function for the initial guess is -0.037 for $\varepsilon = 1$ and $-1.8 \cdot 10^{-9}$ for $\varepsilon = 1 \cdot 10^{-8}$, so that the sensitivity is larger than the optimality tolerance for the first value and smaller for the latter. There is, hence, a descent direction for $\varepsilon = 1$ and the NLP solver moves away from the initial solution.

The adaptive single-shooting algorithm is executed until convergence for each value of ε using the solution of the previous execution as initial guess for the next execution. The results of the optimizations are shown in the Figure 5.

The profiles obtained from the two optimization approaches are substantially different. While the heating profile (Fig. 5a) remains at the initial guess profile for the constant value of the parameter ε, the intuitive solution is obtained for the adaptive approach. The initial solution appears to be a suboptimal stationary point, cf. Section 2.1.3; the vapor fraction is zero for the initial iteration, such that its sensitivity is zero and with it the gradient of the objective function. The initial solution is not a C-stationary point, since all NLP Lagrange multipliers of the CCs are greater than or equal to zero and at least one multiplier would have to be negative in case of a C-stationary solution [17]. The value $\varepsilon = 10^{-8}$ of the non-adaptive approach leads to a too small sensitivity of the vapor fraction and SNOPT converges directly to the initial guess. The heat is used to increase the temperature of the sub-cooled liquid during the first 20 min (Fig. 5c). Then, the vaporization starts and the vapor fraction increases. The desired vapor fraction is obtained by the adaptive strategy after one hour (Fig. 5b). The heat transfer rate is then decreased to zero to keep the vapor fraction constant.

We conclude that the adaptive optimization approach is able to move the solution away from the initial suboptimal stationary point, caused by a small initial sensitivity of the objective function. The first solution (ε non-adaptive in Fig. 5b) leads to an objective function value of 1800, whereas the solution with the adaptation of ε leads to an objective function of 470.

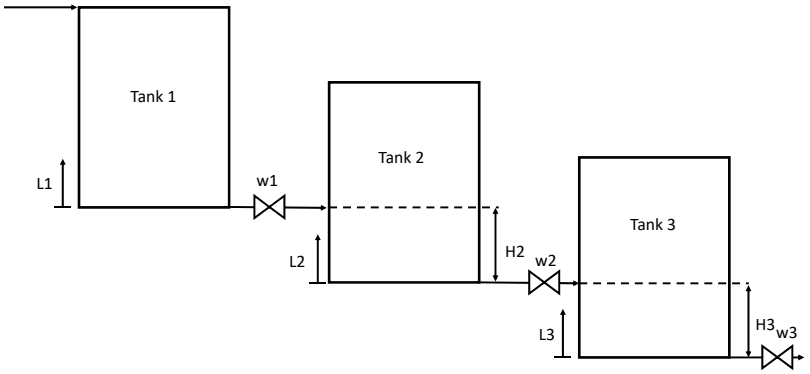


Fig. 6: Flowsheet of cascade tank problem for 3 tanks.

3.3 Tank Cascade Problem

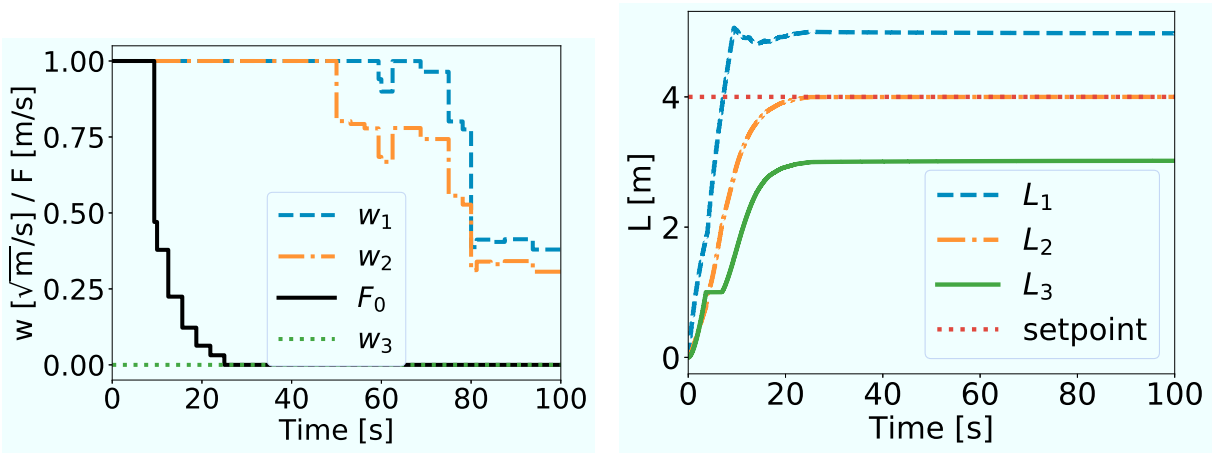
This case study is the cascading tank problem presented in [32, 16]. The discrete behavior results from valves behind each tank preventing a flow inversion and from the hydraulic correlation used to calculate the outlet stream. We compute the optimal control trajectory of the valve position of a series of cascading tanks as shown in Figure 6. The formulation of the optimization problem is provided in the supplementary material. We want to minimize the deviation of the tank holdups L_i from a setpoint of $L^{\text{sp}} = 4$ of a time horizon of $t_f = 100$ s by controlling the feed flowrate to the first tank F_0, and the valve positions w_1, w_2, and w_3., i.e., we minimize

$$\int_0^{t_f} \sum_{i \in \{1,\dots,N_{\text{tank}}\}} (L_i(t) - L_i^{\text{set}})^2 dt.$$

The only constraints of the NLP are the bounds on F_0 and on w_1, w_2, and w_3. The results of the dynamic optimization with three tanks and an adaptive control variable discretization starting with one element are shown in Fig. 7. The first tank is filled at the beginning and the inlet stream is then set to zero (Fig. 7a). The valves are opened at the beginning and reduced until the end (Fig. 7a). The control profiles let the tank holdups achieve approximately their desired setpoints (Fig. 7b). The state variable profiles look similar to the optimization results of [32].

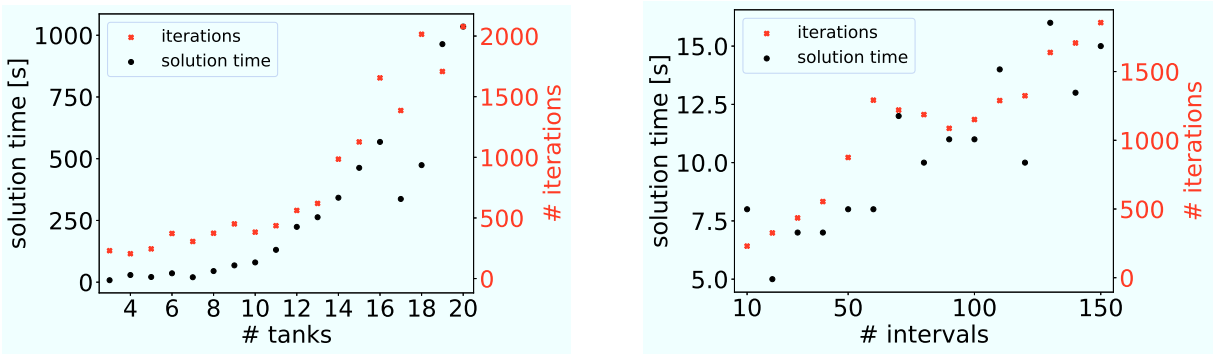
The problem was optimized in [55] using a mixed-integer approach. We compare the computational scaling with the results reported in [55, 32, 16] and investigate the influence of the number of tanks and the number of control variable intervals of the control variable discretization on the solution time.

In this case study, the solution times and NLP solver iterations for the optimizations using increasing



(a) (b)

Fig. 7: Control and state variable profiles of tank cascade optimization with 3 tanks and control grid adaptation. (a) Control variables. (b) Tank levels.



(a) (b)

Fig. 8: Solution times and number of iterations for the optimization of the tank cascade problem. (a) Solution times and number of iterations of NLP solver for increasing number of tanks and a discretization with 10 time intervals. (b) Solution times and number of iterations of NLP solver for increasing number of time intervals and 3 tanks.

number of tanks and time intervals are given in Figure 8. The solution times and NLP solver iterations grow quadratically with number of tanks, which is similar to the performance obtained in [32]. This admits a better time scaling than the solution in [55], which was reported to scale between quadratically and exponentially (Fig. 8). It is remarkable that both the solution times and the number of iterations increase, indicating that the NLP becomes computationally more challenging to solve. Here, not only the DAE integration becomes more challanging but also the solution of the NLP. The solution times and NLP solver iterations grow linearly with increasing number of intervals (Fig. 8), which corresponds to a better time scaling than reported in [32], where a linear to quadratic scaling was resported. The linear dependency between the solution times and the number of intervals is a main advantage of the proposed direct shooting based approach, as the adaptive integration grid of the DAE integrator is independent of the number of intervals of the control variables. On the other hand, DAE integration becomes more demanding only due to the larger sensitivity DAE, but not due to an increased number of time intervals. The step-size of the integrator is changed due to error control and not due to the number of time intervals of the control variable discretization.

3.4 Rectification Column Start-Up

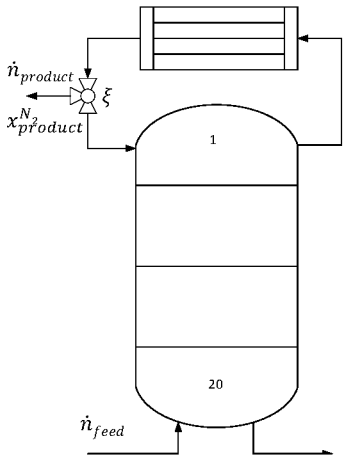


Fig. 9: Flowsheet of rectification column.

The last case study is the optimization of a start-up procedure for a rectification column. We use the rectification column model formulation of Raghunathan and Biegler [31], which also includes the overflow weir formulation presented in Section 3.1. In contrast to [31], we focus on a cryogenic rectification column for the separation of nitrogen, oxygen, and argon, and do not assume ideal thermodynamic behavior. The considered rectification column configuration comprises a column and a total condenser, as shown in Figure 9. This column is motivated by cryogenic air separation unit, see, e.g., [56]. The rectification

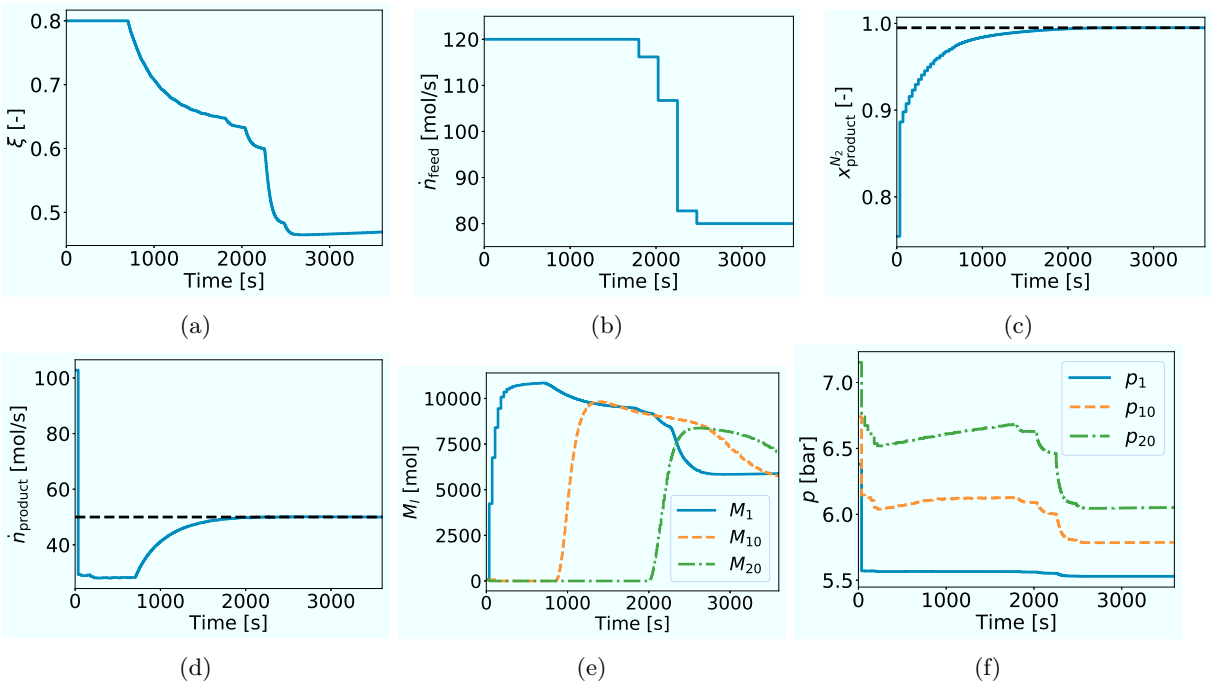


Fig. 10: Selected control and state variable profiles of rectification column optimization. (a) Split factor. (b) Feed flow rate. (c) N_2 Product purity, setpoint dashed. (d) Product flow rate, setpoint dashed. (e) Liquid holdups on trays. (f) Pressures on trays.

column consists of 20 equilibrium trays. During start-up, each tray may contain a liquid phase, a vapor phase, or a VLE. The model equations are given in the supplementary material. The model comprises 81 differential states, i.e., the molar holdups and the energy on the equilibrium trays, and 1665 algebraic variables. Each tray includes two CCs, one for the overflow weir and one for the VLE detection. Solving the problem with a MIDO approach would require at least 40 time-dependent binary variables for the different modes and trays leading to a large number of binary variables after discretization. We want to minimize the integral of the squared deviations of the product stream and product purity from their setpoints over a time horizon of one hour. Hence, we minimize

$$\int_0^{t_f} w_n(\dot{n}_{\text{product}}(t) - \dot{n}^{\text{set}}_{\text{product}})^2 + w_x(x^{N_2}_{\text{product}}(t) - x^{N_2,\text{set}}_{\text{product}})^2 dt$$

where $\dot{n}_{\text{product}}$ is the product flow rate and $\dot{n}^{\text{set}}_{\text{product}}$ the corresponding setpoint, $x^{N_2}_{\text{product}}$ is the product purity and $x^{N_2,\text{set}}_{\text{product}}$ the corresponding setpoint, w_n and w_x are weighting parameters. We weight the two parts of the objective function to lie in the same order of magnitude using $w_n = 1$ and $w_x = 1000$. The target setpoints are $\dot{n}^{\text{set}}_{\text{product}} = 50$ mol/s and $x^{N_2,\text{set}}_{\text{product}} = 0.995$. The initial profiles are time-invariant, with $\dot{n}^0_{\text{feed}}(t) = 80$ mol/s and $\xi^0(t) = 0.0$ and the bounds $\dot{n}_{\text{feed}}(t) \in [80, 120]$ mol/s, $\xi(t) \in [0.0, 0.8]$. The column is initialized with the steady-state corresponding to these initial control variable values, i.e., with

vapor at the dew point on every stage. We apply the adaptive smoothing parameter of the Fischer-Burmeister function starting at $\varepsilon = 1$ and reduce the parameter by the factor 10 at each iteration until $\varepsilon = 10^{-8}$, using adaptive single-shooting in each iteration with the respective parameter value.

The results of the dynamic optimization are given in the Fig. 10. The start-up takes about 2400 s. The split factor is increased in order to provide the column with a higher reflux at the beginning (10a). The feed flow rate is at its upper bound in the beginning and at the lower bound in the end (Fig.10b). The control variables have a fine discretization where needed and a coarse discretization otherwise. The controls make the product stream and product purity reach their desired setpoints (Fig. 10d and 10c). The VLE establishes in the column, beginning from the first tray and propagating down the column. This corresponds to the liquid holdups on the trays, with steep initial increases for trays 1, 10 and 20 (Fig. 10e). The pressures of the lower trays increase first due to increase of the liquid holdup in the upper trays (Fig. 10f). The pressure of a tray quickly decreases when a VLE exists since then a part of the hold up is present in the liquid phase with a higher density and the vapor holdup is decreased (Fig. 10f).

The case study shows that our approach can be used for the optimization of a large-scale dynamic system.

4 Conclusions

We consider optimization problems with smooth differential algebraic equations and CCs of algebraic state pairs. We solve the optimization problems applying direct single-shooting. The DAE integrator directly solves the CCs using NCP functions as model equations. To facilitate the dynamic optimization of the resulting nonsmooth DAE, we use a smoothed NCP function resulting in a smooth DAE, which allows for the application of standard DAE integrators and NLP solvers. We analyze both the nonsmooth DAE and the DAE with the smoothed NCP function and give conditions for the well-posedness of the systems. The generalized index 1 property of the nonsmooth DAE implies the well-posedness of the DAE with the smoothed NCP function. We use a smoothed Fischer-Burmeister NCP function for the solution of the optimization problems with CCs. Other smoothed NCP functions could be used as well.

However, the formulation with CCs may contain suboptimal solutions that are not local minima of the MPCC (e.g., C-stationary points with a descent direction). Therefore, we propose a heuristic approach. This prevents the NLP solver from converging to suboptimal stationary points by adjusting the smoothing parameter of the NCP function. Increasing the smoothing parameter, the sensitivities with respect to the NLP degrees of freedom can be increased and thus enable the NLP solver to move away from suboptimal

stationary points. The optimization problem is repeatedly solved with decreasing smoothing parameter.

The presented four numerical case studies range from simple illustrative examples to a large-scale optimization of a rectification column. In a tank cascade example, the method scales quadratically with increasing system size and linearly with increasing number of control variable intervals. We consider the approach to be particularly beneficial for large-scale problems including many discrete events, where a large number of binary or integer control variables would be required when formulated as MIDO problem.

The solution of the optimization problems with direct sequential methods using CC reformulations different from the Fischer-Burmeister function is left for future work, as well as the comparison of the presented approach with the alternative solution approaches (i) and (ii) as described in Section 2. In addition, the comparison with methods for the optimization of nonsmooth DAEs, the application of methods for global dynamic optimization, and the application of the present approach to model predictive control or more complex examples are considered promising research directions. While the convergence of NLPs with CCs formulated using the Fischer-Burmeister function is known [18], the convergence of solution of the DAE with the smoothed Fischer-Burmeister function to the solution of the DAE with the original Fischer-Burmeister function is also an interesting topic for future work. Furthermore, the comparison of the optimization with a smoothed DAE and the optimization with the original nonsmooth DAE, as in [35], is a relevant field for future research.

Acknowledgement

The authors gratefully acknowledge the financial support of the Kopernikus project SynErgie by the Federal Ministry of Education and Research (BMBF) and the project supervision by the project management organization Projektträger Jülich (PtJ). The authors thank Jan C. Schulze from AVT for proof reading.

References

[1] P. Stechlinski, M. Patrascu, and P. I. Barton, "Nonsmooth differential-algebraic equations in chemical engineering," *Computers & Chemical Engineering*, vol. 114, pp. 52–68, 2018.

[2] P. I. Barton, *The Modelling and Simulation of Combined Discrete/Continuous Processes*. PhD thesis, Imperial College of Science, Technology and Medicine London, 1992.

[3] P. I. Barton and C. C. Pantelides, "Modeling of combined discrete/continuous processes," *AIChE Journal*, vol. 40, no. 6, pp. 966–979, 1994.

[4] A. M. Sahlodin, H. A. J. Watson, and P. I. Barton, "Nonsmooth model for dynamic simulation of phase changes," *AIChE Journal*, vol. 62, no. 9, pp. 3334–3351, 2016.

[5] P. I. Barton and C. K. Lee, "Modeling, simulation, sensitivity analysis, and optimization of hybrid systems," *ACM Transactions on Modeling and Computer Simulation*, vol. 12, pp. 256–289, oct 2002.

[6] M. Avraam, N. Shah, and C. Pantelides, "Modelling and optimisation of general hybrid systems in the continuous time domain," *Computers & Chemical Engineering*, vol. 22, pp. S221–S228, 1998.

[7] R. Allgor and P. Barton, "Mixed-integer dynamic optimization i: problem formulation," *Computers & Chemical Engineering*, vol. 23, no. 4-5, pp. 567–584, 1999.

[8] J. Oldenburg, W. Marquardt, D. Heinz, and D. B. Leineweber, "Mixed-logic dynamic optimization applied to batch distillation process design," *AIChE Journal*, vol. 49, no. 11, pp. 2900–2917, 2003.

[9] K. Kraemer and W. Marquardt, "Continuous reformulation of MINLP problems," in *Recent Advances in Optimization and its Applications in Engineering*, pp. 83–92, Springer Berlin Heidelberg, 2010.

[10] J. E. Cuthrell and L. T. Biegler, "On the optimization of differential-algebraic process systems," *AIChE Journal*, vol. 33, no. 8, pp. 1257–1270, 1987.

[11] R. G. Brusch and R. H. Schapelle, "Solution of highly constrained optimal control problems using nonlinear programing," *AIAA Journal*, vol. 11, no. 2, pp. 135–136, 1973.

[12] H. Bock and K. Plitt, "A multiple shooting algorithm for direct solution of optimal control problems," *IFAC Proceedings Volumes*, vol. 17, no. 2, pp. 1603–1608, 1984.

[13] B. Baumrucker, J. Renfro, and L. Biegler, "MPEC problem formulations and solution strategies with chemical engineering applications," *Computers & Chemical Engineering*, vol. 32, no. 12, pp. 2903–2913, 2008.

[14] H. Scheel and S. Scholtes, "Mathematical programs with complementarity constraints: Stationarity, optimality, and sensitivity," *Mathematics of Operations Research*, vol. 25, no. 1, pp. 1–22, 2000.

[15] R. Fletcher, S. Leyffer, D. Ralph, and S. Scholtes, "Local convergence of SQP methods for mathematical programs with equilibrium constraints," *SIAM Journal on Optimization*, vol. 17, no. 1, pp. 259–286, 2006.

[16] L. T. Biegler, *Nonlinear Programming: Concepts, Algorithms, and Applications to Chemical Processes*. SIAM, 2010.

[17] Y. Kim, S. Leyffer, and T. Munson, "Mpec methods for bilevel optimization problems," *Argonne National Laboratory, MCS Division Preprint*, vol. ANL/MCS-P9195-0719, 2019.

[18] D. Ralph and S. J. Wright, "Some properties of regularization and penalization schemes for MPECs," *Optimization Methods and Software*, vol. 19, no. 5, pp. 527–556, 2004.

[19] R. Fletcher and S. Leyffer, "Solving mathematical programs with complementarity constraints as nonlinear programs," *Optimization Methods and Software*, vol. 19, no. 1, pp. 15–40, 2004.

[20] X. M. Hu and D. Ralph, "Convergence of a penalty method for mathematical programming with complementarity constraints," *Journal of Optimization Theory and Applications*, vol. 123, no. 2, pp. 365–390, 2004.

[21] A. U. Raghunathan and L. T. Biegler, "An interior point method for mathematical programs with complementarity constraints (MPCCs)," *SIAM Journal on Optimization*, vol. 15, no. 3, pp. 720–750, 2005.

[22] M. Anitescu, P. Tseng, and S. J. Wright, "Elastic-mode algorithms for mathematical programs with equilibrium constraints: global convergence and stationarity properties," *Mathematical Programming*, vol. 110, no. 2, pp. 337–371, 2006.

[23] D. Sun and L. Qi *Computational Optimization and Applications*, vol. 13, no. 1/3, pp. 201–220, 1999.

[24] X. Chen and M. Fukushima, "A smoothing method for a mathematical program with p-matrix linear complementarity constraints," *Computational Optimization and Applications*, vol. 27, no. 3, pp. 223–246, 2004.

[25] V. Gopal and L. T. Biegler, "Smoothing methods for complementarity problems in process engineering," *AIChE Journal*, vol. 45, no. 7, pp. 1535–1547, 1999.

[26] Y.-D. Lang and L. T. Biegler, "Distributed stream method for tray optimization," *AIChE Journal*, vol. 48, no. 3, pp. 582–595, 2002.

[27] O. Stein, J. Oldenburg, and W. Marquardt, "Continuous reformulations of discrete–continuous optimization problems," *Computers & Chemical Engineering*, vol. 28, no. 10, pp. 1951–1966, 2004.

[28] A. Fischer, "A special Newton-type optimization method," *Optimization*, vol. 24, no. 3-4, pp. 269–284, 1992.

[29] S. Leyffer, G. López-Calva, and J. Nocedal, "Interior methods for mathematical programs with complementarity constraints," *SIAM Journal on Optimization*, vol. 17, no. 1, pp. 52–77, 2006.

[30] A. U. Raghunathan and L. T. Biegler, "Mathematical programs with equilibrium constraints (MPECs) in process engineering," *Computers & Chemical Engineering*, vol. 27, no. 10, pp. 1381–1392, 2003.

[31] A. U. Raghunathan, M. S. Diaz, and L. T. Biegler, "An MPEC formulation for dynamic optimization of distillation operations," *Comput. Chem. Eng.*, vol. 28, no. 10, pp. 2037–2052, 2004.

[32] B. Baumrucker and L. Biegler, "MPEC strategies for optimization of a class of hybrid dynamic systems," *Journal of Process Control*, vol. 19, no. 8, pp. 1248–1256, 2009.

[33] 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," *Industrial & Engineering Chemistry Research*, vol. 33, no. 9, pp. 2123–2133, 1994.

[34] T. Guo and J. T. Allison, "On the use of mathematical programs with complementarity constraints in combined topological and parametric design of biochemical enzyme networks," *Engineering Optimization*, vol. 49, no. 2, pp. 345–364, 2016.

[35] M. Patrascu and P. I. Barton, "Optimal campaigns in end-to-end continuous pharmaceuticals manufacturing. part 2: Dynamic optimization," *Chemical Engineering and Processing - Process Intensification*, vol. 125, pp. 124–132, 2018.

[36] J.-S. Pang and D. E. Stewart, "Differential variational inequalities," *Mathematical Programming*, vol. 113, no. 2, pp. 345–424, 2007.

[37] J.-S. Pang and J. Shen, "Strongly regular differential variational systems," *IEEE Transactions on Automatic Control*, vol. 52, no. 2, pp. 242–255, 2007.

[38] C. Chen and O. L. Mangasarian, "A class of smoothing functions for nonlinear and mixed complementarity problems," *Computational Optimization and Applications*, vol. 5, pp. 97–138, Mar 1996.

[39] P. G. Stechlinski and P. I. Barton, "Generalized derivatives of differential–algebraic equations," *Journal of Optimization Theory and Applications*, vol. 171, no. 1, pp. 1–26, 2016.

[40] P. G. Stechlinski and P. I. Barton, "Generalized derivatives of optimal control problems with nonsmooth differential-algebraic equations embedded," in *2016 IEEE 55th Conference on Decision and Control (CDC)*, IEEE, 2016.

[41] P. G. Stechlinski and P. I. Barton, "Dependence of solutions of nonsmooth differential-algebraic equations on parameters," *Journal of Differential Equations*, vol. 262, no. 3, pp. 2254–2285, 2017.

[42] K. A. Khan and P. I. Barton, "Generalized derivatives for solutions of parametric ordinary differential equations with non-differentiable right-hand sides," *Journal of Optimization Theory and Applications*, vol. 163, no. 2, pp. 355–386, 2014.

[43] F. H. Clarke, *Optimization and Nonsmooth Analysis*. Society for Industrial and Applied Mathematics, 1987.

[44] M. Schlegel, K. Stockmann, T. Binder, and W. Marquardt, "Dynamic optimization using adaptive control vector parameterization," *Computers & Chemical Engineering*, vol. 29, no. 8, pp. 1731–1751, 2005.

[45] F. Assassa and W. Marquardt, "Optimality-based grid adaptation for input-affine optimal control problems," *Computers & Chemical Engineering*, vol. 92, pp. 189–203, 2016.

[46] F. Assassa and W. Marquardt, "Exploitation of the control switching structure in multi-stage optimal control problems by adaptive shooting methods," *Computers & Chemical Engineering*, vol. 73, pp. 82–101, 2015.

[47] T. Maly and L. R. Petzold, "Numerical methods and software for sensitivity analysis of differential-algebraic systems," *Applied Numerical Mathematics*, vol. 20, no. 1-2, pp. 57–79, 1996.

[48] A. Caspari, J. M. M. Faust, F. Jung, C. Kappatou, S. Sass, Y. Vaupel, R. Hannesmann-Tamás, A. Mhamdi, and A. Mitsos, "Dyos - a framework for optimization of large-scale differential algebraic equation systems," *Computer-Aided Chemical Engineering*, vol. 46, 2019.

[49] P. E. Gill, W. Murray, and M. A. Saunders, "SNOPT: An SQP algorithm for large-scale constrained optimization," *SIAM Rev.*, vol. 47, no. 1, pp. 99–131, 2005.

[50] R. Hannemann, W. Marquardt, U. Naumann, and B. Gendler, "Discrete first- and second-order adjoints and automatic differentiation for the sensitivity analysis of dynamic models," *Procedia Comput. Sci.*, vol. 1, no. 1, pp. 297–305, 2010.

[51] T. Ploch, M. Glass, A. M. Bremen, R. Hannemann-Tamás, and A. Mitsos, "Modeling of dynamic systems with a variable number of phases in liquid-liquid equilibria," *AIChE Journal*, vol. 65, pp. 571–581, nov 2018.

[52] R. Serban, C. Petra, , and A. C. Hindmarsh, *User Documentation for idas v2.2.1.* Center for Applied Scientic Computing, Lawrence Livermore National Laboratory, 2018.

[53] M. Schlegel, W. Marquardt, R. Ehrig, and U. Nowak, "Sensitivity analysis of linearly-implicit differential–algebraic systems by one-step extrapolation," *Applied Numerical Mathematics*, vol. 48, no. 1, pp. 83–102, 2004.

[54] A. Wächter and L. T. Biegler, "On the implementation of an interior-point filter line-search algorithm for large-scale nonlinear programming," *Mathematical Programming*, vol. 106, no. 1, pp. 25–57, 2005.

[55] J. Till, S. Engell, S. Panek, and O. Stursberg, "Applied hybrid system optimization: An empirical investigation of complexity," *Control Engineering Practice*, vol. 12, no. 10, pp. 1291–1303, 2004.

[56] A. Caspari, J. M. Faust, P. Schäfer, A. Mhamdi, and A. Mitsos, "Economic nonlinear model predictive control for flexible operation of air separation units," *IFAC-PapersOnLine*, vol. 51, no. 20, pp. 295–300, 2018.