Journal of Process Control 19 (2009) 1248–1256

Contents lists available at ScienceDirect

Journal of Process Control

journal homepage: www.elsevier.com/locate/jprocont

MPEC strategies for optimization of a class of hybrid dynamic systems

B.T. Baumrucker, L.T. Biegler *

Chemical Engineering Department, Carnegie Mellon University, Pittsburgh, PA 15213, USA

ARTICLE INFO

Article history:
Received 8 October 2008
Received in revised form 14 February 2009
Accepted 28 February 2009

Keywords:
Hybrid systems
Dynamic optimization
MPECs
Complementarity constraints

ABSTRACT

With the development and widespread use of large-scale nonlinear programming (NLP) tools for process optimization, there has been an associated application of NLP formulations with complementarity constraints in order to represent discrete decisions. In particular, these constraints arise frequently in equation-based formulations for real-time optimization. Also known as mathematical programs with equilibrium constraints (MPECs), these formulations can be used to model certain classes of discrete events and can be more efficient than a mixed integer formulation, particularly for large systems with many discrete decisions, such as dynamic systems with switches at any point in time. In this study, we consider and extend MPEC formulations for the optimization of a class of hybrid dynamic models, where the differential states remain continuous over time. These include differential inclusions of the Filippov type. Here, particular care is required in the formulation in order to preserve smoothness properties of the dynamic system. Results on three case studies, including process control examples, illustrate the effectiveness and accuracy of the proposed MPEC optimization methodology for a class of hybrid dynamic systems.

1. Introduction

Mathematical programming methods have proven extremely valuable for the design and operation of chemical processes. This has been made possible by more flexible modeling constructs, increased computing power, and a better fundamental understanding of solution algorithms and problem formulations. Related optimization models have a number of options to represent discrete decisions, as in contact and friction problems in computational mechanics, equilibrium relations in economics, and a wide variety of discrete events in process systems.

Traditionally, modular approaches to simulation problems have handled discrete events through logic embedded in the software (*IF–THEN* conditions). Depending on the value of computed variables, different sets of equations may be used for the simulation. However, this modular approach has its limitations for complex models and information flow reversals. Moreover, its usage in an optimization framework has not been well-supported except for small, structured problems. Instead, equation based flowsheet models are much better suited for use in an optimization framework and have the flexibility to handle complex models and specifications. On the other hand, equation-based approaches must deal with discrete decisions in a different way. Here, embedded logic to change model equations is undesirable, as it leads to nonsmooth relations and usually results in solver failures or convergence at nonphysical solution points of the model equations.

These problems become more pronounced when dynamic problems are considered, where the combination of discrete decisions and dynamics is generally referred to as hybrid dynamics. Pioneering work on differential equations with discontinuous right hand sides was performed by Filippov [11]. These lead to a dynamic system of Differential Inclusions (DIs); a survey along with an analysis of several classes of DIs is given by Dontchev and Lempio [10]. More recently Pang and Stewart [23] introduce and study the notion of differential variational inequalities (DVI). Their study includes development of time-stepping methods along with conditions for existence of solutions, convergence of particular methods and rates of convergence with respect to step sizes. Consistency of time-stepping methods for DVIs was also investigated by Heemels and coworkers [15–17]. DVIs are a broad problem class that unify several other fields including ordinary differential equations (ODEs) with smooth or discontinuous right-hand sides, differential algebraic equations (DAEs), dynamic complementarity systems, and evolutionary variational inequalities. Leine and Nijmeijer [20] consider dynamics and bifurcation of non-smooth systems with applications in mechanics. Control and optimization of hybrid systems have been active application areas for DVIs, and smoothing methods, non-smooth solution techniques, and mixed integer approaches have been considered in a number of studies including [33,5,4,34,3,22]. In particular, the last two approaches deal with optimization formulations with discrete decisions. A direct transcription strategy was used with mixed integer techniques in [3]. More recently, a mixed logic dynamic optimization problem is formulated in [22], using Generalized Disjunctive Programming (GDP) to cover fairly general classes of hybrid systems.

* Corresponding author. Tel.: +1 412 268 2232; fax: +1 412 268 7139.
E-mail address: lb01@andrew.cmu.edu (L.T. Biegler).

0959-1524/$ - see front matter © 2009 Elsevier Ltd. All rights reserved.
doi:10.1016/j.jprocont.2009.02.006

This study adopts a direct transcription, or simultaneous collocation, approach to the treatment of DVIs. This "discretize-then-optimize" strategy also incorporates moving finite elements to track the location of nonsmooth features. In this work a class of hybrid systems is considered where the differential states remain continuous over all time. For this class, the dynamic optimization problem can be formulated using only continuous variables. In the next section we discuss the formulation of MPEC problems and discuss NLP strategies for their solution. Section 3 then develops direct transcription formulations that lead to MPEC problems. Here, some care is needed to ensure accurate, piece-wise smooth solutions. Section 4 presents three case studies that illustrate this approach and demonstrate the advantages of our MPEC formulation. Finally, Section 5 concludes the study and briefly describes areas for future work.

2. Complementarities for non-smooth behavior

Optimization of hybrid dynamic systems requires the treatment of nonsmooth conditions within the problem formulation. As mentioned above, Generalized Disjunctive Programming (GDP) and Mixed Integer Programming (MIP) strategies are general ways to handle logical disjunctions that lead to nonsmoothness. However, the associated computational expense may be high for large systems with many discrete decisions. This is often the case in dynamic systems with switches at any point in time. Because worst-case MIP solution time grows exponentially with the number of discrete decisions, complementarity formulations offer an alternative for some classes of disjunctive problems. Complementarity is a relationship between two variables where either one (or both) must be at its bound. Complementarities are particularly useful in optimization since they can be used to model disjunctions without the use of binary variables. As a result, they can be embedded within a standard nonlinear programming (NLP) solver to obtain fast local solutions, generally in polynomial time. On the other hand, the introduction of complements introduces an inherent non-convexity as well as linear dependence of active constraints at all feasible points.

For hybrid dynamic systems, complementarity models allow a number of options when dealing with disappearance of phases, flow reversal, safety valve operation, sliding modes and other discrete events. For a more complete discussion of chemical engineering applications that can be modeled with complementarities, see [6,24,25]. For conventional differential-algebraic equation (DAE) solvers, models with derivative discontinuities are usually handled through logic embedded in the DAE model. Depending on the value of computed variables, different sets of equations may be used for the dynamic simulation. However, when extended to an optimization formulation, this approach introduces nonsmoothness into the optimization problem, which is difficult to handle with existing NLP algorithms. Moreover, algorithms for non-smooth optimization tend to be less advanced and slower to converge than algorithms for NLPs. As a result, we consider a direct transcription approach, where fully discretized equation-based DAE models allow more flexible formulation of these discrete features, either through disjunctions or complementarity constraints.

A mathematical program with equilibrium constraints (MPEC) is often written as the following optimization problem with complementarity constraints:

$$\min \quad f(z) \tag{1a}$$
$$\text{s.t.} \quad h(z) = 0 \tag{1b}$$
$$g(z) \geqslant 0 \tag{1c}$$
$$0 \leqslant z_1 \perp z_2 \geqslant 0 \tag{1d}$$

where we define complementing variables as $z_1, z_2 \in \mathbb{R}^{n_1}$, as a subset of the variable vector, $z \in \mathbb{R}^{n_z}$ and $\perp$ is the complementarity operator enforcing at least one of the complementing bounds to be active. The complementarity constraint therefore implies the following:

$$z_1^{(j)} = 0 \quad \text{OR} \quad z_2^{(j)} = 0,\; j = 1, \ldots, n_1, \quad z_1 \geqslant 0,\; z_2 \geqslant 0,$$

so that complementarity constraints implicitly model disjunctive behavior. Here the OR operator is inclusive as both variables may be zero. MPEC problems are ill-posed and generally difficult to solve as NLPs, because they do not satisfy constraint qualifications and have unbounded multipliers.

To make the MPEC problem tractable, the MPEC problem can be reformulated and solved in a variety of ways. In particular, formulations that allow the use of standard NLP solvers are particularly attractive. As described in [6,12,29] MPEC reformulations can be classified into relaxation methods and penalty function methods. In relaxation methods [24,26,31,36], the complementarity constraint (1d) is reformulated as a set of equality or inequality constraints and relaxed by a parameter $\eta > 0$ so that the resulting NLP is well-posed [24,26,29,31]. Solving a sequence $\{l\}$ of relaxed problems with $lim_{l\to\infty}\eta_l \to 0$ allows the MPEC solution to be approached asymptotically, under assumptions of strong stationarity. In contrast, we consider (1) through an ℓ_1 penalty formulation [21,14] leading to an NLP given by:

$$\text{PF}(\rho): \quad \min \quad f(z) + \rho z_1^T z_2 \tag{2a}$$
$$\text{s.t.} \quad h(z) = 0 \tag{2b}$$
$$g(z) \geqslant 0 \tag{2c}$$
$$z_1, z_2 \geqslant 0 \tag{2d}$$

Here the complementarity (1d) is moved from the constraints to the objective function and the resulting problem is solved for a suitably large value of ρ. If $\rho \geqslant \rho_c$, where ρ_c is the critical value of the penalty parameter, then the complementarity constraints will be satisfied at the solution. Similarly, Anitescu et al. [1] consider a related "elastic mode" formulation, where artificial variables are introduced to relax the constraints in PF(ρ) and an additional ℓ_∞ constraint penalty term is added. In both cases the resulting NLP formulation has the following properties:

- If z^* is a strongly stationary point for (1), then for all ρ sufficiently large z^* is a stationary point for PF (ρ)[29].
- If z^* is a solution to PF (ρ) and z^* is feasible for (1) with finite ρ, then z^* is a strongly stationary solution to (1) [1].

These properties indicate that the exact penalty-based solution strategy is attracted to strongly stationary points for sufficiently large values of ρ. Although a threshold value $\rho > \rho_c$ is not known *a priori*, it is usually not difficult to choose a sufficiently large value (generally 3–6 orders of magnitude greater than the objective function for well scaled problems) that serves to set the penalty term ($\rho z_1^T z_2$) to zero at the solution to (2).

Note that with a finite value of $\rho > \rho_c$ the ℓ_1 penalization is exact in the sense that all strongly stationary points of (1) are local minimizers to PF(ρ). Therefore, PF(ρ) allows any general nonlinear programming solver to be used to solve a complementarity problem directly, not asymptotically as with relaxation methods. This approach is described in more detail in [1,21]. Additional examples of MPECs applied to dynamic optimization problems can be found in [27,18,28].

Finally, to solve PF(ρ) we apply both the active set NLP solver, CONOPT [7], and the barrier method, IPOPT [37]. These solvers have competing advantages; CONOPT quickly detects active sets and handles dependent constraints efficiently, while IPOPT has low computational complexity in handling large-scale problems with many inequality constraints and degrees of freedom. Both sets of advantages will be exploited in Section 4.

3. Dynamic optimization with direct transcription

To illustrate a hybrid form with complements, we consider a switched system with ordinary differential equation models. This system has discontinuous right hand sides and is of the form considered by Filippov [11].

$$\dot{x} = \begin{cases} f_-(x,u,t) & \sigma(x(t)) < 0 \\ v(t)f_-(x,u) + (1-v(t))f_+(x,u,t)\ v(t) \in [0,1] & \sigma(x(t)) = 0 \\ f_+(x,u,t) & \sigma(x(t)) > 0 \end{cases} \tag{3}$$

In this representation, $t, x(t), u(t), v(t)$ and $\sigma(x(t))$ are time, differential state variables, control variables, switching profiles, and guard (or switching) function, respectively. The scalar switching function, $\sigma(x(t))$ determines transitions to different state models, represented by a scalar switching profile $v(t)$ set to zero or one. At the transition point, where $\sigma(t) = 0$, a convex combination of the two models is allowed with $v(t) \in [0,1]$. Note that if $\sigma(x(t)) = 0$ over a nonzero period of time, $v(t)$ can be determined from smoothness properties of $\sigma(x(t))$, i.e.,

$$\frac{d\sigma}{dt} = \nabla_x \sigma(x)^T [v(t)f_-(x,u) + (1-v(t))f_+(x,u,t)] = 0 \tag{4}$$

Also, for this problem class, we assume that the differential states $x(t)$ remain continuous over time. Existence and uniqueness properties of (3) have been presented in [11,20]. We can express (3) equivalently through the addition of slack variable profiles and complementarity constraints as:

$$\sigma(x(t)) = s^p(t) - s^n(t) \tag{5a}$$
$$\dot{x} = v(t)f_-(x,u) + (1-v(t))f_+(x,u,t), \quad v(t) \in [0,1] \tag{5b}$$
$$0 \leqslant s^p(t) \perp v(t) \geqslant 0 \tag{5c}$$
$$0 \leqslant s^n(t) \perp (1-v(t)) \geqslant 0 \tag{5d}$$

Using direct transcription, the DAE is converted into an NLP by approximating state and control profiles by a family of polynomials on finite elements, defined by $t_0 < t_1 < \cdots < t_N$, as described in [19,27]. In addition, the differential equation is discretized using K-point Radau collocation on finite elements. This method is equivalent to an implicit Runge–Kutta (IRK) integration scheme of order $2K-1$. Here, we use a monomial basis representation for the differential profiles, as follows:

$$x(t) = x_{i-1} + \sum_{k=1}^{K} \Omega_k\left(\frac{t-t_{i-1}}{h_i}\right)\dot{x}_{i,k}, \quad t \in [t_{i-1}, t_i] \tag{6}$$

where x_{i-1}, t_{i-1} are values of the differential variable and time at the beginning of element $i, h_i = t_i - t_{i-1}$ is the length of element $i, \dot{x}_{i,k}$ is the value of its first derivative in element i at the collocation point k, and Ω_k is the polynomial of order K, satisfying

$$\Omega_k(0) = 0 \quad \text{for } k = 1,\ldots,K$$
$$\Omega'_k(\tau_{k'}) = \delta_{k,k'} \quad \text{for } k,k' = 1,\ldots,K$$

where τ_k is the location of the kth collocation point within each element, and $\tau_K = 1$. At each collocation point, the corresponding state variables are represented by:

$$x_{ik} = x_{i-1} + \sum_{k'=1}^{K} \Omega_{k'}(\tau_k)\dot{x}_{i,k'}, \quad k = 1,\ldots K \tag{7}$$

Continuity of the differential profiles is enforced by

$$x_i = x_{i-1} + h_i \sum_{k=1}^{K} \Omega_K(1)\dot{x}_{i,k} \tag{8}$$

The switching and control profiles are approximated using a similar monomial basis representation which takes the form:

$$v(t) = \sum_{k=1}^{K} \psi_k\left(\frac{t-t_{i-1}}{h_i}\right)v_{i,k} \tag{9}$$

$$u(t) = \sum_{k=1}^{K} \psi_k\left(\frac{t-t_{i-1}}{h_i}\right)u_{i,k} \tag{10}$$

where ψ_k is the Lagrange polynomial of order K satisfying

$$\psi_k(\rho_{k'}) = \delta_{k,k'} \quad \text{for } k,k' = 1,\ldots,K$$

From (7), the differential variables are required to be continuous throughout the time horizon, while the control and algebraic variables are allowed to have discontinuities at the boundaries of the elements.

A straightforward substitution of (7)–(10) allows us to rewrite (5) as:

$$\dot{x}_{ik} = v_{ik}f_-(x_{ik},u_{ik}) + (1-v_{ik})f_+(x_{ik},u_{ik}), \quad v_{ik} \in [0,1]$$
$$\sigma(x_{ik}) = s^p_{ik} - s^n_{ik}, \quad i = 1,\ldots,N,\ k = 1,\ldots,K$$
$$0 \leqslant s^p_{ik} \perp v_{ik} \geqslant 0, \quad i = 1,\ldots,N,\ k = 1,\ldots,K$$
$$0 \leqslant s^n_{ik} \perp (1-v_{ik}) \geqslant 0, \quad i = 1,\ldots,N,\ k = 1,\ldots,K$$

However, this formulation is not sufficient to enforce sufficient smoothness in the Runge–Kutta approximation for $x(t), v(t)$ and $u(t)$. For this we allow a variable length $h_i \in [h_L, h_U]$ for each finite element (determined by the NLP) and we allow sign changes in $\sigma(x(t))$ only at t_i, the boundary of the finite element. A small, positive lower bound on h_i is required to ensure that h_i does not go to zero, thus creating a degeneracy in the problem. An upper bound on h_i is chosen to limit the integration error associated with the finite element. Permitting sign changes in $\sigma(t)$ only at the boundary of a finite element is enforced by choosing $v(t)$ to complement the L_1 norm, $\int_{t_{i-1}}^{t_i} |s^p(t)|\, dt$, and $1 - v(t)$ to complement $\int_{t_{i-1}}^{t_i} |s^n(t)|\, dt$. With this modification, the discretized formulation becomes:

$$\dot{x}_{ik} = v_{ik}f_-(x_{ik},u_{ik}) + (1-v_{ik})f_+(x_{ik},u_{ik}), \quad v_{ik} \in [0,1] \tag{11a}$$
$$\sigma(x_{ik}) = s^p_{ik} - s^n_{ik}, \quad i = 1,\ldots,N,\ k = 1,\ldots,K \tag{11b}$$
$$0 \leqslant \sum_{k'=0}^{K} s^p_{ik'} \perp v_{ik} \geqslant 0, \quad i = 1,\ldots,N,\ k = 1,\ldots,K \tag{11c}$$
$$0 \leqslant \sum_{k'=0}^{K} s^n_{ik'} \perp (1-v_{ik}) \geqslant 0, \quad i = 1,\ldots,N,\ k = 1,\ldots,K \tag{11d}$$

where we define $s^p_{i0} = s^p_{i-1,K}$ and $s^n_{i0} = s^n_{i-1,K}$ for $i = 2,\ldots,N$. Note that the complementarities are now formulated such that only one branch of the piecewise function is allowed over element i, i.e.,

- $v(t) = 0$ only if $\sigma(x(t)) \geqslant 0$ for the entire finite element,
- $v(t) = 1$ only if $\sigma(x(t)) \leqslant 0$ for the entire finite element,
- $v(t) \in (0,1)$ only if $\sigma(x(t)) = 0$ over the entire finite element.

Moreover, for the last condition, we have an index-2 path constraint $\sigma(x(t)) = 0$ over the finite element. In our direct transcription approach the Radau collocation scheme allows us to handle the high-index constraint directly as:

$$\sigma(x_{ik}) = 0, \quad i = 1,\ldots,N,\ k = 1,\ldots,K$$

and to obtain $v(t)$ implicitly through the solution of (11). As noted in [2], Radau collocation is stable and accurate for index-2 systems; the error in the differential variables is $O(h^{2K-1})$ and the error in the algebraic variables is only reduced to $O(h^K)$.

We now generalize this formulation to multiple guard functions and switches that define NE epochs of positive length, indexed by $l = 1, \dots NE$, along with hybrid index-1 DAE models given by:

$$\left.\begin{array}{l} F(\dot{x}(t), x(t), y(t), u(t), v(t), p) = 0 \\ g(x(t), y(t), u(t), v(t), p) \leqslant 0 \end{array}\right\}, \quad t \in [t_{l-1}, t_l], \; l = 1, \dots NE$$

with algebraic variables $y(t)$, time independent optimization variables p and inequality constraints g that may include the vector of switching functions. For this problem class, we further assume that epochs can be of variable time, that transitions occur only at epoch boundaries and that differential states remain continuous over all time. This leads to a set of switching profiles, indexed by $m \in \mathscr{M}$, and represented through a relation between guard functions, $\sigma_m(x(t))$, as follows:

$$\left.\begin{array}{l} F(\dot{x}, x(t), y(t), u(t), v(t), p) = 0 \\ g(x(t), y(t), u(t), v(t), p) \leqslant 0 \\ \sigma_m(x(t)) < 0 \Rightarrow v_m(t) = 1 \\ \sigma_m(x(t)) > 0 \Rightarrow v_m(t) = 0 \\ \sigma_m(x(t)) = 0 \Rightarrow v_m(t) \in [0, 1] \end{array}\right\}, \quad t \in (t_{l-1}, t_l], \; m \in \mathscr{M} \tag{12a}$$

$$x(t_0) = x_0, \quad x(t_f) = x(t_{NE}^+) = x_f \tag{12b}$$

$$x(t_l^-) = x(t_l^+), \quad l = 1, \dots, NE - 1 \tag{12c}$$

In a manner similar to reformulating (5) to form the complementarity system (11), we apply Radau collocation on finite elements. Because one or more finite elements are allowed to make up an epoch l, we re-adopt the finite element index i for epochs as well and apply the complementarity conditions within each finite element. This leads to the following formulation:

$$F(\dot{x}_{ik}, x_{ik}, y_{ik}, u_{ik}, v_{ik}, p) = 0 \tag{13a}$$

$$g(x_{ik}, y_{ik}, u_{ik}, v_{ik}, p) \leqslant 0 \tag{13b}$$

$$x(t_i^-) = x(t_i^+) \tag{13c}$$

$$\sigma_m(x_{ik}) = s^p_{m,ik} - s^n_{m,ik} \tag{13d}$$

$$0 \leqslant \left(\sum_{k'=0}^{K} s^p_{m,ik'}\right) \perp v_{m,ik} \geqslant 0 \tag{13e}$$

$$0 \leqslant \left(\sum_{k'=0}^{K} s^n_{m,ik'}\right) \perp (1 - v_{m,ik}) \geqslant 0 \tag{13f}$$

$$i = 1, \dots, N, \quad k = 1, \dots, K, \quad m \in \mathscr{M} \tag{13g}$$

$$x(t_0) = x_0, \quad x(t_N^+) = x_f \tag{13h}$$

Eqs. (13) constitute the constraints for the discretized hybrid dynamic optimization problem represented by the MPEC (1). Moreover, because a higher order IRK discretization is used within smooth finite elements, we are able to enforce accurate solutions to the hybrid dynamic system, with a relatively small number of finite elements.

4. Case studies

We now consider three case studies that illustrate our systematic strategy for developing complementarity formulations for hybrid systems. The first example considers frequent discrete decisions in a dynamic system through the reformulation of the *signum* function. This case demonstrates the advantage of the MPEC reformulation strategy to obtain accurate optimal solutions. In addition, this example demonstrates and compares the computational complexity of the MPEC and MINLP approaches. The second case study considers a hybrid dynamic system in a cascading tank process. This problem is posed in a way that naturally lends itself to Model Predictive Control. Through our proposed MPEC formulation, we demonstrate that optimal solutions can be found reliably with polynomial complexity. Finally, the third case study deals with optimization of a vehicle dynamic system with hybrid elements due to modeling of frictional forces. This problem involves a race car completing a course in minimum time and is generally referred to as the "Michael Schumacher" problem [32]. This problem is used to demonstrate how high index control problems arise from the proposed methodology as well as the numerical accuracy of our proposed MPEC approach under these circumstances.

4.1. Reformulation of a differential inclusion

The first example is a dynamic optimization with a differential inclusion:

$$\begin{array}{ll} \min & \phi = (x_f - 5/3)^2 + \int_{t_0}^{t_f} x^2 \cdot dt \\ \text{s.t.} & \dot{x} \in 2 - sgn(x), \quad x(t_0) = -2 \end{array}$$

As stated, this problem has a unique, well-defined solution with no degrees of freedom. This problem can be redefined by a complementarity system [33] as the following optimal control problem:

$$\min \quad \phi = (x_f - 5/3)^2 + \int_{t_0}^{t_f} x^2 \cdot dt \tag{14a}$$

$$\text{s.t.} \quad \dot{x} = 2 - v, \quad x(t_0) = -2 \tag{14b}$$

$$x = s^p - s^n \tag{14c}$$

$$0 \leqslant 1 - v \perp s^p \geqslant 0 \tag{14d}$$

$$0 \leqslant v + 1 \perp s^n \geqslant 0 \tag{14e}$$

Applying Radau collocation with $t_0 = 0, t_f = 2$ and with variable element lengths restricted to $1.6/N \leqslant h_i \leqslant 2.4/N$, the discretized problem can be written as the following MPEC:

$$\min \quad (x_f - 5/3)^2 + \sum_{i=1}^{N} \sum_{k=1}^{K} \omega_k h_i \cdot x_{ik}^2 \tag{15a}$$

$$\text{s.t.} \quad \dot{x}_{ik} = 2 - v_{ik} \quad i = 1, \dots, N, \; k = 1, \dots, K \tag{15b}$$

$$x_i = x_{i-1} + h_i \sum_{k=1}^{K} \Omega_K(1) \dot{x}_{i,k} \quad i = 1, \dots, N \tag{15c}$$

$$x_{ik} = x_{i-1} + \sum_{k'=1}^{K} \Omega_{k'}(\tau_k) \dot{x}_{i,k}, \quad i = 1, \dots, N, \; k = 1, \dots, K \tag{15d}$$

$$x_{ik} = s^p_{ik} - s^n_{ik}, \quad i = 1, \dots, N, \; k = 1, \dots, K \tag{15e}$$

$$0 \leqslant 1 - v_{ik} \perp \sum_{k'=0}^{K} s^p_{ik'} \geqslant 0, \quad i = 1, \dots, N \tag{15f}$$

$$0 \leqslant v_{ik} + 1 \perp \sum_{k'=0}^{K} s^n_{ik'} \geqslant 0, \quad i = 1, \dots, N, \; k = 1, \dots, K \tag{15g}$$

$$1.6/N \leqslant h_i \leqslant 2.4/N, \quad i = 1, \dots, N \tag{15h}$$

$$\sum_{i=1}^{N} h_i = 2 \tag{15i}$$

where ω_k are Radau quadrature weights. For this formulation, choosing $K = 1$ leads to an implicit Euler method with first order accuracy. Since the differential equation is piecewise constant, implicit Euler integrates the differential equation exactly and leads to exact identification of the switching locations. On the other hand, the integrand in the objective function is of higher order and is not exact with quadrature of $K = 1$. The resulting discretized problem can be scaled up to be arbitrarily large by increasing the number of finite elements N. The complementarities can also be modeled by using a discrete decision in each element, leading to a Mixed Integer Nonlinear Program (MINLP). Using concepts from Generalized Disjunctive Programming (GDP) one can introduce binary variables, b_i, and replace (15e)–(15g) by the following constraints:

$$x_L(b_i - 1) \leqslant x_{ik} \leqslant x_U\, b_i, \quad i = 1, \ldots, N,\ k = 1, \ldots K \tag{16a}$$
$$v_{ik} = 2b_i - 1, \quad i = 1, \ldots, N \tag{16b}$$
$$b_i \in \{0, 1\}, \quad i = 1, \ldots, N \tag{16c}$$

However, finding a solution to this MINLP is NP-Hard and with finer discretizations the problem can become expensive. In contrast, solving (15) through an NLP formulation generally leads to only polynomial complexity.

The discretized MPEC was solved using the NLP penalty reformulation with CONOPT using $\rho = 1000$. The results are shown in Table 1. It can be seen that the number of iterations increases almost linearly with the number of finite elements and this behavior is typical for active set strategies. In addition, the computational time per iteration increases nearly linearly with the number of finite elements, and the total computational time for CONOPT grows approximately quadratically with problem size. To solve the MINLP formulation both DICOPT [9,13], an outer approximation method that uses CONOPT and CPLEX [8,13] for the NLP and MILP subproblems, respectively, and SBB [13,30] were applied. From the results in Table 1 we can observe the NP-hard complexity of solving the MINLP formulation. The problem with 100 finite elements cannot be solved within 10,000 CPUs with DICOPT, while the problem with 1000 finite elements cannot be solved within 10,000 CPUs with SBB.

The analytic solution for the hybrid dynamic system is plotted in Fig. 1. Starting from $x(0) = -2, x(t)$ is piecewise linear and the influence of $sgn(x)$ can be seen clearly from the plot of dx/dt. Moreover, from the analytic solution, it can be shown that $x(t)$ and the objective function, ϕ are both differentiable in $x(0)$. On the other hand, a discretized problem with h_i fixed does not locate the transition point accurately and this leads to inaccurate profiles for $x(t)$ and $v(t)$. Moreover, as discussed in [33], the application of fixed finite elements also leads to a nonsmooth dependence of the solution on $x(0)$, as seen in Fig. 2. The plot with $N = 100$ shows a sawtooth behavior of ϕ vs. $x(0)$ when the elements remain fixed. In contrast with variable finite elements the objective function varies smoothly with $x(0)$. This occurs because the NLP solver can now locate the switching points accurately, and the complementarity formulation requires the differential state $x(t)$ to remain smooth within an element.

Moreover, the Euler discretization captures the piecewise linear $x(t)$ profile exactly and varies smoothly with $x(0)$. On the other hand, because ϕ still has local errors of $O(h_i^2)$, Fig. 2 shows that the plot of the *analytically determined* objective function still differs from the Euler discretization, despite the accurate determination of $x(t)$. (When the problem is resolved with $K = 3$, Radau collocation has 5th order accuracy and the numeric solution matches the analytical solution within the numerical precision of the machine). Nevertheless, in both of these cases smoothness is still maintained in Fig. 2 because the discontinuity is located exactly, and the integral is evaluated over the piecewise smooth portions, not across the discontinuity.

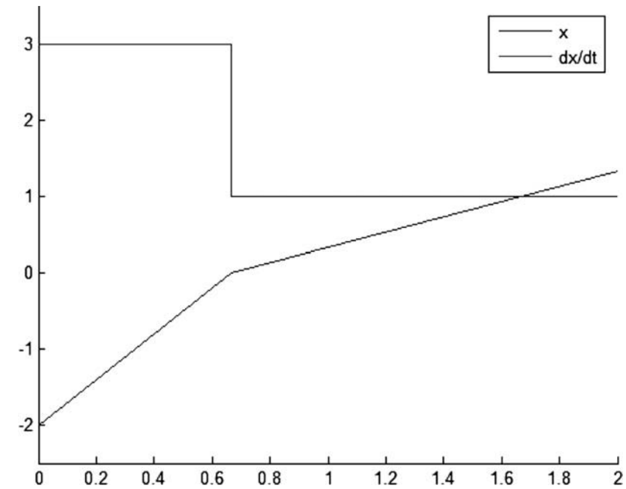

Fig. 1. Solution profiles of hybrid dynamic system.

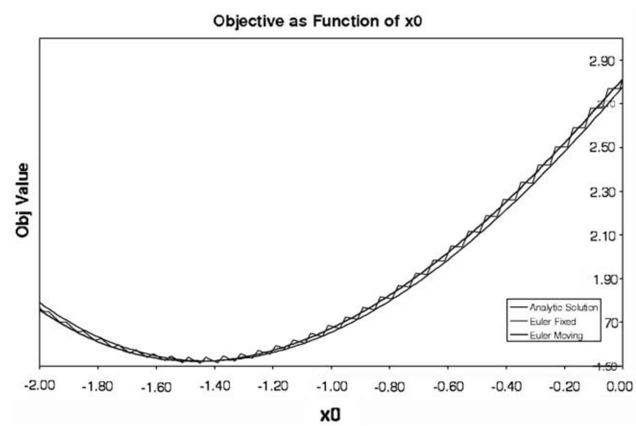

Fig. 2. Sensitivity of objective function with respect to x_0.

4.2. Cascading tank problem

The second case study computes an optimal control trajectory of valve positions for a series of cascading tanks. The problem is made arbitrarily large by changing various model parameters to investigate the growth of solution times. This problem is a modified version of the problem presented in [35], where it was solved using an MILP approach. The system is illustrated in Fig. 3. Here the inlet to the first tank is located at the top of the tank and the inlet for each subsequent tank is located at an intermediate height H_i,

Table 1
Solution times (Pentium 4, 1.8 GHz, 992 MB RAM) comparing MINLP and MPEC formulations.

	MINLP formulation			MPEC formulation		
N	Objective	SBB (CPU s)	DICOPT (CPU s)	Objective	Iters.	CPU s
10	1.5364	0.188	0.905	1.5364	25	0.063
100	1.7581	58.469	> 10000	1.7766	97	0.766
1000	–	> 10000	–	1.7889	698	23.266
2000	–	–	–	1.7895	1345	77.188
3000	–	–	–	1.7897	2009	166.781
4000	–	–	–	1.7898	2705	343.016

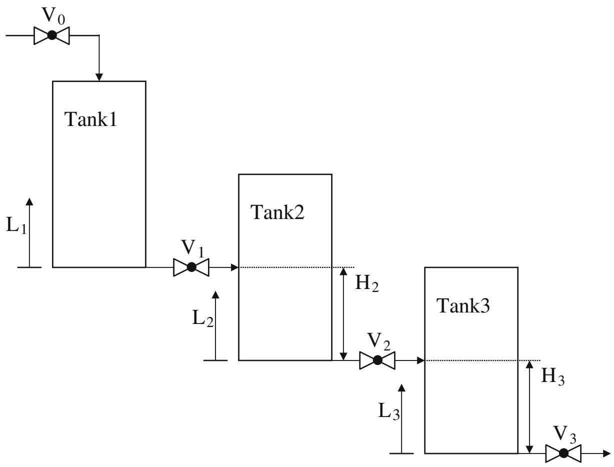

Fig. 3. Diagram of base case system.

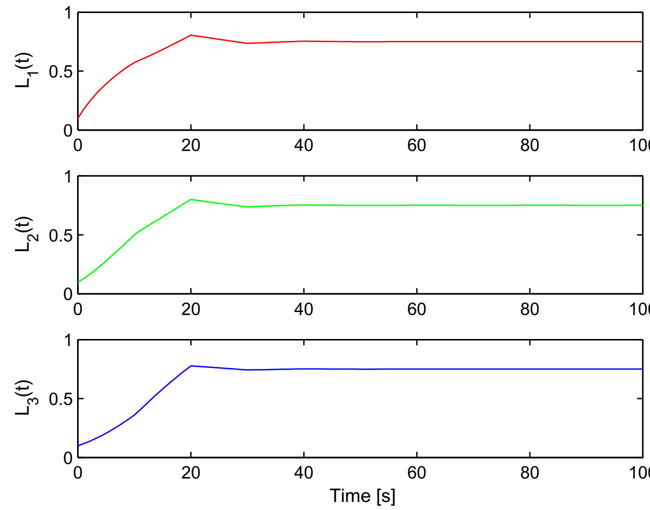

Fig. 4. Plot of state trajectories of tank levels L_i for base case.

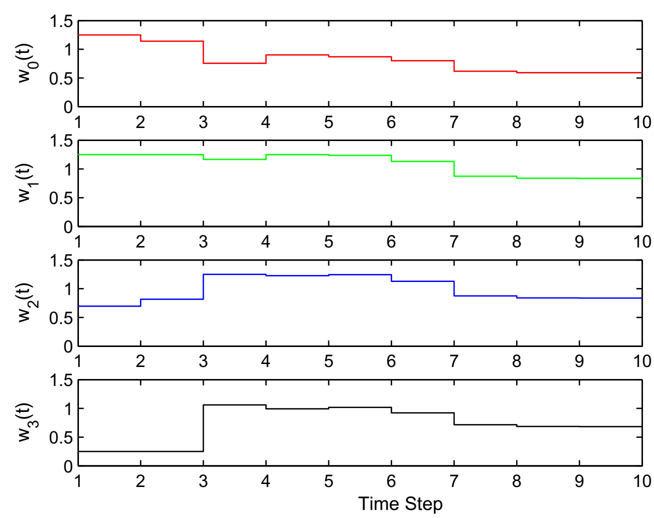

Fig. 5. Plot of optimal control profiles of valve openings $w(t)$.

where $i \in \{1, 2, \ldots, n\}$ is the tank index, n is the total number of tanks and L_i is the level in tank i. The outlet for all tanks is located at the bottom of the tank. The initial feed inlet to the first tank and all outlets are equipped with valves V_i to control the flowrates.

$$\frac{dL_i(t)}{dt} = \frac{1}{A_i}(F_{i-1}(t) - F_i(t)) \quad \forall i \tag{17}$$

$$F_0(t) = w_0(t)k_0 \tag{18}$$

$$F_i(t) = w_i(t)k_i(t)\sqrt{\Delta L_i(t)} \quad \forall i \tag{19}$$

$$\Delta L_i(t) = \begin{cases} L_i(t) - L_{i+1}(t) - H_{i+1} & L_{i+1}(t) > H_{i+1} \\ L_i(t) & L_{i+1}(t) < H_{i+1} \end{cases} \quad i \in \{1, 2, \ldots, n-1\} \tag{20}$$

$$\Delta L_n(t) = L_n(t) \tag{21}$$

Eq. (17) is a material balance equation for each tank, while (18) and (19) are valve flowrate equations describing the flowrate entering the first tank and leaving each tank, respectively. The flowrate out of tank i is modeled using Torricelli's principle. When the level in the next tank L_{i+1} is above its inlet level H_{i+1}, it influences the dynamics of L_i. This leads to the piecewise function (20) describing the difference in tank levels $\Delta L_i(t)$. Additionally, reverse flow (flow from tank i to tank $i-1$) is prohibited. While it appears that the physics of this problem will naturally prevent this, we include $\Delta L_i(t) \geqslant 0$ as an inequality constraint, along the lines of the restriction imposed in [35].

In the formulation in [35] an L_1 penalty was used in the objective function to penalize state deviations from the setpoint and to move the tank levels into the target region [0.7, 0.8] from an initial tank level of 0.1. Here the quadratic penalty is used instead,

$$\phi = \sum_i \int_0^T (L_i(t) - 0.75)^2 dt \tag{22}$$

This is more consistent with a traditional Model Predictive Control application, which our formulation closely resembles. Implicit Euler $(K = 1)$ was used to discretize the differential equations. Since this problem is likely to be solved in an on-line context where solution times must be kept as short as possible, IPOPT was used to solve this set of problems. The proposed methodology can also be extended in a straightforward manner to include control inputs for fixed length time intervals, as used in digital control. Here, an additional set of time intervals and state variables can be defined with continuity enforced for state variables between time intervals.

The model was initially solved with 10 finite elements per time interval, 10 intervals, and 3 tanks. This serves as the base case scenario. The state trajectories for the solution of the base case problem are plotted in Fig. 4, while the corresponding control profiles are given in Fig. 5. The tank levels reach the target region during the first two time intervals and remain in the target region for all subsequent intervals.

From this base case the number of finite elements per time interval, number of time intervals, and the number of tanks were varied independently, with the solution time (Pentium 4, 1.8 GHz, 992 MB RAM) and iteration results given in Tables 2–4, respectively. The solution time grows linearly with increasing the number of tanks, and between linearly and quadratically both with the number of finite elements, and the number of time steps. The increase in solution times is largely the result of the increased cost of the sparse linear factorization of the KKT matrix, which increases linearly with the size of the problem.

The base case solved in 1.703 CPU s. In contrast, the study in [35] used a slightly faster computer (Pentium 4, 2.4 GHz, 1 GB RAM) and solved the equivalent base case in 2.75 CPU s. While our base case result represents an improvement, the major improvement occurs when looking at how the problems scale. In [35] the solution time increases between cubically and exponen-

Table 2
Scaling of solution time with the number of finite elements (N).

N	Time (s)	Iterations
10	1.703	28
20	7.297	42
30	12.125	41
40	23.641	48
50	28.078	43
60	40.219	47
70	46.063	43
80	63.094	45
90	80.734	48
100	90.234	44

Table 3
Scaling of solution time with the number of time intervals (TI).

TI	Time (s)	Iterations
10	1.703	28
20	6.781	36
30	12.031	39
40	26.922	54
50	22.437	34
60	71.703	75
70	90.828	73
80	126.875	84
90	80.891	47

Table 4
Scaling of solution time with the number of tanks.

Tanks	Time (s)	Iterations
3	1.703	28
4	4.157	45
5	9.984	69
6	7.484	40
7	14.968	61
8	11.156	40
9	13.906	41
10	78.703	122
11	20.266	43
12	28.203	52

tially as the number of time steps increases and between quadratically and exponentially as the number of tanks increases, while our proposed methodology increases never more than quadratically. Finally, we note that the effect of both continuous and discrete valve position openings was also considered in [35], while this study is limited to continuous valve position openings.

4.3. Michael Schumacher problem

The third case study computes an optimal control trajectory for a simplified model of a race car that covers a track in minimum time. This problem is a modification of the original "Michael Schumacher" problem contained in CPNET, the Complementarity Problem Network [32] as described in [33]. The state space consists of vectors $\mathbf{x}, \mathbf{v} \in \mathbb{R}^2$ which represent the position and velocity of the car, respectively. The individual components of the position vector are represented as standard Cartesian coordinates, i.e. $\mathbf{x} = \begin{bmatrix} x \\ y \end{bmatrix}$. Also the state $\theta \in \mathbb{R}$ represents the direction of motion, $a(t)$ represents the acceleration and deceleration control and $s(t)$ represents the steering control. Additionally, auxiliary functions $bft(\theta) = [\cos\theta, \sin\theta]^T$ and $\mathbf{n}(\theta) = [-\sin\theta, \cos\theta]^T$ are used, which represent the tangential and normal directions of the car's path of motion, respectively. Fig. 6 shows the direction of acceleration

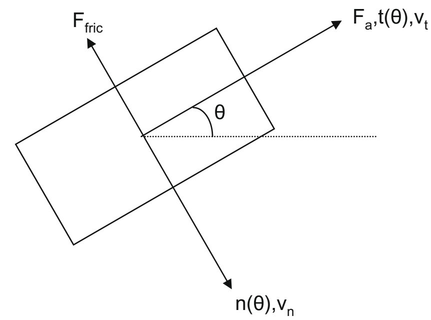

Fig. 6. Acceleration and friction forces acting on the car and their relation to tangential and normal components of the car's velocity.

and friction forces acting on the car as well as their relation to tangential and normal components of the car's velocity.

The optimization problem minimizes T, the total time the car needs to reach the final target position $\mathbf{x}_{tgt}$. This target is imposed as a penalty term in the objective function as opposed to an equality constraint to help ensure feasibility of the problem and maintain consistency with [33]. The force due to friction F is calculated using an inclusion (Eq. 23e), where μ is the product of the coefficient of friction and the normal force. This inclusion sets $F = -\mu$ for positive values of the normal velocity, $\mathbf{n}(\theta)^T\mathbf{v}$, and $F = \mu$ for negative values of the normal velocity. When the normal velocity is zero, F is bounded $[-\mu, \mu]$ and can be determined from differentiability of the normal velocity. In this manner both static and dynamic friction forces are accounted, although more complex friction models could also be considered [20]:

$$\min \quad T + 1000\|\mathbf{x}(T) - \mathbf{x}_{tgt}\|^2 \tag{23a}$$
$$\text{s.t.} \quad \dot{\mathbf{x}} = \mathbf{v} \tag{23b}$$
$$\dot{\mathbf{v}} = a(t)\mathbf{t}(\theta) + F\mathbf{n}(\theta) \tag{23c}$$
$$\dot{\theta} = s(t)(\mathbf{t}(\theta)^T\mathbf{v}) \tag{23d}$$
$$F \in -\mu\, sgn(\mathbf{n}(\theta)^T\mathbf{v}) \tag{23e}$$
$$\mathbf{x}(t) \in C = \{(x,y) \| y - y_{cl}(x)| \leqslant w/2\} \tag{23f}$$
$$y_{cl}(x) = \begin{cases} sin(x) & x \leqslant \pi \\ \pi - x & \pi \leqslant x \leqslant 2\pi \\ -\pi - sin(x) & 2\pi \leqslant x \end{cases} \tag{23g}$$
$$|a(t)| \leqslant a_{max} \quad \forall t \tag{23h}$$
$$|s(t)| \leqslant s_{max} \quad \forall t \tag{23i}$$
$$\mathbf{x}(0) = \mathbf{0} \tag{23j}$$
$$\mathbf{v}(0) = \mathbf{0} \tag{23k}$$
$$\theta(0) = 0 \tag{23l}$$

For this problem $a_{max} = 2, s_{max} = 2, \mathbf{x}_{tgt} = [3\pi, -\pi]^T, \mu = 4$, and $w = 0.5$. Also, the inclusion in Eq. (23e) is reformulated as $F = -\mu\, sgn(\mathbf{n}(\theta)^T\mathbf{v})$, where the *sgn* function is modeled as in the Section 4.1 and in [6]. Eq. (23g) is also reformulated using the piecewise function formulation described in [6].

This problem is significantly more challenging than the prior case study due to the nonlinear differential equations. To ease the solution of this case study, the MPEC problem was solved in three stages. Initially the problem is solved with an implicit Euler discretization for a fixed final time $T = 2\pi$ using IPOPT. IPOPT was used because of its strong global convergence properties and fast performance. This provides a feasible initialization to the second stage, where we maintain an implicit Euler discretization and optimize the problem with CONOPT, but without fixing the final time. CONOPT, an active set method, has advantages in identifying and dealing with degenerate active constraints and is also significantly aided by the feasible initialization. For 100 finite elements,

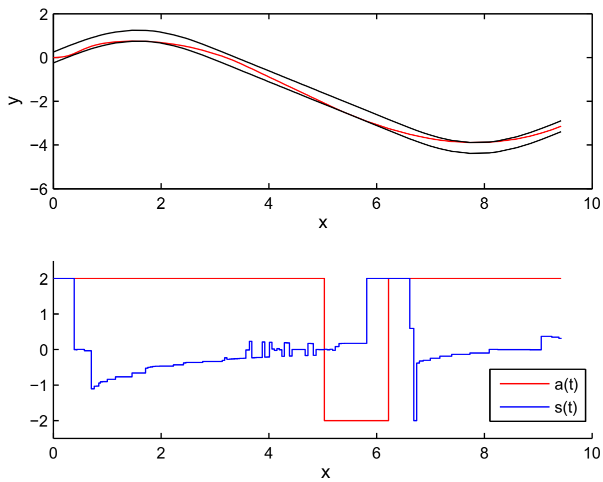

Fig. 7. Plot of optimal trajectory (above) and optimal control profiles (below) for Schumacher problem.

this provides a low fidelity solution to the problem. In the third stage, the problem is solved again using CONOPT and the prior stage as an initialization, but with 3 point collocation, instead of implicit Euler. This last stage significantly improves the integration error.

Fig. 7 shows the optimal solution to the problem and displays the path of the car through the track and the control profiles. Here we depart from standard convention and give the control profiles as a function of the car's position. This makes interpretation of the controls more intuitive as they may be directly compared against the path of the car on the track. The car initially performs a hard steering maneuver to the left in order to stay on the track. Then the car steers straight until it comes to the first bend in the track. At that point the car makes a sharp turn to the right that gradually decreases. As the car approaches the second bend, the car makes a hard left that it holds through the turn. Initially, the car maintains acceleration until just before the car reaches the second turn, when the car brakes after the car enters the turn. After this single braking maneuver, the car returns to full acceleration until reaching the end of the track.

To assess the accuracy of the optimal solution, we compared the errors in final position predicted by the optimizer to the final position calculated from a simulation of the model using the optimizer's control profile. The simulation was performed using the error controlled integration method RK45 in Matlab. Because the integration error is estimated numerically, the best expected error would be the square root of machine precision, approximately 10^{-8}. Table 5 gives the estimated error for increasing number of finite elements (N). Error(a,s,u) is the error in final position by considering control profiles $a(t)$ and $s(t)$ at their optimal values. Furthermore, we treat the calculated values of $u(t) = sgn(\mathbf{n}(\theta)^T\mathbf{v})$ as fixed. This allows us to integrate the dynamic system as a set of ODEs instead of an index 2 DAE. The integration error ranges from 2.31×10^{-4} with $N = 100$ to 4.50×10^{-8} for $N = 1600$. For comparison, the L_1 penalty is the estimated error in the complementarity calculation of the *sgn* function, assuming accurate integration.

We also consider the solution accuracy over the time period when the normal velocity $\mathbf{n}(\theta)^T\mathbf{v}$ is zero in (23e). To assess whether we have identified the index-2 arc correctly, we consider the property of the guard function in (4). As a result, we examine this constraint by differentiating the normal velocity, setting the resulting expression to zero, and applying the differential equations in (23) to obtain:

Table 5
Error in final position of car for increasing number of finite elements (N).

N	Objective value	L_1 penalty	Error(a,s,u)	Error for Index-2 Arc
100	4.0209	0.0	2.31×10^{-4}	8.80×10^{-3}
200	4.0172	0.0	8.58×10^{-6}	3.53×10^{-3}
400	4.0178	0.0	2.16×10^{-7}	8.68×10^{-4}
800	4.0174	0.0	1.09×10^{-7}	2.20×10^{-4}
1600	4.0169	9.85×10^{-10}	4.50×10^{-8}	8.07×10^{-5}

$$0 = \dot{\mathbf{n}}(\theta)^T\mathbf{v} + \mathbf{n}(\theta)^T\dot{\mathbf{v}} = -\mathbf{t}(\theta)^T\mathbf{v}\dot{\theta} + \mathbf{n}(\theta)^T\dot{\mathbf{v}} \tag{24}$$

$$\Rightarrow F = (\mathbf{t}(\theta)^T\mathbf{v})\dot{\theta}(t) = (\mathbf{t}(\theta)^T\mathbf{v})^2 s(t). \tag{25}$$

For the MPEC solution we evaluate the residual error (using the L_1 norm) in this expression at collocation points where $|\mathbf{n}(\theta)^T\mathbf{v}| < 10^{-8}$. As shown in Table 5, this "index 2" error is only slightly greater than the estimated integration error for 100 finite elements and it decreases to 8.07×10^{-5} at 1600 elements. Finally, the results in Table 5 are consistent with the index-2 order properties of Radau collocation [2] and indicate an excellent approximation to the index-2 arc in this solution.

Finally, we compare this solution with the one reported in [33], where the minimum track time was 5.34341 s. In [33], the authors apply smoothing functions to the nonsmooth terms in order to solve the discretized MPEC problem. They also provide a detailed analysis to ensure convergence to the solution of the original problem. In particular, they require the finite element grid to converge at a faster rate ($h_i \to 0$) than convergence of the smoothing parameters to their asymptotic values.

Our result with a minimum track time of 4.0169 s (for the most accurate run) is clearly a significant improvement; we believe at least two possible reasons may exist for this improvement. First, it is possible that the difference in results is due to finding different local solutions of this nonconvex optimization problem. Second, the result in [33] is based on approaching a solution asymptotically using a smoothing method. Their results are given for the tightest smoothing approximation considered in [33]. It is possible that the tightest smoothing approximation is not sufficiently tight to accurately recover the true solution. On the other hand, the ℓ_1 penalty formulation, $PF(\rho)$, is exact with respect to satisfying the complementarity constraints.

5. Conclusions

We have proposed an MPEC-based formulation to handle a class of hybrid dynamic optimization problems, where the differential states are continuous over time. These include differential inclusions of the Filippov type. Our approach differs from prior work using direct transcription [33,3] because it identifies the location of switching events using only continuous variables. The method is able to do this by allowing the finite elements to move and by enforcing only one branch of the disjunction to be active across an entire finite element. Any nonsmooth behavior from a change in the disjunctions is thus forced to occur only at the finite element boundaries. Furthermore, high order IRK discretization is used to ensure accurate solutions in each finite element. The case studies illustrate the effectiveness and accuracy of this methodology.

In the first case study, it was shown that if the lengths of the finite elements are fixed the problem will exhibit non-smoothness in the parameter sensitivities, while moving finite elements lead to smooth solutions. This study also demonstrated that the solution times grow polynomially for the proposed MPEC method. If an equivalent MINLP formulation is used that also identifies the switching locations, its solution time grows exponentially.

In the second case study an optimal control profile was determined for a set of cascading tanks. It was shown that the solution time grows no worse then quadratically for the three parameters varied: number of tanks, number of time steps and number of finite elements per time step. This was contrasted with an MILP approach that grew between quadratically and exponentially as the number of tanks and time steps increased. While the MILP strategy is more general, the proposed MPEC approach was shown to be significantly faster for this restricted problem class.

The third case study is the "Michael Schumacher" problem which exhibits a index-2 arc as part of the solution profile. It was shown that the proposed solution methodology was able to find a better solution, 4.0169 s, than the previously reported solution of 5.34341 s. Furthermore, integration error estimates indicate that the problem was solved to an accuracy of 4.5×10^{-8} with 1600 finite elements and 3 point collocation, and the error in the index-2 arc portion of the control problem was estimated to 8.07×10^{-5}, indicating that the arc was correctly identified.

Future work will deal with recovering accurate high index solutions, exploiting structure and developing faster solutions strategies for MPECs. Additionally, we intend to extend these formulations to on-line applications, particularly for NMPC strategies. Finally, we plan to tackle larger and more challenging applications in process control and hybrid systems optimization.

Acknowledgement

Funding from the National Science Foundation under Grants CTS-0438279 and CTS-0457379 is gratefully acknowledged.

References

[1] M. Anitescu, P. Tseng, S.J. Wright, Elastic-mode algorithms for mathematical programs with equilibrium constraints: global convergence and stationarity properties, Math. Program. 110 (2007) 337–371.
[2] U.M. Ascher, L.R. Petzold, Computer Methods for Ordinary Differential Equations and Differential-Algebraic Equations, SIAM, Philadelphia, PA, 1998.
[3] M.P. Avraam, N. Shah, C.C. Pantelides, Modelling and optimisation of general hybrid systems in the continuous time domain, Comput. Chem. Eng. 22 (1998) S221–S228.
[4] P.I. Barton, C.K. Lee, M. Yunt, Optimization of hybrid systems, Comput. Chem. Eng. 30 (2006) 1576–1589.
[5] P.I. Barton, J.R. Banga, S. Gala, Optimization of hybrid discrete:continuous dynamic systems, Comput. Chem. Eng. 24 (2000) 2171–2182.
[6] B.T. Baumrucker, L.T. Biegler, MPEC problem formulations and solution strategies with chemical engineering applications, Comput. Chem. Eng. 32 (2008) 2903–2913.
[7] A. Drud, CONOPT Solver Manual, GAMS Development Corporation, Washington, DC, 2004.
[8] CPLEX 10 Solver Manual, GAMS Development Corporation, Washington, DC, 2004.
[9] I.E. Grossmann, J. Viswanathan, A. Vecchietti, R. Raman, E. Kalvelagen, DICOPT Solver Manual, GAMS Development Corporation, Washington, DC, 2004.
[10] A. Dontchev, F. Lempio, Difference methods for differential inclusions: a survey, SIAM Rev. 34 (1992) 263–294.
[11] A.F. Filippov, Differential equations with discontinuous right-hand side, Matematicheskii Sbornik Novaya Seriya 5 (1960) 99–127; A.F. Filippov, Differential equations with discontinuous right-hand side, Am. Math. Soc. Transl. 42 (1964) 199–231.
[12] R. Fletcher, S. Leyffer, Solving mathematical programs with complementarity constraints as nonlinear programs, Optim. Methods Softw. 19 (1) (2004) 15–40.
[13] GAMS: The Solver Manuals, GAMS Development Corporation, Washington, DC, 2004.
[14] X.M. Hu, D. Ralph, Convergence of a penalty method of mathematical programming with complementarity constraints, J. Optim. Theory Appl. 123 (2) (2004) 365–390.
[15] M. Heemels, Ph.D. Thesis, Technische Universiteit Eindoven, 1999.
[16] M. Heemels, B. DeSchutter, A. Bemporad, On the equivalence of hybrid dynamical models, In: 40th IEEE Conference on Decision and Control, 2001, pp. 364–369.
[17] W.P.M.H. Heemels, B. Brogliato, The complementarity class of hybrid dynamical systems, Eur. J. Cont. 9 (2–3) (2003) 322–360.
[18] S. Kameswaran, G. Staus, L.T. Biegler, Parameter estimation of core flood and reservoir models, Comput. Chem. Eng. 29 (8) (2005) 1787–1800.
[19] S. Kameswaran, Lorenz T. Biegler, Simultaneous dynamic optimization strategies: recent advances and challenges, Comput. Chem. Eng. 30 (2006) 1560–1575.
[20] R.I. Leine, H. Nijmeijer, Dynamics and bifurcations of non-smooth mechanical systems, Lecture Notes in Applied and Computational Mechanics 18 (2006).
[21] S. Leyffer, G. Lopez-Calva, J. Nocedal, Interior methods for mathematical programs with complementarity constraints, SIAM J. Optim. 17 (1) (2006) 52–77.
[22] J. Oldenburg, W. Marquardt, Disjunctive modeling for optimal control of hybrid systems, Comput. Chem. Eng. 32 (10) (2008) 2346–2364.
[23] J.S. Pang, D. Stewart, Differential variational inequalities, Math. Program. 113 (2) (2008) 345–424.
[24] A. Raghunathan, L.T. Biegler, MPEC formulations and algorithms in process engineering, Comput. Chem. Eng. 27 (2003) 1381–1392.
[25] A. Raghunathan, Ph.D. Thesis, Department of Chemical Engineering, Carnegie Mellon University, 2004.
[26] A.U. Raghunathan, L.T. Biegler, Interior point methods for mathematical programs with complementarity constraints (MPCCs), SIAM J. Optim. 15 (3) (2005) 720–750.
[27] A. Raghunathan, M.S. Diaz, L.T. Biegler, An MPEC formulation for dynamic optimization of distillation operation, Comput. Chem. Eng. 28 (10) (2004) 2037–2052.
[28] A.U. Raghunathan, J.R. Perez-Correa, E. Agosin, L.T. Biegler, Parameter estimation in metabolic flux balance models for batch fermentation – formulations and solution using differential variational inequalities, Ann. Oper. Res. 148 (2006) 251–270.
[29] D. Ralph, S.J. Wright, Some properties of regularization and penalization schemes for MPECs, Optim. Methods Softw. 19 (5) (2004) 527–556.
[30] SBB Solver Manual, GAMS Development Corporation, Washington, DC, 2004.
[31] O. Stein, J. Oldenburg, W. Marquardt, Continuous reformulations of discrete-continuous optimization problems, Comput. Chem. Eng. 28 (2004) 1951–1966.
[32] D.E. Stewart, The Michael Schumacher problem, June 1999. <http://www.cs.wisc.edu/cpnet/cpnetmeetings/iccp99/race-car/race-car.html>.
[33] D. Stewart, M. Anitescu, Optimal Control of Systems with Discontinuous Differential Equations, Preprint number: ANL/MCS-P1258-0605, Mathematics and Computer Science Division, Argonne National Laboratory, June 2005.
[34] O. Stursberg, S. Panek, J. Till, S. Engell, Generation of optimal control policies for systems with switched hybrid dynamics, Model. Anal. Des. Hybrid Syst. 279 (2002) 337–352.
[35] J. Till, S. Engell, S. Panek, O. Stursberg, Applied hybrid system optimization: an empirical investigation of complexity, Control Eng. Pract. 12 (2004) 1291–1303.
[36] S. Veelken, M. Ulbrich, A new regularization method for MPECs, presented at Second Mathematical Programming Society International Conference on Continuous Optimization, Hamilton, Ont., Canada, August, 2007.
[37] A. Wächter, L.T. Biegler, On the implementation of an interior point filter line search algorithm for large-scale nonlinear programming, Math. Program. 106 (1) (2006) 25–57.