Nonlinear Analysis: Hybrid Systems 54 (2024) 101518

Contents lists available at ScienceDirect

Nonlinear Analysis: Hybrid Systems

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

Finite Elements with Switch Detection for numerical optimal control of nonsmooth dynamical systems with set-valued heaviside step functions☆

Armin Nurkanović [a,*], Anton Pozharskiy [a], Jonathan Frey [a,b], Moritz Diehl [a,b]

[a] *Department of Microsystems Engineering (IMTEK), University of Freiburg, Germany*
[b] *Department of Mathematics, University of Freiburg, Germany*

ARTICLE INFO

Keywords:
Numerical simulation
Filippov systems
Numerical optimal control
Discontinuous differential equations
Differential inclusions

ABSTRACT

This paper develops high-accuracy methods for the numerical solution of optimal control problems subject to nonsmooth differential equations with set-valued Heaviside step functions. An important subclass of these systems are Filippov systems. By writing the Heaviside step function as the solution map of a linear program and using its optimality conditions, the initial nonsmooth system is rewritten into an equivalent Dynamic Complementarity System (DCS). The Finite Elements with Switch Detection (FESD) method (Nurkanović et al., 2024) was originally developed for Filippov systems transformed via Stewart's reformulation into DCS (Stewart, 1990). This paper extends it to the above mentioned class of nonsmooth systems. The key ideas are to start with a standard Runge–Kutta method for the DCS and to let the integration step sizes to be degrees of freedom. Then, additional conditions are introduced to allow implicit but accurate switch detection and to remove possible spurious degrees of freedom when no switches occur. The theoretical properties of the FESD method are studied. The motivation for these developments is to obtain a computationally tractable formulation of nonsmooth optimal control problems. Numerical simulations and optimal control examples are used to illustrate the favorable properties of the proposed approach. All methods introduced in this paper are implemented in the open-source software package `nosnoc` (Nurkanović and Diehl, 2022).

1. Introduction

The set-valued version of the Heaviside step function, denoted as $\gamma : \mathbb{R} \to \mathcal{P}(\mathbb{R})$, is defined as follows:

$$\gamma(y) = \begin{cases} \{1\}, & y > 0, \\ [0,1], & y = 0, \\ \{0\}, & y < 0. \end{cases} \tag{1}$$

Here $\mathcal{P}(\mathbb{R})$ represents the power set of $\mathbb{R}$. The set-valued Heaviside step function, often referred to simply as the step function, provides an intuitive way to model Boolean relationships within a dynamical system.

☆ This research was supported by the DFG via Research Unit FOR 2401 and project 424107692, by the EU via ELO-X 953348, by the German Federal Ministry for Economic Affairs and Climate Action (BMWK.IIC6) via the project WOpS with the project number 03EN3054A.

* Corresponding author.

E-mail addresses: armin.nurkanovic@imtek.uni-freiburg.de (A. Nurkanović), anton.pozharskiy@imtek.uni-freiburg.de (A. Pozharskiy), jonathan.frey@imtek.uni-freiburg.de (J. Frey), moritz.diehl@imtek.uni-freiburg.de (M. Diehl).

https://doi.org/10.1016/j.nahs.2024.101518
Received 30 June 2023; Received in revised form 23 May 2024; Accepted 1 June 2024
Available online 8 June 2024

In the modeling process, *switching functions* $\psi_i : \mathbb{R}^{n_x} \to \mathbb{R}$, for $i = 1, \dots, n_\psi$, are commonly used as arguments of the step functions. We denote the concatenation of all scalar Heaviside step functions as $\Gamma(\psi(x)) := (\gamma(\psi_1(x)), \dots, \gamma(\psi_{n_\psi}(x)))$, where $\psi(x) = (\psi_1(x), \dots, \psi_{n_\psi}(x))$.

In this paper, we study nonsmooth dynamical systems of the form:

$$\dot{x} \in \mathcal{F}(x, u, \Gamma(\psi(x))). \tag{2}$$

Here, $u \in \mathbb{R}^{n_u}$ represents a known control function that could be obtained, for example, by solving an optimal control problem. The Eq. (2) is an instance of a differential inclusion (DI) since the right-hand side is set-valued due to the function Γ, which can enter $\mathcal{F}$ in a nonlinear way. Thus, the set $\mathcal{F}(x, u, \Gamma(\psi(x)))$ can be convex or nonconvex.

The nonsmooth and set-valued nature of $\mathcal{F}(x, u, \Gamma(\psi(x)))$ in the DI (2) introduces complexities in its numerical treatment that we address in this paper. In particular, we develop accurate and efficient numerical methods for solving simulation and optimal control problems subject to such DIs. An important subclass of nonsmooth systems with Heaviside step functions are Filippov systems [1–3]. Moreover, some classes of systems with state jumps can be reformulated into Filippov systems via the time-freezing reformulation [4–8]. Therefore, this modeling approach enables us to treat a broad class of practical problems.

In a Piecewise Smooth System (PSS), the state space is split into nonempty regions where each of them is equipped with a different vector field. Step functions are used in PSS modeling to determine in which region or on what boundaries the trajectory is. Smoothed single-valued versions of the Heaviside step function are often used in the numerical treatment of a PSS [9–11]. A prominent application example of step functions are gene-regulatory networks [1,11]. Another common application of step functions is to compute selections of Filippov sets in sliding modes [2,12]. The event of $x(t)$ becoming nondifferentiable is called a *switch*. In our case, this corresponds that one or more components of $\psi(x)$ become zero, or if they were zero, they become nonzero.

Many mature numerical simulation methods to treat ODEs with switches exist, and the corresponding theory is well-established [13]. However, numerical methods for solving Optimal Control Problems (OCPs) subject to nonsmooth dynamical systems are not yet at such a mature stage. The key ingredient to high-accuracy methods and efficient numerical optimal control is to detect the time points when the switches occur [14]. In the control community, a popular approach to deal with the nonsmoothness is to introduce integer variables to label all modes of the nonsmooth system [15]. This leads to mixed-integer optimization problems for solving OCPs. However, they often become computationally intractable when nonconvexities appear or exact switching times need to be computed.

On the one hand, the usually computationally less favorable indirect methods for optimal control are not widely used, since Pontryagin-like conditions are not established for many classes of nonsmooth systems, cf. [16,17] for an overview. On the other hand, the application of direct methods, *i.e.*, in a first-discretize-then-optimize approach, is not as straightforward as for OCPs with smooth ODEs and can lead to spurious solutions and wrong conclusions. This was first rigorously explained in the seminal paper of Stewart and Anitescu [18].

They show that the derivatives of an integration map obtained by time-stepping methods (*e.g.*, the implicit Euler method) with respect to initial values and controls (called numerical sensitivities) do not converge to corresponding continuous-time values (called sensitivities), no matter how small the integrator step size is. In practice, this can result in termination at feasible, but nonoptimal points [19]. Moreover, they show that the numerical sensitivities of the smoothed approximations of a nonsmooth system are only correct if the step size shrinks faster than the smoothing parameter. This makes the smoothing approach impractical, as very small step sizes are needed even for moderate accuracy.

The limitations of direct methods based on time-stepping were recently overcome by the method of Finite Elements with Switch Detection (FESD) [14]. This method is based on standard Runge–Kutta (RK) discretizations of the Dynamic Complementarity systems (DCS), where the integrator step sizes are left as degrees of freedom as first proposed by Baumrucker and Biegler [20]. Additional constraints are introduced to have a well-defined system with exact switch detection. The discretization yields Mathematical Programs with Complementarity Constraints (MPCC). They are nonregular and nonsmooth Nonlinear Programs (NLP) [21,22]. Still, with suitable reformulations and homotopy procedures, they can be solved efficiently using techniques from smooth optimization.

Initially, we developed FESD for Stewart's reformulation of Filippov systems into DCS [23]. In this paper, we extend these ideas to nonsmooth systems of the form of (2) and corresponding OCPs. This enables us to cover a more general class of nonsmooth ODEs with a discontinuous r.h.s.

Contributions. We provide a detailed study of the transformation of a Filippov system into a Dynamic Complementarity System (DCS) via set-valued Heaviside step functions. The key step in this transformation is to view the Heaviside step function as the solution of a parametric linear program and to use its optimality conditions. Furthermore, if the active set in the DCS is fixed, we obtain locally a smooth ODE or Differential Algebraic Equation (DAE). We study the well-posedness of these systems. In the theoretical analysis and algorithmic development, we focus on the Filippov systems. Simple tutorial examples accompany all developments. Most importantly, we present the extension of the FESD method to nonsmooth systems that are described with step functions. We also adapt the convergence and well-posedness results of [14] to this case.

If Filippov systems are modeled via step functions, one ends up with multi-affine expressions, consisting of products of step functions, in the r.h.s. of the ODE. We propose a lifting algorithm that introduces auxiliary variables and makes these expressions "less nonlinear". This can improve the convergence of the proposed method [24].

The dynamical system (2) is more general than the Filippov system. Therefore, the FESD method developed here applies to this broader class of systems. The performance of the new method is compared to the original FESD [14] and standard RK discretizations in terms of accuracy and computational time. All methods are implemented in the open-source software package `nosnoc` [25].

Outline. In Section 2, we define and relate various classes of nonsmooth systems, including Dynamic Complementarity Systems (DCS), Filippov systems, and systems with set-valued Heaviside step functions. It is also demonstrated that the latter is equivalent to a DCS. Section 3 explores the properties of this DCS for a fixed active set and during active-set changes. In Section 4, we introduce the FESD method for this class of problems. Section 5 presents convergence and well-posedness results for the new method. Additionally, in Section 6, we demonstrate how to efficiently model piecewise smooth systems with step functions and introduce a lifting algorithm to reduce nonlinearity in the DCS.

In Section 7, we showcase the developments using several numerical examples.

Notation. The complementarity conditions for two vectors $a, b \in \mathbb{R}^n$ read as $0 \leq a \perp b \geq 0$, where $a \perp b$ means $a^\top b = 0$. For two scalar variables a, b the so-called C-functions [26] have the property $\phi(a, b) = 0 \iff a \geq 0, b \geq 0, ab = 0$. Examples are the natural residual functions $\phi_{\mathrm{NR}}(a, b) = \min(a, b)$ or the Fischer–Burmeister function $\phi_{\mathrm{FB}}(a, b) = a + b - \sqrt{a^2 + b^2}$. If $a, b \in \mathbb{R}^n$, we use $\phi(\cdot)$ component-wise and define $\Phi(a, b) = (\phi(a_1, b_1), \ldots, \phi(a_n, b_n))$. All vector inequalities are to be understood element-wise, $\mathrm{diag}(x) \in \mathbb{R}^{n\times n}$ returns a diagonal matrix with $x \in \mathbb{R}^n$ containing the diagonal entries. The concatenation of two column vectors $a \in \mathbb{R}^{n_a}$, $b \in \mathbb{R}^{n_b}$ is denoted by $(a, b) := [a^\top, b^\top]^\top$, the concatenation of several column vectors is defined analogously. A column vector with all ones is denoted by $e = (1, 1, \ldots, 1) \in \mathbb{R}^n$ and its dimension is clear from the context. The closure of a set C is denoted by $\overline{C}$, its boundary as ∂C.

For a given vector $a \in \mathbb{R}^n$ and set $\mathcal{I} \subseteq \{1, \ldots, n\}$, we define the projection matrix $P_{\mathcal{I}} \in \mathbb{R}^{|\mathcal{I}|\times n}$, which has zeros or ones as entries. It selects all component $a_i, i \in \mathcal{I}$ from the vector a, *i.e.*, $a_{\mathcal{I}} = P_{\mathcal{I}} a \in \mathbb{R}^{|\mathcal{I}|}$ and $a_{\mathcal{I}} = [a_i \mid i \in \mathcal{I}]$.

Given a matrix $M \in \mathbb{R}^{n\times m}$, its ith row is denoted by $M_{i,\bullet}$ and its jth column is denoted by $M_{\bullet,j}$. For the left and right limits, we use the notation $x(t_{\mathrm{s}}{}^+) = \lim_{t\to t_{\mathrm{s}},\ t>t_{\mathrm{s}}} x(t)$ and $x(t_{\mathrm{s}}{}^-) = \lim_{t\to t_{\mathrm{s}},\ t<t_{\mathrm{s}}} x(t)$, respectively. For an overview of the notation in this paper see Table 1.

2. Overview and relations between different classes of nonsmooth dynamical systems

This section defines, reviews, and relates several classes of nonsmooth dynamical systems relevant to the algorithmic development in this paper. The class of Dynamic Complementarity Systems (DCS) is defined in Section 2.1. In Section 2.2, generic ODEs with a Discontinuous Right-Hand Side (DRHS) and their Filippov extension are considered. In Section 2.3, these results are applied to a more structured ODE with DRHS, namely to piecewise smooth systems. This is followed by Section 2.4, where another class of structured ODEs with DRHS, *i.e.*, nonsmooth systems where Heaviside step functions appear in the dynamics. This class is the focus of this paper, but it is closely related to other classes that we review. Section 2.5 relates these systems to Filippov and DCSs. The key tool is to rewrite the set-valued Heaviside step function in terms of the optimality conditions of an appropriate linear program. For comparison, we review Stewart's way of rewriting a Filippov system as a DCS in Section 2.6. We conclude with Section 2.7, where we summarize the relations between all system classes regarded in this section.

This section assumes that a continuous control function $u : [0, T] \to \mathbb{R}^{n_u}$ is given. Discontinuous $u(t)$ can be considered by partitioning the considered time interval $[0, T]$ into pieces where $u(t)$ is locally continuous, and considering a sequence of ODEs with continuous control functions.

2.1. Dynamic complementarity systems

A Dynamic Complementarity System (DCS) [27,28] is the problem:

$$\dot{x}(t) = f(x(t), y(t), z(t), u(t)), \tag{3a}$$

$$0 = g_{\mathrm{e}}(x(t), y(t), z(t)), \tag{3b}$$

$$0 \leq y(t) \perp g_{\mathrm{c}}(x(t), y(t), z(t)) \geq 0 \text{ for almost all } t, \tag{3c}$$

where $x \in \mathbb{R}^{n_x}$ are the differential states, $y \in \mathbb{R}^{n_y}$ and $z \in \mathbb{R}^{n_z}$ are the algebraic states of the DCS. The DCS consists of an ODE (defined by the function $f : \mathbb{R}^{n_x} \times \mathbb{R}^{n_y} \times \mathbb{R}^{n_z} \times \mathbb{R}^{n_u} \to \mathbb{R}^{n_x}$ in the right-hand side) coupled with algebraic equality constraints (3b) (defined by $g_{\mathrm{e}} : \mathbb{R}^{n_x} \times \mathbb{R}^{n_y} \times \mathbb{R}^{n_z} \to \mathbb{R}^{n_z}$) and complementarity constraints (3c) (defined by $y : [0, T] \to \mathbb{R}^{n_y}$ and $g_{\mathrm{c}} : \mathbb{R}^{n_x} \times \mathbb{R}^{n_y} \times \mathbb{R}^{n_z} \to \mathbb{R}^{n_y}$). We assume that the problem functions f, g_{e} and g_{c} are at least twice continuously differentiable in all arguments. In the complementarity conditions (3c), both components y_i and $g_{\mathrm{c},i}(x, y, z)$ have to be non-negative for $i = 1, \ldots, n_y$, but only one of them is allowed to be strictly positive at a time t. These conditions encode combinatorial structure in the problem and make it nonsmooth. In contrast to many other nonsmooth system formulations, complementarity conditions can be efficiently treated with Newton-type methods, which makes them attractive from a computational point of view [26].

The formulation in Eq. (3) is very general and further assumptions on f, g_{e} and g_{c} determine the existence, uniqueness, and qualitative properties of x, y and z. Several important models fit into the form of (3), most prominently rigid bodies with friction and impact [27,29] and electric circuits with electronic devices [30]. We refer the reader to [27–29] for an extensive collection of theoretical results and application examples.

If one writes (3c) equivalently via a C-function Φ, *i.e.*, $\Phi(y, g_{\mathrm{c}}(x, y, z)) = 0$, the DCS (3) can be seen as a nonsmooth Differential Algebraic Equation (DAE). The DCS is an instance of a more general class of problems called Differential Variational Inequalities (DVI), formally introduced by Pang and Stewart [28]. One way to classify DCS (and DVIs) is their *index* [28], that is, how many times $g_{\mathrm{c}}(x, y, z) = 0$ must be differentiated with respect to time t to find z and y as a function of x. A similar concept is the *relative degree*, cf. [27, Appendix C]. Index zero DVIs (and DCS) have continuously differentiable solutions [28, Proposition 5.1]. The

Table 1
Key symbols used in this paper.

Symbol	Description
x	differential state, Eq. (8)
u	control function, Eq. (8)
θ	Filippov's convex multipliers
S	sign matrix defining regions R_i, (9)
α	selection of set-valued step function, (20)
λ^{p}	Lagrange multiplier in step DCS, (22)
λ^{n}	Lagrange multiplier in step DCS, (22)
β	Lifting variable in the step DCS, Section 6.3
$\psi(x)$	switching functions, Section 2.5.2
$\gamma(\psi_j(x))$	set-valued Heaviside step function, Eq. (1)
$\Gamma(\psi(x))$	concatenation of all regarded step functions , Eq. (1)
$f_i(x,u)$	ith mode of the total n_f modes of the PSS system, Eq. (8)
$F(x,u)$	matrix collecting all PSS modes, F(x,u) $= [f_1(x,u),\dots,f_{n_f}(x,u)] \in \mathbb{R}^{n_x \times n_f}$, (22)
$G(x,\theta,\alpha,\lambda^{\mathrm{p}},\lambda^{\mathrm{n}})$	algebraic eq. in the step DCS as nonsmooth DAE (23)
$g(x)$	Stewart's indicator function, (25)
$W_{\mathcal{K},\mathcal{I}}(x,u)$	auxiliary matrix used in the study of the DCS (22), Section 3.2
$B_{\mathcal{K},\mathcal{I}}(x,u)$	auxiliary matrix used in the study of the DCS (22), Section 3.2
R_i	regions of the PSS, Eq. (8)
$\tilde{R}_i$	base sets, Definition 1
$\mathcal{J}$	index set of PSS modes, Eq. (8)
$\mathcal{F}(x,u,\Gamma(\psi(x)))$	r.h.s. of the nonsmooth ODE with set-valued Heaviside step functions, Eq. (2)
$\mathcal{F}_{\mathrm{F}}(x,u)$	r.h.s. of the Filippov differential inclusion, Eqs. (6), (11)
$\mathcal{F}_{\mathrm{AP}}(x,u)$	r.h.s. of the Aizerman–Pyatnitskii differential inclusion, Eq. (14)
$\mathcal{F}_{\mathrm{H}}(x,u)$	Filippov set obtained with set-valued Heaviside step functions, Eq. (20)
$\mathcal{I}(\cdot)$	active set for Filippov systems, Eq. (12)
$\mathcal{T}_n$	$n-$th time interval with fixed active set $\mathcal{T}_n = (t_{\mathrm{s},n}, t_{\mathrm{s},n+1})$
$\mathcal{I}_n$	the fixed active set $\mathcal{I}_n = \mathcal{I}(x(t)), t \in I_n$
$\mathcal{I}_n^0$	active set at $t_{\mathrm{s},n}$, *i.e.* $\mathcal{I}_{n,0} = \mathcal{I}(t_{\mathrm{s},n})$
$\mathcal{C}$	index set of all switching functions $\psi_i(x)$
$\mathcal{K}$	index set of all switching functions $\psi_i(x)$ that are zero for a given $\mathcal{I}$, Section 3.2

existence and uniqueness of solutions of index zero systems are proven in [28]. Index one DVIs have in general absolutely continuous solutions [29]. Uniqueness for some index one DVIs is proven by Stewart [31]. Index two systems are systems with state jumps [29, Chapter 6]. For index two DVIs only existence can be proven [32]. In general, higher index DVIs may not have solutions [29], and if they exist they are distributions [27]. In this paper, we are interested in DCS where $x(t)$ and $y(t)$ are continuous functions of time, and $z(t)$ is allowed to be discontinuous.

2.2. Discontinuous ODEs and Filippov systems

Consider an ODE with a discontinuous right-hand side (DRHS):

$$\dot{x}(t) = f(x(t), u(t)), \tag{4}$$

with a given initial value $x(0) = x_0$. More precisely, f is discontinuous in x but continuous in u. Classical solutions $x(t)$ to (4) on an interval $[0,T]$ are continuously differentiable, which is impossible with a discontinuous f. Carathéodory solutions of an ODE satisfy the ODE in integral form, *i.e.*, $x(t) = x(0) + \int_0^t f(x(t), u(t))\mathrm{d}t$ for $t > 0$, where the integral is the Lebesgue integral. This allows discontinuities of f in x and in some situations, a Carathéodory solution might be of use, as x is absolutely continuous, and it requires (4) to be satisfied for almost all $t \in [0,T]$. However, this relaxation is not always sufficient as can be seen from the following example [33, Example 5]:

$$\dot{x} = \begin{cases} 1, & x < 0, \\ -1, & x \geq 0, \end{cases} \tag{5}$$

with $x(0) = x_0$. For $x_0 > 0$, there exist the solution $x(t) = x(0) - t$ for $t \in \left[0, x_0\right)$. Similarly, for For $x_0 < 0$, there exist the solution $x(t) = x(0) + t$ for $t \in \left[0, -x_0\right)$. For t larger than $|x_0|$ in both cases, each solution reaches the point $x(t) = 0$ and cannot leave it as the vector fields from both sides push towards it. However, since $\dot{x} = 0 \neq -1$, we have no solution in the classical or Carathéodory sense, and a more general concept is necessary.

To have a meaningful ODE, now it is necessary to choose a more general notion of solution. A popular option in the control community is Fillipov's extension, which suggests embedding the ODE (4) into the following differential inclusion [34]:

$$\dot{x}(t) \in \mathcal{F}_{\mathrm{F}}(x(t), u(t)) := \bigcap_{\delta > 0} \bigcap_{\mu(N)=0} \overline{\mathrm{conv}}\, f(x(t) + \delta \mathcal{B}(x(t)) \setminus N, u(t)). \tag{6}$$

The right-hand side is now a closed convex set. Given a point $x(t)$ and the corresponding $u(t)$, the idea behind this definition is to regard the closed convex hull of all neighboring values in a ball $x(t) \in \delta B(x(t))$ instead of only $f(x(t), u(t))$ and thereby ignoring all values of $f(x(t), u(t))$ on sets of measure zero. In practice, these sets are surfaces on which f becomes discontinuous or points/surfaces where f is not even defined. If $f(\cdot)$ is continuous at $x(t)$ we obtain $\mathcal{F}_{\mathrm{F}}(x(t), u(t)) = \{f(x(t), u(t))\}$. This DI (6) is a so-called Filippov system. An absolutely continuous function x is said to be a Filippov solution if it is a solution to the initial value problem defined (6). It is guaranteed that at least one solution exists to the Filippov DI (6) [3,33]. For the uniqueness of solutions, more assumptions must be made, for explicit statements see [3,27,33].

For illustration, we revisit the example in (5) and apply Filippov's extension to it. We obtain

$$\dot{x} \in \begin{cases} \{1\}, & x < 0, \\ [-1, 1], & x = 0, \\ \{-1\}, & x \geq 0, \end{cases} \tag{7}$$

It can be seen that this DI has a solution for any $x(0)$ and for all $t \in [0, \infty)$, since for $t > |x_0|$ we have that $\dot{x} = 0 \in [-1, 1]$.

2.3. *Piecewise smooth systems as Filippov systems*

In the modeling of physical systems and control applications the discontinuities in (4) usually appear in a very structured way. Such an example are piecewise smooth systems, which read as:

$$\dot{x}(t) = f_i(x(t), u(t)),\ \text{if}\ x(t) \in R_i \subset \mathbb{R}^{n_x},\ i \in \mathcal{J} := \{1, \dots, n_f\}, \tag{8}$$

where R_i are disjoint, connected, and open sets. They are assumed to be nonempty and to have piecewise-smooth boundaries ∂R_i. We assume that $\overline{\bigcup_{i \in \mathcal{J}} R_i} = \mathbb{R}^{n_x}$ and that $\mathbb{R}^{n_x} \setminus \bigcup_{i \in \mathcal{J}} R_i$ is a set of measure zero. The functions $f_i(\cdot)$ are assumed to be at least twice continuously differentiable and Lipschitz continuous on an open neighborhood of $\overline{R}_i$. We regard PSS where the regions R_i are defined via a finite number of switching functions $\psi_j(x)$, $j = 1, \dots, n_\psi$. It is assumed that the functions $\psi_j(x)$ are Lipschitz continuous and at least twice continuously differentiable.

Definition 1 (*Base Regions*)**.** Given n_ψ scalar switching functions $\psi_j(x)$, $j \in C := \{1, \dots, n_\psi\}$, we define the $n_f = 2^{n_\psi}$ base regions:

$$\begin{aligned}
\tilde{R}_1 &= \{x \in \mathbb{R}^{n_x} \mid \psi_1(x) > 0, \psi_2(x) > 0, \dots, \psi_{n_\psi - 1}(x) > 0, \psi_{n_\psi}(x) > 0\}, \\
\tilde{R}_2 &= \{x \in \mathbb{R}^{n_x} \mid \psi_1(x) > 0, \psi_2(x) > 0, \dots, \psi_{n_\psi - 1}(x) > 0, \psi_{n_\psi}(x) < 0\}, \\
&\vdots \\
\tilde{R}_{n_f} &= \{x \in \mathbb{R}^{n_x} \mid \psi_1(x) < 0, \psi_2(x) < 0, \dots, \psi_{n_\psi - 1}(x) < 0, \psi_{n_\psi}(x) < 0\},
\end{aligned}$$

such that $\mathbb{R}^{n_x} = \cup_{i=1}^{n_f} \tilde{R}_i$. These definitions are compactly expressed via a dense sign matrix $S \in \mathbb{R}^{n_f \times n_\psi}$:

$$S = \begin{bmatrix} 1 & 1 & \dots & 1 & 1 \\ 1 & 1 & \dots & 1 & -1 \\ \vdots & \vdots & \cdots & \vdots & \vdots \\ -1 & -1 & \dots & -1 & -1 \end{bmatrix}. \tag{9}$$

The matrix S has no repeating rows and no zero entries. The sets $\tilde{R}_i$ are defined using the rows of the matrix S:

$$\tilde{R}_i = \{x \in \mathbb{R}^{n_x} \mid S_{i,j} \psi_j(x) > 0, j \in C\},\ i = 1, \dots, n_f. \tag{10}$$

Note that the boundaries of the regions $\partial \tilde{R}_i$ are unions of subsets of the zero-level sets of corresponding functions $\psi_j(x)$. For notational convenience and ease of exposition, we assume that the PSS regions are equal to the base regions, *i.e.*, $n_f = 2^{n_\psi}$ and $R_i = \tilde{R}_i$ for $i = 1, \dots, n_f$. Other cases are discussed later in Section 6.

Note that the ODE (8) is not defined on the region boundaries ∂R_i. However, if the vector fields from both sides of ∂R_i push towards it, the solution needs to evolve on ∂R_i. The cases when $x(t)$ must evolve on ∂R_i [3,33] are so-called *sliding modes*. To define the sliding mode dynamics, one can apply Filippov's extension to the PSS (8). Furthermore, the special structure of the PSS allows a more explicit definition of (6) via a finite number of multipliers θ_i [3,23]. The corresponding DI reads as

$$\dot{x}(t) \in \mathcal{F}_{\mathrm{F}}(x,u) = \Big\{ \sum_{i\in\mathcal{J}} f_i(x,u)\,\theta_i \mid \sum_{i\in\mathcal{J}} \theta_i = 1, \theta_i \geq 0,\ \ \theta_i = 0 \text{ if } x \notin \overline{R}_i, i \in \mathcal{J} \Big\}, \tag{11}$$

with $\mathcal{F}_{\mathrm{F}} : \mathbb{R}^{n_x} \times \mathbb{R}^{n_u} \to \mathcal{P}(\mathbb{R}^{n_x})$. An important notion for Filippov PSS (11) is the active set, which is defined as the set

$$\mathcal{I}(x(t)) = \{ i \in \mathcal{J} \mid \theta_i(t) > 0 \}. \tag{12}$$

Note that if $x(t)$ is in the interior of a region, then $\mathcal{I}$ is a singleton, and for sliding modes, it consists of the indices of regions neighboring the current sliding surface.

Remark 2. A solution of the Filippov PSS (11) is an absolutely continuous function. However, systems with state jumps (and thus discontinuous $x(t)$) are often of practical interest. One way to transform such systems into a PSS, and thus to apply the methods developed in this paper, is to use the time-freezing reformulation [4,5,7,8,35]. In several classes of systems with state jumps some parts of the state space are forbidden (*e.g.*, a hooping robot is not allowed to penetrate the ground). The main idea in the time-freezing is to define auxiliary regions R_i in the forbidden region and corresponding auxiliary dynamics $f_i(x)$. By construction, the endpoints of the trajectory parts evolving in the auxiliary regions satisfy the state jump law. Furthermore, a clock state $t(\tau)$ is introduced, which evolves in the feasible region of the initial system ($\frac{\mathrm{d}t}{\mathrm{d}\tau} > 0$) and is frozen in the initially forbidden region ($\frac{\mathrm{d}t}{\mathrm{d}\tau} = 0$). Here τ is the time of the new relaxed system. Now, by taking only the parts of the trajectory where the clock state $t(\tau)$ evolved, we recover the discontinuous solution of the original system. An example of such a system is treated in Section 7.2.

2.4. Aizerman–Pyatnitskii differential inclusions

So far we have introduced dynamic complementarity and Filippov systems. Now we introduce another special case of the discontinuous ODE (4) and relate it to the others later with the help of step functions. Regard an ODE with DRHS of the form of:

$$\dot{x} = f(x, u, v(x)), \tag{13}$$

where $f : \mathbb{R}^{n_x} \times \mathbb{R}^{n_u} \in \mathbb{R}^{n_v}$ is continuous in all arguments, but $v(x)$ is discontinuous, *e.g.* it could be the single-valued Heaviside step function. Such systems appear commonly in sliding mode control or the modeling of gene-regulatory networks [1]. It assumed that for each $v_i \in V_i(x)$, where $V_i(x)$ is a closed convex set and for some x the set $V_i(x)$ is not a singleton. Hence, one can define the following differential inclusion:

$$\dot{x} \in \mathcal{F}_{\mathrm{AP}}(x,u) := \Big\{ f(x,u,v) \mid v_i \in V_i(x), i = 1, \dots, n_v \Big\}. \tag{14}$$

Recall that $\mathcal{F}_{\mathrm{AP}} : \mathbb{R}^{n_x} \times \mathbb{R}^{n_u} \to \mathcal{P}(\mathbb{R}^{n_x})$. The set $\mathcal{F}_{\mathrm{AP}}(x,u)$ is in general nonconvex, except in some special cases. For example, $\mathcal{F}_{\mathrm{AP}}(x,u)$ is convex if $v(x)$ enters the r.h.s. of (13) linearly and $V(x)$ is closed convex. The DI (14) does not have as rich a theory as Filippov DIs, and there are fewer results available on the existence of solutions and convergence of numerical methods [1]. These systems are called Aizerman–Pyatnitskii DIs, cf. [1] and [3, page 55, Definition c]. Time-stepping methods for such systems were developed in [1]. In this paper, we focus on the case where the functions $V_i(x)$ are given by set-valued Heaviside step functions. In particular, we regard the DI (2) introduced at the beginning of the paper, where $V(x) = \Gamma(\psi(x))$.

2.5. Rewriting Filippov PSS as DCS via set-valued Heaviside step functions

The next question we address is: How can we find an (implicit) function of x for computing the Filippov multipliers θ in (11)? To derive such a function, we will make use of the set-valued step function and the algebraic representations of the regions R_i in (10). Moreover, expressing the step function via the Karush–Kuhn–Tucker (KKT) conditions of a suitable Linear Program (LP), we can write the Filippov DI (11) as an equivalent DCS. This DCS will be the main formulation used in the development of the FESD method.

2.5.1. Heaviside step functions via linear programming

We start the described procedure with a closer look at the step functions.

Let us denote by $\alpha \in \mathbb{R}^{n_\psi}$ a selection $\alpha \in \Gamma(\psi(x))$. A well-known way to express the function $\Gamma(\psi(x))$ [1,20] is the use of the solution map of the parametric linear program:

$$\Gamma(\psi(x)) = \arg \min_{\alpha\in\mathbb{R}^{n_\psi}} \ -\psi(x)^\top \alpha \tag{15a}$$

$$\text{s.t.} \quad 0 \leq \alpha_i \leq 1,\ \ i = 1, \dots, n_\psi. \tag{15b}$$

Note that all components of α are decoupled in this LP, *i.e.*, every α_i can be obtained by solving a one-dimensional LP with the objective $-\psi_i(x)\alpha_i$ and the feasible set $0 \leq \alpha_i \leq 1$. Let $\lambda^{\mathrm{n}}, \lambda^{\mathrm{p}} \in \mathbb{R}^{n_\psi}$ be the Lagrange multipliers for the lower and upper bound on α in (15b), respectively. The KKT conditions of (15) read as

$$\psi(x) = \lambda^{\mathrm{p}} - \lambda^{\mathrm{n}}, \tag{16a}$$

$$0 \leq \lambda^{\mathrm{n}} \perp \alpha \geq 0, \tag{16b}$$

$$0 \leq \lambda^{\mathrm{p}} \perp e - \alpha \geq 0, \tag{16c}$$

Now we have a purely algebraic representation of the set-valued Heaviside step function. Let us look at a single component α_j and the associated functions $\psi_j(x)$. From the LP (15) and its KKT conditions, one can see that for $\psi_j(x) > 0$, we have $\alpha_j = 1$. Since the upper bound is active, we have that $\lambda_j^{\mathrm{n}} = 0$ and from (16a) that $\lambda_{\mathrm{p},j} = \psi_j(x) > 0$. Likewise, for $\psi_j(x) < 0$, we obtain $\alpha_j = 0$, $\lambda_j^{\mathrm{p}} = 0$ and $\lambda_j^{\mathrm{n}} = -\psi_j(x) > 0$. On the other hand, $\psi_j(x) = 0$ implies that $\alpha_j \in [0,1]$ and $\lambda_j^{\mathrm{p}} = \lambda_j^{\mathrm{n}} = 0$. From this discussion, it can be seen that $\psi(x)$, λ^{n} and λ^{p} are related by the following expressions:

$$\lambda^{\mathrm{p}} = \max(\psi(x), 0),\ \ \lambda^{\mathrm{n}} = -\min(\psi(x), 0). \tag{17}$$

That is, λ^{p} collects the positive parts of $\psi(x)$ and λ^{n} the absolute value of the negative parts of $\psi(x)$. From this relation, we can immediately conclude the following:

Lemma 3. *Let $\psi(x(t))$ be a continuous function of time, and then the functions $\lambda^{\mathrm{p}}(t)$ and $\lambda^{\mathrm{n}}(t)$ are continuous in time.*

The continuity of the Lagrange multipliers $\lambda^{\mathrm{p}}(t)$ and $\lambda^{\mathrm{n}}(t)$ plays a crucial role in the switch detection in the FESD method. The exact switch detection is necessary for high-order accuracy and the correct computation of numerical sensitivities.

2.5.2. Filippov system as a dynamic complementarity systems

For the sake of clarity, we start by illustrating what we want to achieve on a simple example and give in the sequel the general expression.

Example 1 (*Step Representation*)**.** We regard a PSS with four regions defined via two scalar switching functions $\psi_1(x)$ and $\psi_2(x)$. The regions are equal to the base sets from Definition 1 and read as $R_1 = \{x \in \mathbb{R}^{n_x} \mid \psi_1(x) > 0, \psi_2(x) > 0\}$, $R_2 = \{x \in \mathbb{R}^{n_x} \mid \psi_1(x) > 0, \psi_2(x) < 0\}$, $R_3 = \{x \in \mathbb{R}^{n_x} \mid \psi_1(x) < 0, \psi_2(x) > 0\}$ and $R_4 = \{x \in \mathbb{R}^{n_x} \mid \psi_1(x) < 0, \psi_2(x) < 0\}$.

The corresponding sign matrix $S \in \mathbb{R}^{4\times 2}$ read as

$$S = \begin{bmatrix} 1 & 1 \\ 1 & -1 \\ -1 & 1 \\ -1 & -1 \end{bmatrix}.$$

The corresponding Filippov system (defined by (11) reads as:

$$\dot{x} \in \{\sum_{i=1}^{4} \theta_i f_i(x) \mid \theta \geq 0, \sum_{i=1}^{4} \theta_i = 1,\ \theta_i = 0, if\ \ x \notin \overline{R_i}\}. \tag{18}$$

Let $x \in R_1$, then $\alpha_1 \in \gamma(\psi_1(x)) = \{1\}$ and $\alpha_2 \in \gamma(\psi_2(x)) = \{1\}$, thus $\theta_1 = \alpha_1\alpha_2 = 1$. On the other hand, by direct evaluation of the step functions $\gamma(\psi_1(x))$ and $\gamma(\psi_2(x))$ for any $x \notin \overline{R_1}$, one can see that $\theta_1 = \alpha_1\alpha_2 = 0$, since at least one of the selections $\alpha_1 \in \gamma(\psi_1(x))$ or $\alpha_2 \in \gamma(\psi_2(x))$ is zero. Similarly, for $x \in R_2$, we observe that $\alpha_1 \in \underline{\gamma(\psi_1}(x)) = \{1\}$ and $\alpha_2 \in \gamma(\psi_2(x)) = \{0\}$ and we can set $\theta_2 = \alpha_1(1-\alpha_2) = 1$. By direct evaluation of we can see that $\theta_2 > 0$ if $x \in \overline{R_2}$ and $\theta_2 = 0$, otherwise. Following the same pattern, we conclude that $\theta_3 = (1-\alpha_1)\alpha_2$ and $\theta_4 = (1-\alpha_1)(1-\alpha_2)$. Thus, we can define the system

$$\begin{aligned}\dot{x} \in \Big\{ & \alpha_1\alpha_2 f_1(x) + \alpha_1(1-\alpha_2) f_2(x) + (1-\alpha_1)\alpha_2 f_3(x) + (1-\alpha_1)(1-\alpha_2) f_4(x) \\ & \mid \alpha_1 \in \gamma(\psi_1(x)), \alpha_2 \in \gamma(\psi_2(x)) \Big\}.\end{aligned} \tag{19}$$

Since $\alpha_1, \alpha_2 \in [0,1]$ it is clear that $\theta_i \in [0,1], i \in \{1,\dots,4\}$. Moreover, direct calculation shows that $\sum_{i=1}^{4} \theta_i = 1$. Therefore, we conclude that the sets in the r.h.s. of (18) and (19) are the same sets, *i.e.*, we can express the Filippov set via set-valued Heaviside step functions.

Furthermore, we can observe how the sign pattern in S determines how α_j enters the expression for θ_i. For $S_{i,j} = 1$ we have α_j, for $S_{i,j} = -1$ we have $(1-\alpha_j)$. In summary, the definition of θ_i consists of products of α_j and $(1-\alpha_k)$, *i.e.*, it is multi-affine (*i.e.*, polynomial up to certain degree) in the selections α_j, $j = 1,\dots,n_\psi$.

We generalize the patterns observed in the previous example and define the set

$$\mathcal{F}_{\mathrm{H}}(x,u) := \Big\{ \sum_{i=1}^{n_f} \prod_{j=1}^{n_\psi} \Big(\frac{1-S_{i,j}}{2} + S_{i,j}\alpha_i \Big) f_i(x,u) \mid \alpha \in \Gamma(\psi(x)) \Big\}. \tag{20}$$

Note that we have

$$\frac{1-S_{i,j}}{2}+S_{i,j}\alpha_i=\begin{cases}\alpha_i, & \text{if } S_{i,j}=1,\\ 1-\alpha_i, & \text{if } S_{i,j}=-1.\end{cases}$$

Similar definitions of $\mathcal{F}_{\mathrm{H}}(x,u)$ as in (20) can be found in [2, Section 4.2] and [10, Section 2.1]. Observe that this set has the same form as the r.h.s. $\mathcal{F}_{\mathrm{AP}}(x,u)$ in (14). Next, we show that $\mathcal{F}_{\mathrm{H}}(x,u)$ is indeed the same set as $\mathcal{F}_{\mathrm{F}}(x,u)$, *i.e.*, the set in the r.h.s. of (11).

Lemma 4 (*Lemma 1.5 in [2]*). *Let $a_1, a_2, \dots, a_m \in \mathbb{R}$. Consider the 2^m non-repeated products of the form $p_i = (1 \pm a_1)(1 \pm a_2)\cdots(1 \pm a_m)$, then it holds that $\sum_{i=1}^{2^m} p_i = 2^m$.*

Proposition 5. *Let*

$$\theta_i=\prod_{j=1}^{n_\psi}\Big(\frac{1-S_{i,j}}{2}+S_{i,j}\alpha_j\Big),\ \textit{for all } i\in\mathcal{J}=\{1,\dots,n_f\},\tag{21}$$

then it holds that $\mathcal{F}_{\mathrm{F}}(x,u)=\mathcal{F}_{\mathrm{H}}(x,u)$.

Proof. We only need to show that $\theta_i \geq 0$ for all $i \in \mathcal{J}$ and $\sum_{i\in\mathcal{J}}\theta_i = 1$. It is easy to see that $\theta_i \in [0,1]$ since it consists of a product of terms that takes value in $[0,1]$.

Without loss of generality, regard θ_1 and suppose that $x \notin \overline{R_1}$. This means that $x \in R_i, i \neq 1$ and that at least one $\psi_j(x) < 0, j \in \mathcal{C}$, which implies that $\alpha_j = 0$. From (21) it follows that $\theta_1 = 0$ if $x \notin \overline{R_1}$. By similar arguments it follows that $\theta_i = 0$ if $x \notin \overline{R_i}$ for $i = 1,\dots,n_f$.

Next we show that $\sum_{i\in\mathcal{J}}\theta_i = 1$. We introduce the change of variables:

$$\frac{1+b_j}{2}=\alpha_j,\ \frac{1-b_j}{2}=1-\alpha_j.$$

Then all θ_i are of the form

$$\theta_i=2^{-n_\psi}\prod_{j=1}^{n_\psi}(1\pm b_j).$$

By applying Lemma 4, we conclude that $\sum_{i\in\mathcal{J}}\theta_i = 1$ and the proof is complete. □

To pass from the definition in Eq. (20) to a dynamic complementarity system, we state the KKT conditions of (15) to obtain an algebraic expression for $\Gamma(\psi(x))$. Combining this with the definition of the Filippov set in (20) and the expression for θ_i in (21), we obtain the following DCS:

$$\dot{x}=F(x,u)\,\theta,\tag{22a}$$

$$0=\theta_i-\prod_{j=1}^{n_\psi}\Big(\frac{1-S_{i,j}}{2}+S_{i,j}\alpha_j\Big),\ \text{for all } i\in\mathcal{J},\tag{22b}$$

$$0=\psi(x)-\lambda^{\mathrm{p}}+\lambda^{\mathrm{n}},\tag{22c}$$

$$0\leq\lambda^{\mathrm{n}}\perp\alpha\geq 0,\tag{22d}$$

$$0\leq\lambda^{\mathrm{p}}\perp e-\alpha\geq 0,\tag{22e}$$

where $F(x,u) = [f_1(x,u),\dots,f_{n_f}(x,u)] \in \mathbb{R}^{n_x\times n_f}$, $\theta = (\theta_1,\dots,\theta_{n_f}) \in \mathbb{R}^{n_f}$ and $\lambda^{\mathrm{p}},\lambda^{\mathrm{n}},\alpha \in \mathbb{R}^{n_\psi}$. We group all algebraic equations into a single function and use a C-function $\Psi(\cdot,\cdot)$ for the complementarity condition to obtain a more compact expression:

$$G(x,\theta,\alpha,\lambda^{\mathrm{p}},\lambda^{\mathrm{n}}):=\begin{bmatrix}\theta_1-\prod_{j=1}^{n_\psi}\Big(\frac{1-S_{1,j}}{2}+S_{1,j}\alpha_j\Big)\\ \vdots\\ \theta_{n_f}-\prod_{j=1}^{n_\psi}\Big(\frac{1-S_{n_f,j}}{2}+S_{n_f,j}\alpha_j\Big)\\ \psi(x)-\lambda^{\mathrm{p}}+\lambda^{\mathrm{n}}\\ \Psi(\lambda^{\mathrm{n}},\alpha)\\ \Psi(\lambda^{\mathrm{p}},e-\alpha)\end{bmatrix}.\tag{23}$$

Finally, we obtain a compact representation of (22) in the form of a nonsmooth DAE:

$$\dot{x}=F(x,u)\theta,\tag{24a}$$

$$0=G(x,\theta,\alpha,\lambda^{\mathrm{p}},\lambda^{\mathrm{n}}).\tag{24b}$$

We can see that (22) is an instance of the generic DCS (3). Set $z = (\theta,\alpha)$ and $y = (\lambda^{\mathrm{n}},\lambda^{\mathrm{p}})$. It follows that $f(x,y,z) = F(x,u)\theta$, the function $g_{\mathrm{e}}(x,y,z)$ is define by the expressions in (22b) and (22c), and $g_{\mathrm{c}}(x,z,y) = (\alpha, e-\alpha)$.

Remark 6. In this section, we have focused on PSS, their Filippov extension, and their relation to Aizerman-Pyatnitskii DIs (14). However, we can also reformulate generic Aizerman-Pyatnitskii DIs with Heaviside step functions (2) into DCS by replacing the step function by (16). There is no need for equivalence to Filippov systems to apply this procedure, and the numerical method developed in this paper can also be applied directly to such DCS. Such a more general example is treated in Section 7.1.

2.6. Stewart's representation

The FESD method [14] was originally developed for Stewart's representation [23] of Filippov systems. Stewart's representation assumes a specific definition of the regions R_i and uses a LP to transform the Filippov system into an equivalent DCS, which is not the same DCS as (22). For comparison and completeness, we briefly introduced Stewart's DCS.

In Stewart's representation [23], the regions R_i are defined via so-called indicator functions $g_i(x)$ for all $i \in \mathcal{J} = \{1, \dots, n_f\}$. The definition reads as

$$R_i = \{x \in \mathbb{R}^{n_x} \mid g_i(x) < \min_{j \in \mathcal{J} \setminus \{i\}} g_j(x)\}. \tag{25}$$

Arguably, this definition of the regions might be less intuitive than (10). However, if the regions R_i match the base sets $\tilde{R}_i$ from Definition 1, it was shown in [14, Proposition 2] that the function $g(x) = (g_1(x), \dots, g_{n_f}(x))$ can be obtained as:

$$g(x) = -S\psi(x). \tag{26}$$

The multiplier vector θ is expressed as the solution of an LP parameterized by x:

$$\theta(x) \in \arg \min_{\tilde{\theta} \in \mathbb{R}^{n_f}} \quad g(x)^\top \tilde{\theta} \tag{27a}$$

$$\text{s.t.} \quad e^\top \tilde{\theta} = 1 \tag{27b}$$

$$\tilde{\theta} \geq 0. \tag{27c}$$

Using its KKT condition, one can obtain a DCS equivalent to (24), which reads as:

$$\dot{x} = F(x, u)\theta, \tag{28a}$$

$$0 = g(x) - \lambda + \mu e, \tag{28b}$$

$$0 = e^\top \theta - 1, \tag{28c}$$

$$0 \leq \lambda \perp \theta \geq 0, \tag{28d}$$

where $\mu \in \mathbb{R}$ and $\lambda \in \mathbb{R}^{n_f}$ are the Lagrange multipliers for the constraints (27b) and (27c), respectively.

The DCS (28) is also an instance of the generic DCS (3). Set $z = (\theta, \mu)$ and $y = \lambda$. It follows that $f(x, y, z) = F(x, u)\theta$, $g_{\mathrm{e}}(x, y, z) = (g(x) - \lambda + \mu e, e^\top \theta - 1)$ and $g_{\mathrm{c}}(x, z, y) = \theta$.

2.7. Summary and relations between different formalisms

The diagram in Fig. 1 summarizes the relationships between the nonsmooth systems studied in this paper. The first column consists of the different ODEs with DRHS that we have treated. After treating a generic ODE with DRHS in (4), we specialize in two structured cases: PSS in (8) and ODEs (13), where their dynamics contain some discontinuous expressions of x, *e.g.* as Heaviside step functions.

These ODEs may not have a classical or Carathéodory solution in some cases, so we embed them in differential inclusions and obtain more general notions, such as Filippov solutions. These concepts are summarized in the second column. A generic ODE with DRHS is generalized to Filippov DIs (6). If the ODE is more structured as a piecewise smooth system, then its Filippov extensions become more explicit (11). In both cases, the r.h.s. is now a convex and compact set. If the discontinuous function in the structured ODE (13) is replaced by set-valued extensions, we obtain an Aizerman–Pyatnitskii DI (14). The set on the r.h.s. may even be nonconvex. In Proposition 5 we show that when we use PSS and Heaviside step functions, the Aizerman-Pyatnitskii DI and the Filippov extension for PSS are equivalent.

The third column consists of dynamic complementarity systems obtained from the DIs, which are useful representations for numerical computations. The Heaviside step DCS (22) and the Stewart DCS (28) are both instances of a generic DCS (3). They are derived from the corresponding differential inclusions using the KKT conditions of parametric LPs, as shown in Sections 2.6 and 2.5.2, respectively. The numerical methods developed in [14] and in this paper exploit the continuity properties of the Lagrange multipliers in the KKT conditions of these LPs. We conclude this section by illustrating the different formalisms with an example.

Example 2. Let us illustrate the different classes of nonsmooth systems on the discontinuous ODE (which is in the form of (4)):

$$\dot{x} = \begin{cases} 1, & x > 0, \\ 3, & x < 0. \end{cases}$$

$$\begin{tikzcd}
\text{ODE with DRHS (4)} \arrow[r, Rightarrow] \arrow[d, Rightarrow] \arrow[dd, Rightarrow, bend right] & \text{Filippov DI (6)} \arrow[d, Rightarrow] & \text{DCS (3)} \arrow[d, Rightarrow] \arrow[dd, Rightarrow, bend left] \\
\text{PSS (8)} \arrow[r, Rightarrow] & \text{Filippov PSS (11)} \arrow[r, Rightarrow] \arrow[dr, Leftrightarrow] & \text{Heaviside DCS(22)} \\
\text{Structured ODE with DRHS (13)} \arrow[r, Rightarrow] & \text{Aizerman–Pyatnitskii DI (14)} \arrow[u, Rightarrow] & \text{Stewart DCS(28)}
\end{tikzcd}$$

Fig. 1. Summary of relations between nonsmooth systems treated in this paper.

This is a PSS (as in (8)) with the switching function $\psi(x) = x$, with the regions $R_1 = \{x \mid x > 0\}$ and $R_2 = \{x \mid x < 0\}$. The Filippov extensions of this PSS (Eq. (11)) reads as: $\dot{x} \in \{\theta_1 + 3\theta_2 \mid \theta \geq 0, \theta_1 + \theta_2 = 1\}$, with $\theta = (\theta_1, \theta_2)$. In the form of an Aizerman–Pyatnitskii DI (2), *i.e.* (14), the system reads as $\dot{x} \in 3 - 2\gamma(x)$. We can write also as a DCS of the form of (22):

$$\begin{aligned}
&\dot{x} = \begin{bmatrix} 3 & 1 \end{bmatrix} \theta, \\
&\theta_1 = \alpha,\ \theta_2 = 1 - \alpha,\ x = \lambda^{\mathrm{p}} - \lambda^{\mathrm{n}}, \\
&0 \leq \lambda^{\mathrm{p}} \perp \alpha \geq 0,\ 0 \leq \lambda^{\mathrm{n}} \perp 1 - \alpha \geq 0.
\end{aligned}$$

Similarly, by defining the indicator function $g(x) = (-x, x)$, we can state Stewart DCS (28):

$$\begin{aligned}
&\dot{x} = \begin{bmatrix} 3 & 1 \end{bmatrix} \theta, \\
&-x = \lambda_1 - \mu,\ x = \lambda_2 - \mu,\ \theta_1 + \theta_2 = 1, \\
&0 \leq \theta_1 \perp \lambda_1 \geq 0,\ 0 \leq \lambda_2 \perp \theta_2 \geq 0.
\end{aligned}$$

3. Properties of the step representation DCS

In this section, we study some properties of the DCS obtained via Heaviside step functions (22) for a fixed active set and at active-set changes. These properties will be useful for algorithmic development in the subsequent sections. For a fixed active set, *i.e.* $\mathcal{I}(x(t)) = \{i \mid \theta_i(t) > 0\}$ being constant on some time interval, there are no switches and the dynamics have locally no discontinuities. On the other hand, at an active-set change, there is a switch and a discontinuity in the dynamics.

3.1. Active-set changes and continuity of λ^{p} and λ^{n}

Active-set changes are paired with discontinuities in some of the algebraic variables. We have seen in Lemma 3 that λ^{p} and λ^{n} are continuous functions of time.

At an active-set change, at least one of the switching functions $\psi_j(x)$ either becomes zero or if it was zero it becomes nonzero. It follows from $\psi_j(x(t)) = \lambda_j^{\mathrm{p}}(t) - \lambda_j^{\mathrm{n}}(t)$ (in (17)), that also both $\lambda_j^{\mathrm{p}}(t)$ and $\lambda_j^{\mathrm{n}}(t)$ must be zero at an active-set change.

We use the DCS formulation via step functions in (22) to illustrate the different switching cases that arise in Filippov systems.

Example 3. There are four possible switching cases which we illustrate with the following examples:

(a) crossing a surface of discontinuity, $\dot{x}(t) \in 2 - \mathrm{sign}(x(t))$ (same system as in Example 2),
(b) entering a sliding mode, $\dot{x}(t) \in -\mathrm{sign}(x(t)) + 0.2\sin(5t)$,
(c) leaving a sliding mode $\dot{x}(t) \in -\mathrm{sign}(x(t)) + t$,
(d) spontaneous switch, $\dot{x}(t) \in \mathrm{sign}(x(t))$.

In case (a), for $x(0) < 0$ the trajectory reaches $x = 0$ and crosses it. In example (b), for any finite $x(0)$, the trajectory reaches $x = 0$ and stays there. On the other hand, in example (c), for $x(0) = 0$, the DI has a unique solution and leaves $x = 0$ at $t = 1$. In the last example, the DI has infinitely many solutions for $x(0) = 0$, and $x(t)$ can spontaneously leave $x = 0$ at any $t \geq 0$. The trajectories are illustrated in Fig. 2.

Whenever $x(t)$ has a kink, which corresponds to a switch and discontinuity in the dynamics, both the Lagrange multipliers $\lambda^{\mathrm{p}}(t)$ and $\lambda^{\mathrm{n}}(t)$ are zero at that time.

3.2. Fixed active set in the step formulation

We study the properties of the DCS (22) for a fixed active set $\mathcal{I}(x(t))$. Without loss of generality, the corresponding time interval is $\mathcal{T} = (0, T)$. For a fixed active set, the DCS (22) reduces either to an ODE or to a Differential Algebraic Equation (DAE).

We start with the simpler ODE case. Let $\psi_j(x) \neq 0$ for all $j \in \mathcal{C} := \{1, \dots, n_\psi\}$, then $x(t)$ is in the interior of some region R_i. It can be seen from the LP (15) that $\alpha_j \in \{0, 1\}$ for all $j \in \mathcal{C}$. This implies that $\theta_i = 1$ and $\theta_k = 0, k \neq i$. Therefore, $\mathcal{I}(x(t)) = \{i\}$ and the Filippov DI reduces to $\dot{x} \in F_{\mathrm{F}}(x) = \{f_i(x)\}$, *i.e.*, we have locally an ODE.

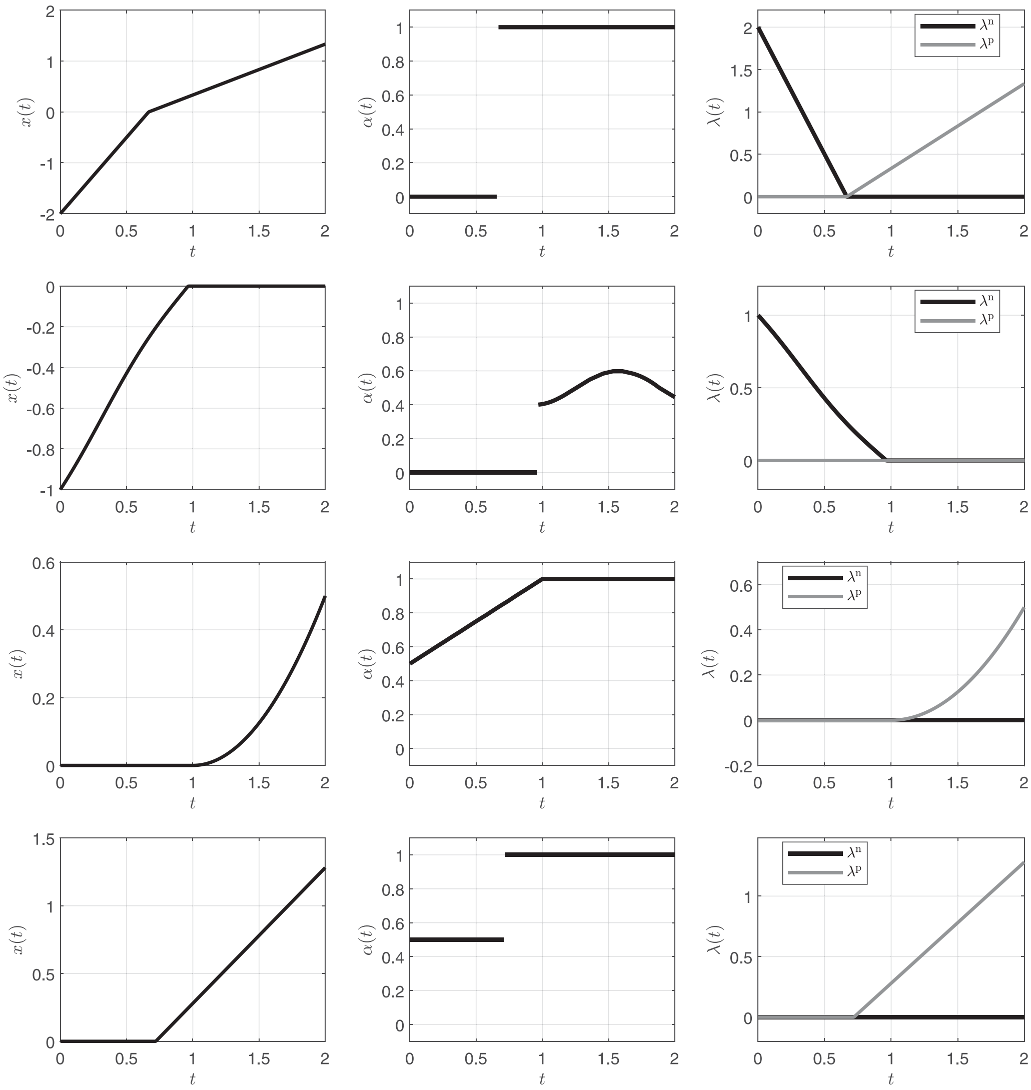


Fig. 2. Illustration of example solution trajectories for different switching cases. The rows from top to bottom show $x(t)$, $\alpha(t)$, $\lambda^{\mathrm{p}}(t)$ and $\lambda^{\mathrm{n}}(t)$ for the cases (a)–(d) in Example 3, respectively.

Next, we regard the case when $\mathcal{I}(x(t))$ is not a singleton, *i.e.*, the trajectory evolves at the boundary of two or more regions. Consequently, we have at least one $\psi_j(x) = 0$. Let us associate with $\mathcal{I}(x(t))$ the index set $\mathcal{K}(x(t)) = \{j \in \mathcal{C} \mid \psi_j(x) = 0\}$, *i.e.*, the set of indices of all switching functions that are zero for a given active set $\mathcal{I}(x(t))$. In the sequel, we make use of the following notation. For a given vector $a \in \mathbb{R}^n$ and set $\mathcal{I} \subseteq \{1, \dots, n\}$, we define the projection matrix $P_{\mathcal{I}} \in \mathbb{R}^{|\mathcal{I}| \times n}$, which has zeros or ones as entries. It selects all component $a_i, i \in \mathcal{I}$ from the vector a, *i.e.*, $a_{\mathcal{I}} = P_{\mathcal{I}} a \in \mathbb{R}^{|\mathcal{I}|}$ and $a_{\mathcal{I}} = [a_i \mid i \in \mathcal{I}]$.

Following the discussion from the previous section, for all nonzero $\psi_j(x)$, *i.e.*, $j \in \mathcal{C} \setminus \mathcal{K}$, we can compute $\alpha_j \in \{0, 1\}$ via the LP (15) and $\lambda_{\mathrm{p},j}, \lambda_{\mathrm{n},j}$ via (17). Next, we have that $\lambda_{\mathrm{p},j} = \lambda_{\mathrm{n},j} = 0$ for all $j \in \mathcal{K}$. It is left to determine α_j for all $j \in \mathcal{K}$ and thus implicitly all θ_i, for all $i \in \mathcal{I}$. Recall that $\theta_i = 0$ for all $i \notin \mathcal{I}$. By fixing the already known variables in (22) we obtain the DAE:

$$\dot{x} = F_{\mathcal{I}}(x, u)\, \theta_{\mathcal{I}}, \tag{29a}$$

$$\theta_i - \prod_{j=1}^{n_\psi} \Big(\frac{1 - S_{i,j}}{2} + S_{i,j} \alpha_j \Big) = 0,\ i \in \mathcal{I}, \tag{29b}$$

$$\psi_j(x) = 0,\ j \in \mathcal{K}, \tag{29c}$$

where we define $F_{\mathcal{I}}(x,u) := F(x,u)P_{\mathcal{I}}^{\top} \in \mathbb{R}^{n_x \times |\mathcal{I}|}$, *i.e.*, we select only the columns of $F(x,u)$ with the index $i \in \mathcal{I}$. Note that α_j for all $j \in \mathcal{C} \setminus \mathcal{K}$ are known and thus no degrees of freedom. We keep them for ease of notation. Thus we have a DAE with $|\mathcal{I}| + |\mathcal{K}|$ unknowns, namely $\theta_{\mathcal{I}} \in \mathbb{R}^{|\mathcal{I}|}$ and $\alpha_{\mathcal{K}} \in \mathbb{R}^{|\mathcal{K}|}$, and $|\mathcal{I}| + |\mathcal{K}|$ algebraic equations in (29b) and (29c).

Next, we investigate conditions under which the DAE (29) is well-posed. For this purpose, we define the matrix

$$W_{\mathcal{K},\mathcal{I}}(x,u) = \nabla\psi_{\mathcal{K}}(x)^{\top} F_{\mathcal{I}}(x,u) \in \mathbb{R}^{|\mathcal{K}| \times |\mathcal{I}|},$$

where $\nabla\psi_{\mathcal{K}}(x) = \left[\nabla\psi_j(x) \mid j \in \mathcal{K}\right] \in \mathbb{R}^{n_x \times |\mathcal{K}|}$ is a matrix, whose columns are the gradients of the switching functions that are zero for the given active set $\mathcal{I}$.

Moreover, we define a compact notation for the partial Jacobian $B_{\mathcal{K},\mathcal{I}}(\alpha) \in \mathbb{R}^{|\mathcal{I}| \times |\mathcal{K}|}$ of (29b) w.r.t. to $\alpha_{\mathcal{K}}$, with the elements:

$$B_{i,j}(\alpha) := \frac{\partial}{\partial \alpha_j}\Big(\prod_{l \in C} \frac{1 - S_{i,l}}{2} + S_{i,l}\alpha_l\Big),\ i \in \mathcal{I}, j \in \mathcal{K}.$$

Assumption 7. Given a fixed active set $\mathcal{I}(x(t)) = \mathcal{I}$ for $t \in \mathcal{T}$, it holds that the matrix functions $W_{\mathcal{K},\mathcal{I}}(x,u)$ and $B_{\mathcal{K},\mathcal{I}}(\alpha)$ are Lipschitz continuous in all their arguments, and that $W_{\mathcal{K},\mathcal{I}}(x,u)B_{\mathcal{K},\mathcal{I}}(\alpha)$ has rank $|\mathcal{K}|$, *i.e.* it is full rank, for all $t \in \mathcal{T}$.

Proposition 8. *Suppose that Assumption 7 holds. Given an initial value $x(0)$, the DAE (29) has a unique solution for all $t \in \mathcal{T}$.*

Proof. First, we differentiate (29c) with respect to t, such that algebraic variables appear explicitly in the algebraic equations (this corresponds to so-called index reduction in the theory of DAEs, cf. [36]):

$$\dot{x} = F_{\mathcal{I}}(x,u)\,\theta_{\mathcal{I}}, \tag{30a}$$

$$\theta_i - \prod_{j \in C}\Big(\frac{1 - S_{i,j}}{2} + S_{i,j}\alpha_j\Big) = 0,\ i \in \mathcal{I}, \tag{30b}$$

$$W_{\mathcal{K},\mathcal{I}}(x,u)\theta_{\mathcal{I}} = 0. \tag{30c}$$

Next, we prove that the partial Jacobian of (30b)–(30c) w.r.t. to the algebraic variables $(\theta_{\mathcal{I}}, \alpha_{\mathcal{K}})$ has rank $|\mathcal{I}| + |\mathcal{K}|$, *i.e.*, it is an invertible matrix.

We omit the dependencies on α and x for brevity. The Jacobian of (30b)–(30c) w.r.t. to $(\theta_{\mathcal{I}}, \alpha_{\mathcal{K}})$ has the form

$$A = \begin{bmatrix} I_{|\mathcal{I}|} & -B_{\mathcal{K},\mathcal{I}} \\ W_{\mathcal{K},\mathcal{I}} & \mathbf{0} \end{bmatrix}.$$

To prove that this matrix has full rank, we show that the only solution $(v,w) \in \mathbb{R}^{|\mathcal{I}|} \times \mathbb{R}^{|\mathcal{K}|}$, to the following linear system is the zero vector:

$$\begin{bmatrix} I_{|\mathcal{I}|} & -B_{\mathcal{I},\mathcal{K}} \\ W_{\mathcal{K},\mathcal{I}} & \mathbf{0} \end{bmatrix} \begin{bmatrix} v \\ w \end{bmatrix} = 0. \tag{31}$$

From the first line we have $v = B_{\mathcal{I},\mathcal{K}}w$ and substituting this into the second line we have $W_{\mathcal{I},\mathcal{K}}B_{\mathcal{I},\mathcal{K}}w = 0$. Since the matrix $W_{\mathcal{I},\mathcal{K}}B_{\mathcal{I},\mathcal{K}} \in \mathbb{R}^{|\mathcal{K}| \times |\mathcal{K}|}$ has rank $|\mathcal{K}|$, the only solution to (31) is $w = 0$, and $v = Bw = 0$. Hence, A has full rank.

Now we can apply the implicit function theorem [37, Theorem 1B.1] to (30b)–(30c), which guarantees the existence of continuously differentiable functions $\theta_{\mathcal{I}}(x)$ and $\alpha_{\mathcal{K}}(x)$. Since the function $\theta_{\mathcal{I}}(x)$ is continuously differentiable, it is also Lipschitz continuous for a fixed $\mathcal{I}(x(t))$ on $t \in \mathcal{T}$. By substituting $\theta_{\mathcal{I}}(x)$ into (30a), we have a product of two Lipschitz continuous functions (all columns of $F_{\mathcal{I}}(x,u)$, are Lipschitz by assumption), and the DAE (30) reduces to an ODE with a Lipschitz continuous r.h.s. This enables us to apply the Picard-Lindelöf Theorem to obtain the assertion of the proposition. □

We make a few comments on Assumption 7. The rank condition can be checked explicitly since one can compute the matrix $W_{\mathcal{K},\mathcal{I}}(x,u)B_{\mathcal{K},\mathcal{I}}(\alpha)$. We have already assumed Lipschitz continuity of all columns of $f_i(x,u)$ and the gradients $\nabla\psi_j(x)$. Here, we additionally assume it for the matrix $W_{\mathcal{K},\mathcal{I}}(x,u)$, whose entries are computed as inner products on these vectors. The entries of the matrix $B_{\mathcal{K},\mathcal{I}}(\alpha)$ are multi-affine terms, which are also Lipschitz, at least on the bounded domains we consider here. In Stewart's reformulation, we consider a square matrix with entries $\nabla g_i(x)^{\top} f_j(x,u)$ (which is structurally similar to $W_{\mathcal{K},\mathcal{I}}(x,u)$). For well-posedness with a fixed active set, the invertibility of this matrix is assumed [14,23].

In [2], the authors make some assumptions on the signs of the entries of $W_{\mathcal{K},\mathcal{I}}(x)$ and prove the existence, but not uniqueness, of solutions with a fixed-point argument. For the case of $|\mathcal{K}| \leq 2$, *i.e.*, sliding modes with co-dimension one or two, and with additional assumptions on the signs of the entries of $W_{\mathcal{K},\mathcal{I}}(x)$ they even prove the uniqueness of solutions.

Observe that for a given $x(t)$, there might be several $\mathcal{I}(x(t))$ that yield meaningful DAEs of the form of Eq. (29). This may happen when the Filippov DI does not have unique solutions, such as in Example 3 case (d). The trajectory may stay in a sliding mode or leave it any time. The trajectory pieces in either scenario are well-posed, even though the overall trajectory is not unique.

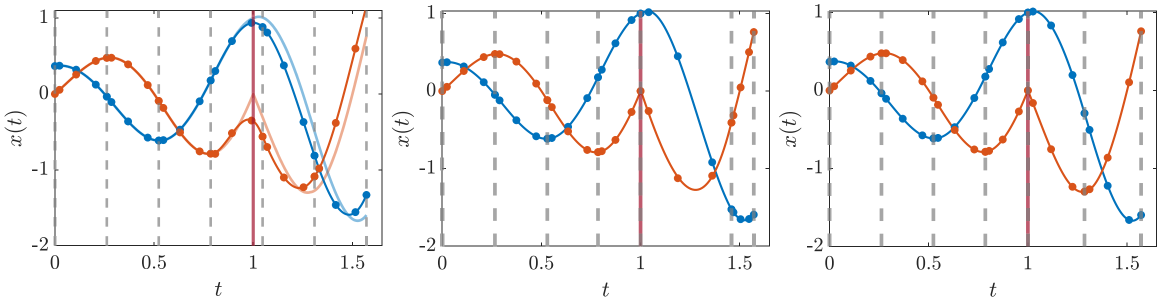


Fig. 3. Example trajectories of an ODE with DRHS, with a switch at $t = 1$. The solid lines correspond to numerical approximations and the transparent ones to analytic solutions. The circle markers correspond to the stage values of an underlying Runge–Kutta (RK) method, and the vertical dashed lines to the boundaries of the integration intervals. The left plots show an approximation obtained with a time-stepping RK method, the middle one with FESD but without step equilibration, and the right one with FESD including step equilibration.

4. Finite Elements with Switch Detection (FESD) for the step representation

4.1. Main ideas

We outline the main ideas in the derivation of the FESD discretization, and in the following sections, we explain each step in more detail. The starting point is a standard time-stepping Runge–Kutta (RK) discretization of the DCS (22). In this case, a fixed integration step size is assumed. If the switch occurs within an integration interval, the high accuracy properties of these methods are lost. This is shown in the left plot of Fig. 3. The switch occurred at an RK stage point inside the integration interval, and thereafter there is a large discrepancy between the numerical approximation and the exact solution. Nevertheless, the standard RK methods are a starting point for the development of the FESD method and we recall them in Section 4.2.

If the switches always coincided with the integration interval boundaries, the RK method would not lose its accuracy. This can be achieved by detecting the switches and adjusting the step size. To achieve this, in the FESD method [14], the integration step sizes are left as degrees of freedom. This idea was first proposed by Baumrucker and Biegler in [20] and extended and theoretically analyzed in [14]. Moreover, additional complementarity conditions, called cross complementarities, prohibit switches within an integration interval (finite element), which leads to their isolation at the boundaries and recovery of high accuracy. An illustration is given in the middle plot of Fig. 3. In contrast to the fixed step size discretization, the numerical and exact solutions are indistinguishable. We study these extensions in Section 4.3.

However, when there are no switches, the step sizes are not uniquely determined. As can be seen in the middle plot of Fig. 3, the integration intervals all have different and somewhat random lengths. In Section 4.4, we introduce the step equilibration conditions that make neighboring finite elements have the same length if no switches occur, and thus determine them uniquely. The right plot in Fig. 3 illustrates this effect, where now before and after the switch has equidistant grids. This is also a key ingredient for having locally unique solutions, as we will show later.

In Section 4.5, we summarize the developments of the previous sections and compactly summarize the full FESD discretization. We emphasize that in the FESD method, switch detection is fully implicit and based only on algebraic conditions. This makes it suitable for the discretization of optimal control problems. We conclude with Section 4.6, where we show how to apply the FESD method to discretize an optimal control problem subject to a nonsmooth dynamical system with Heaviside step functions.

4.2. Standard implicit Runge–Kutta discretization

As a starting point in our analysis, we regard a standard Runge–Kutta (RK) discretization of the DCS (22). For ease of notation, we work with the nonsmooth DAE formulation of the DCS (24), which we restate for convenience

$$\dot{x} = F(x,u)\theta, \tag{32a}$$

$$0 = G(x,\theta,\alpha,\lambda^{\mathrm{p}},\lambda^{\mathrm{n}}) \tag{32b}$$

In the sequel, one should keep in mind that (32b) collects all algebraic equations including the complementarity conditions (22d) and (22e).

In a discretized Optimal Control Problem (OCP) we usually have several control intervals. In our derivations here, it is sufficient to consider a single control interval $[0,T]$ with a constant control input $q \in \mathbb{R}^{n_u}$, *i.e.*, we set $u(t) = q$ for $t \in [0,T]$. The extension to more elaborate control parameterizations is straightforward [38]. In Section 4.6 we show how to discretize OCPs with multiple control intervals.

Let $x(0) = s_0$ be the given initial value. The control interval $[0, T]$ is divided into N_{FE} finite elements (*i.e.*, integration intervals) $[t_n, t_{n+1}]$ via the grid points $0 = t_0 < t_1 < \dots < t_{N_{\mathrm{FE}}} = T$. On each of the finite elements we regard an n_{s}-stage Runge–Kutta method which is characterized by the Butcher tableau entries $a_{i,j}, b_i$ and c_i with $i, j \in \{1, \dots, n_{\mathrm{s}}\}$ [36]. The step-sizes read as $h_n = t_{n+1} - t_n$, $n = 0, \dots, N_{\mathrm{FE}} - 1$. The approximation of the differential state at the grid points t_n is denoted by $x_n \approx x(t_n)$.

We regard the so-called differential representation of the Runge–Kutta method [36]. Thus, the derivatives of the states at the RK stage points $t_{n,i} := t_n + c_i h_n$, $i = 1, \dots, n_{\mathrm{s}}$, are the degrees of freedom. For a single finite element, we summarize them in the vector $V_n := (v_{n,1}, \dots, v_{n,n_{\mathrm{s}}}) \in \mathbb{R}^{n_{\mathrm{s}} n_x}$. The stage values for the algebraic variables are collected in the vectors: $\Theta_n := (\theta_{n,1}, \dots, \theta_{n,n_{\mathrm{s}}}) \in \mathbb{R}^{n_{\mathrm{s}} \cdot n_f}$, $A_n := (\alpha_{n,1}, \dots, \alpha_{n,n_{\mathrm{s}}}) \in \mathbb{R}^{n_{\mathrm{s}} \cdot n_c}$, $\Lambda_n^{\mathrm{p}} := (\lambda_{n,1}^{\mathrm{p}}, \dots, \lambda_{n,n_{\mathrm{s}}}^{\mathrm{p}}) \in \mathbb{R}^{n_{\mathrm{s}} \cdot n_c}$ and $\Lambda_n^{\mathrm{n}} := (\lambda_{n,1}^{\mathrm{n}}, \dots, \lambda_{n,n_{\mathrm{s}}}^{\mathrm{n}}) \in \mathbb{R}^{n_{\mathrm{s}} \cdot n_c}$.

We collect all *internal* variables in the vector $Z_n = (x_n, \Theta_n, A_n, \Lambda_n^{\mathrm{p}}, \Lambda_n^{\mathrm{n}}, V_n)$. The vector x_n^{next} denotes the state value at t_{n+1}, which is obtained after a single integration step. Now, we can state the RK equations for the DCS (32) for a single finite element as

$$0 = G_{\mathrm{rk}}(x_n^{\mathrm{next}}, Z_n, h_n, q) := \begin{bmatrix} v_{n,1} - F(x_n + h_n \sum_{j=1}^{n_{\mathrm{s}}} a_{1,j} v_{n,j}, q)\theta_{n,1} \\ \vdots \\ v_{n,n_{\mathrm{s}}} - F(x_n + h_n \sum_{j=1}^{n_{\mathrm{s}}} a_{n_{\mathrm{s}},j} v_{n,j}, q)\theta_{n,n_{\mathrm{s}}} \\ G(x_n + h_n \sum_{j=1}^{n_{\mathrm{s}}} a_{1,j} v_{n,j}, \theta_{n,1}, \alpha_{n,1}, \lambda_{n,1}^{\mathrm{p}}, \lambda_{n,1}^{\mathrm{n}}) \\ \vdots \\ G(x_n + h_n \sum_{j=1}^{n_{\mathrm{s}}} a_{n_{\mathrm{s}},j} v_{n,j}, \theta_{n,n_{\mathrm{s}}}, \alpha_{n,n_{\mathrm{s}}}, \lambda_{n,n_{\mathrm{s}}}^{\mathrm{p}}, \lambda_{n,n_{\mathrm{s}}}^{\mathrm{n}}) \\ x_n^{\mathrm{next}} - x_n - h_n \sum_{i=1}^{n_{\mathrm{s}}} b_i v_{n,i} \end{bmatrix}. \tag{33}$$

Next, we summarize the equations for all N_{FE} finite elements over the entire interval $[0, T]$ in a discrete-time system format. To make it more manageable, we use some additional shorthand notation and group all variables of all finite elements for a single control interval into the following vectors: $\mathbf{x} = (x_0, \dots, x_{N_{\mathrm{FE}}}) \in \mathbb{R}^{(N_{\mathrm{FE}}+1)n_x}$, $\mathbf{V} = (V_0, \dots, V_{N_{\mathrm{FE}}-1}) \in \mathbb{R}^{N_{\mathrm{FE}} n_{\mathrm{s}} n_x}$ and $\mathbf{h} := (h_0, \dots, h_{N_{\mathrm{FE}}-1}) \in \mathbb{R}^{N_{\mathrm{FE}}}$. Recall that the simple continuity condition $x_{n+1} = x_n^{\mathrm{next}}$ holds. We collect the stage values of the Filippov multipliers in the vector $\mathbf{\Theta} = (\Theta_0, \dots, \Theta_{N_{\mathrm{FE}}-1}) \in \mathbb{R}^{n_\theta}$ and $n_\theta = N_{\mathrm{FE}} n_{\mathrm{s}} n_f$. Similarly, we group the stage values of the algebraic variables in the vectors $\mathbf{A}, \mathbf{\Lambda}^{\mathrm{p}}, \mathbf{\Lambda}^{\mathrm{n}} \in \mathbb{R}^{n_\alpha}$, where $n_\alpha = N_{\mathrm{FE}} n_{\mathrm{s}} n_c$. Finally, we collect all internal variables in the vector $\mathbf{Z} = (\mathbf{x}, \mathbf{V}, \mathbf{\Theta}, \mathbf{A}, \mathbf{\Lambda}^{\mathrm{p}}, \mathbf{\Lambda}^{\mathrm{n}}) \in \mathbb{R}^{n_{\mathbf{Z}}}$, where $n_{\mathbf{Z}} = (N_{\mathrm{FE}} + 1)n_x + N_{\mathrm{FE}} n_{\mathrm{s}} n_x + n_\theta + 3n_\alpha$.

All computations over a single control interval of the standard discretization (denoted by the subscript std in the corresponding functions) are summarized in the following equations:

$$s_1 = F_{\mathrm{std}}(\mathbf{Z}), \tag{34a}$$

$$0 = G_{\mathrm{std}}(\mathbf{Z}, \mathbf{h}, s_0, q), \tag{34b}$$

where $s_1 \in \mathbb{R}^{n_x}$ is the approximation of $x(T)$ and

$$F_{\mathrm{std}}(\mathbf{Z}) = x_{N_{\mathrm{FE}}},$$

$$G_{\mathrm{std}}(\mathbf{Z}, \mathbf{h}, s_0, q) := \begin{bmatrix} x_0 - s_0 \\ G_{\mathrm{rk}}(x_1, Z_0, h_0, q) \\ \vdots \\ G_{\mathrm{rk}}(x_{N_{\mathrm{FE}}}, Z_{N_{\mathrm{FE}}-1}, h_{N_{\mathrm{FE}}-1}, q) \end{bmatrix}.$$

In (34), $\mathbf{h}$ is a given parameter and implicitly fixes the discretization grid. In contrast to standard RK discretizations, we will now proceed by letting $\mathbf{h}$ be degrees of freedom and introduce the cross-complementarity conditions.

4.3. Cross complementarity

For ease of exposition, suppose that the underlying RK scheme satisfies $c_{n_{\mathrm{s}}} = 1$. This means that the right boundary point of a finite element is a stage point, since $t_{n+1} = t_n + c_{n_{\mathrm{s}}} h_n$. For example, this assumption is satisfied by Radau and Lobatto schemes [36]. We provide extensions for $c_{n_{\mathrm{s}}} \neq 1$ at the end of the section.

The goal is to derive additional constraints that will allow active-set changes only at the boundary of a finite element, and compare left and middle plots in Fig. 3. Moreover, in this case, the step size h_n should adapt such that the switch is detected exactly. Recall that for the step reformulation at every stage point we have the complementarity conditions:

$$0 \leq \lambda_{n,m}^{\mathrm{n}} \perp \alpha_{n,m} \geq 0, \qquad n = 1, \dots, N_{\mathrm{FE}}, m = 1, \dots, n_{\mathrm{s}}, \tag{35a}$$

$$0 \leq \lambda_{n,m}^{\mathrm{p}} \perp e - \alpha_{n,m} \geq 0, \qquad \text{for all } n = 1, \dots, N_{\mathrm{FE}}, m = 1, \dots, n_{\mathrm{s}}, \tag{35b}$$

where n is the index of the finite elements (integration interval) and m the index of the RK-stage. We exploit the continuity of the Lagrange multipliers λ^{p} and λ^{n}, cf. Lemma 3. We regard the boundary values of the approximation of λ^{p} and λ^{n} on an interval $[t_n, t_{n+1}]$. They are denoted by $\lambda_{n,0}^{\mathrm{p}}$, $\lambda_{n,0}^{\mathrm{n}}$ (which we define below) at t_n and $\lambda_{n,n_{\mathrm{s}}}^{\mathrm{p}}$, $\lambda_{n,n_{\mathrm{s}}}^{\mathrm{n}}$ at t_{n+1}.

Next, we impose a continuity condition for the discrete-time versions of λ^{p} and λ^{n} for all $n \in \{0, \dots, N_{\mathrm{FE}} - 1\}$:

$$\lambda^{\mathrm{p}}_{n+1,0} = \lambda^{\mathrm{p}}_{n,n_{\mathrm{s}}},\ \lambda^{\mathrm{n}}_{n+1,0} = \lambda^{\mathrm{n}}_{n,n_{\mathrm{s}}}. \tag{36}$$

Note that $\lambda^{\mathrm{p}}_{0,0}$ and $\lambda^{\mathrm{n}}_{0,0}$ are not defined via (36), as we do not have a preceding finite element for $n = 0$. Nevertheless, they are crucial for determining the active set in the first finite element. They are not degrees of freedom but parameters determined by a given s_0. Using (17) we obtain $\lambda^{\mathrm{p}}_{0,0} = \max(\psi(s_0), 0)$ and $\lambda^{\mathrm{n}}_{0,0} = -\min(\psi(s_0), 0)$.

We have seen in Section 3.1 that, due to continuity, $\lambda^{\mathrm{p}}_i(t)$ and $\lambda^{\mathrm{n}}_i(t)$ must be zero at an active set change, see also Fig. 2. Moreover, on an interval $t \in (t_n, t_{n+1})$ with a fixed active set, the components of these multipliers are either zero or positive on the whole interval. The discrete-time counterparts, *i.e.*, the stage values $\lambda^{\mathrm{p}}_{n,m}$ and $\lambda^{\mathrm{n}}_{n,m}$ should satisfy these properties as well. We achieve these goals via the cross complementarity conditions, which read as, for all $n \in \{0, \dots, N_{\mathrm{FE}}-1\}$:

$$0 = \operatorname{diag}(\lambda^{\mathrm{n}}_{n,m'})\alpha_{n,m}, \qquad m = 1, \dots, n_{\mathrm{s}},\ m' = 0, \dots, n_{\mathrm{s}},\ m \neq m', \tag{37a}$$

$$0 = \operatorname{diag}(\lambda^{\mathrm{p}}_{n,m'})(e - \alpha_{n,m}), \qquad m = 1, \dots, n_{\mathrm{s}},\ m' = 0, \dots, n_{\mathrm{s}},\ m \neq m'. \tag{37b}$$

In contrast to (36), here we have conditions relating variables corresponding to different RK stages within a finite element. Eq. (37) extends the complementarity conditions for the same RK-stage, *i.e.*, for $m = m'$, which are part of the standard RK equations, cf. (35). Some of the claims about the constraints (37) are formalized by the next lemma. Recall that in our notation $\alpha_{n,m,j}$ is the jth component of the vector $\alpha_{n,m}$.

Lemma 9. *Regard a fixed $n \in \{0, \dots, N_{\mathrm{FE}} - 1\}$ and a fixed $j \in \mathcal{C}$. If any $\alpha_{n,m,j}$ with $m \in \{1, \dots, n_{\mathrm{s}}\}$ is positive, then all $\lambda^{\mathrm{n}}_{n,m',j}$ with $m' \in \{0, \dots, n_{\mathrm{s}}\}$ must be zero. Conversely, if any $\lambda^{\mathrm{n}}_{n,m',j}$ is positive, then all $\alpha_{n,m,j}$ are zero.*

Proof. Let $\alpha_{n,m,i}$ be positive, and suppose $\lambda^{\mathrm{n}}_{n,j,i} = 0$ and $\lambda^{\mathrm{n}}_{n,k,i} > 0$ for some $k, j \in \{0, \dots, n_{\mathrm{s}}\}, k \neq j$, then $\alpha_{n,m,i}\lambda^{\mathrm{n}}_{n,k,i} > 0$, which violates (37), thus all $\lambda^{\mathrm{n}}_{n,m',i} = 0,\ m' \in \{0, \dots, n_{\mathrm{s}}\}$. The converse is proven similarly. □

An according statement holds for $\lambda^{\mathrm{p}}_{n,m}$ and $(e - \alpha_{n,m})$.

Lemma 10. *Regard a fixed $n \in \{0, \dots, N_{\mathrm{FE}}-1\}$ and a fixed $j \in \mathcal{C}$. If any $1 - \alpha_{n,m,j}$ with $m \in \{1, \dots, n_{\mathrm{s}}\}$ is positive, then all $\lambda^{\mathrm{p}}_{n,m',j}$ with $m' \in \{0, \dots, n_{\mathrm{s}}\}$ must be zero. Conversely, if any $\lambda^{\mathrm{p}}_{n,m',j}$ is positive, then all $1 - \alpha_{n,m,j}$ are zero.*

It is now left to discuss why the boundary points $\lambda^{\mathrm{p}}_{n+1,0} = \lambda^{\mathrm{p}}_{n,n_{\mathrm{s}}}$ and $\lambda^{\mathrm{n}}_{n+1,0} = \lambda^{\mathrm{n}}_{n,n_{\mathrm{s}}}$ of the previous finite element are included in the cross complementarity conditions (37). It turns out, they are the key to switch detection. A consequence of Lemmata 9 and 10 is that, if the active set changes in the jth component between the nth and $(n+1)$th finite element, then it must hold that $\lambda^{\mathrm{p}}_{n,n_{\mathrm{s}},j} = \lambda^{\mathrm{p}}_{n+1,0,j} = 0$ and $\lambda^{\mathrm{n}}_{n,n_{\mathrm{s}},j} = \lambda^{\mathrm{n}}_{n+1,0,j} = 0$. Since $x^{\mathrm{next}}_n = x_{n+1}$, we have from (33) the condition

$$\psi_j(x_{n+1}) = 0,$$

which defines exactly the switching surface between two regions. Therefore, we have implicitly a constraint that forces h_n to adapt such that the switch is detected exactly.

Given $a, b \in \mathbb{R}^p$, the complementarity conditions $0 \leq a \perp b \geq 0$ mean that $a, b \geq 0$ and $a_i b_i = 0, i = 1, \dots, p$. Due to the non-negativity of a and b, the last conditions can be replaced by $a^\top b = 0$. Similar aggregations can be made with the cross complementarity conditions. For clarity, we stated (37) in their most sparse form, without any aggregation. However, the nonnegativity of $\alpha_{n,m}$, $\lambda^{\mathrm{p}}_{n,m}$ and $\lambda^{\mathrm{n}}_{n,m}$ allows more compact forms. In the sequel, we use a formulation such that, together with the constraint $\sum_{n=0}^{N_{\mathrm{FE}}-1} h_n = T$, we have the same number of new equations as new degrees of freedom by varying h_n. Thus, we combine the constraints of two neighboring finite elements and have a compact formulation

$$G_{\mathrm{cross}}(\mathbf{A}, \mathbf{\Lambda^p}, \mathbf{\Lambda^n}) = 0, \tag{38}$$

whose entries are for all $n \in \{0, \dots, N_{\mathrm{FE}} - 2\}$ given by

$$G_{\mathrm{cross},n}(\mathbf{A}, \mathbf{\Lambda^p}, \mathbf{\Lambda^n}) = \sum_{k=n}^{n+1}\Big(\sum_{m=1}^{n_{\mathrm{s}}} \sum_{\substack{m'=0,\\ m'\neq m}}^{n_{\mathrm{s}}} \alpha_{k,m}^\top \lambda^{\mathrm{n}}_{k,m'} + (e - \alpha_{k,m})^\top \lambda^{\mathrm{p}}_{k,m'}\Big).$$

We remind the reader that we use this seemingly complicated form to obtain a square system of equations. This simplifies the study of the well-posedness of the FESD equations later. However, in an implementation one can use any of the equivalent more sparse, or dense, formulations. Many possible variants are implemented in `nosnoc` [39], and the user can control the sparsity.

4.4. Step size equilibration

To complete the derivation of the FESD method for the Heaviside step reformulation (32), we need to derive the step equilibration conditions. Here, *step* refers to the integration step size and should not be confused with the set-valued Heaviside step function.

If no active-set changes happen, the cross complementarity constraints (37) are implied by the standard complementarity conditions (35). This can easily be verified by looking at the stage point in the middle plot of Fig. 3. Therefore, we end up with a

Table 2
Overview of switching cases for the step size equilibration.

Switching case	$\sigma_n^{\lambda^{\mathrm{n}},\mathrm{B}}$	$\sigma_n^{\lambda^{\mathrm{n}},\mathrm{F}}$	$\sigma_n^{\lambda^{\mathrm{p}},\mathrm{B}}$	$\sigma_n^{\lambda^{\mathrm{p}},\mathrm{F}}$	$\pi_n^{\lambda^{\mathrm{n}}}$	$\pi_n^{\lambda^{\mathrm{p}}}$	υ_n
No switch	1	1	0	0	1	0	1
Crossing	1	0	0	1	0	0	0
Entering sliding mode	1	0	0	0	0	0	0
Leaving sliding mode	0	0	0	1	0	0	0
Spontaneous switch	0	0	0	1	0	0	0

system of equations with more degrees of freedom than conditions. The step equilibration constraints aim to remove the degrees of freedom in the appropriate h_n if no switches happen. This results in a piecewise uniform discretization grid for the differential and algebraic states on the considered time interval.

We achieve the goals outlined above via the equation:

$$0 = G_{\mathrm{eq}}(\mathbf{h}, \mathbf{\Lambda}, \mathbf{\Lambda^p}, \mathbf{\Lambda^n}) := \begin{bmatrix} (h_1 - h_0)\eta_1(\mathbf{\Lambda}, \mathbf{\Lambda^p}, \mathbf{\Lambda^n}) \\ \vdots \\ (h_{N_{\mathrm{FE}}-1} - h_{N_{\mathrm{FE}}-2})\eta_{N_{\mathrm{FE}}-1}(\mathbf{\Lambda}, \mathbf{\Lambda^p}, \mathbf{\Lambda^n}) \end{bmatrix}, \tag{39}$$

where η_n is an indicator function that is zero only if a switch occurs, otherwise its value is strictly positive. This provides a condition that removes the spurious degrees of freedom. In the remainder of this section, we derive a possible expression for η_n.

The derivations below are motivated by the following facts. Let t_n be a switching point with $\psi_j(x(t_n)) = 0$ for some $j \in \mathcal{C}$. Consequently, it holds that $\lambda_j^{\mathrm{p}}(t_n) = \lambda_j^{\mathrm{n}}(t_n) = 0$. If, for example, a switch occurs at t_n such that $\psi(x(t_n^-)) < 0$ and $\psi(x(t_n^+)) > 0$, we have that $\dot{\lambda}_j^{\mathrm{n}}(t_n^-) < 0$, $\dot{\lambda}_j^{\mathrm{n}}(t_n^-) = 0$ and that $\dot{\lambda}_j^{\mathrm{p}}(t_n^+) = 0$, $\lambda_j^{\mathrm{p}}(t_n^+) > 0$. This can be verified by looking at the plots in Fig. 2. The symmetric case is possible as well. The absolute values of these directional derivatives help us to encode the switching logic.

Now, instead of looking at the time derivatives, in the discrete-time case, we exploit the non-negativity of $\lambda_{n,m}^{\mathrm{p}}$, $\lambda_{n,m}^{\mathrm{n}}$, and the fact no switches occur within a finite element. For $n \in \{1, \dots, N_{\mathrm{FE}} - 1\}$, we define the following backward and forward sums of the stage values over the neighboring finite elements $[t_{n-1}, t_n]$ and $[t_n, t_{n+1}]$:

$$\sigma_n^{\lambda^{\mathrm{p}},\mathrm{B}} = \sum_{m=0}^{n_{\mathrm{s}}} \lambda_{n-1,m}^{\mathrm{p}}, \quad \sigma_n^{\lambda^{\mathrm{p}},\mathrm{F}} = \sum_{m=0}^{n_{\mathrm{s}}} \lambda_{n,m}^{\mathrm{p}},$$
$$\sigma_n^{\lambda^{\mathrm{n}},\mathrm{B}} = \sum_{m=0}^{n_{\mathrm{s}}} \lambda_{n-1,m}^{\mathrm{n}}, \quad \sigma_n^{\lambda^{\mathrm{n}},\mathrm{F}} = \sum_{m=0}^{n_{\mathrm{s}}} \lambda_{n,m}^{\mathrm{n}}.$$

They are zero if the left and right time derivatives are zero, respectively. Likewise, they are positive when the left and right time derivatives are nonzero.

Moreover, for all $n \in \{1, \dots, N_{\mathrm{FE}} - 1$ we define the following variables to summarize the logical dependencies:

$$\pi_n^{\lambda^{\mathrm{n}}} = \mathrm{diag}(\sigma_n^{\lambda^{\mathrm{n}},\mathrm{B}})\sigma_n^{\lambda^{\mathrm{n}},\mathrm{F}} \in \mathbb{R}^{n_\psi},$$
$$\pi_n^{\lambda^{\mathrm{p}}} = \mathrm{diag}(\sigma_n^{\lambda^{\mathrm{p}},\mathrm{B}})\sigma_n^{\lambda^{\mathrm{p}},\mathrm{F}} \in \mathbb{R}^{n_\psi},$$

and

$$\upsilon_n = \pi_n^{\lambda^{\mathrm{n}}} + \pi_n^{\lambda^{\mathrm{p}}} \in \mathbb{R}^{n_\psi}.$$

The switching cases and sign logic are summarized in Table 2. For readability, we put in the table a one if a variable is positive and a zero if it is zero. Let us discuss how the variables above encode the switching logic, and for this purpose, we regard the jth switching functions $\psi_j(x)$. If no switch occurs, and for example, we have that $\psi_j(x(t)) < 0$ during the regard time interval, it follows that $\lambda_j^{\mathrm{n}}(t) > 0$ and $\lambda_j^{\mathrm{p}}(t) = 0$ during this time interval. In the discrete time setting, we have $\sigma_{n,j}^{\lambda^{\mathrm{n}},\mathrm{B}}, \sigma_{n,j}^{\lambda^{\mathrm{n}},\mathrm{F}} > 0$ and $\sigma_{n,j}^{\lambda^{\mathrm{p}},\mathrm{B}} = \sigma_{n,j}^{\lambda^{\mathrm{p}},\mathrm{F}} = 0$. This means that $\pi_{n,j}^{\lambda^{\mathrm{n}}} > 0$, $\pi_{n,j}^{\lambda^{\mathrm{n}}} = 0$ and $\upsilon_{n,j} > 0$. It can be seen that the symmetric case with $\psi_j(x(t)) > 0$ leads also to $\upsilon_{n,j} > 0$, hence we do not enumerate all symmetric cases in Table 2.

On the other hand, if we have a switch of the crossing type (cf. top plots in Fig. 2 with $\psi_j(x(t)) < 0$ for $t < t_{\mathrm{s},n}$ and $\psi_j(x(t)) > 0$ for $t > t_{\mathrm{s},n}$, it follows that $\lambda_j^{\mathrm{n}}(t) > 0, \lambda_j^{\mathrm{p}}(t) = 0$ for $t < t_{\mathrm{s},n}$ and $\lambda_j^{\mathrm{n}}(t) = 0, \lambda_j^{\mathrm{p}}(t) > 0$ for $t > t_{\mathrm{s},n}$. In the discrete-time setting we obtain the sign pattern as in the second row of Table 2, with $\upsilon_{n,j} = 0$.

In general, if there is an active-set change in the jth complementarity pair, then at most one of the jth components of $\sigma_n^{\lambda^{\mathrm{p}},\mathrm{B}}$ and $\sigma_n^{\lambda^{\mathrm{p}},\mathrm{F}}$, or $\sigma_n^{\lambda^{\mathrm{n}},\mathrm{B}}$ and $\sigma_n^{\lambda^{\mathrm{n}},\mathrm{F}}$ is nonzero. In these cases, we obtain that $\upsilon_{n,j} = 0$, and if now switch happens we have that $\upsilon_{n,j} > 0$.

In other words, $\upsilon_{n,j}$ is only zero if there is an active-set change in the jth complementarity pair at t_n, otherwise, it is strictly positive. We summarize all logical relations for all switching functions into a single scalar expression and define

$$\eta_n(\mathbf{\Lambda}, \mathbf{\Lambda^p}, \mathbf{\Lambda^n}) := \prod_{i=1}^{n_\psi} (\upsilon_n)_i.$$

It is zero only if an active-set change happens at the boundary point t_n, otherwise, it is strictly positive.

4.5. The FESD discretization

We have now introduced all extensions needed to pass from a standard RK discretization (34) to the FESD discretization for the step reformulation. With a slight abuse of notation, we collect all equations in a discrete-time system form:

$$s_1 = F_{\mathrm{fesd}}(\mathbf{Z}), \tag{40a}$$

$$0 = G_{\mathrm{fesd}}(\mathbf{Z}, \mathbf{h}, s_0, q, T), \tag{40b}$$

where $F_{\mathrm{fesd}}(\mathbf{x}) = x_{N_{\mathrm{FE}}}$ is the state transition map and $G_{\mathrm{fesd}}(\mathbf{x}, \mathbf{h}, \mathbf{Z}, q, T)$ collects all other internal computations including all RK steps within the regarded time interval:

$$G_{\mathrm{fesd}}(\mathbf{Z}, \mathbf{h}, s_0, q, T) := \begin{bmatrix} G_{\mathrm{std}}(\mathbf{Z}, \mathbf{h}, s_0, q, T) \\ G_{\mathrm{cross}}(\mathbf{A}, \mathbf{\Lambda^p}, \mathbf{\Lambda^n}) \\ G_{\mathrm{eq}}(\mathbf{h}, \mathbf{A}, \mathbf{\Lambda^p}, \mathbf{\Lambda^n}) \\ \sum\limits_{n=0}^{N_{\mathrm{FE}}-1} h_n - T \end{bmatrix}. \tag{41}$$

Here, the control variable q, the horizon length T, and the initial value s_0 are given parameters.

Remark on RK methods with $c_{n_s} \neq 1$. The extension for the case of an RK method with $c_{n_s} \neq 1$ follows similar lines as in Stewart's formulation [14]. We have that $t_n + c_{n_s} h_n < t_{n+1}$. Hence, the variables $\lambda^{\mathrm{p}}_{n,n_s}$ and $\lambda^{\mathrm{n}}_{n,n_s}$ do not correspond to the boundary values $\lambda^{\mathrm{n}}(t_{n+1})$ and $\lambda^{\mathrm{p}}(t_{n+1})$ anymore. We denote the true boundary points by $\lambda^{\mathrm{p}}_{n,n_s+1}$ and $\lambda^{\mathrm{n}}_{n,n_s+1}$. They can be computed from the KKT conditions of the step reformulation LP (15). For all $n \in \{1, \dots, N_{\mathrm{FE}} - 2\}$ we have

$$\begin{bmatrix} \psi(x_{n+1}) - \lambda^{\mathrm{p}}_{n,n_s+1} + \lambda^{\mathrm{n}}_{n,n_s+1} \\ \Psi(\lambda^{\mathrm{n}}_{n,n_s+1}, \alpha_{n,n_s+1}) \\ \Psi(\lambda^{\mathrm{p}}_{n,n_s+1}, e - \alpha_{n,n_s+1}) \end{bmatrix} = 0. \tag{42}$$

These equations are appended to the FESD equation in (41).

However, to make the switch detection work, we must update the continuity conditions for the discrete-time versions of the Lagrange multipliers and adapt the cross-complementarity conditions accordingly. For all $n = \{0, \dots, N_{\mathrm{FE}} - 1\}$, (36) is replaced by:

$$\lambda^{\mathrm{p}}_{n,n_s+1} = \lambda^{\mathrm{p}}_{n+1,0}, \ \lambda^{\mathrm{n}}_{n,n_s+1} = \lambda^{\mathrm{n}}_{n+1,0}. \tag{43}$$

We append to the vectors $\mathbf{A}, \mathbf{\Lambda^p}$ and $\mathbf{\Lambda^n}$ the new variables $\alpha_{n,n_s+1}, \lambda^{\mathrm{p}}_{n,n_s+1}$ and $\lambda^{\mathrm{n}}_{n,n_s+1}$ accordingly. For the whole control interval, we have in total $3(N_{\mathrm{FE}} - 1)n_c$ new variables. It is only left to state the modified cross complementarity conditions, including the expressions' $(n_s + 1)$-th point. More explicitly, the nth component of (38) reads now for all $n \in \{0, \dots, N_{\mathrm{FE}} - 2\}$ as

$$G_{\mathrm{cross},n}(\mathbf{A}, \mathbf{\Lambda^p}, \mathbf{\Lambda^n}) = \sum_{k=n}^{n+1} \sum_{m=1}^{n_s} \sum_{\substack{m'=0,\\ m' \neq m}}^{n_s+1} \alpha_{k,m}^{\top} \lambda^{\mathrm{n}}_{k,m'} + (e - \alpha_{k,m})^{\top} \lambda^{\mathrm{p}}_{k,m'}.$$

4.6. Discretizing optimal control problems with FESD

Regard an optimal control problem subject to a nonsmooth dynamical system (2): of the following form:

$$\min_{x(\cdot),u(\cdot)} \quad \int_0^T L(x(t), u(t))\mathrm{d}t + R(x(T)) \tag{44a}$$

$$\text{s.t.} \quad x(0) = s_0, \tag{44b}$$

$$\dot{x}(t) \in \mathcal{F}(x(t), u(t), \Gamma(\psi(x(t)))), \qquad \text{for a.a. } t \in [0,T], \tag{44c}$$

$$0 \geq G_{\mathrm{path}}(x(t), u(t)), \qquad t \in [0, T], \tag{44d}$$

$$0 \geq G_{\mathrm{terminal}}(x(T)), \tag{44e}$$

where $L : \mathbb{R}^{n_x} \times \mathbb{R}^{n_u} \to \mathbb{R}$ is the running cost and $R : \mathbb{R}^{n_x} \to \mathbb{R}$ is the terminal cost, $s_0 \in \mathbb{R}^{n_x}$ is a given initial value. The path and terminal constraints are grouped into the functions $G_{\mathrm{path}} : \mathbb{R}^{n_x} \times \mathbb{R}^{n_u} \to \mathbb{R}^{n_{G_{\mathrm{p}}}}$ and $G_{\mathrm{terminal}} : \mathbb{R}^{n_x} \to \mathbb{R}^{n_{G_{\mathrm{t}}}}$, respectively.

In this paper, we consider a direct approach [38], *i.e.*, we first discretize the continuous-time OCP (44) and then solve a finite-dimensional nonlinear program (NLP). Here we discretize the OCP using the FESD method. First, for the discretization of the control function, we consider $N \geq 1$ control intervals of equal length, indexed by k. We use a piecewise constant control discretization, where the control variables are collected $\mathbf{q} = (q_0, \dots, q_{N-1}) \in \mathbb{R}^{N n_u}$. Such a discretization is typically used in feedback control, but extensions to more sophisticated control are straightforward. All internal variables are additionally indexed by k.

Second, we discretize the cost function (44a) and the dynamics (44c). To apply the FESD method, the differential inclusion in (44c) is transformed into an equivalent DCS, as described in Section 2.5. The DCS, now of the from of (22), is discretized with the

FESD method summarized in the previous section. For each control interval k we use (40) with N_{FE} internal finite elements. The state values at the control interval boundaries are grouped in the vector $\mathbf{s} = (s_0, \dots, s_N) \in \mathbb{R}^{(N+1)n_x}$. In $\mathcal{Z} = (\mathbf{Z}_0, \dots, \mathbf{Z}_{N-1})$ we collect all internal variables and in $\mathcal{H} = (\mathbf{h}_0, \dots, \mathbf{h}_{N-1})$ all step sizes. For the cost discretization, one can derive a quadrature formula [38, Chapter 8], or introduce a scalar quadrature state $\dot{\ell}(t) = L(x(t), u(t)), \ell(0) = 0$, integrate it with the dynamics equations, and use $\ell(T)$ in the objective, which approximates the integral term in (44a).

Third, the path constraints are relaxed and evaluated only at the control interval boundary points, *i.e.*, at the variables s_k and $q_k, k = 0, \dots, N-1$. If necessary, the path constraint can be evaluated on a finer grid using the values computed in the internal integration intervals within a control interval or at the RK stage points. The terminal constraint and cost are simply evaluated at s_N, which is an approximation of $x(T)$. In summary, we obtain a discrete-time variant of (44):

$$\min_{\mathbf{s},\mathbf{q},\mathcal{Z},\mathcal{H}} \quad \sum_{k=0}^{N-1} \hat{L}(s_k, \mathbf{x}_k, q_k) + R(s_N) \tag{45a}$$

$$\text{s.t.} \quad s_0 = \bar{x}_0, \tag{45b}$$

$$s_{k+1} = F_{\mathrm{fesd}}(\mathbf{x}_k), \qquad k = 0, \dots, N-1, \tag{45c}$$

$$0 = G_{\mathrm{fesd}}(\mathbf{x}_k, \mathbf{Z}_k, q_k), \qquad k = 0, \dots, N-1, \tag{45d}$$

$$0 \geq G_{\mathrm{ineq}}(s_k, q_k), \qquad k = 0, \dots, N-1, \tag{45e}$$

$$0 \geq G_{\mathrm{terminal}}(s_N), \tag{45f}$$

where $\hat{L} : \mathbb{R}^{n_x} \times \mathbb{R}^{(N_{\mathrm{FE}}+1)n_s n_x} \times \mathbb{R}^{n_u} \to \mathbb{R}$ is the discretized running cost.

Due to the complementarity constraints in the FESD discretization, this NLP is a mathematical program with complementarity constraints. These are degenerate NLPs since the complementarity constraints lead to the violation of standard constraint qualifications at all feasible points. In practice, they can often be efficiently solved by solving a sequence of related and relaxed NLPs within a homotopy approach. Such an approach with some of the standard reformulations [21,22,40] is implemented in `nosnoc`. A survey and comparison of state-of-the-art methods for solving NLPs in nonsmooth optimal control problems is given in [41]. All these homotopy solution methods are implemented in `nosnoc` [39]. In practice, Scholtes' relaxations [22] together with `IPOPT` [42], called via its `CasADi` interface [43], often work very well. We use this method in the numerical experiments in the paper when solving problems like (45).

5. Convergence theory of FESD for the step representation

In this section, we present the main convergence result of the FESD method for the step representation outlined in (40). Specifically, we show that: (1) the solutions to the FESD problem are locally isolated; (2) both the solution approximations and (3) the numerical sensitivities obtained via the FESD method converge with the same order of accuracy as the underlying RK method. The proofs are similar to those used in Stewart's case in [14], hence, we will not repeat them here. The main difference is in the assumptions we make.

5.1. Main assumptions

We start by stating all assumptions. The first assumption relates to the underlying RK methods [44]:

Assumption 11 (*Runge–Kutta Method*)**.** A Butcher tableau with the entries $a_{i,j}, b_i$ and c_i, $i, j \in \{1, \dots, n_{\mathrm{s}}\}$ related to an n_{s}-stage Runge–Kutta (RK) method is used in the FESD (40). Moreover, we assume that:

(a) If the same RK method is applied to the differential algebraic Eq. (29) on an interval $[t_a, t_b]$, with a fixed active set, it has a global accuracy of $O(h^p)$ for the differential states, where p is a positive integer.
(b) The RK equations applied to (29) have a locally isolated solution for a sufficiently small $h_n > 0$.

This assumption describes the properties of an underlying RK method used in FESD when applied to a smooth ODE or DAE. Both requirements (a) and (b) are standard and are satisfied by many RK methods used in practice, *e.g.* Radau IIA or Gauss–Legendre methods, cf. [36]. The second assumption concerns the existence of solutions to the FESD problem outlined in (40).

Assumption 12 (*Solution Existence*)**.** For given parameters s_0, q and T, there exists a solution to the FESD problem (40), such that for all $n \in \{0, \dots, N_{\mathrm{FE}} - 1\}$ it holds that $h_n > 0$.

The problem (40) is a nonlinear complementarity problem [26]. The proof of the existence of solutions for the standard RK Eqs. (34) can probably be done with standard tools from complementarity theory [26], as it was done for example in [1, Proposition 15] for the implicit Euler method. With the additional cross complementarity and step equilibration conditions appearing in (40), the same proof technique is no longer applicable. The existence of solutions is still an open problem. Motivated by empirical observations, we assume the existence of solutions in this paper.

The next assumption is slightly more technical and relates to the regularity of the problem under consideration.

Assumption 13 (*Regularity*)**.** Given the complementarity pairs $\Psi(\alpha_{n,m}, \lambda^{\mathrm{n}}_{n,m}) = 0$ and $\Psi(e - \alpha_{n,m}, \lambda^{\mathrm{p}}_{n,m}) = 0$, for all $n \in \{0, \dots N_{\mathrm{FE}} - 1\}$ there exists an $m \in \{1, \dots, n_{\mathrm{s}}\}$ and $i \in \{1, \dots, n_f\}$, such that the strict complementarity property holds, *i.e.*, $\alpha_{n,m,i} + \lambda^{\mathrm{n}}_{n,m,i} > 0$ and $e - \alpha_{n,m,i} + \lambda^{\mathrm{p}}_{n,m,i} > 0$. Moreover, for the RK Eqs. (33), it holds for all $n \in \{0, \dots N_{\mathrm{FE}} - 1\}$, that at least one entry of the vector $\nabla_{h_n} G_{\mathrm{rk}}(x_{n+1}, Z_n, h_n, q)$ is nonzero.

This assumption is made to ensure the correct rank of partial Jacobians of (40) (with a fixed active set). It is used to prove the local uniqueness of solutions to the FESD problem, and similar assumptions are made in [14]. The first part of the assumption requires that at least one complementarity pair on stage points within an integration interval (finite element) satisfies strict complementarity. Looking at the common switching cases in Fig. 2, one can see that this assumption always holds and is thus not restrictive. The second part requires that at least one term, $G_{\mathrm{rk}}(x_{n+1}, Z_n, h_n, q)$ multiplied by h_n, is nonzero. Both assumptions can be checked computationally once a candidate solution has been computed.

Before starting the final assumption, let us introduce some notation. We use a suitable interpolation scheme to construct a continuous-time approximation of the solution based on the stage values obtained by solving Eq. (40). We denote the continuous-time approximation on every finite element $[t_n, t_{n+1}]$ by $\hat{x}_n(t; h_n)$. To approximate the solution over the entire time interval $[0, T]$, we append the local approximations from each finite element:

$$\hat{x}_h(t) = \hat{x}_n(t; h_n) \text{ if } t \in [t_n, t_{n+1}], \tag{46}$$

where $h = \max_{n \in \{0, \dots N_{\mathrm{FE}}-1\}} h_n$. The subscript h for a solution approximation $\hat{x}_h(t)$ indicates that we get different solution approximations on the entire interval $[0, T]$ as the maximum step size changes, and not that the entire approximation is parameterized by a single step size h. As the maximum step size shrinks, we expect $\hat{x}_h(t)$ to converge to an exact solution $x(t)$. The set of all grid points is defined as $\mathcal{G} = \{t_0, \dots, t_{N_{\mathrm{FE}}}\}$. We can also use this approach to construct continuous-time representations for the algebraic variables, which we denote by $\hat{\theta}_h, \hat{\alpha}_h, \hat{\lambda}^{\mathrm{n}}_h$ and $\hat{\lambda}^{\mathrm{p}}_h$. The nth switching point of the true solution is denoted by $t_{\mathrm{s},n}$ and one corresponding to a solution approximation by $\hat{t}_{\mathrm{s},n}$. Similarly, the active sets (cf. (12)) of the solution approximation are denoted by $\mathcal{I}(\hat{x}_h(t)) = \hat{\mathcal{I}}_n,\ t \in (\hat{t}_{\mathrm{s},n}, \hat{t}_{\mathrm{s},n+1})$ and the active set at switching point $\hat{t}_{\mathrm{s},n}$ by $\mathcal{I}(\hat{x}_h(\hat{t}_{\mathrm{s},n})) = \hat{\mathcal{I}}^0_n$.

Assumption 14. Let $\mathcal{I}^0_n$ be the active set at switching point $x(t_{\mathrm{s},n})$ of true solution and $\hat{\mathcal{I}}^0_n$ the active set at the switching point $\hat{x}(\hat{t}_{\mathrm{s},n})$ a solution approximation. If $\hat{t}_{\mathrm{s},n}$ is sufficiently close to $t_{\mathrm{s},n}$ and $\mathcal{I}^0_n = \hat{\mathcal{I}}^0_n$, then $\mathcal{I}_{n+1} = \hat{\mathcal{I}}_{n+1}$. Furthermore, if there are several possible new active sets, they are identical for both the true solution and its approximation.

This assumption requires that a given solution approximation and the corresponding true solution predict the same active sets. In other words, the solution approximation enters the same region or sliding mode as the true solution after a switching event. In Stewart's reformulation, this statement can be directly proved using an auxiliary linear complementarity problem constructed with the problem data [23]. In the case of DCS (22), such auxiliary problems are not available and the property is assumed directly. The requirement that a sufficiently good solution approximation predicts the same active set as the true solution is needed to prove the high accuracy convergence of a switch detection method such as FESD, cf. [23, Theorem 4.3] and [14, Theorem 16].

5.2. Solutions of the FESD problem are locally isolated

In this section, we present a theorem on the regularity of solutions to the FESD problem (40). For brevity, we focus on the case where $c_{n_{\mathrm{s}}} = 1$. For the reader's convenience we restate the FESD problem

$$G_{\mathrm{fesd}}(\mathbf{Z}, \mathbf{h}, s_0, q, T) := \begin{bmatrix} G_{\mathrm{std}}(\mathbf{Z}, \mathbf{h}, s_0, q, T) \\ G_{\mathrm{cross}}(\mathbf{A}, \boldsymbol{\Lambda}^{\mathbf{p}}, \boldsymbol{\Lambda}^{\mathbf{n}}) \\ G_{\mathrm{eq}}(\mathbf{h}, \mathbf{A}, \boldsymbol{\Lambda}^{\mathbf{p}}, \boldsymbol{\Lambda}^{\mathbf{n}}) \\ \sum\limits_{n=0}^{N_{\mathrm{FE}}-1} h_n - T \end{bmatrix}. \tag{47}$$

Furthermore, we provide a summary of the dimensions of all key functions and variables:

- Degrees of freedom: $\mathbf{Z} = (\mathbf{x}, \mathbf{V}, \boldsymbol{\Theta}, \mathbf{A}, \boldsymbol{\Lambda}^{\mathrm{p}}, \boldsymbol{\Lambda}^{\mathrm{n}})$ and $\mathbf{h}$.
- Total number of degrees of freedom: $n_{\mathbf{Z}} + N_{\mathrm{FE}}$, where $n_{\mathbf{Z}} = (N_{\mathrm{FE}} + 1)n_x + N_{\mathrm{FE}} n_{\mathrm{s}} n_x + n_\theta + 3n_\alpha$.
- Dimension of $\boldsymbol{\Theta}$: $n_\theta = N_{\mathrm{FE}} n_{\mathrm{s}} n_f$.
- Dimension of $\mathbf{A}, \boldsymbol{\Lambda}^{\mathrm{p}}, \boldsymbol{\Lambda}^{\mathrm{n}}$: $n_\alpha = N_{\mathrm{FE}} n_{\mathrm{s}} n_c$
- Parameters: $(s_0, q, T) \in \mathbb{R}^{n_x + n_u + 1}$,
- Standard RK equations: $G_{\mathrm{std}} : \mathbb{R}^{n_{\mathbf{Z}}} \times \mathbb{R}^{N_{\mathrm{FE}}} \times \mathbb{R}^{n_x} \times \mathbb{R}^{n_u} \times \mathbb{R} \to \mathbb{R}^{n_{\mathbf{Z}}}$,
- Cross complementarity: $G_{\mathrm{cross}} : \mathbb{R}^{n_\alpha} \times \mathbb{R}^{n_\alpha} \times \mathbb{R}^{n_\alpha} \to \mathbb{R}^{N_{\mathrm{FE}}-1}$,
- Step equilibration: $G_{\mathrm{eq}} : \mathbb{R}^{n_{N_{\mathrm{FE}}}} \times \mathbb{R}^{n_\alpha} \times \mathbb{R}^{n_\alpha} \times \mathbb{R}^{n_\alpha} \to \mathbb{R}^{N_{\mathrm{FE}}-1}$ and
- FESD equations: $G_{\mathrm{fesd}} : \mathbb{R}^{n_{\mathbf{Z}}} \times \mathbb{R}^{N_{\mathrm{FE}}} \times \mathbb{R}^{n_x} \times \mathbb{R}^{n_u} \times \mathbb{R} \to \mathbb{R}^{n_{\mathbf{Z}} + 2N_{\mathrm{FE}} - 1}$.

The vectors $s_0 \in \mathbb{R}^{n_x}$, $q \in \mathbb{R}^{n_u}$ and $T \in \mathbb{R}$ are given parameters. As a result, the system of Eqs. (47) has a total of $n_{\mathbf{Z}} + N_{\mathrm{FE}}$ unknowns and $n_{\mathbf{Z}} + 2N_{\mathrm{FE}} - 1$ equations, which means it is always over-determined.

Theorem 15. *Suppose that Assumptions 11–13 hold. Let s_0, q_0 and T be some fixed parameters such that $G_{\mathrm{fesd}}(\mathbf{Z}^*, \mathbf{h}^*, s_0, q, T) = 0$ in (47). Let $P^* \subseteq \mathbb{R}^{n_x} \times \mathbb{R}^{n_u} \times \mathbb{R}$ be the set of all parameters $(\hat{s}_0, \hat{q}, \hat{T})$ such that $\mathbf{Z} \in \mathbb{R}^{n_{\mathbf{Z}}}$, which is the solution of $G_{\mathrm{fesd}}(\mathbf{Z}, \mathbf{h}, \hat{s}_0, \hat{q}, \hat{T}) = 0$, has the same active set as $\mathbf{Z}^*$. Additionally, suppose that $G_{\mathrm{fesd}}(\cdot)$ is continuously differentiable in s_0, q and T for all $(s_0, q, T) \in P^*$. Then there exists a neighborhood $P \subseteq P^*$ of (s_0, q_0, T) such that there exist continuously differentiable single valued functions $\mathbf{Z}^* : P \to \mathbb{R}^{n_{\mathbf{Z}}}$ and $\mathbf{h}^* : P \to \mathbb{R}^{N_{\mathrm{FE}}}$.*

Proof. The proof follows similar lines as the proof of [14, Theorem 14] and we omit it for brevity. □

5.3. Convergence of the FESD method

We proceed by stating the results of the convergence of the FESD method. We show that under suitable assumptions, the sequence of approximations $\hat{x}_h(\cdot)$ generated by the FESD method converges to a solution of (8). In particular, the FESD method has the same order as the underlying RK method for smooth ODE.

Theorem 16. *Let $x(\cdot)$ be a solution of (8) with finitely many active set changes for $t \in [0, T]$ with $x(0) = x_0$. Suppose the following is true:*

(a) The Assumptions 7 and 14 are satisfied.
(b) The Assumptions 11, 12 and 13 hold for the FESD problem (40).

Then $x(\cdot)$ is a limit point of the sequence of approximations $\hat{x}_h(\cdot)$, defined in Eq. (46) as $h \downarrow 0$. Moreover, for sufficiently small $h > 0$, the solution of (40) generates a solution approximation $\hat{x}_h(t)$ on $[0, T]$ such that:

$$|\hat{t}_{\mathrm{s},n} - t_{\mathrm{s},n}| = O(h^p) \text{ for every } n \in \{0, \dots, N_{\mathrm{sw}}\}, \tag{48a}$$

$$\|\hat{x}_h(t) - x(t)\| = O(h^p), \text{ for all } t \in \mathcal{G}. \tag{48b}$$

Proof. The proof follows similar lines as the proof of [14, Theorem 16]. The primary distinction lies in the prediction of new active sets. In [14, Theorem 16], we use [14, Assumption 8] to be able to apply [23, Lemma A.2] and demonstrate that both the approximation and the exact solution share the same active set in the vicinity of a switching point. In this theorem, the assertion emerges directly from Assumption 14. □

The proof of [14, Theorem 16] is inspired by the proof of [23, Theorem 4.3] and is quite involved. The main idea is to consider intervals $[t_{\mathrm{s},n}, t_{\mathrm{s},n+1}]$ with fixed active sets $\mathcal{I}_n$ where the dynamics are locally smooth, and we recover the accuracy of the underlying RK method. At a switching point, $\hat{t}_{\mathrm{s},n}$, respectively $t_{\mathrm{s},n}$, one has to prove (48a) and that the solution approximation can continue to evolve with the same active set $\mathcal{I}_{n+1}$ as the true solution. Then the argument can be used inductively.

Some distinction must be made for the case where the approximation switches before or after the true solution to obtain the results. Once (48a) is proved, (48b) can be obtained from some algebraic manipulations and the Lipschitz continuity of the local dynamics and its solution. The error is dominated by the maximum step size h; hence, it is used in the error estimate.

5.4. Convergence of discrete-time sensitivities

This section concludes by demonstrating that the numerical sensitivities (cf. Section 1 for a definition) obtained using the FESD method for the step reformulation converge to the correct values with a high order of accuracy. We remind the reader that numerical sensitivities of standard time-stepping methods, *e.g.* (34), do not converge to the correct values, no matter how small the step size becomes [18]. The convergence of the sensitivities is crucial for the success of direct optimal control methods [19]. Before stating the result, we assume the time derivatives of the solution approximation $\hat{x}_h$ converge accordingly. This assumption extends Assumption 11 and allows us to consider a wide range of RK methods.

Assumption 17 (*RK Derivatives*)**.** Regard the RK methods from Assumption 11 applied to the differential algebraic Eqs. (29). The derivatives of the numerical approximation for the same RK method converge with order $1 \leq q \leq p$, *i.e.*, $\|\dot{\hat{x}}_h(t) - \dot{x}(t)\| = O(h^q)$, $t \in \mathcal{G}$.

We remind the reader that for collocation-based implicit RK methods for ODE in general it holds that $q = p - 1$ [45, Theorem 7.10].

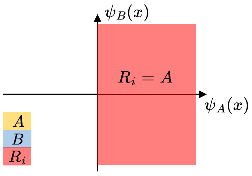
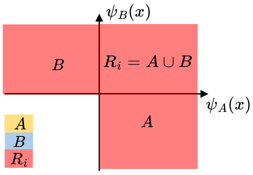
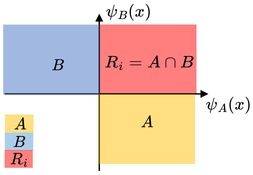
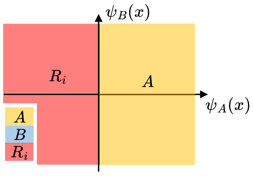
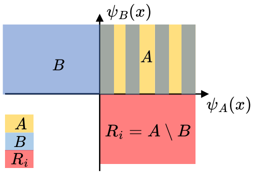
Theorem 18 (*Convergence to Exact Sensitivities*)**.** *Suppose the assumptions of Theorem 16 and Assumption 17 hold. Assume that a single active-set change happens at time $t_{\mathrm{s},n}$, i.e., $||\mathcal{I}_n| - |\mathcal{I}_{n+1}|| \leq 1, n \in \{0, \dots, N_{\mathrm{sw}}\}$. Then for $h \downarrow 0$ it holds that $\frac{\partial \hat{x}_h(t,x_0)}{\partial x_0} \to \frac{\partial x(t,x_0)}{\partial x_0}$ with the convergence rate*

$$\left\|\frac{\partial \hat{x}_h(t, x_0)}{\partial x_0} - \frac{\partial x(t, x_0)}{\partial x_0}\right\| = O(h^q), \text{ for all } t \in \mathcal{G}. \tag{49}$$

Proof. The proof is essentially the same as the proof of [14, Theorem 18], one has only to replace the local switching functions $\psi_{i,j}(x)$ by an appropriate switching function $\psi_k(x)$. □

The key to this proof is the fact that the cross complementarity conditions imply $\psi_j(\hat{x}_h(\hat{t}_{\mathrm{s}})) = 0$ at a switching point $\hat{t}_{\mathrm{s}}$, cf. Section 4.3. Most of the proof then applies the chain rule and the implicit function theorem to obtain a similar expression for the discrete-time numerical sensitivities as for the continuous-time case.

Table 3
Expressions of θ_i for different definitions of R_i.

Definition of R_i	Expression for θ_i	Sketch
$R_i = A$	$\theta_i = \alpha_A$	 $\psi_B(x)$ $R_i = A$ $\psi_A(x)$ A B R_i
$R_i = A \cup B$	$\theta_i = \alpha_A + \alpha_B$	 $\psi_B(x)$ B $R_i = A \cup B$ $\psi_A(x)$ A A B R_i
$R_i = A \cap B$	$\theta_i = \alpha_A \alpha_B$	 $\psi_B(x)$ B $R_i = A \cap B$ $\psi_A(x)$ A A B R_i
$R_i = \mathrm{int}(\mathbb{R}^{n_x} \setminus A) = \{x \mid \psi_A(x) < 0\}$	$\theta_i = 1 - \alpha_A$	 $\psi_B(x)$ R_i A $\psi_A(x)$ A B R_i
$R_i = A \setminus B$	$\theta_i = \alpha_A - \alpha_B$	 $\psi_B(x)$ B A $\psi_A(x)$ $R_i = A \setminus B$ A B R_i

6. Efficient modeling with set-valued Heaviside step functions

In this section, we show how to efficiently represent common geometries of the PSS regions with the use of Heaviside step functions. This is useful for reducing the complexity of the modeling process. Moreover, we introduce a lifting algorithm, which makes the multi-affine expressions for θ_i in (22b) "less nonlinear" with the help of auxiliary variables.

6.1. Overview of expressions for θ via Heaviside step functions

We regard the following two sets: $A = \{x \mid \psi_A(x) > 0\}, B = \{x \mid \psi_B(x) > 0\}$, and let $\alpha_A \in \gamma(\psi_A(x))$ and $\alpha_B \in \gamma(\psi_B(x))$. Note that these sets do not have to correspond to the base regions in Definition 1, but we can use them to define such regions. Table 3 provides an overview of how the elementary algebraic expressions for the multipliers θ_i are related to the geometric definition of a region R_i. More complicated expressions can be obtained by combining the ones listed in Table 3.

Remark 19 (*Sum of Filippov Systems*). In practice, one often encounters DIs that arise from the sum of several Filippov systems. This occurs, for example, if we have multiple surfaces with friction, or multiple objects touching the same frictional surface [46]. All developments from this paper can be extended to this case and are implemented in `nosnoc`. For brevity, we omit the corresponding equations and refer the reader to [14, Section 2.3].

6.2. Representing unions of sets

The regions R_i may be given as unions of base sets $\tilde{R}_j$. Consequently, the number of multipliers $\theta \in \mathbb{R}^{n_f}$ decreases and it holds that $n_f < 2^{n_\psi}$. For example, in the extreme case, we may have $R_1 = \tilde{R}_i$ and $R_2 = \cup_{j=1,j\neq i}^{2^{n_\psi}} \tilde{R}_j$, for some i, which significantly reduces the number of variables, since we are left with only two regions, *i.e.*, $n_f = 2$. We illustrate such a case with by a simple example and discuss the general case in the sequel.

Example 4 (*Union of Sets*)**.** We regard an example with two scalar switching functions $\psi_1(x)$ and $\psi_2(x)$, with the basis regions $\tilde{R}_1 = \{x \in \mathbb{R}^{n_x} \mid \psi_1(x) > 0, \psi_2(x) > 0\}$, $\tilde{R}_2 = \{x \in \mathbb{R}^{n_x} \mid \psi_1(x) > 0, \psi_2(x) < 0\}$, $\tilde{R}_3 = \{x \in \mathbb{R}^{n_x} \mid \psi_1(x) < 0, \psi_2(x) > 0\}$ and $\tilde{R}_4 = \{x \in \mathbb{R}^{n_x} \mid \psi_1(x) < 0, \psi_2(x) < 0\}$. The PSS is defined by the two regions:

$$\begin{aligned}
R_1 &= \cup_{i=1}^{3} \tilde{R}_i = \{x \in \mathbb{R}^{n_x} \mid \psi_1(x) > 0\} \cup \{x \in \mathbb{R}^{n_x} \mid \psi_1(x) \leq 0,\ \psi_2(x) > 0\},\\
R_2 &= \tilde{R}_4 = \{x \in \mathbb{R}^{n_x} \mid \psi_1(x) < 0,\ \psi_2(x) < 0\}.
\end{aligned}$$

According to the second row of Table 3, if a region R_i consists of the union of two sets, the expression for θ_i is the sum of two corresponding indicators. In the current example, R_1 consists of the union of three base regions, so the expressions for θ_1 must be a sum of three terms corresponding to the three base sets. On the other hand, all base sets are constructed from the intersection of two sets defined by $\psi_1(x)$ and $\psi_2(x)$, see the third row in Table 3. Thus, the products in the expressions refer to the intersection of the sets. Therefore, the related Filippov system reads as $\dot{x} = \theta_1 f_1(x) + \theta_2 f_2(x)$, where

$$\begin{aligned}
\theta_1 &= \alpha_1\alpha_2 + \alpha_1(1 - \alpha_2) + (1 - \alpha_1)\alpha_2 = \alpha_1 + (1 - \alpha_1)\alpha_2,\\
\theta_2 &= (1 - \alpha_1)(1 - \alpha_2).
\end{aligned}$$

By direct calculation, we verify that $\theta_1, \theta_2 \geq 0$ and $\theta_1 + \theta_2 = 1$. In contrast to the previous examples, the union of sets introduces a sum in the expressions for θ_1.

We generalize the reasoning above as follows. Given n_f total regions and the matrix $F(x,u) \in \mathbb{R}^{n_x} \times \mathbb{R}^{n_F}$, where $n_F = 2^{n_\psi}$ is the number of possibly repeating columns. We define the index sets $\mathcal{R}_k = \{i \mid F_{\bullet,i}(x,u) = f_k(x,u)\}$, *i.e.*, the set if the indices of the columns of $F(x,u)$ equal to $f_k(x,u)$. Note that if we have no unions, then $n_F = n_f$ and $\mathcal{R}_k = \{k\}$ for all $k \in \mathcal{J}$. Using these definitions, the expression for θ_i in (21) reduces to:

$$\theta_i = \sum_{k \in \mathcal{R}_i} \prod_{j \in C} \frac{1 - S_{k,j}}{2} + S_{k,j}\alpha_j.$$

6.3. A lifting algorithm for the multi-affine terms

It can be seen from (21), the expression of θ_i consists of a product of n_ψ affine terms (of the form of α_j and $1-\alpha_k$). If n_ψ is large, then this expression is very nonlinear. To reduce the nonlinearity, we introduce auxiliary lifting variables as in the *lifted* Newton's method [24], which iterates on a larger but less nonlinear problem. Whenever there are more than two terms in the multi-affine expression for θ_i, we introduce lifting variables β_k and derive an equivalent formulation, which has only bilinear terms. Now, instead of having n_f multi-affine expressions, we have $n_f + n_\beta$ expressions, but which are less nonlinear. We exploit the structure of the matrix S and derive an easy-to-implement algorithm that automates the lifting procedure. To give an idea of the final results we aim to obtain, we illustrate the lifting procedure with an example.

Example 5. Regard a PSS with $n_\psi = 3$ switching functions and $n_f = 8$ modes, *i.e.*, the PSS regions match the basis sets, $R_i = \tilde{R}_i$, $i = \{1,\dots,8\}$. The matrix $S \in \mathbb{R}^{8\times3}$, and the expression for the multipliers $\theta \in \mathbb{R}^8$ read as

$$S = \begin{bmatrix} 1 & 1 & 1\\ 1 & 1 & -1\\ 1 & -1 & 1\\ 1 & -1 & -1\\ -1 & 1 & 1\\ -1 & 1 & -1\\ -1 & -1 & 1\\ -1 & -1 & -1 \end{bmatrix},\ G_{\mathrm{F}}(\theta,\alpha) = \begin{bmatrix} \theta_1 - \alpha_1\alpha_2\alpha_3\\ \theta_2 - \alpha_1\alpha_2(1 - \alpha_3)\\ \theta_3 - \alpha_1(1 - \alpha_2)\alpha_3\\ \theta_4 - \alpha_1(1 - \alpha_2)(1 - \alpha_3)\\ \theta_5 - (1 - \alpha_1)\alpha_2\alpha_3\\ \theta_6 - (1 - \alpha_1)\alpha_2(1 - \alpha_3)\\ \theta_7 - (1 - \alpha_1)(1 - \alpha_2)\alpha_3\\ \theta_8 - (1 - \alpha_1)(1 - \alpha_2)(1 - \alpha_3) \end{bmatrix} = 0.$$

We can introduce the lifting variable $\beta \in \mathbb{R}^4$ and obtain

$$G_\beta(\alpha,\beta) = \begin{bmatrix} \beta_1 - \alpha_1\alpha_2\\ \beta_2 - \alpha_1(1 - \alpha_2)\\ \beta_3 - (1 - \alpha_1)\alpha_2\\ \beta_4 - (1 - \alpha_1)(1 - \alpha_2) \end{bmatrix} = 0,\ G_\theta(\theta,\alpha,\beta) = \begin{bmatrix} \theta_1 - \beta_1\alpha_3\\ \theta_2 - \beta_1(1 - \alpha_3)\\ \theta_3 - \beta_2\alpha_3\\ \theta_4 - \beta_2(1 - \alpha_3)\\ \theta_5 - \beta_3\alpha_3\\ \theta_6 - \beta_3(1 - \alpha_3)\\ \theta_7 - \beta_4\alpha_3\\ \theta_8 - \beta_4(1 - \alpha_3) \end{bmatrix} = 0.$$

The equation $G_{\beta}(\alpha,\beta)=0$ relates the lifting variables β_i with the variables α_j, whereas $G_{\theta}(\theta,\alpha,\beta)$ provides expressions for θ_i via β_i and the remaining α_j. By replacing $G_{\mathrm{F}}(\theta,\alpha)=0$ with $G_{\mathrm{lift}}(\theta,\alpha,\beta) := (G_{\beta}(\alpha,\beta), G_{\theta}(\theta,\alpha,\beta))=0$, we obtain an equivalent system of equations that only consists of bilinear terms.

Algorithm 1 Lifting algorithm for the step DCS (22)

```
1:  Input: $S, n_{\mathrm{d}}$
2:  Initialize: $\tilde{S} \leftarrow S$, $k \leftarrow 0$; $\tilde{\theta} \leftarrow e \in \mathbb{R}^{n_f}$, $G_{\theta}(\theta,\alpha,\beta) \leftarrow [\,]$, $G_{\beta}(\alpha,\beta) \leftarrow [\,]$.
3:  for $j = 1 : n_{\psi}$ do
4:      $\tilde{N} \leftarrow \sum_{j=1}^{n_{\psi}} |\tilde{S}_{\bullet,j}|$
5:      $\mathcal{I}_j \leftarrow \{i \mid \tilde{N}_i = j\}$
6:      $\tilde{\theta} \leftarrow \tilde{\theta} \cdot \left(\frac{e-\tilde{S}_{\bullet,j}}{2} + S^{\mathrm{temp}}_{\bullet,j} \cdot \alpha_j\right)$
7:      if $\mathcal{I}_j \neq \emptyset$ then
8:          $G_{\theta}(\theta,\alpha,\beta) \leftarrow (G_{\theta}(\theta,\alpha,\beta), \theta_{k+\mathcal{I}_j} - \tilde{\theta}_{\mathcal{I}_j})$
9:          Remove entries of $\tilde{\theta}$ with index in $\mathcal{I}_j$
10:         Remove rows of $\tilde{S}$ with index in $\mathcal{I}_j$
11:         $k \leftarrow k + \max(\mathcal{I}_j)$
12:     end if
13:     if $j \in \{n_{\mathrm{d}}, \dots, n_{\psi}-1\}$ then
14:         $\{\mathcal{I}_{\mathrm{red}}, \mathcal{I}_{\mathrm{full}}\} = \mathtt{unique}(\tilde{S}_{\bullet,\{1,\dots,j\}})$,
15:         $\beta \leftarrow (\beta, \beta^j)$ where $\beta^j \in \mathbb{R}^{|\mathcal{I}_{\mathrm{red}}|}$
16:         $G_{\beta}(\alpha,\beta) \leftarrow (G_{\beta}(\alpha,\beta)\beta^j - \tilde{\theta}_{\mathcal{I}_{\mathrm{red}}})$
17:         $\tilde{\theta} \leftarrow \beta^j_{\mathcal{I}_{\mathrm{full}}}$
18:     end if
19: end for
20: $G_{\mathrm{Lift}}(\theta,\alpha,\beta) := (G_{\theta}(\theta,\alpha,\beta), G_{\beta}(\alpha,\beta))$
21: Output: $G_{\mathrm{Lift}}(\theta,\alpha,\beta)$ , $\beta$
```

We proceed by outlining a general lifting algorithm. The expressions for θ_i consist of the product of n_{ψ} affine terms. Our goal is to have at most n_{d} terms in the multi-affine expression for θ_i. For example, if we pick $n_{\mathrm{d}}=2$, we have only bilinear expressions in the equations defining θ and β. Thus, the parameter $n_{\psi} \geq n_{\mathrm{d}} \geq 2$, controls the number of terms in the multi-affine expressions and implicitly the number of new lifting variables $\beta \in \mathbb{R}^{n_{\beta}}$. Given the matrix S, our goal is to automatically obtain the constraint $G_{\mathrm{lift}}(\theta,\alpha,\beta)=0$.

The algorithm outlined above can be implemented using a symbolic framework such as `CasADi` [43]. We provide the pseudo code in Algorithm 1, which introduces the lifting algebraic variables β and new *lifted* expressions for θ_i, namely $G_{\mathrm{lift}}(\theta,\alpha,\beta)$. Note that we make use of three helper variables, the matrix $\tilde{S}$ and the vectors $\tilde{\theta}$ and $\tilde{N}$. The matrix $\tilde{S}$ is a submatrix of S, where we have removed the rows with index i, for which we already have a (lifted) expression for θ_i. The vector $\tilde{N}$, defined in line 4, collects the number of nonzero entries of every row $\tilde{S}$. In other words, it keeps track of how many terms are in the initial expressions for θ_i, that are not yet lifted.

The main loop iterates from $j=1$ to n_{ψ} and provides in every iteration the expressions for all θ_i that have exactly j terms in their multi-affine expression. The index set $\mathcal{I}_j = \{i \mid \tilde{N}_i = j\}$, defined in line 5, contains the indices of θ, that have exactly j entries in their corresponding multi-affine expression. In line 6, we define the auxiliary variable $\tilde{\theta}$, which stores the intermediate expressions for θ with up to j terms in the product. The index k stores the index of the last θ_k for which a lifted expression was derived. For $j \leq n_{\mathrm{d}}$ the expressions for θ_i are unaltered. This is treated in lines 7–11.

As soon as $j > n_{\mathrm{d}}$, the algorithm introduces new lifting variables β^j (line 15) and changes the expression for $\tilde{\theta}$ accordingly. This is done in lines 13–17. A key tool is the function `unique` in line 14. It is available in `MATLAB` and the `numpy` package in `python`. It works as follows: given a matrix $A \in \mathbb{R}^{m\times n}$ it returns a matrix $\tilde{A} \in \mathbb{R}^{p\times n}$, with $p \leq m$. This is the matrix constructed from A by removing its repeating rows. More importantly for our needs, it returns the index sets $\mathcal{I}_{\mathrm{red}}$ and $\mathcal{I}_{\mathrm{full}}$, with $|\mathcal{I}_{\mathrm{red}}|=p$ and $|\mathcal{I}_{\mathrm{full}}|=m$. The index sets have the properties $A=\left[\tilde{A}_{i,\bullet} \mid i \in \mathcal{I}_{\mathrm{full}}\right] \in \mathbb{R}^{m\times n}$ and $\tilde{A}=\left[A_{i,\bullet} \mid i \in \mathcal{I}_{\mathrm{red}} \in \mathbb{R}^{p\times n}\right]$.

This enables the use of the same β^j for several θ_i if they share the same terms in the corresponding multi-affine expressions, cf. lines 15–17. After the loop is finished, the algorithm outputs $G_{\mathrm{Lift}}(\theta,\alpha,\beta)$ and β. One can verify that Algorithm 1 produces the same output as Example 5. It can be shown, that for a given $n_{\mathrm{d}} < n_{\psi}$, the total number of new lifting variables is $n_{\beta} = 2^{n_{\psi}} - 2^{n_{\mathrm{d}}}$.

6.4. Comparisons of Stewart's and the Heaviside step reformulation

We compare Stewart's reformulation (28) and the Heaviside step reformulation (22) based on their total number of algebraic variables, complementarity constraints, and equality constraints for a given number of switching functions n_{ψ}. The total number of regions (and multipliers θ_i) in Stewart's reformulation is always $n_f = 2^{n_{\psi}}$.

In contrast to the Heaviside step reformulation, we cannot reduce the number of variables if the regions R_i are defined as unions of the base regions $\tilde{R}_i$. On the other hand, in the Heaviside step reformulation, depending on the geometry of the regions R_i, n_f

Table 4
Comparison of the problem sizes in Stewart's and the step reformulation for a fixed n_ψ.

Ref.	n_f	n_β	n_{alg}	n_{comp}	n_{eq}
Stewart	2^{n_ψ}	0	$2\cdot 2^{n_\psi}+1$	2^{n_ψ}	$2^{n_\psi}+1$
Step	$[2, 2^{n_\psi}]$	$\begin{cases} 2^{n_\psi}-2^{n_{\mathrm{d}}}, & n_{\mathrm{d}} \le n_\psi \\ 0, & n_{\mathrm{d}} > n_\psi \end{cases}$	$n_f+3n_\psi+n_\beta$	$2n_\psi$	$n_\psi+n_\beta+n_f$

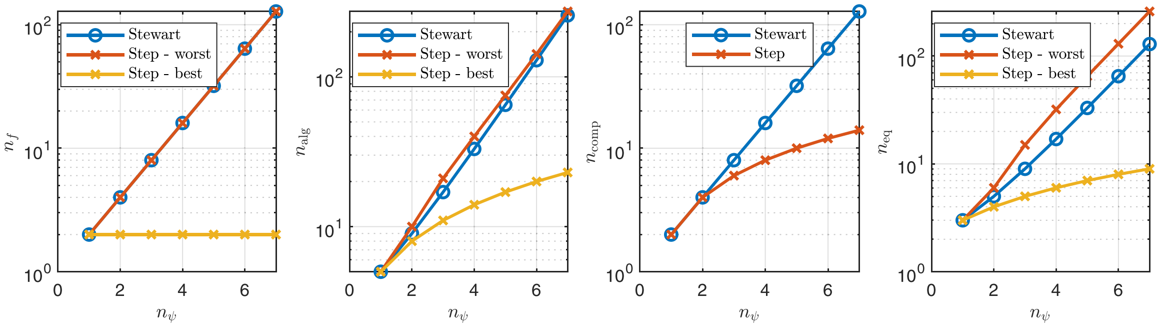


Fig. 4. Comparison of the complexities of Stewart's and the step reformulation.

is an integer in $[2, 2^{n_\psi}]$. In the step reformulation, we may introduce n_β lifting variables to reduce the nonlinearity. If $n_{\mathrm{d}} > n_\psi$, this leads to $n_\beta = 2^{n_\psi} - 2^{n_{\mathrm{d}}}$ additional lifting variables and equations.

We compare now the number of algebraic variables. In Stewart's reformulation, we have $\lambda \in \mathbb{R}^{2^{n_\psi}}$ and $\mu \in \mathbb{R}$. In the Heaviside step reformulation, we have $\alpha, \lambda^{\mathrm{p}}, \lambda^{\mathrm{n}} \in \mathbb{R}^{n_\psi}$. Thus, the total number of algebraic variables in the former case is $n_{\mathrm{alg}}^{\mathrm{S}} = 2\cdot 2^{n_\psi}+1$, and in the later case $n_{\mathrm{alg}}^{\mathrm{H}} = n_f + 3n_\psi + n_\beta$. The number of complementarity constraints n_{comp} in Stewart's case is $n_{\mathrm{comp}}^{\mathrm{S}} = 2^{n_\psi}$, and in the Heaviside step case $n_{\mathrm{comp}}^{\mathrm{H}} = 2n_\psi$, *i.e.*, we have exponential versus linear complexity. Finally, in Stewart's reformulation, we have in total $n_{\mathrm{eq}}^{\mathrm{S}} = 2^{n_\psi}+1$ equality constraints ($g_i(x) = \lambda_i - \mu$ and $e^\top\theta = 1$). In the Heaviside step case, there are $n_{\mathrm{eq}}^{\mathrm{H}} = n_f + n_\beta + n_\psi$ equality constraints, for the definitions of θ_i, β_i and the constraints $\psi_i(x) = \lambda_i^{\mathrm{p}} - \lambda_i^{\mathrm{n}}$, respectively. The numbers of variables and constraints are summarized in Table 4.

Fig. 4 illustrates the different quantities for several n_ψ. We plot for the Heaviside step reformulation two extreme scenarios:

1. Worst complexity case - every basis set defines a PSS region, $n_f = 2^{n_\psi}$, we lift to have only bilinear terms, *i.e.*, $n_{\mathrm{d}} = 2$ (maximizes the number of lifting variables) - red line in the plots.
2. Best complexity case - no lifting and only two regions ($n_f = 2$) for all n_ψ -yellow line in the plots.

Note that in both cases the Heaviside step reformulation has the same number of complementarity constraints. For smaller values of n_ψ, both reformulations have similar complexity. For a large number of switching functions, the step reformulation has fewer variables. However, if there is no lifting, the problem can become very nonlinear for large n_ψ.

An extensive numerical comparison of the use of the two approaches in simulation and optimal control problems has been done in [47, Section 5.8]. It turns out that statistically both formulations have similar performance, but on specific examples one can outperform the other. In addition, the Heaviside step reformulation provides more modeling flexibility.

7. Numerical examples

In this section, we consider two numerical examples. First, we consider a numerical simulation example of a gene regulatory network and empirically demonstrate that the FESD method has higher-order integration accuracy. Second, we consider an optimal control example of a planar two-link monoped that must reach a certain goal in the horizontal direction. We compare the Heaviside step reformulation with the Stewart reformulation in terms of total computational time and cost per iteration.

7.1. Gene regulatory networks

In the simulation experiment, we consider a gene regulatory network model, which is described by a DI of the form (14). The model is not a Filippov system, but a more general DI. More numerical examples with FESD for the step reformulation can be found in [6]. In this reference, we have generated integration order plots for a Filippov system and confirmed empirically the results of Theorem 16.

We regard the IRMA example, a synthetic network composed of five genes, originally proposed in [48]. This example is inspired by [1], which includes more examples with Heaviside step functions that are implemented in nosnoc [49].

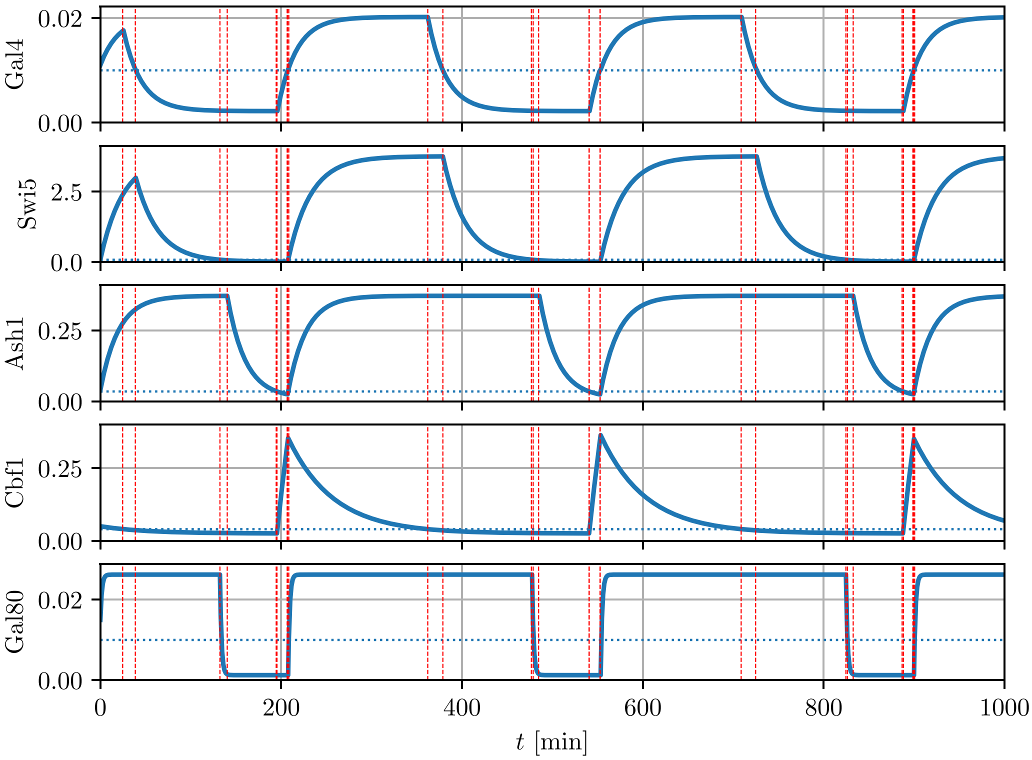


Fig. 5. State trajectory of the IRMA example with red dotted vertical lines indicating the switches. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

IRMA model. In Fig. 5, we reproduce the state trajectories from [1, Fig. 11].Additionally, the vertical lines show the switching times of the selection variables α_i. The states of this system are the protein concentrations of Gal4, Swi5, Ash1, Cbf1, and Gal80, which are denoted by $x_1, \dots, x_5$. There are seven switching functions, which are defined by the states crossing certain thresholds, which are plotted as horizontal lines in Fig. 5. Specifically, the switching functions are

$$\psi(x) = (x_1 - 0.01, x_2 - 0.01, x_2 - 0.06, x_2 - 0.08, x_3 - 0.035, x_4 - 0.04, x_5 - 0.01).$$

Note that Swi5 (x_2) has three threshold values. The continuous-time dynamics of the system are given by

$$\dot{x}_1 \in -p_1 x_1 + \kappa_1^1 + \kappa_1^2 \gamma(\psi_6(x)), \tag{50a}$$

$$\dot{x}_2 \in -p_2 x_2 + \kappa_2^1 + \kappa_2^2 \gamma(\psi_1(x))(1-u)\gamma(\psi_7(x)), \tag{50b}$$

$$\dot{x}_3 \in -p_3 x_3 + \kappa_3^1 + \kappa_3^2 \gamma(\psi_3(x)), \tag{50c}$$

$$\dot{x}_4 \in -p_4 x_4 + \kappa_4^1 + \kappa_4^2 \gamma(\psi_2(x))(1 - \gamma(\psi_5(x))), \tag{50d}$$

$$\dot{x}_5 \in -p_5 x_5 + \kappa_5^1 + \kappa_5^2 \gamma(\psi_4(x)). \tag{50e}$$

Note that $u \in \{0, 1\}$ is an external input, which is set to $u = 1$ in the scenario considered here. The initial state is given by $x_0 = (0.011, 0.09, 0.04, 0.05, 0.015)$. The parameter values are given as

$$\begin{aligned} p &= (0.05, 0.04, 0.05, 0.02, 0.6), \\ \kappa^1 &= (1.1 \cdot 10^{-4}, 3 \cdot 10^{-4}, 6 \cdot 10^{-4}, 5 \cdot 10^{-4}, 7.5 \cdot 10^{-4}) \\ \kappa^2 &= (9 \cdot 10^{-4}, 0.15, 0.018, 0.03, 0.015). \end{aligned}$$

Integration order experiment. To showcase that the FESD step reformulation preserves the integration order of the underlying Runge–Kutta method, we simulate the first 100 min of the trajectory depicted in Fig. 5. This time interval contains exactly two switches. We use $N_{\mathrm{FE}} = 3$, to capture those switches even with a single FESD step. We plot the integration error over the average step size $\bar{h} = T_{\mathrm{sim}}/N_{\mathrm{sim}} N_{\mathrm{FE}}$, where N_{sim} is the total number of simulation steps. The results are summarized in Fig. 6 for FESD with underlying Radau IIA of orders 1, 3, 5, and 7, and Gauss–Legendre methods of orders 2, 4, 6, and 8. It can be seen that the FESD step formulation preserves the integration order of the underlying Runge–Kutta method, while with the standard time-stepping approach,

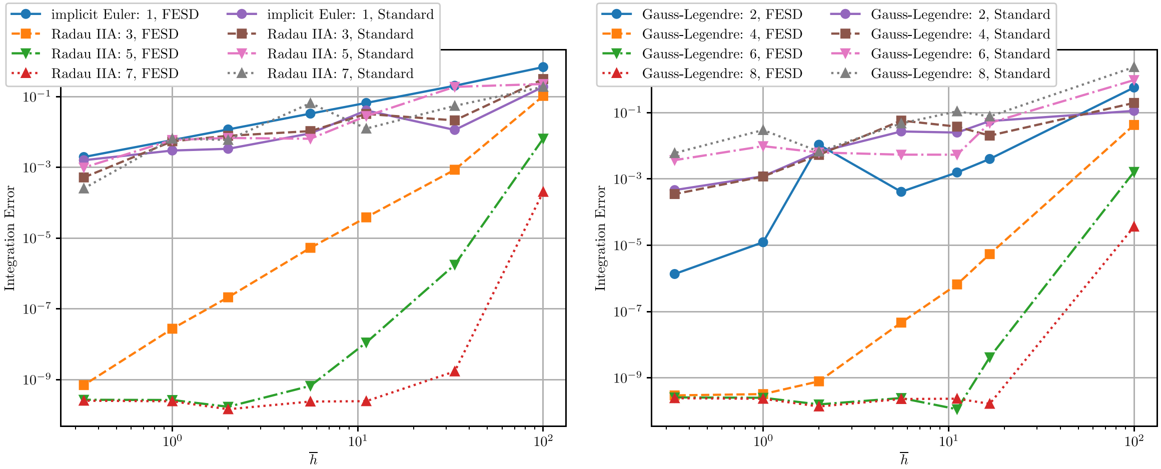


Fig. 6. Accuracy vs. step size: Simulation of first 100 min of the trajectory in Fig. 5 with different RK schemes and step sizes.

using methods that typically have higher-order integration accuracy degrade to order one without FESD. This experiment shows, that if a feasible solution is found the integration order of the underlying Runge–Kutta method is preserved. The implementation of the example is publicly available.[1]

7.2. Optimal control example with state jumps

Now we further investigate the use of FESD for the Heaviside step reformulation by applying it to an optimal control problem. We regard the problem of synthesizing dynamic motions of the two-link *Capler* robot with state jumps and friction [50]. Systems with state jumps do not directly fit the form of ODEs with set-valued Heaviside step functions (14). However, using the time-freezing reformulation we can transform the system with state jumps into a PSS of the form of (8) [8] and use the methods developed in this paper, see also Remark 2.

The monoped's configuration is described by four degrees of freedom $q = (q_x, q_z, \phi_{\text{knee}}, \phi_{\text{hip}})$, where (q_x, q_z) are the coordinates of the monoped's base at the hip and ϕ_{knee}, ϕ_{hip} are the angles of the hip and knee. The robot is actuated by two direct-drive motors at the hip and knee joints. The control variables are the torques of these motors $u(t) = (u_{\text{knee}}(t), u_{\text{hip}}(t))$. Denote by $p_{\text{foot}}(q) = (p_{\text{foot},x}(q), p_{\text{foot},z}(q))$ and $p_{\text{knee}}(q) = (p_{\text{knee},x}(q), p_{\text{knee},z}(q))$ the kinematic position of the robot's foot and knee, respectively. We model a single contact point, the tip of the robot's foot touching the ground, which is expressed via the unilateral constraint

$$f_c(q) = p_{\text{foot},z}(q) \geq 0.$$

Moreover, we denote the expression summarizing the Coriolis, control, and gravitational forces by $f_v(q,u)$, the inertia matrix by $M(q)$, the normal contact Jacobian by $J_{\text{n}}(q) \in \mathbb{R}^{n_q}$, and the contact tangent by $J_{\text{t}}(q) \in \mathbb{R}^{n_q}$. The detailed derivation of all mentioned functions, *i.e.*, the model equations, kinematic expressions, and all parameters for the robot can be found in [51, Appendix A].

After the time-freezing reformulation, the PSS state consists of $x(\tau) = (q(\tau), v(\tau), t(\tau)) \in \mathbb{R}^9$, with $q(\tau)$ being the position, $v(\tau)$ the velocity, $t(\tau)$ is a clock state needed for the time-freezing reformulation and τ is the time of the ODE, cf. [8]. For the time-freezing PSS, we have in total three switching functions: the gap function, as well as the normal and tangential contact velocities [8]:

$$\psi(x) = (f_c(q), J_{\text{n}}(q)^\top v, J_{\text{t}}(q)^\top v).$$

This allows a definition of eight base sets via the sign matrix (cf. Definition 1):

$$S = \begin{bmatrix} 1 & 1 & 1 \\ 1 & 1 & -1 \\ 1 & -1 & 1 \\ 1 & -1 & -1 \\ -1 & 1 & 1 \\ -1 & 1 & -1 \\ -1 & -1 & 1 \\ -1 & -1 & -1 \end{bmatrix}$$

[1] https://github.com/FreyJo/nosnoc_py/blob/main/examples/Acary2014/irma_integration_order_experiment.py

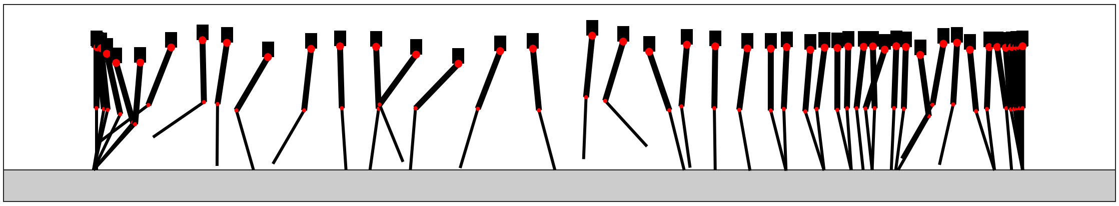


Fig. 7. Illustration of several frames of the solution of the discretized OCP.

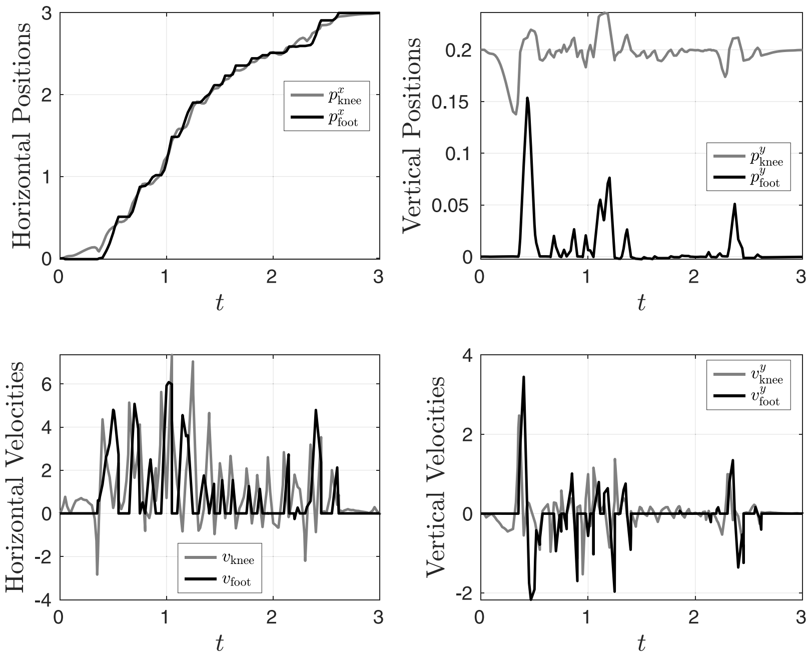


Fig. 8. The monoped's vertical and horizontal positions (top plots), and velocities (bottom plots).

However, the time-freezing PSS has only three regions, where the first one consists of the union of the first six base sets, and the other two match the two remaining base sets, *i.e.*:

$$\begin{aligned}
R_1 &= \cup_{i=1}^{6} \tilde{R}_i = \{x \in \mathbb{R}^{n_x} \mid f_c(q) > 0\} \cup \{x \in \mathbb{R}^{n_x} \mid f_c(q) < 0, J_{\mathrm{n}}(q)^\top v > 0\},\\
R_2 &= \tilde{R}_7 = \{x \in \mathbb{R}^{n_x} \mid f_c(q) < 0, J_{\mathrm{n}}(q)^\top v < 0, J_{\mathrm{t}}(q)^\top v > 0\},\\
R_3 &= \tilde{R}_8 = \{x \in \mathbb{R}^{n_x} \mid f_c(q) < 0, J_{\mathrm{n}}(q)^\top v < 0, J_{\mathrm{t}}(q)^\top v < 0\}.
\end{aligned}$$

In region R_1, we define the unconstrained (free flight) dynamics of the monoped, and in regions, R_2 and R_3, auxiliary ODEs that mimic state jumps in normal and tangential directions due to frictional impacts:

$$\begin{aligned}
f_1(x,u) &= (q, M(q)^{-1} f_v(q,u), 1),\\
f_2(x) &= (\mathbf{0}_{4,1}, M(q)^{-1}(J_{\mathrm{n}}(q) - J_{\mathrm{t}}(q)\mu)a_{\mathrm{n}}, 0),\\
f_3(x) &= (\mathbf{0}_{4,1}, M(q)^{-1}(J_{\mathrm{n}}(q) + J_{\mathrm{t}}(q)\mu)a_{\mathrm{n}}, 0).
\end{aligned}$$

Here, $a_{\mathrm{n}} = 200$ is the auxiliary ODE's constant [8] and the coefficient of friction is $\mu = 0.8$. Observe that the clock state dynamics are $\frac{\mathrm{d}t}{\mathrm{d}\tau} = 1$ in R_1, and $\frac{\mathrm{d}t}{\mathrm{d}\tau} = 0$ in R_2 and R_3. A solution trajectory of a PSS is continuous in time. By taking the pieces of the trajectory where $\frac{\mathrm{d}t}{\mathrm{d}\tau} > 0$, we recover the solution of the original system with state jumps [8].

It follows from the discussion in Section 6, that with the Heaviside step reformulation, we have $\theta \in \mathbb{R}^3$. Since we have $n_\psi = 3$, the step DCS (22) in this example has three complementarity constraints and six equality constraints. For Stewart's reformulation, we cannot exploit the union of basis sets but must define eight regions, which are equal to the basis sets. The first six regions are equipped with $f_1(x,u)$ and the remaining two with $f_2(x)$ and $f_3(x)$ respectively. In this case, we have $\theta \in \mathbb{R}^8$, eight complementarity constraints, and nine equality constraints in the DCS (28). Using the Heaviside step reformulation for this problem allows us to reduce the number of regions we need to define from eight to only three.

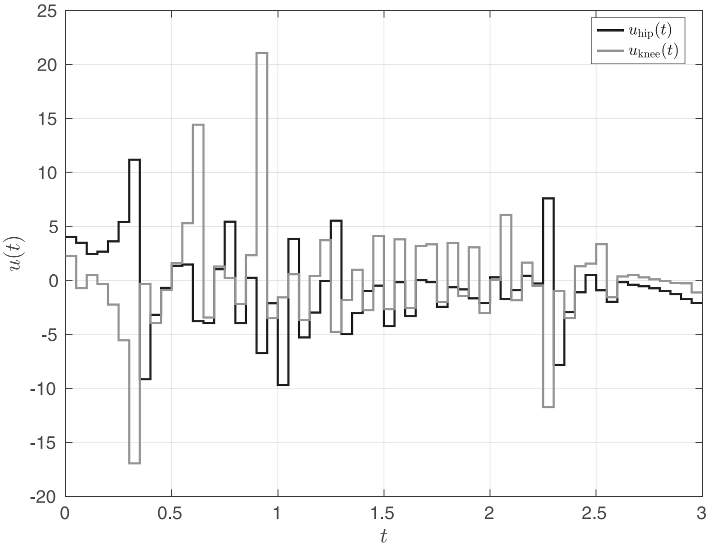


Fig. 9. The optimal controls for the example of $N = 60$ control stages.

To compare the performances of the two reformulations, we run an experiment in which the robot reaches a target position $q_{\rm target} = (3, 0.4, 0, 0)$ with zero velocity $v_{\rm target} = \mathbf{0}_{4,1}$ in $T = 3.0$ seconds. The initial state is $x(0) = (0, 0.4, 0, 0, 0, 0, 0, 0, 0)$. Given a reference $x^{\rm ref}(t)$, which is a spline interpolation between the initial and final position, and includes two jumps, we define the least-squares objective with the running and terminal costs

$$L(x(\tau), u(\tau)) = (x(\tau) - x^{\rm ref}(\tau))^\top Q(x(\tau) - x^{\rm ref}(\tau)) + \rho_u u(\tau)^\top u(\tau),$$
$$R(x(T)) = (x(T) - x^{\rm ref}(T))^\top Q_T(x(T) - x^{\rm ref}(T)),$$

where the weight matrices are $Q = \mathrm{diag}(1, 1, 10, 1, 10^{-6}, 10^{-6}, 10^{-6}, 10^{-6}, 0)$, $\rho_u = 0.01$, and $Q_T = \mathrm{diag}(10^3, 10^3, 10^3, 10^3, 10, 10, 10, 10, 0)$. We define the state and control bound constraints:

$$x_{\rm lb} \leq x(\tau) \leq x_{\rm ub},$$
$$u_{\rm lb} \leq u(\tau) \leq u_{\rm ub},$$

where $x_{\rm ub} = (3.5, 10, \pi, \pi, 100, 100, 100, 100, \infty)$, $x_{\rm lb} = (-0.5, 0, -\pi, -\pi, -100, -100, -100, -100, -\infty)$, $u_{\rm ub} = (100, 100)$, and $u_{\rm lb} = -u_{\rm ub}$.

Collecting all the above, we can define an OCP of the form of (44), which we discretized with the FESD Radau IIA scheme of order 3 ($n_{\rm s} = 2$), with $N_{\rm FE} = 3$ finite elements on every control interval. This OCP is discretized and solved with nosnoc in a homotopy loop with IPOPT [42]. Fig. 7 illustrates several frames of an example solution ($N = 60$). Fig. 8 shows the relevant vertical and horizontal positions and velocities, and Fig. 9 shows the optimal controls.

From these plots, we can see that the robots make several small jumps, mostly landing with low vertical velocities to reduce energy dissipation, and successfully reach the target position with zero terminal velocity.

Next, we solve this OCP for different values N (number of control intervals) from 30 to 90 in increments of 10 and compare FESD for the Heaviside step reformulation (this paper) to FESD derived for Stewart's reformulation [14]. We compare the CPU time per NLP iteration and total CPU time for both approaches in Fig. 10. As expected, due to the smaller number of variables and constraints, the Heaviside step reformulation leads to faster NLP iterations than the Stewart reformulation. In our experiments, we try different linear solvers from the HSL library [52]. The linear solver choice has a particularly strong influence on performance for both reformulations. After evaluation, we select the best linear solver for each one, which is ma27 for Stewart reformulation and ma97 for the Heaviside step reformulation. We can see the linear algebra costs increase significantly, for $N > 60$, as the linear solvers need more time to factorize larger systems. The total computation time is influenced by several factors: homotopy algorithm, initialization, NLP solver performance, linear algebra solver, etc., and as such shows a less clear trend. In this example, Stewart's reformulation achieves a consistently lower computation time despite a higher cost per iteration. On the other hand, in our experiments in [6], we used a similar setting, and the Heaviside step reformulation led to lower total computational times. We can conclude that which reformulation is the better depends strongly on the example and the chosen optimization algorithm parameters.

Further comparisons of the two approaches can be found in [47, Section 5.2].

8. Summary

This paper extends the Finite Elements with Switch Detection (FEDS) method to nonsmooth dynamical systems with set-valued Heaviside step functions.

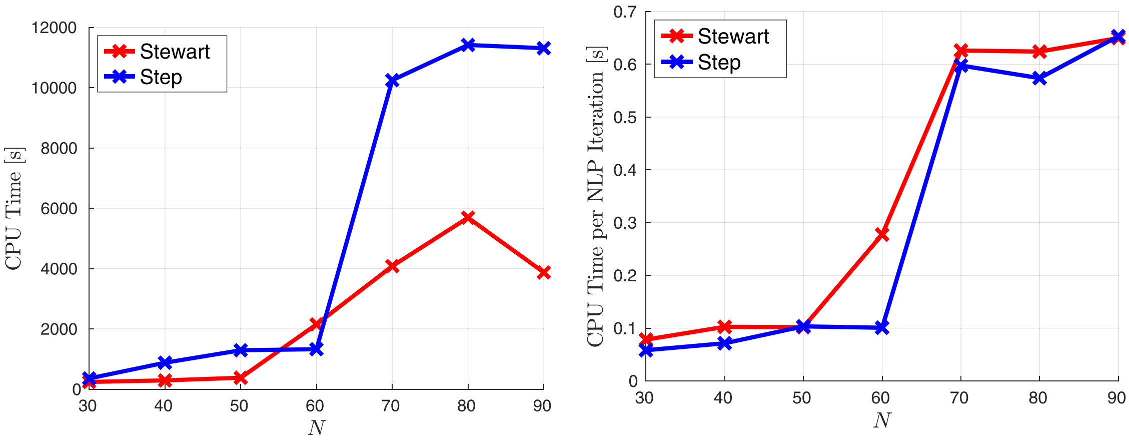


Fig. 10. The total CPU time of the homotopy loop for different N (left plot) and the CPU time per NLP solver iteration (right plot).

The step functions allow to encode logical relations within a dynamical system. These systems cam be equivalent to Filippov systems, and this paper focuses on this case. The set-valuedness of the Heaviside step functions leads to a differential inclusion. However, by using the optimality condition of a parametric linear program whose solution map corresponds to the step function, one obtains an equivalent Dynamic Complementarity System (DCS). Now the nonsmoothness and combinatorial structure is expressed algebraically and is thus suitable for numerical computations.

In the derivation of FESD, we exploit the continuity of the Lagrange multipliers in the DCS, which in turns enables the accurate detection the nonsmooth transition in time. This is also necessary for the correct computation of sensitivities of the discretized nonsmooth system.

We show how to apply the FESD method to discretize optimal control problems subject to nonsmooth systems with set-valued step functions and provide some convergence results of the method in simulation problems. Compared to Stewart's reformulation used in [14], the reformulation considered here allows for a more compact reformulation of the same system, but with a smaller number of algebraic variables. The comparison on an optimal control example shows that the new approach has a lower cost per iteration. In summary, we extend the FESD method to a problem class that allows more modeling flexibility than in our initial study in [14]. Our claims are verified by numerical examples. An implementation of the new method is provided in the open-source software package `nosnoc` [39].

CRediT authorship contribution statement

Armin Nurkanović: Writing – original draft, Visualization, Software, Methodology, Formal analysis, Conceptualization. **Anton Pozharskiy:** Writing – review & editing, Validation, Software, Investigation. **Jonathan Frey:** Writing – review & editing, Visualization, Validation, Software, Investigation. **Moritz Diehl:** Writing – review & editing, Supervision, Resources, Project administration, Funding acquisition.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Data availability

Data will be made available on request.

Acknowledgments

The authors thank two anonymous referees for detailed and constructive comments on earlier versions of this article. We also thank Florian Messerer for carefully reading the article and making suggestions that improved its readability.

References

[1] Vincent Acary, Hidde De Jong, Bernard Brogliato, Numerical simulation of piecewise-linear models of gene regulatory networks using complementarity systems, Physica D 269 (2014) 103–119.

[2] Luca Dieci, Luciano Lopez, Sliding motion on discontinuity surfaces of high co-dimension. a construction for selecting a filippov vector field, Numer. Math. 117 (4) (2011) 779–811.

[3] Alexei F. Filippov, Differential Equations with Discontinuous Righthand Sides: Control Systems, vol. 18, Springer Science & Business Media, 1988.

[4] Mathew Halm, Addressing Stiffness-Induced Challenges in Modeling and Identification for Rigid-Body Systems With Friction and Impacts (Ph.D. thesis), University of Pennsylvania, 2023.

[5] Mathew Halm, Michael Posa, Set-valued rigid-body dynamics for simultaneous, inelastic, frictional impacts, The International Journal of Robotics Research (2024) 02783649241236860.

[6] Armin Nurkanović, Sebastian Albrecht, Bernard Brogliato, Moritz Diehl, The time-freezing reformulation for numerical optimal control of complementarity Lagrangian systems with state jumps, Automatica 158 (2023) 111295.

[7] Armin Nurkanović, Moritz Diehl, Continuous optimization for control of hybrid systems with hysteresis via time-freezing, IEEE Control Syst. Lett. (2022).

[8] Armin Nurkanović, Tommaso Sartor, Sebastian Albrecht, Moritz Diehl, A time-freezing approach for numerical optimal control of nonsmooth differential equations with state jumps, IEEE Control Syst. Lett. 5 (2) (2021) 439–444.

[9] Mario Bernardo, Chris Budd, Alan Richard Champneys, Piotr Kowalczyk, Piecewise-Smooth Dynamical Systems: Theory and Applications, vol. 163, Springer Science & Business Media, 2008.

[10] Nicola Guglielmi, Ernst Hairer, An efficient algorithm for solving piecewise-smooth dynamical systems, Numer. Algorithms 89 (3) (2022) 1311–1334.

[11] Anna Machina, Arcady Ponosov, Filippov solutions in the analysis of piecewise linear models describing gene regulatory networks, Nonlinear Anal. TMA 74 (3) (2011) 882–900.

[12] Luca Dieci, Luciano Lopez, Sliding motion in filippov differential systems: theoretical results and a computational approach, SIAM J. Numer. Anal. 47 (3) (2009) 2023–2051.

[13] Vincent Acary, Bernard Brogliato, Numerical methods for nonsmooth dynamical systems: applications in mechanics and electronics, Springer Science & Business Media, 2008.

[14] Armin Nurkanović, Mario Sperl, Sebastian Albrecht, Moritz Diehl, Finite elements with switch detection for direct optimal control of nonsmooth systems, Numerische Mathematik (2024) 1–48.

[15] A. Bemporad, M. Morari, Control of systems integrating logic, dynamics, and constraints, Automatica 35 (3) (1999) 407–427.

[16] Lei Guo, Jane J. Ye, Necessary optimality conditions for optimal control problems with equilibrium constraints, SIAM J. Control Optim. 54 (5) (2016) 2710–2733.

[17] Anas Bouali, Hybrid optimal control: optimality conditions and applications (Ph.D. thesis), Avignon Université, 2023.

[18] David E. Stewart, Mihai Anitescu, Optimal control of systems with discontinuous differential equations, Numer. Math. 114 (4) (2010) 653–695.

[19] Armin Nurkanović, Sebastian Albrecht, Moritz Diehl, Limits of MPCC formulations in direct optimal control with nonsmooth differential equations, in: 2020 European Control Conference, ECC, 2020, pp. 2015–2020.

[20] Brian T. Baumrucker, Lorenz T. Biegler, MPEC strategies for optimization of a class of hybrid dynamic systems, J. Process Control 19 (8) (2009) 1248–1256.

[21] Mihai Anitescu, Paul Tseng, Stephen J. Wright, Elastic-mode algorithms for mathematical programs with equilibrium constraints: global convergence and stationarity properties, Math. Program. 110 (2) (2007) 337–371.

[22] Stefan Scholtes, Convergence properties of a regularization scheme for mathematical programs with complementarity constraints, SIAM J. Optim. 11 (4) (2001) 918–936.

[23] David Stewart, A high accuracy method for solving ODEs with discontinuous right-hand side, Numer. Math. 58 (1) (1990) 299–328.

[24] J. Albersmeyer, M. Diehl, The lifted Newton method and its application in optimization, SIAM J. Optim. 20 (3) (2010) 1655–1684.

[25] NOSNOC, 2022, https://github.com/nurkanovic/nosnoc.

[26] F. Facchinei, J.-S. Pang, Finite-Dimensional Variational Inequalities and Complementarity Problems, vol. 1–2, Springer-Verlag, New York, 2003.

[27] Bernard Brogliato, Aneel Tanwani, Dynamical systems coupled with monotone set-valued operators: Formalisms, applications, well-posedness, and stability, SIAM Rev. 62 (1) (2020) 3–129.

[28] Jong-Shi Pang, David E. Stewart, Differential variational inequalities, Math. Program. 113 (2) (2008) 345–424.

[29] David E. Stewart, Dynamics with inequalities: impacts and hard constraints, SIAM, 2011.

[30] Vincent Acary, Olivier Bonnefon, Bernard Brogliato, Nonsmooth Modeling and Simulation for Switched Circuits, vol. 69, Springer Science & Business Media, 2010.

[31] David E. Stewart, Uniqueness for index-one differential variational inequalities, Nonlinear Anal. Hybrid Syst. 2 (3) (2008) 812–818.

[32] David E. Stewart, Existence of solutions to rigid body dynamics and the painlevé paradoxes, C. R. Acad. Sci.-Ser. I-Math. 325 (6) (1997) 689–693.

[33] Jorge Cortes, Discontinuous dynamical systems, IEEE Control Syst. Mag. 28 (3) (2008) 36–73.

[34] A.F. Filippov, Differential equations with discontinuous right hand side, AMS Transl. 42 (1964) 199–231.

[35] Wim Van Roy, Armin Nurkanović, Ramin Abbasi-Esfeden, Jonathan Frey, Anton Pozharskiy, Jan Swevers, Moritz Diehl, Continuous optimization for control of finite-state machines with cascaded hysteresis via time-freezing, in: IEEE Conference on Decision and Control, 2023.

[36] E. Hairer, G. Wanner, Solving Ordinary Differential Equations II – Stiff and Differential-Algebraic Problems, second ed., Springer, Berlin Heidelberg, 1991.

[37] Asen L. Dontchev, R. Tyrrell Rockafellar, Implicit functions and solution mappings: A view from variational analysis, Springer, 2014.

[38] J.B. Rawlings, D.Q. Mayne, M.M. Diehl, Model Predictive Control: Theory, Computation, and Design, second ed., Nob Hill, 2017.

[39] Armin Nurkanović, Moritz Diehl, NOSNOC: A software package for numerical optimal control of nonsmooth systems, IEEE Control Syst. Lett. 6 (2022) 3110–3115.

[40] Daniel Ralph, Stephen J. Wright, Some properties of regularization and penalization schemes for MPECs, Optim. Methods Softw. 19 (5) (2004) 527–556.

[41] Armin Nurkanović, Anton Pozharskiy, Moritz Diehl, Solving mathematical programs with complementarity constraints arising in nonsmooth optimal control, Vietnam J. Math. (2024) (in press).

[42] Andreas Wächter, Lorenz 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.

[43] J.A.E. Andersson, J. Gillis, G. Horn, J.B. Rawlings, M. Diehl, CasADi – a software framework for nonlinear optimization and optimal control, Math. Program. Comput. 11 (1) (2019) 1–36.

[44] E. Hairer, S.P. Nørsett, G. Wanner, Solving Ordinary Differential Equations II – Stiff and Differential-Algebraic Problems, second ed., in: Springer Series in Computational Mathematics, Springer, Berlin, 1996.

[45] E. Hairer, S.P. Nø rsett, G. Wanner, Solving Ordinary Differential Equations I, second ed., in: Springer Series in Computational Mathematics, Springer, Berlin, 1993.

[46] David E. Stewart, A numerical method for friction problems with multiple contacts, ANZIAM J. 37 (3) (1996) 288–308.

[47] Anton Pozharskiy, Evaluating Methods for Solving Mathematical Programs With Complementarity Constraints Arising From Nonsmooth Optimal Control., (Master's thesis), Albert-Ludwigs-University Freiburg, 2023.

[48] Irene Cantone, Lucia Marucci, Francesco Iorio, Maria Aurelia Ricci, Vincenzo Belcastro, Mukesh Bansal, Stefania Santini, Mario di Bernardo, Diego di Bernardo, Maria Pia Cosma, A yeast synthetic network for in vivo assessment of reverse-engineering and modeling approaches, Cell 137 (1) (2009) 172–181.

[49] nosnoc_py, 2022, https://github.com/FreyJo/nosnoc_py.

[50] Jan Carius, René Ranftl, Vladlen Koltun, Marco Hutter, Trajectory optimization with implicit hard contacts, IEEE Robot. Autom. Lett. 3 (4) (2018) 3316–3323.

[51] Christian Gehring, Operational Space Control of Single Legged Hopping, (Master's thesis), Eidgenössische Technische Hochschule Zürich, Autonomous Systems Lab, 2011.

[52] HSL, A collection of Fortran codes for large scale scientific computation., 2011, http://www.hsl.rl.ac.uk.