Nonsmooth numerical optimal control

Armin Nurkanović

Systems Control and Optimization Laboratory
Department of Microsystems Engineering
University of Freiburg, Germany

based on joint work with
Moritz Diehl, Jonathan Frey, Mario Sperl, Sebastian Albrecht, Bernard Brogliato

2nd Seasonal School of the ELO-X project
University of Freiburg
6th – October, 2022
Nonsmooth Dynamics (NSD) - a classification

Regard ordinary differential equation (ODE) with a **nonsmooth** right-hand side (RHS). Distinguish three cases:

**NSD1:** non-differentiable RHS, e.g., \( \dot{x} = 1 + |x| \)

**NSD2:** state dependent switch of RHS, e.g., \( \dot{x} = 2 - \text{sign}(x) \)

**NSD3:** state dependent jump, e.g., bouncing ball, \( v(t_+) = -0.9 \, v(t_-) \)
Nonsmooth numerical optimal control - overview

Sys. with state jumps (NSD3)

Filippov system (NSD2)

Dynamic comp. system

Optimal control problem

Math. prog. with comp. constraints

Nonlinear program

Solution

User input

FESD

Time-freezing

Stewart or Step

relaxation

homotopy
Nonsmooth numerical optimal control - overview

Toolchain implemented in our open-source package NOSNOC
NOSNOC: NOnSmooth Numerical Optimal Control
Open-source package based on MATLAB, CasADi and IPOPT

Key features

1. automatic reformulation of systems with state jumps into switched systems via the time-freezing reformulation
2. automatic discretization of the OCP via FESD (high accuracy)
3. solution methods for the resulting discrete-time OCP via continuous optimization in a homotopy (no integers)

NOSNOC: https://github.com/nurkanovic/nosnoc
NOSNOC: NOnSmooth Numerical Optimal Control
Open-source package based on MATLAB, CasADi and IPOPT

NOSNOC: https://github.com/nurkanovic/nosnoc
NOSNOC: NOnSmooth Numerical Optimal Control
Open-source package based on MATLAB, CasADi and IPOPT

NOSNOC: https://github.com/nurkanovic/nosnoc

Armin Nurkanović
NOSNOC: NOnSmooth Numerical Optimal Control

Open-source package based on MATLAB, CasADi and IPOPT

NOSNOC: https://github.com/nurkanovic/nosnoc
Videos and gifs.
NSD2 - Piecewise smooth / Filippov systems
Consider the ODE

\[ \dot{x} = 2 - \text{sign}(x) \]
Motivating examples - crossing a discontinuity

Consider the ODE

\[ \dot{x} = 2 - \text{sign}(x) \]

More explicitly...

\[ \dot{x} = \begin{cases} 
3, & \text{if } x < 0 \\
1, & \text{if } x > 0 
\end{cases} \]
Motivating examples - sliding mode (simpler)

Consider the ODE

\[ \dot{x} = -\text{sign}(x) \]

And let

\[ \text{sign}(x) = \begin{cases} 
-1, & \text{if } x < 0 \\
0, & \text{if } x = 0 \\
1, & \text{if } x > 0 
\end{cases} \]

Then...

\[ \dot{x} = \begin{cases} 
1, & \text{if } x < 0 \\
0, & \text{if } x = 0 \\
-1, & \text{if } x > 0 
\end{cases} \]
Motivating examples - sliding mode (simpler)

Consider the ODE

\[
\dot{x} = -\text{sign}(x)
\]

And let

\[
\text{sign}(x) = \begin{cases} 
-1, & \text{if } x < 0 \\
0, & \text{if } x = 0 \\
1, & \text{if } x > 0
\end{cases}
\]

Then...

\[
\dot{x} = \begin{cases} 
1, & \text{if } x < 0 \\
0, & \text{if } x = 0 \\
-1, & \text{if } x > 0
\end{cases}
\]
Motivating examples - sliding mode

Consider the ODE
\[ \dot{x} = -\text{sign}(x) + 0.5 \sin(t) \]

And let
\[ \text{sign}(x) = \begin{cases} 
-1, & \text{if } x < 0 \\
0, & \text{if } x = 0 \\
1, & \text{if } x > 0 
\end{cases} \]
Consider the ODE

\[ \dot{x} = -\text{sign}(x) + 0.5 \sin(t) \]

And let

\[
\text{sign}(x) = \begin{cases} 
-1, & \text{if } x < 0 \\
0, & \text{if } x = 0 \\
1, & \text{if } x > 0 
\end{cases}
\]
Motivating examples - sliding mode

Consider the ODE

\[ \dot{x} = -\text{sign}(x) + 0.5 \sin(t) \]

And let

\[ \text{sign}(x) = \begin{cases} 
-1, & \text{if } x < 0 \\
0, & \text{if } x = 0 \\
1, & \text{if } x > 0 
\end{cases} \]

We have for some \( t > t^* \) that \( x(t) = 0 \) and \( \dot{x}(t) = 0 \)
Motivating examples - sliding mode

Consider the ODE

\[ \dot{x} = -\text{sign}(x) + 0.5 \sin(t) \]

And let

\[ \text{sign}(x) = \begin{cases} 
-1, & \text{if } x < 0 \\
0, & \text{if } x = 0 \\
1, & \text{if } x > 0 
\end{cases} \]

We have for some \( t > t^* \) that \( x(t) = 0 \) and \( \dot{x}(t) = 0 \)

That is \( \text{sign}(0) = 0 = 0.5 \sin(t) \)
Motivating examples - sliding mode

Consider the ODE

\[ \dot{x} = -\text{sign}(x) + 0.5 \sin(t) \]

And let

\[ \text{sign}(x) = \begin{cases} 
-1, & \text{if } x < 0 \\
0, & \text{if } x = 0 \\
1, & \text{if } x > 0 
\end{cases} \]

We have for some \( t > t^* \) that \( x(t) = 0 \)
and \( \dot{x}(t) = 0 \)
That is \( \text{sign}(0) = 0 = 0.5 \sin(t) \)

WHAT HAPPENED?
Consider the ODE

\[ \dot{x} \in -\text{sign}(x) + 0.5 \sin(t) \]

And let

\[ \text{sign}(x) \in \begin{cases} 
{-1}, & \text{if } x < 0 \\
[-1, 1], & \text{if } x = 0 \\
{1}, & \text{if } x > 0 
\end{cases} \]

We have for some \( t > t^* \) that \( x(t) = 0 \) and \( \dot{x}(t) = 0 \)
Consider the ODE
\[ \dot{x} \in -\text{sign}(x) + 0.5 \sin(t) \]
And let
\[ \text{sign}(x) \in \begin{cases} 
{-1}, & \text{if } x < 0 \\
[-1, 1], & \text{if } x = 0 \\
1, & \text{if } x > 0
\end{cases} \]
We have for some \( t > t^* \) that \( x(t) = 0 \) and \( \dot{x}(t) = 0 \)
That is \( \text{sign}(0) = [-1, 1] \ni 0.5 \sin(t) \)
Consider the ODE

\[ \dot{x} \in -\text{sign}(x) + 0.5 \sin(t) \]

And let

\[
\text{sign}(x) \in \begin{cases} 
-1, & \text{if } x < 0 \\
[-1, 1], & \text{if } x = 0 \\
1, & \text{if } x > 0 
\end{cases}
\]

We have for some \( t > t^* \) that \( x(t) = 0 \) and \( \dot{x}(t) = 0 \)

That is \( \text{sign}(0) = [-1, 1] \ni 0.5 \sin(t) \)

It works! Thanks to A.F. Filippov
Regard **discontinuous** right-hand side, piecewise smooth on disjoint open regions $R_i \subset \mathbb{R}^{n_x}$

### Discontinuous ODE (NSD2)

\[
\dot{x} = f_i(x, u), \text{ if } x \in R_i, \\
i \in \{1, \ldots, n_f\}
\]

**Numerical aims:**
1. exactly detect switching times
2. obtain exact sensitivities across regions
Regard **discontinuous** right-hand side, piecewise smooth on disjoint open regions $R_i \subset \mathbb{R}^n$.

**Discontinuous ODE (NSD2)**

\[
\dot{x} = f_i(x, u), \text{ if } x \in R_i, \\
i \in \{1, \ldots, n_f\}
\]

Numerical aims:

1. exactly detect switching times
2. obtain exact sensitivities across regions
3. appropriately treat evolution on boundaries (sliding mode $\rightarrow$ Filippov convexification)
Dynamics not yet well-defined on region boundaries $\partial R_i$. Idea by A.F. Filippov (1923-2006): replace ODE by differential inclusion, using convex combination of neighboring vector fields.

**Filippov Differential Inclusion**

\[
\dot{x} \in F_F(x, u) := \left\{ \sum_{i=1}^{n_f} f_i(x, u) \theta_i \mid \sum_{i=1}^{n_f} \theta_i = 1, \quad \theta_i \geq 0, \quad i = 1, \ldots, n_f, \quad \theta_i = 0, \quad \text{if } x \notin \overline{R_i} \right\}
\]

Aleksei F. Filippov (1923-2006)
image source: wikipedia
Dynamics not yet well-defined on region boundaries $\partial R_i$. Idea by A.F. Filippov (1923-2006): replace ODE by differential inclusion, using convex combination of neighboring vector fields.

**Filippov Differential Inclusion**

$$\dot{x} \in F_{F}(x, u) := \left\{ \sum_{i=1}^{n_f} f_i(x, u) \theta_i \mid \sum_{i=1}^{n_f} \theta_i = 1, \theta_i \geq 0, i = 1, \ldots, n_f, \theta_i = 0, \text{ if } x \notin R_i \right\}$$

- for interior points $x \in R_i$ nothing changes: $F_{F}(x, u) = \{f_i(x, u)\}$
- Provides meaningful generalization on region boundaries. E.g. on $R_1 \cap R_2$ both $\theta_1$ and $\theta_2$ can be nonzero
A high accuracy method for solving ODEs with discontinuous right-hand side

David Stewart
Department of Mathematics, University of Queensland, St. Lucia, Australia 4067

Received August 1, 1987/January 16, 1990

Summary. Ordinary Differential Equations with discontinuities in the state variables require a differential inclusion formulation to guarantee existence [8]. From this formulation a high accuracy method for solving such initial value problems is developed which can give any order of accuracy and “tracks” the discontinuities. The method uses an “active set” approach, and determines appropriate active sets from solutions to Linear Complementarity Problems. Convergence results are established under some non-degeneracy assumptions. The method has been implemented, and results compare favourably with previously published methods [7, 21].
How to compute convex multipliers $\theta$?

Assume sets $R_i$ given by [cf. Stewart, 1990]

$$R_i = \{ x \in \mathbb{R}^n | g_i(x) < \min_{j \neq i} g_j(x) \}$$
How to compute convex multipliers $\theta$?

Assume sets $R_i$ given by [cf. Stewart, 1990]

$$R_i = \{ x \in \mathbb{R}^n | g_i(x) < \min_{j \neq i} g_j(x) \}$$

Linear program (LP) Representation

$$\dot{x} = \sum_{i=1}^{n_f} f_i(x, u) \theta_i \quad \text{with}$$

$$\theta \in \arg \min_{\tilde{\theta} \in \mathbb{R}^{n_f}} \sum_{i=1}^{n_f} g_i(x) \tilde{\theta}_i$$

$$\text{s.t.} \quad \sum_{i=1}^{n_f} \tilde{\theta}_i = 1$$

$$\tilde{\theta} \geq 0$$

Note that the boundary between $R_i$ and $R_j$ is defined by $\{x \in \mathbb{R}^n | 0 = g_i(x) - g_j(x)\}$. 

Nonsmooth numerical optimal control
From Filippov to dynamic complementarity systems

Using the KKT conditions of the parametric LP

**LP representation**

\[
\dot{x} = F(x, u) \theta
\]

with \( \theta \in \arg\min_{\tilde{\theta} \in \mathbb{R}^{n_f}} g(x)^\top \tilde{\theta} \)

s.t. \( 0 \leq \tilde{\theta} \)

\( 1 = e^\top \tilde{\theta} \)

where

\[
F(x, u) := [f_1(x, u), \ldots, f_{n_f}(x, u)] \in \mathbb{R}^{n_x \times n_f}
\]

\[
g(x) := [g_1(x), \ldots, g_{n_f}(x)]^\top \in \mathbb{R}^{n_f}
\]

\[
e := [1, 1, \ldots, 1]^\top \in \mathbb{R}^{n_f}
\]
From Filippov to dynamic complementarity systems
Using the KKT conditions of the parametric LP

**LP representation**

\[
\dot{x} = F(x, u) \theta
\]

with \( \theta \in \arg\min_{\tilde{\theta} \in \mathbb{R}^{n_f}} g(x)^\top \tilde{\theta} \)

\[
\begin{align*}
0 & \leq \tilde{\theta} \\
1 & = e^\top \tilde{\theta}
\end{align*}
\]

where

\[
F(x, u) := [f_1(x, u), \ldots, f_{n_f}(x, u)] \in \mathbb{R}^{n_x \times n_f}
\]

\[
g(x) := [g_1(x), \ldots, g_{n_f}(x)]^\top \in \mathbb{R}^{n_f}
\]

\[
e := [1, 1, \ldots, 1]^\top \in \mathbb{R}^{n_f}
\]

Express equivalently by optimality conditions:

**Dynamic Complementarity System (DCS)**

\[
\dot{x} = F(x, u) \theta \quad (1a)
\]

\[
0 = g(x) - \lambda - e\mu 
\]

\[
0 \leq \theta \perp \lambda \geq 0 
\]

\[
1 = e^\top \theta 
\]

**Compact notation**

\[
\dot{x} = F(x, u) \theta 
\]

\[
0 = G_{LP}(x, \theta, \lambda, \mu),
\]

- \( \mu \in \mathbb{R} \) and \( \lambda \in \mathbb{R}^{n_f} \) are Lagrange multipliers
- \((1c) \iff \min\{\theta, \lambda\} = 0 \in \mathbb{R}^{n_f} \)
- Together, (1b), (1c), (1d) determine the \((2n_f + 1)\) variables \( \theta, \lambda, \mu \) uniquely
Optimal control needs to solve Nonlinear Programs (NLPs)

Original optimal control problem in continuous time

\[
\min_{x(\cdot), u(\cdot), \theta(\cdot), \lambda(\cdot), \mu(\cdot)} \int_0^T L(x, u)dt + E(x(T))
\]

s.t. \( x(0) = \bar{x}_0 \)
\[
\dot{x}(t) = \sum_{i=1}^{nf} f_i(x(t), u(t)) \theta_i(t)
\]
\[
0 = G_{LP}(x(t), \theta(t), \lambda(t), \mu(t))
\]
\[
0 \geq h(x(t), u(t)), \quad t \in [0, T]
\]
\[
0 \geq r(x(T))
\]

Assume smooth (convex) \( L, E, h, r \)

Nonsmooth dynamics make problem nonconvex

Direct methods discretize, then optimize

E.g., collocation or multiple shooting
Optimal control needs to solve Nonlinear Programs (NLPs)

**Original optimal control problem in continuous time**

\[
\begin{align*}
\min_{x(\cdot), u(\cdot), \theta(\cdot), \lambda(\cdot), \mu(\cdot)} & \quad \int_0^T L(x, u) dt + E(x(T)) \\
\text{s.t.} & \quad x(0) = \bar{x}_0 \\
& \quad \dot{x}(t) = \sum_{i=1}^{n_f} f_i(x(t), u(t)) \theta_i(t) \\
& \quad 0 = G_{LP}(x(t), \theta(t), \lambda(t), \mu(t)) \\
& \quad 0 \geq h(x(t), u(t)), \ t \in [0,T] \\
& \quad 0 \geq r(x(T))
\end{align*}
\]

Assume smooth (convex) \( L, E, h, r \)
Nonsmooth dynamics make problem nonconvex
Direct methods discretize, then optimize
E.g., collocation or multiple shooting

**Goal: discretized optimal control problem (an NLP)**

\[
\begin{align*}
\min_{x, z, u} & \quad \sum_{k=0}^{N-1} \Phi_L(x_k, z_k, u_k) + E(x_N) \\
\text{s.t.} & \quad x_0 = \bar{x}_0 \\
& \quad x_{k+1} = \Phi_{\text{dif}}^f(x_k, z_k, u_k) \\
& \quad 0 = \Phi_{\text{alg}}^f(x_k, z_k, u_k) \\
& \quad 0 \geq \Phi_h(x_k, z_k, u_k), \ k = 0, \ldots, N-1 \\
& \quad 0 \geq r(x_N)
\end{align*}
\]

Smooth convex \( \Phi_L, E, \Phi_h, r \)
Variables \( x = (x_0, \ldots), \ z = (z_0, \ldots) \) and \( u = (u_0, \ldots, u_{N-1}) \) summarized in vector \( w \in \mathbb{R}^{n_w} \)
Nonsmooth \( \Phi_{\text{alg}}^f \)
Why not use standard discretization methods?
Motivating examples - sliding mode - explicit Euler

Consider the ODE

\[ \dot{x} = -\text{sign}(x) \]

And let

\[ \text{sign}(x) = \begin{cases} 
-1, & \text{if } x < 0 \\
0, & \text{if } x = 0 \\
1, & \text{if } x > 0 
\end{cases} \]

We use

\[ x_{k+1} = x_k - h \cdot \text{sign}(x_k) \]
Consider the ODE

\[ \dot{x} = -\text{sign}(x) \]

And let

\[ \text{sign}(x) = \begin{cases} 
-1, & \text{if } x < 0 \\
0, & \text{if } x = 0 \\
1, & \text{if } x > 0 
\end{cases} \]

We use

\[ x_{k+1} = x_k - h \cdot \text{sign}(x_k) \]
Consider the ODE

\[ \dot{x} = -\text{sign}(x) \]

And let

\[ \text{sign}(x) = \begin{cases} 
-1, & \text{if } x < 0 \\
0, & \text{if } x = 0 \\
1, & \text{if } x > 0 
\end{cases} \]

We use

\[ x_{k+1} = x_k - h \cdot \text{sign}(x_k) \]
Numerical simulation example: unstable switched oscillator

Regard an unstable nonsmooth oscillator

\[ \dot{x}(t) = \begin{cases} 
A_1 x, & c(x) < 0, \\
A_2 x, & c(x) > 0,
\end{cases} \]

with

\[ A_1 = \begin{bmatrix} 1 & \omega \\ -\omega & 1 \end{bmatrix}, \quad A_2 = \begin{bmatrix} 1 & -\omega \\ \omega & 1 \end{bmatrix}, \]

\[ c(x) = x_1^2 + x_2^2 - 1, \quad \omega = 2\pi, \quad x(0) = [e^{-1} \ 0]^T \]
Numerical simulation example: importance of switch detection

- Use Implicit Runge-Kutta Gauss Legendre method of order 10!
- If implicit Euler has an accuracy of \( \approx 10^{-1} \), then this method has for the same step-size an accuracy of \( \approx 10^{-10} \)
- 4 integration steps over \( T = \frac{\pi}{2}, h = 0.3928\ldots \)

Standard IRK vs FESD IRK

Analytic solution
Numerical simulation example: importance of switch detection

- Use Implicit Runge-Kutta Gauss Legendre method of order 10!
- If implicit Euler has an accuracy of $\approx 10^{-1}$, then this method has for the same step-size an accuracy of $\approx 10^{-10}$
- 4 integration steps over $T = \frac{\pi}{2}$, $h = 0.3928$...

Standard IRK vs FESD IRK

Analytic solution
Numerical simulation example: importance of switch detection

- Use Implicit Runge-Kutta Gauss Legendre method of order $10$!
- If implicit Euler has an accuracy of $\approx 10^{-1}$, then this method has for the same step-size an accuracy of $\approx 10^{-10}$
- $4$ integration steps over $T = \frac{\pi}{2}$, $h = 0.3928$...

Standard IRK vs FESD IRK

Analytic solution
Numerical simulation example: importance of switch detection

- Use Implicit Runge-Kutta Gauss Legendre method of order 10!
- If implicit Euler has an accuracy of \( \approx 10^{-1} \), then this method has for the same step-size an accuracy of \( \approx 10^{-10} \)
- 4 integration steps over \( T = \frac{\pi}{2}, h = 0.3928\ldots \)

Standard IRK vs FESD IRK

Analytic solution
Why not smooth everything?
Continuous-time OCP

\[
\min_{x(\cdot) \in C^0([0,2])} \int_0^2 x(t)^2 dt + (x(2) - \frac{5}{3})^2 \\
\text{s.t. } \dot{x}(t) = 2 - \text{sign}(x(t)), \quad t \in [0, 2]
\]

Free initial value $x(0)$ is the effective degree of freedom.
Denote by $V_*(x_0)$ the nonsmooth objective value for the unique feasible trajectory starting at $x(0) = x_0$.

Equivalent reduced problem

\[
\min_{x_0 \in \mathbb{R}} V_*(x_0)
\]
Direct optimal control with a standard IRK discretization - smoothing
Tutorial example inspired by [Stewart & Anitescu, 2010]

Continuous-time OCP

\[
\min_{x(\cdot) \in C^0([0,2])} \int_0^2 x(t)^2 dt + (x(2) - 5/3)^2 \\
\text{s.t. } \dot{x}(t) = 2 - \tanh\left(\frac{x(t)}{\sigma}\right), \quad t \in [0,2]
\]

Free initial value \(x(0)\) is the effective degree of freedom.
IRK Radau of order 5 with \(h = 0.1\); i.e., \(N = 20\) steps

Equivalent reduced problem

\[
\min_{x_0 \in \mathbb{R}} V_\sigma(x_0)
\]
Direct optimal control with a standard IRK discretization - smoothing

Tutorial example inspired by [Stewart & Anitescu, 2010]

Continuous-time OCP

\[
\min_{x(\cdot) \in C^0([0,2])} \int_0^2 x(t)^2 dt + (x(2) - 5/3)^2
\]

s.t. \( \dot{x}(t) = 2 - \tanh\left(\frac{x(t)}{\sigma}\right), \quad t \in [0, 2] \)

Free initial value \( x(0) \) is the effective degree of freedom.
IRK Radau of order 5 with \( h = 0.1 \); i.e., \( N = 20 \) steps

Equivalent reduced problem

\[
\min_{x_0 \in \mathbb{R}} V_\sigma(x_0)
\]
Direct optimal control with a standard IRK discretization - smoothing
Tutorial example inspired by [Stewart & Anitescu, 2010]

Continuous-time OCP

<table>
<thead>
<tr>
<th>Objective</th>
<th>Nonsmooth numerical optimal control</th>
</tr>
</thead>
<tbody>
<tr>
<td>$\min_{x(\cdot) \in C^0([0,2])} \int_0^2 x(t)^2 dt + (x(2) - 5/3)^2$</td>
<td>Armin Nurkanović</td>
</tr>
<tr>
<td>s.t. $\dot{x}(t) = 2 - \tanh\left(\frac{x(t)}{\sigma}\right)$, $t \in [0,2]$</td>
<td></td>
</tr>
<tr>
<td>Free initial value $x(0)$ is the effective degree of freedom.</td>
<td></td>
</tr>
<tr>
<td>IRK Radau of order 5 with $h = 0.1$; i.e., $N = 20$ steps</td>
<td></td>
</tr>
</tbody>
</table>

Equivalent reduced problem

$\min_{x_0 \in \mathbb{R}} V_\sigma(x_0)$
Direct optimal control with a standard IRK discretization - smoothing

Tutorial example inspired by [Stewart & Anitescu, 2010]

Continuous-time OCP

\[
\min_{x(\cdot) \in C^0([0,2])} \int_0^2 x(t)^2 dt + (x(2) - 5/3)^2
\]

s.t. \( \dot{x}(t) = 2 - \tanh\left(\frac{x(t)}{\sigma}\right) \), \( t \in [0, 2] \)

Free initial value \( x(0) \) is the effective degree of freedom.
IRK Radau of order 5 with \( h = 0.1 \); i.e., \( N = 20 \) steps

Equivalent reduced problem

\[
\min_{x_0 \in \mathbb{R}} V_\sigma(x_0)
\]
Direct optimal control with a standard IRK discretization - Let us try again
See "Limits of MPCC formulations in direct optimal control with nonsmooth differential equations", N., Albrecht, Diehl, ECC 2020

Continuous-time OCP

\[
\min_{x(\cdot) \in C^0([0,2])} \int_0^2 x(t)^2 dt + (x(2) - 5/3)^2
\]

s.t. \( \dot{x}(t) = 2 - \tanh\left( \frac{x(t)}{\sigma} \right), \quad t \in [0,2] \)

Free initial value \( x(0) \) is the effective degree of freedom.
IRK Radau of order 5 with \( h = 0.01 \); i.e., \( N = 200 \) steps

Equivalent reduced problem

\[
\min_{x_0 \in \mathbb{R}} V_\sigma(x_0)
\]
Direct optimal control with a standard IRK discretization - Let us try again

See "Limits of MPCC formulations in direct optimal control with nonsmooth differential equations", N., Albrecht, Diehl, ECC 2020

**Continuous-time OCP**

\[
\begin{align*}
\min_{x(\cdot) \in C^0([0,2])} & \int_0^2 x(t)^2 \, dt + (x(2) - 5/3)^2 \\
\text{s.t.} & \quad \dot{x}(t) = 2 - \tanh\left( \frac{x(t)}{\sigma} \right), \quad t \in [0, 2]
\end{align*}
\]

Free initial value \(x(0)\) is the effective degree of freedom.

IRK Radau of order 5 with \(h = 0.01\); i.e., \(N = 200\) steps

**Equivalent reduced problem**

\[
\min_{x_0 \in \mathbb{R}} V_{\sigma}(x_0)
\]
Direct optimal control with a standard IRK discretization - Let us try again

See "Limits of MPCC formulations in direct optimal control with nonsmooth differential equations", N., Albrecht, Diehl, ECC 2020

**Continuous-time OCP**

\[
\min_{x(\cdot) \in C^0([0,2])} \int_0^2 x(t)^2 \, dt + (x(2) - 5/3)^2
\]

s.t. \( \dot{x}(t) = 2 - \tanh \left( \frac{x(t)}{\sigma} \right), \quad t \in [0,2] \)

Free initial value \( x(0) \) is the effective degree of freedom.

IRK Radau of order 5 with \( h = 0.01 \); i.e., \( N = 200 \) steps

**Equivalent reduced problem**

\[
\min_{x_0 \in \mathbb{R}} V_{\sigma}(x_0)
\]
Direct optimal control with a standard IRK discretization - Let us try again

See "Limits of MPCC formulations in direct optimal control with nonsmooth differential equations", N., Albrecht, Diehl, ECC 2020

Continuous-time OCP

\[
\begin{align*}
\min_{x(\cdot) \in C^0([0,2])} \int_0^2 x(t)^2 \, dt + (x(2) - 5/3)^2 \\
\text{s.t. } \dot{x}(t) = 2 - \tanh\left(\frac{x(t)}{\sigma}\right), \quad t \in [0,2]
\end{align*}
\]

Free initial value \(x(0)\) is the effective degree of freedom.

IRK Radau of order 5 with \(h = 0.01\); i.e., \(N = 200\) steps

Equivalent reduced problem

\[
\min_{x_0 \in \mathbb{R}} V_\sigma(x_0)
\]
Direct optimal control with a standard IRK discretization - Let us try again

See "Limits of MPCC formulations in direct optimal control with nonsmooth differential equations", N., Albrecht, Diehl, ECC 2020

Continuous-time OCP

\[
\min_{x(\cdot) \in C^0([0,2])} \int_0^2 x(t)^2 dt + (x(2) - 5/3)^2
\]

s.t. \( \dot{x}(t) = 2 - \tanh\left(\frac{x(t)}{\sigma}\right), \quad t \in [0, 2] \)

Free initial value \( x(0) \) is the effective degree of freedom.

IRK Radau of order 5 with \( h = 0.01 \); i.e., \( N = 200 \) steps

Equivalent reduced problem

\[
\min_{x_0 \in \mathbb{R}} V_{\sigma}(x_0)
\]
Direct optimal control with a standard IRK discretization
Tutorial example inspired by [Stewart & Anitescu, 2010]

- discretize the OCP with standard IRK for DCS
- numerical sensitivities wrong independent of the step-size
- smoothing works only if step-size smaller than smoothing parameter
Direct optimal control with a standard IRK discretization

Tutorial example inspired by [Stewart & Anitescu, 2010]

Optimal control of systems with discontinuous
differential equations

David E. Stewart · Mihai Anitescu

(another remarkable paper by D. Stewart)

- discretize the OCP with standard IRK for DCS
- numerical sensitivities wrong independent of the step-size
- smoothing works only if step-size smaller than smoothing parameter

Objective

\[ V_{Std}(x_0) \quad V^*(x_0) \]
Direct optimal control with a standard IRK discretization
Tutorial example inspired by [Stewart & Anitescu, 2010]

- Spurious local minima, optimizer gets trapped close to initialization
- Sensitivity correct if step-sizes smaller than smoothing parameter [Stewart & Anitescu, 2010] $\Rightarrow$ homotopy improves convergence
- Integrator NOT differentiable even when they appear to be so!
- Still, at best $O(h)$ accuracy can be expected
1. Convergence of the FESD method to a Filippov solution of the underlying system with accuracy $O(h^p)$ is proven. Here, $p$ is the order of the underlying smooth IRK scheme.

2. Convergence of numerical sensitivities to the true value with $O(h^p)$ is given. The Stewart & Anitescu problem is resolved.

3. An FESD problem needs to solve a nonlinear complementarity problem (NCP) to advance the integration. The solutions of these NCP are locally unique.
Revisiting the OCP example - now with FESD

Tutorial example inspired by [Stewart & Anitescu, 2010]

- No spurious local minima, correct sensitivities
- Convergence to the "true" local minima, both with homotopy and without it
- In contrast to the standard approach with accuracy $O(h)$, now we have $O(h^p)$
Revisiting the OCP example - now with FESD

Tutorial example inspired by [Stewart & Anitescu, 2010]

- No spurious local minima, correct sensitivities
- Convergence to the "true" local minima, both with homotopy and without it
- In contrast to the standard approach with accuracy $O(h)$, now we have $O(h^p)$
- FESD resolves the accuracy and convergence issues
Nonsmooth Dynamics (NSD) - a classification

Regard ordinary differential equation (ODE) with a **nonsmooth** right-hand side (RHS). Distinguish three cases:

**NSD1**: non-differentiable RHS, e.g., \( \dot{x} = 1 + |x| \)

**NSD2**: state dependent switch of RHS, e.g., \( \dot{x} = 2 - \text{sign}(x) \)

**NSD3**: state dependent jump, e.g., bouncing ball, \( v(t_+ = -0.9 \cdot v(t_-) \)
NSD3 state jump example: bouncing ball

Bouncing ball with state \(x = (q, v)\):

\[
\dot{q} = v, \quad m \dot{v} = -mg, \quad \text{if } q > 0
\]

\[
v(t^+) = -0.9 \, v(t^-), \quad \text{if } q(t^-) = 0 \text{ and } v(t^-) < 0
\]

Time plot of bouncing ball trajectory:

Phase plot of bouncing ball trajectory:

**Question:** could we transform NSD3 systems into (easier) NSD2 systems?
Three ideas:

1. mimic state jump by **auxiliary dynamic system** $\dot{x} = f_{aux}(x)$ on prohibited region
2. introduce a **clock state** $t(\tau)$ that stops counting when the auxiliary system is active
3. adapt speed of time, $\frac{dt}{d\tau} = s$ with $s \geq 1$, and **impose terminal constraint** $t(T) = T$
Augmented state \((x, t) \in \mathbb{R}^{n+1}\) evolves in \textbf{numerical time} \(\tau\). Augmented system is nonsmooth, of NSD2 type:

\[
\frac{d}{d\tau} \begin{bmatrix} x \\ t \end{bmatrix} = \begin{cases} 
  s \begin{bmatrix} f(x) \\ 1 \end{bmatrix}, & \text{if } c(x) \geq 0 \\
  s f_{aux}(x) \begin{bmatrix} 0 \\ 1 \end{bmatrix}, & \text{if } c(x) < 0 
\end{cases}
\]

- During normal times, system and clock state evolve with adapted speed \(s \geq 1\).
- Auxiliary system \(\frac{dx}{d\tau} = f_{aux}(x)\) mimics state jump while time is frozen, \(\frac{dt}{d\tau} = 0\).
Time-freezing for bouncing ball example

Evolution of physical time (clock state) during augmented system simulation ($s = 1$).

We can recover the true solution by plotting $x(\tau)$ vs. $t(\tau)$ and disregarding "frozen pieces".
A tracking OCP example with Time-Freezing and FESD in NOSNOC

Regard bouncing ball in two dimensions driven by bounded force: \( \ddot{q} = u \)

\[
\begin{align*}
\min_{x(.), u(.), s(.), 
\theta(.), \lambda(.), \mu(.)} & \quad \int_0^T (q - q_{\text{ref}}(\tau))^\top (q - q_{\text{ref}}(\tau)) s(\tau) \, d\tau \\
\text{s.t.} & \quad x(0) = x_0, \quad t(T) = T, \\
& \quad x'(\tau) = \sum_{i=1}^{n_f} \theta_i(\tau) f_i(x(\tau), u(\tau), s(\tau)), \\
& \quad 0 = g(x(\tau)) - \lambda(\tau) - \mu(\tau)e, \\
& \quad 0 \leq \lambda(\tau) \perp \theta(\tau) \geq 0, \\
& \quad 1 = e^\top \theta(\tau), \\
& \quad \|u(\tau)\|_2^2 \leq u_{\text{max}}^2, \\
& \quad 1 \leq s(\tau) \leq s_{\text{max}}, \quad \tau \in [0, T].
\end{align*}
\]

\( q_{\text{ref}}(\tau) = (R \cos(\omega t(\tau)), R \sin(\omega t(\tau))) \).

- augmented state \( x = (q, \dot{q}, t) \in \mathbb{R}^5 \)
- \( n_f = 9 \) regions (8 with auxiliary dynamics for state jumps)
Results with slowly moving reference

For $\omega = \pi$, tracking is easy: no jumps occur in optimal solution.

- Regard time horizon of two periods
- $N = 25$ equidistant control intervals
- use FESD with $N_{FE} = 3$ finite elements with Radau 3 on each control interval
- each FESD interval has one constant control $u$ and one speed of time $s$
- MPCC solved via $\ell_\infty$ penalty reformulation and homotopy
- For homotopy convergence: in total 4 NLPs solved with IPOPT via CasADi

States and controls in physical time.
Results with slowly moving reference - movie

For $\omega = \pi$, tracking is easy: no jumps occur in optimal solution.
Results with fast reference

For $\omega = 2\pi$, tracking is only possible if ball bounces against walls.

States and controls in numerical time. States and controls in physical time.
Results with fast reference - movie

For $\omega = 2\pi$, tracking is only possible if ball bounces against walls.
Homotopy: first iteration vs converged solution

Geometric trajectory

After the first homotopy iteration

The solution trajectory after convergence
Physical vs. Numerical Time

for $\omega = \pi$

for $\omega = 2\pi$
Hybrid systems and finite automaton

\[ \dot{x} = f_A(x) \]
\[ w = 0 \]

\[ \psi(x) \geq 1 \]

\[ \dot{x} = f_B(x) \]
\[ w = 1 \]

\[ \psi(x) \leq 0 \]
Hybrid systems and finite automaton

Hybrid system with hysteresis (*incomplete description*)

\[ \dot{x} = f(x, w) = (1 - w)f_A(x) + wf_B(x) \]
Tutorial example: thermostat with hysteresis

\[ \dot{x} = -0.2x, \quad w = 0 \]

\[ x \leq 18 \]

\[ \dot{x} = -0.2x + u_h, \quad w = 1 \]

\[ x \geq 20 \]
Tutorial example: thermostat with hysteresis

\[
\begin{align*}
\dot{x} &= -0.2x \\
w &= 0 \\
x &\leq 18 \\
\dot{x} &= -0.2x + u_h \\
w &= 1 \\
x &\geq 20
\end{align*}
\]
Hysteresis: a system with state jumps

Hybrid system with hysteresis

\begin{align*}
\dot{x} &= f(x, w) = (1 - w)f_A(x) + w f_B(x) \\
\dot{w} &= 0
\end{align*}

The State Jump Law

1. if \( w(t - s) = 0 \) and \( \psi(x(t - s)) = 1 \), then \( x(t + s) = x(t - s) \) and \( w(t + s) = 1 \)
2. if \( w(t - s) = 1 \) and \( \psi(x(t - s)) = 0 \), then \( x(t + s) = x(t - s) \) and \( w(t + s) = 0 \)

Remember: \( w(t) \) is now a discontinuous differential state!
Hysteresis: a system with state jumps

Hybrid system with hysteresis

\[ \dot{x} = f(x, w) = (1 - w)f_A(x) + wf_B(x) \]
\[ \dot{w} = 0 \]

The State Jump Law

1. if \( w(t_s^-) = 0 \) and \( \psi(x(t_s^-)) = 1 \), then \( x(t_s^+) = x(t_s^-) \) and \( w(t_s^+) = 1 \)
2. if \( w(t_s^-) = 1 \) and \( \psi(x(t_s^-)) = 0 \), then \( x(t_s^+) = x(t_s^-) \) and \( w(t_s^+) = 0 \)

Remember: \( w(t) \) is now a discontinuous differential state!
Tutorial example: thermostat and time-freezing

\[ x(\tau) \]

\[ w(\tau) \]

\[ t(\tau) \]

\[ \tau \text{ [numerical time]} \]

\[ t \text{ [physical time]} \]

Nonsmooth numerical optimal control

Armin Nurkanović
Everything except the blue solid curve is prohibited in the $\psi(x), w -$ space (use 1st principle of time-freezing)

The evolution happens in a lower-dimensional space $\Rightarrow$ sliding mode (use 4th principle of time-freezing)
Time-freezing: partitioning of the space

An efficient partition leads to less variables in FESD

Partition the state space into Voronoi regions:

\[ R_i = \{ z \mid \|z - z_i\|^2 < \|z - z_j\|^2, \ j = 1, \ldots, 4, j \neq i \}, \ \ z = (\psi(x), w) \]
Partition the state space into Voronoi regions:

\[ R_i = \{ z \mid \| z - z_i \|^2 < \| z - z_j \|^2, j = 1, \ldots, 4, j \neq i \}, \ z = (\psi(x), w) \]

Feasible region for initial hybrid system with hysteresis on the region boundaries
Time-freezing: auxiliary dynamics
To mimic state jumps in finite numerical time

Use regions $R_2$ and $R_3$ to define auxiliary dynamics for the state jumps of $w(\cdot)$
Use regions $R_2$ and $R_3$ to define auxiliary dynamics for the state jumps of $w(\cdot)$.

Evolution in $w$–direction happens only for $\psi \in \{0, 1\}$. 
Time-freezing: auxiliary dynamics

To mimic state jumps in finite numerical time

- Use regions $R_2$ and $R_3$ to define auxiliary dynamics for the state jumps of $w(\cdot)$
- Evolution in $w$–direction happens only for $\psi \in \{0, 1\}$
- Zoom in: with a naive approach one has locally nonunique solutions
Time-freezing: auxiliary dynamics

The new state space of the system is \( y = (x, w, t) \in \mathbb{R}^{n_x+2} \)

**Auxiliary dynamics**

\[
\frac{dy}{d\tau} = f_{aux,A}(y) := \begin{bmatrix} 0 \\ -\gamma(\psi(x)) \\ 0 \end{bmatrix}
\]

\[
\frac{dy}{d\tau} = f_{aux,B}(y) := \begin{bmatrix} 0 \\ \gamma(\psi(x) - 1) \\ 0 \end{bmatrix}
\]

\[
\gamma(x) = \frac{ax^2}{1 + x^2}
\]
Time-freezing: auxiliary dynamics

● Smart choice of auxiliary dynamics resolves the nonuniqueness issue
Time-freezing: auxiliary dynamics

- Smart choice of auxiliary dynamics resolves the nonuniqueness issue
- Zoom in: escape only in one direction possible
Time-freezing: DAE forming dynamics

Stop the state jump and construct suitable sliding mode

▶ Dynamics in $R_1$ and $R_4$ stops evolution of auxiliary ODE - similar to inelastic impacts
Time-freezing: DAE forming dynamics
Stop the state jump and construct suitable sliding mode

▶ Dynamics in $R_1$ and $R_4$ stops evolution of auxiliary ODE - similar to inelastic impacts
▶ Sliding modes on $R_A := \partial R_1 \cap \partial R_2$ and $R_B := \partial R_3 \cap \partial R_4$ match $f_A(y)$ and $f_B(y)$, resp.
Time-freezing: summary

DAE-forming dynamics

\[ y = (x, w, t) \]
\[ \frac{dy}{d\tau} = f_{df,A}(y) := \begin{bmatrix} 2f_A(x) \\ \gamma(\psi(x)) \\ 2 \end{bmatrix} \]
\[ \frac{dy}{d\tau} = f_{df,B}(y) := \begin{bmatrix} 2f_B(x) \\ -\gamma(\psi(x) - 1) \\ 2 \end{bmatrix} \]

- In total four regions \( R_i \), \( i = 1, 2, 3, 4 \) and evolution of original system is the **sliding mode**.
### DAE-forming dynamics

\[
y = (x, w, t)
\]

\[
\frac{dy}{d\tau} = f_{df,A}(y) := \begin{bmatrix} 2f_A(x) \\ \gamma(\psi(x)) \\ 2 \end{bmatrix}
\]

\[
\frac{dy}{d\tau} = f_{df,B}(y) := \begin{bmatrix} 2f_B(x) \\ -\gamma(\psi(x) - 1) \\ 2 \end{bmatrix}
\]

- In total four regions \( R_i \), \( i = 1, 2, 3, 4 \) and evolution of original system is the **sliding mode**
- Regions \( R_2 \) and \( R_3 \) equipped with aux. dynamics \( y' = f_2(y) = f_{aux,A}(y) \) and \( y' = f_3(y) = f_{aux,B}(y) \), resp., to mimic state jump
Time-freezing: summary

DAE-forming dynamics

\[ y = (x, w, t) \]
\[ \frac{dy}{d\tau} = f_{df,A}(y) := \begin{bmatrix} 2f_A(x) \\ \gamma(\psi(x)) \\ 2 \end{bmatrix} \]
\[ \frac{dy}{d\tau} = f_{df,B}(y) := \begin{bmatrix} 2f_B(x) \\ -\gamma(\psi(x) - 1) \\ 2 \end{bmatrix} \]

- In total four regions \( R_i \), \( i = 1, 2, 3, 4 \) and evolution of original system is the **sliding mode**

- Regions \( R_2 \) and \( R_3 \) equipped with aux. dynamics \( y' = f_2(y) = f_{aux,A}(y) \) and \( y' = f_3(y) = f_{aux,B}(y) \), resp., to mimic state jump

- Regions \( R_1 \) and \( R_4 \) equipped with DAE-forming dynamics \( y' = f_1(y) = f_{df,A}(y) \) and \( y' = f_4(y) = f_{df,B}(y) \), resp., to recover original dynamics in sliding mode
Time-freezing: summary

**DAE-forming dynamics**

\[ y = (x, w, t) \]

\[
\frac{dy}{d\tau} = f_{df,A}(y) := \begin{bmatrix}
2f_A(x) \\
\gamma(\psi(x)) \\
2
\end{bmatrix}
\]

\[
\frac{dy}{d\tau} = f_{df,B}(y) := \begin{bmatrix}
2f_B(x) \\
-\gamma(\psi(x) - 1) \\
2
\end{bmatrix}
\]

- In total four regions \( R_i \), \( i = 1, 2, 3, 4 \) and evolution of original system is the **sliding mode**
- Regions \( R_2 \) and \( R_3 \) equipped with aux. dynamics \( y' = f_2(y) = f_{aux,A}(y) \) and \( y' = f_3(y) = f_{aux,B}(y) \), resp., to mimic state jump
- Regions \( R_1 \) and \( R_4 \) equipped with DAE-forming dynamics \( y' = f_1(y) = f_{df,A}(y) \) and \( y' = f_4(y) = f_{df,B}(y) \), resp., to recover original dynamics in sliding mode
- E.g., \( w' = 0 \Rightarrow \theta_1 f_{df,A}(y) + \theta_2 f_{aux,A}(y) = f_A(y) \) (sliding mode)
- Conclusion: we have a PSS and can treat it with FESD
Time optimal control of a car with a turbo accelerator

Example from [Avraam, 2000] solved with NOSNOC

\[
\begin{align*}
\dot{q} &= v \\
\dot{v} &= u \\
\dot{L} &= c_N \\
w &= 0
\end{align*}
\]

\[
\begin{align*}
\dot{q} &= v \\
\dot{v} &= 3u \\
\dot{L} &= c_T \\
w &= 1
\end{align*}
\]

\[
v \geq 15
\]

\[
v \leq 10
\]
Time optimal control of a car with a turbo accelerator

Example from [Avraam, 2000] solved with NOSNOC

\[
\begin{align*}
\dot{q} &= v \\
\dot{v} &= u \\
\dot{L} &= c_N \\
w &= 0
\end{align*}
\]

\[
\begin{align*}
\dot{q} &= v \\
\dot{v} &= 3u \\
\dot{L} &= c_T \\
w &= 1
\end{align*}
\]

\[v \geq 15\]

\[v \leq 10\]

\[\begin{align*}
\min_{y(\cdot), u(\cdot), s(\cdot)} & \quad t(\tau_f) + L(\tau_f) \\
\text{s.t.} & \quad y(0) = (z_0, 0) \\
& \quad y'(\tau) \in s(\tau)F_{TF}(y(\tau), u(\tau)) \\
& \quad -\bar{u} \leq u(\tau) \leq \bar{u} \\
& \quad \bar{s}^{-1} \leq s(\tau) \leq \bar{s} \\
& \quad -\bar{v} \leq v(\tau) \leq \bar{v} \quad \tau \in [0, \tau_f] \\
& \quad (q(\tau_f), v(\tau_f)) = (q_f, v_f)
\end{align*}\]
Scenario 1: turbo and nominal cost the same

\[ c_N = c_T \]
Scenario 2: Turbo is Expensive

$c_N < c_T$

\[ \begin{align*}
  v(t) & \quad t \\
  0 & \quad 0 \quad 5 \quad 10 \\
  0 & \quad 5 \quad 10 \\
  0 & \quad 5 \quad 10 \\
  0 & \quad 5 \quad 10 \\
  0 & \quad 5 \quad 10 \\
  0 & \quad 5 \quad 10 \\
  0 & \quad 5 \quad 10
\end{align*} \]

\[ \begin{align*}
  u(t) & \quad t \\
  0 & \quad 0 \quad 5 \quad 10 \\
  0 & \quad 5 \quad 10 \\
  0 & \quad 5 \quad 10 \\
  0 & \quad 5 \quad 10 \\
  0 & \quad 5 \quad 10 \\
  0 & \quad 5 \quad 10 \\
  0 & \quad 5 \quad 10
\end{align*} \]

\[ \begin{align*}
  w(t) & \quad t \\
  0 & \quad 0 \quad 5 \quad 10 \\
  0 & \quad 5 \quad 10 \\
  0 & \quad 5 \quad 10 \\
  0 & \quad 5 \quad 10 \\
  0 & \quad 5 \quad 10 \\
  0 & \quad 5 \quad 10 \\
  0 & \quad 5 \quad 10
\end{align*} \]

Nonsmooth numerical optimal control

Armin Nurkanović
NOSNOC vs MILP/MINLP formulations

Benchmark on time-optimal control problem of a car with turbo

- compare CPU time as function of number of control intervals $N$ (left) and solution accuracy (right)
- MILP (Gurobi): solve problem with fixed $T$ until indefeasibly happens with grid search in $T$
- MILP/MINLP and NOSNOC-Std no switch detection = low accuracy
Conclusions and outlook

Conclusions

- Finite Elements with Switch Detection (FESD) allow highly accurate simulation and optimal control for nonsmooth systems of level NSD2
- FESD resolves many of the issues that standard methods have: integration accuracy, convergence of sensitivities
- Main difficulty: solving the Mathematical Programs with Complementarity Constraints (MPCC)

Outlook

- Improve on MPCC methods, test other existing relaxation methods (work in progress, soon available in NOSNOC)
- Properties of FESD-MPCC solutions. Are all stationary points strongly stationary points?
- Combinatorial methods for MPCC arising in nonsmooth optimal control
- Efficient NCP solvers for FESD subproblems
Conclusions and outlook

Conclusions

▶ Finite Elements with Switch Detection (FESD) allow highly accurate simulation and optimal control for nonsmooth systems of level NSD2
▶ FESD resolves many of the issues that standard methods have: integration accuracy, convergence of sensitivities
▶ Main difficulty: solving the Mathematical Programs with Complementarity Constraints (MPCC)

Outlook

▶ Improve on MPCC methods, test other existing relaxation methods (work in progress, soon available in NOSNOC)
▶ Properties of FESD-MPCC solutions. Are all stationary points strongly stationary points?
▶ Combinatorial methods for MPCC arising in nonsmooth optimal control
▶ Efficient NCP solvers for FESD subproblems
A time-freezing approach for numerical optimal control of nonsmooth differential equations with state jumps.

NOSNOC: A software package for numerical optimal control of nonsmooth systems.

Continuous optimization for control of hybrid systems with hysteresis via time-freezing

Finite Elements with Switch Detection for Direct Optimal Control of Nonsmooth Systems.
References 2 (external, theory and algorithms)

- Differential equations with discontinuous right-hand side.  

- MPEC strategies for optimization of a class of hybrid dynamic systems.  

- A high accuracy method for solving ODEs with discontinuous right-hand side.  

- Optimal control of systems with discontinuous differential equations.  
On the implementation of an interior-point filter line-search algorithm for large-scale nonlinear programming.

CasADi: a software framework for nonlinear optimization and optimal control.
Thank you very much for your attention!