Lecture 9: Summary - formulating nonsmooth optimal control problems

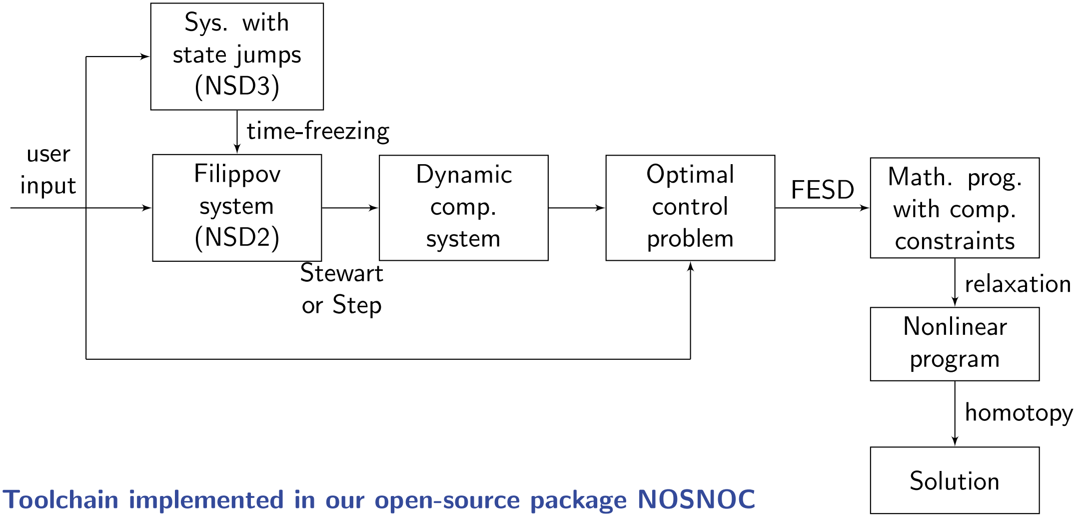
Moritz Diehl and Armin Nurkanović

Systems Control and Optimization Laboratory (syscop)

Summer School on Direct Methods for Optimal Control of Nonsmooth Systems
September 11-15, 2023

universität freiburg

Direct optimal control for nonsmooth system - overview



Toolchain implemented in our open-source package NOSNOC

NOSNOC: NOnSmooth Numerical Optimal Control

Open-source package based on CasADi

Key features

1. automatic reformulation NSD3 (state jumps) to NSD2 (switched systems) via the time-freezing reformulation
2. automatic discretization of the OCP via FESD - high accuracy[4]
3. solution methods for the discrete-time OCP via continuous optimization

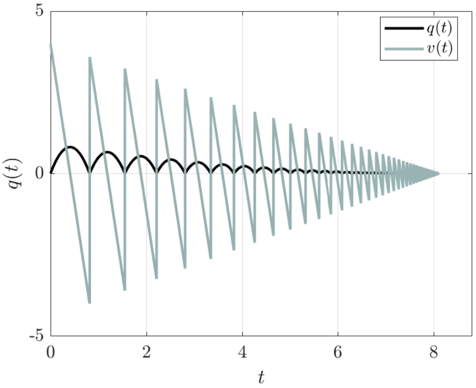
NOSNOC: `https://github.com/nurkanovic/nosnoc`

nosnoc_py: `https://github.com/FreyJo/nosnoc_py`

Your contributions and feedback are welcome!

Nonsmooth system exhibit phenomenon not seen in smooth systems

- Zeno = infinitely many switches in finite time
- Reduction of dimensions, sliding mode
- Stability and instability due to switching
- Numerical chattering
- Nonsmooth sensitivities
- Switches between ODEs and DAEs of different index



Nonsmooth systems switch between ODEs and DAEs of different index

DAEs in Hessenberg form

DAE of index 1

$$\dot{x}(t) = f(t, x(t), z(t), u(t))$$
$$0 = g(t, x(t), z(t), u(t))$$

with $\frac{\partial g}{\partial z}$ nonsingular for all t

DAE of index 2

$$\dot{x}(t) = f(t, x(t), z(t), u(t))$$
$$0 = g(t, x(t), u(t))$$

with $\frac{\partial g}{\partial x}\frac{\partial f}{\partial z}$ nonsingular for all t

DAE of index 3

$$\dot{x}(t) = f_x(t, x(t), y(t))$$
$$\dot{y}(t) = f_y(t, x(t), y(t), z(t), u(t))$$
$$0 = g(t, x(t), u(t))$$

with $\frac{\partial g}{\partial x}\frac{\partial f_x}{\partial y}\frac{\partial f_y}{\partial z}$ nonsingular for all t

- RK methods most often stated for DAEs in a canonical form
- We can get an idea of the differential index by looking at the arguments of $g(\cdot)$
- Nonsmooth system switch between ODEs and DAEs of different index:
  - Entering sliding mode: from ODE to index 2 DAE
  - CLS with impacts: from ODE to index 3 DAE
  - leaving sliding modes: from DAE to ODE
- Numerical methods should be suitable for all occurring scenarios, Radau IIA is often a good choice.

Order reduction of Runge-Kutta methods in higher index DAEs

- RK methods experience **order reduction** for higher index DAEs
- Different components of the solution may have different accuracy
- Index reduction requires consistent initialization and drift handling
- Condition number of Newton matrix $O(h^{-k})$ where k is the index

Method	n_s	ODE	DAE index 1		DAE index 2	
		x	x	z	x	z
Gauss-Legendre	odd	$2n_s$	$2n_s$	n_s	n_s+1	n_s-1
	even	$2n_s$	$2n_s$	n_s+1	n_s	n_s-2
Radau IA	odd/even	$2n_s-1$	$2n_s-1$	n_s	n_s	n_s-1
Radau IIA	odd/even	$2n_s-1$	$2n_s-1$	$2n_s-1$	$2n_s-1$	n_s
Lobatto IIIA	odd	$2n_s-2$	$2n_s-2$	$2n_s-2$	$2n_s-2$	n_s-1
	even	$2n_s-2$	$2n_s-2$	$2n_s-2$	$2n_s-2$	n_s
Lobatto IIIC	odd/even	$2n_s-2$	$2n_s-2$	$2n_s-2$	$2n_s-2$	n_s-1

The vast field of nonsmooth dynamical systems

Switches and jumps can be:

1. external, integer controls, triggered anywhere in the state space
2. **internal**, state depended, modeled with nonsmooth ODEs - this course

The vast field of nonsmooth dynamical systems

Switches and jumps can be:

1. external, integer controls, triggered anywhere in the state space
2. **internal**, state depended, modeled with nonsmooth ODEs - this course

Many mathematical formalisms to model them:

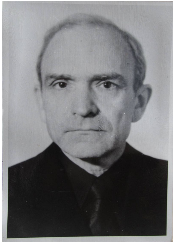
1. differential inclusions, measure differential inclusions (Filippov systems)
2. differential variational inequalities
3. **dynamic complementarity systems** (CLS, Stewart's and the Heaviside step reformulation)
4. piecewise smooths systems, **Filippov systems**
5. project dynamical systems
6. hybrid automaton
7. Moreau's sweeping processes
8. ...

Recap on Filippov Convexification

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

Filippov Differential Inclusion

$$\dot{x} \in F_{\mathrm{F}}(x,u) := \Big\{ \sum_{i=1}^{n_f} f_i(x,u)\,\theta_i \;\Big|\; \sum_{i=1}^{n_f} \theta_i = 1, \quad \theta_i \geq 0, \quad i = 1,\dots n_f, \quad \theta_i = 0, \quad \text{if } x \notin \overline{R_i} \Big\}$$



Aleksei F. Filippov (1923-2006)
image source: wikipedia

- for interior points $x \in R_i$ nothing changes: $F_{\mathrm{F}}(x,u) = \{f_i(x,u)\}$
- Provides meaningful generalization on region boundaries. E.g. on $\overline{R_1} \cap \overline{R_2}$ both θ_1 and θ_2 can be nonzero

How to compute convex multipliers θ?

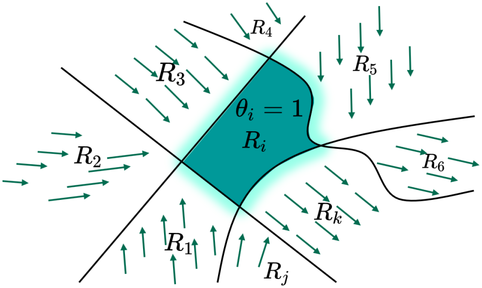
Assume sets R_i given by [cf. Stewart, 1990]

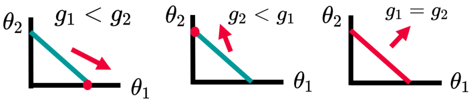
$$\boxed{R_i = \left\{ x \in \mathbb{R}^n \middle| g_i(x) < \min_{j \neq i} g_j(x) \right\}}$$

Linear program (LP) Representation

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

$$\begin{aligned} \theta \in \arg \min_{\tilde{\theta} \in \mathbb{R}^{n_f}} \quad & \sum_{i=1}^{n_f} g_i(x)\,\tilde{\theta}_i \\ \text{s.t.} \quad & \sum_{i=1}^{n_f} \tilde{\theta}_i = 1 \\ & \tilde{\theta} \geq 0 \end{aligned}$$





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

Motivating example

Consider two switching functions $\psi_1(x)$ and $\psi_2(x)$ and four regions

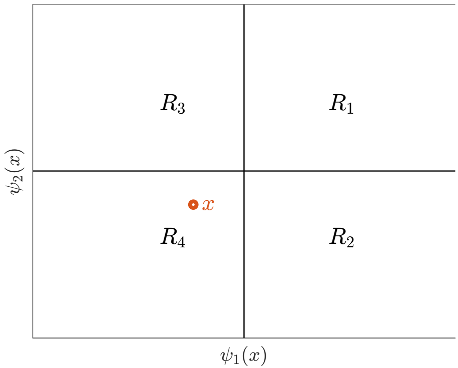
Nonsmooth system

$$\dot{x} = \alpha_1\alpha_2 f_1(x) + \alpha_1(1-\alpha_2) f_2(x) + (1-\alpha_1)\alpha_2 f_3(x) + (1-\alpha_1)(1-\alpha_2) f_4(x)$$

Step representation

$\theta_i = 1$ if $x \in R_i$:

$$\begin{aligned}\theta_1 &= \alpha_1\alpha_2\\ \theta_2 &= \alpha_1(1-\alpha_2)\\ \theta_3 &= (1-\alpha_1)\alpha_2\\ \theta_4 &= (1-\alpha_1)(1-\alpha_2)\end{aligned}$$



$$R_4 = \{x \in \mathbb{R}^{n_x} \mid \psi_1(x) < 0,\ \psi_2(x) < 0\}$$

Stewart vs. Heaviside step

Dynamic complementarity system

$$
\begin{aligned}
\dot{x} &= F(x,u)\,\theta \\
0 &= g_i(x) - \lambda_i - \mu,\ i = 1,\dots,2^{n_\psi}, \\
0 &\leq \theta \perp \lambda \geq 0 \\
1 &= e^\top \theta
\end{aligned}
$$

Heaviside step DCS

$$
\begin{aligned}
\dot{x} &= F(x,u)\,\theta \\
\theta_i &= \prod_{j=1}^{n_\psi} \Big(\frac{1-S_{i,j}}{2} + S_{i,j}\alpha_j\Big), i = 1,\dots,2^{n_\psi} \\
\psi(x) &= \lambda^{\mathrm{p}} - \lambda^{\mathrm{n}} \\
0 &\leq \lambda^{\mathrm{n}} \perp \alpha \geq 0 \\
0 &\leq \lambda^{\mathrm{p}} \perp e - \alpha \geq 0
\end{aligned}
$$

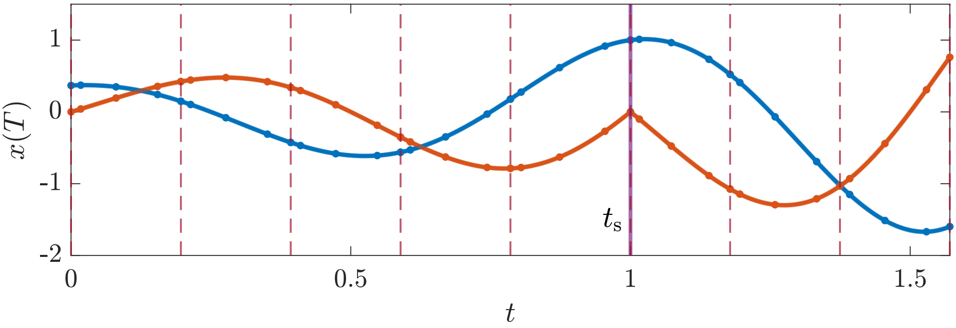
Table: Comparisons of the problem sizes in Stewart's and the step reformulation for a fixed n_ψ.

Method	Number of systems	n_{alg}	n_{comp}	n_{eq}
Stewart	2^{n_ψ}	$2 \cdot 2^{n_\psi} + 1$	2^{n_ψ}	$2^{n_\psi} + 1$
Heaviside step	2^{n_ψ}	$2^{n_\psi} + 3n_\psi$	$2n_\psi$	$n_\psi + n_f$

Time discretization methods for nonsmooth ODEs

Approaches to discretize and simulate a nonsmooth ODE

1) event-capturing, time-stepping methods (can handle Zeno, low accuracy)
2) smoothing and penalty methods (low accuracy, easy to implement)
3) **event-driven, switch-detecting, active-set methods** (cannot handle Zeno, high accuracy)



Direct optimal control with a time stepping IRK discretization

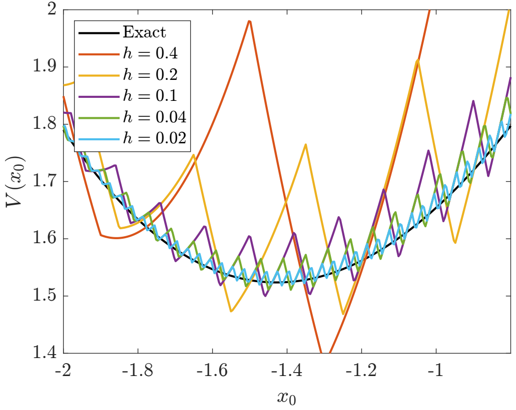
Tutorial example inspired by [Stewart & Anitescu, 2010]

Continuous-time OCP

$$\min_{x(\cdot),\lambda(\cdot),s(\cdot)} \int_0^2 x(t)^2 \mathrm{d}t + (x(2) - 5/3)^2$$

$$\begin{aligned} \text{s.t.} \quad & \dot{x}(t) = 2 - s(t) \\ & 0 \leq \lambda(t) - x(t) \perp 1 + s(t) \geq 0 \\ & 0 \leq \lambda(t) \perp 1 - s(t) \geq 0, \; t \in [0,2] \end{aligned}$$

- Discretize the DCS with fixed step size IRK methods
- E.g., midpoint rule, Gauss-Legendre IRK with $n_{\mathrm{s}} = 1$, accuracy $O(h^2)$
- decreasing the step size might worsen the situation



Many artificial local minima and wrong derivatives

Direct optimal control with a time stepping IRK discretization

Tutorial example inspired by [Stewart & Anitescu, 2010]

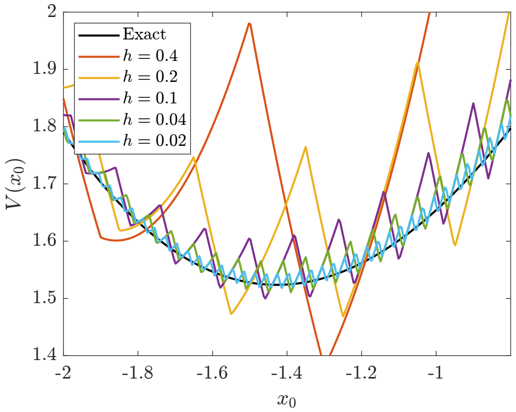
Discrete-time OCP

$$\min_{\mathbf{x},\mathbf{z}} \quad \sum_{n=0}^{N-1} \ell_n(x_n) + (x_N - 5/3)^2$$

$$\text{s.t.} \quad x_{n+1} = \phi_f(x_n, z_n)$$

$$0 = \phi_{\text{int}}(x_n, z_n), \;\; n = 0, \ldots N-1$$

- Discretize the DCS with fixed step size IRK methods
- E.g., midpoint rule, Gauss-Legendre IRK with $n_\text{s} = 1$, accuracy $O(h^2)$
- decreasing the step size might worsen the situation



Many artificial local minima and wrong derivatives

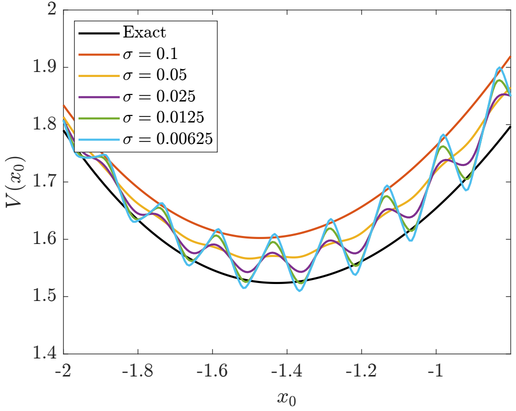
Direct optimal control with a standard IRK discretization - smoothing

Tutorial example inspired by [Stewart & Anitescu, 2010]

Continuous-time OCP

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

- midpoint rule, with $h = 0.05;\ N = 40$



If $h \gg \sigma$, then the smooth approximation behaves **the same as the nonsmooth problem!**

Direct optimal control with a standard IRK discretization - smoothing

Tutorial example inspired by [Stewart & Anitescu, 2010]

Smoothed continuous-time OCP

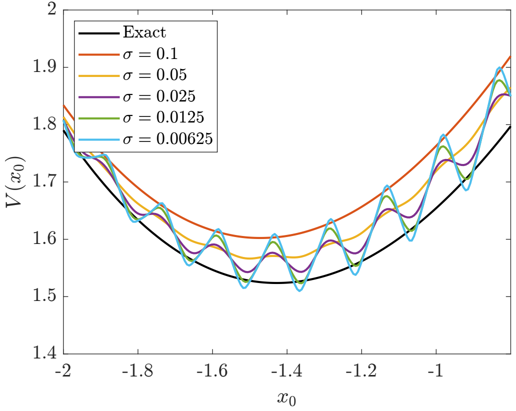
$$\min_{x(\cdot)\in\mathcal{C}^\infty([0,2])} \int_0^2 x(t)^2 \mathrm{d}t + (x(2)-5/3)^2$$

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

Equivalent reduced problem

$$\min_{x_0\in\mathbb{R}} V_\sigma(x_0)$$

- midpoint rule, with $h = 0.05$; $N = 40$



If $h \gg \sigma$, then the smooth approximation behaves **the same as the nonsmooth problem!**

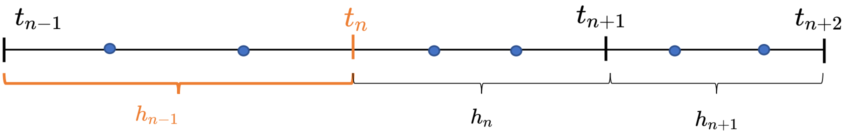
Main ideas of FESD

Based on [Baumrucker & Biegler, 2009], [N. et. al, 2022, 2022a, 2023]

FESD overview

1. Transform Filippov DI into equivalent DCS - Stewart or Heaviside step (Lecture 5)
2. Consider at least two integration intervals = finite elements
3. Use general implicit Runge-Kutta methods (Lectures 2 and 3)
4. Let step sizes h_n be degrees of freedom
5. Cross complementarity conditions - adapt h_n for switch detection
6. Step equilibration - remove degrees of freedom if no switch

$$\dot{x} \in F_{\mathrm{F}}(x,u) \quad \Longleftrightarrow \quad \begin{aligned} \dot{x} &= F(x,u)\theta \\ 0 &= G_{\mathrm{DCS}}(x,z,\theta) \end{aligned}$$



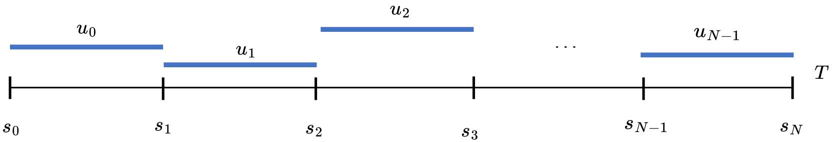
Discretize optimal control problem with FESD

Discretized optimal control problem

$$
\begin{aligned}
\min_{s,z,u}\ & \sum_{k=0}^{N-1} \Phi_L(s_k, z_k, u_k) + E(s_N) \\
\text{s.t.}\quad & s_0 = \bar{x}_0 \\
& s_{k+1} = \Phi_f(s_k, z_k, u_k) \\
& 0 = \Phi_{\mathrm{int}}(s_k, z_k, u_k) \\
& 0 \geq h(s_k, u_k),\ k = 0, \dots, N-1 \\
& 0 \geq r(s_N)
\end{aligned}
$$

- States at control grid points $s = (s_0, \dots, s_N)$
- Piecewise controls $u = (u_0, \dots, u_{N-1})$
- FESD with N_{FE} finite elements applied on every control interval

Control horizon $[0, T]$ with N control stages



Discretize optimal control problem with FESD

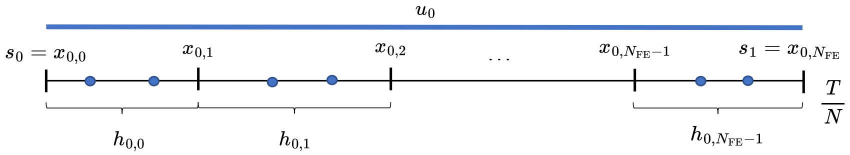
Discretized optimal control problem

$$\min_{s,z,u} \sum_{k=0}^{N-1} \Phi_L(s_k, z_k, u_k) + E(s_N)$$

$$\begin{aligned} \text{s.t.} \quad s_0 &= \bar{x}_0 \\ s_{k+1} &= \Phi_f(s_k, z_k, u_k) \\ 0 &= \Phi_{\text{int}}(s_k, z_k, u_k) \\ 0 &\geq h(s_k, u_k), \; k = 0, \ldots, N-1 \\ 0 &\geq r(s_N) \end{aligned}$$

- States at control grid points $s = (s_0, \ldots, s_N)$
- Piecewise controls $u = (u_0, \ldots, u_{N-1})$
- FESD with N_{FE} finite elements applied on every control interval
- Φ_{int} summarizes internal FESD equations: RK, cross complementarity, step equilibration,...

Control horizon $[0, T]$ with N control stages



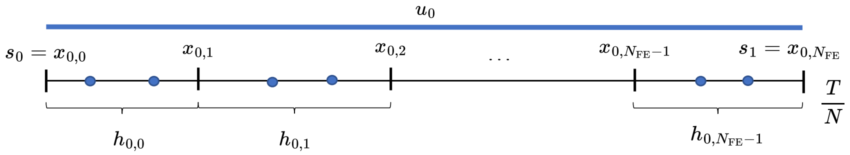
Discretize optimal control problem with FESD

Discretized optimal control problem

$$
\begin{aligned}
\min_{s,z,u} \quad & \sum_{k=0}^{N-1} \Phi_L(s_k, z_k, u_k) + E(s_N) \\
\text{s.t.} \quad & s_0 = \bar{x}_0 \\
& s_{k+1} = \Phi_f(s_k, z_k, u_k) \\
& 0 = \Phi_{\text{int}}(s_k, z_k, u_k) \\
& 0 \geq h(s_k, u_k), \; k = 0, \dots, N-1 \\
& 0 \geq r(s_N)
\end{aligned}
$$

- States at control grid points $s = (s_0, \dots, s_N)$
- Piecewise controls $u = (u_0, \dots, u_{N-1})$
- FESD with N_{FE} finite elements applied on every control interval
- Φ_{int} summarizes internal FESD equations: RK, cross complementarity, step equilibration,...
- $z = (z_0, \dots, z_{N-1})$ - all interval variables: internal states, stage values of states and algebraice, step sizes...

Control horizon $[0, T]$ with N control stages



Discretize optimal control problem with FESD lead to MPCCs

Discretized optimal control problem

$$
\begin{aligned}
\min_{s,z,u}\ & \sum_{k=0}^{N-1} \Phi_L(s_k, z_k, u_k) + E(s_N) \\
\text{s.t.}\quad & s_0 = \bar{x}_0 \\
& s_{k+1} = \Phi_f(s_k, z_k, u_k) \\
& 0 = \Phi_{\text{int}}(s_k, z_k, u_k) \\
& 0 \geq h(s_k, u_k),\ k = 0, \ldots, N-1 \\
& 0 \geq r(s_N)
\end{aligned}
$$

Collect $w = (s, z, u) \in \mathbb{R}^{n_w}$
Mathematical programs with complementarity constraints (MPCCs) are more difficult than standard NLPs

NLP with Complementarity Constraints

$$
\begin{aligned}
\min_{w \in \mathbb{R}^{n_w}}\ & F(w) \\
\text{s.t.}\quad & 0 = G(w) \\
& 0 \geq H(w) \\
& 0 \leq G_1(w) \perp G_2(w) \geq 0
\end{aligned}
$$

Standard and cross complementarity constraints summarized in

$$0 \leq G_1(w) \perp G_2(w) \geq 0$$

Three ideas:

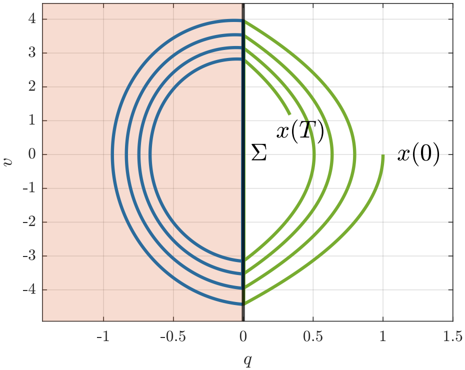
1. mimic state jump by **auxiliary dynamic system** $\dot{x} = f_{\mathrm{aux}}(x)$ on prohibited region
2. introduce a **clock state** $t(\tau)$ that stops counting when the auxiliary system is active
3. adapt speed of time, $\frac{\mathrm{d}t}{\mathrm{d}\tau} = s$ with $s \geq 1$, and **impose terminal constraint** $t(T) = T$

The time-freezing reformulation

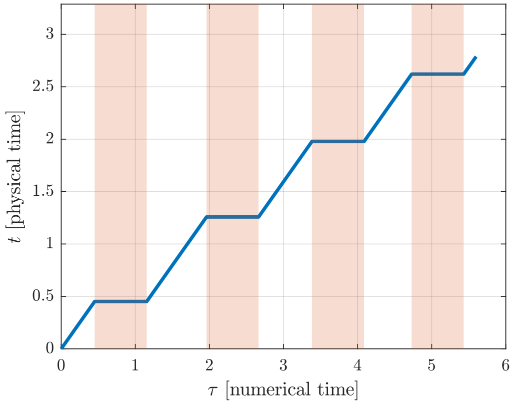
Augmented state $(x, t) \in \mathbb{R}^{n+1}$ evolves in **numerical time** τ. Augmented system is nonsmooth, of NSD2 type:

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

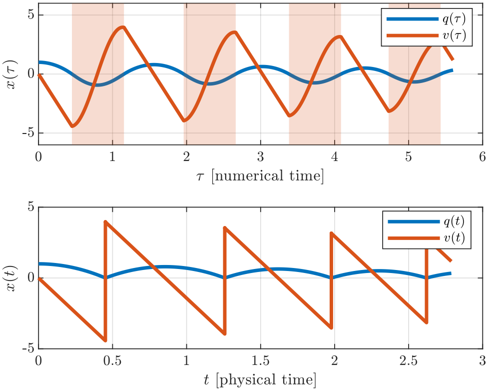
- During normal times, system and clock state evolve with adapted speed $s \geq 1$.
- Auxiliary system $\frac{\mathrm{d}x}{\mathrm{d}\tau} = f_{\mathrm{aux}}(x)$ mimics state jump while time is frozen, $\frac{\mathrm{d}t}{\mathrm{d}\tau} = 0$.



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".

Conclusions and summary

- There exist many mathematical formalisms to model switches in jumps in dynamical systems.
- Nonsmoothness leads to occurrences and difficulties not seen in nonsmooth systems.
- Filippov systems and dynamic complementarity systems have nice practical and theoretical properties.
- Naive discretization and smoothing can lead to non-obvious and severe failures.

Conclusions and summary

- There exist many mathematical formalisms to model switches in jumps in dynamical systems.
- Nonsmoothness leads to occurrences and difficulties not seen in nonsmooth systems.
- Filippov systems and dynamic complementarity systems have nice practical and theoretical properties.
- Naive discretization and smoothing can lead to non-obvious and severe failures.
- Dedicated discretization methods, such as FESD, detect the switches and lead to high accuracy.
- Switch detection not only important for accuracy, but also for correct sensitivities = avoid convergence at spurious solutions.
- Time-freezing enables enables to transform systems with jumps into systems with switches.

Conclusions and summary

- There exist many mathematical formalisms to model switches in jumps in dynamical systems.
- Nonsmoothness leads to occurrences and difficulties not seen in nonsmooth systems.
- Filippov systems and dynamic complementarity systems have nice practical and theoretical properties.
- Naive discretization and smoothing can lead to non-obvious and severe failures.
- Dedicated discretization methods, such as FESD, detect the switches and lead to high accuracy.
- Switch detection not only important for accuracy, but also for correct sensitivities = avoid convergence at spurious solutions.
- Time-freezing enables enables to transform systems with jumps into systems with switches.
- In nonsmooth optimal control one needs to solve mathematical programs with complementarity constraints (MPCC).
- Solving MPCCs can be significantly more difficult than solving smooth nonlinear programs.

Outlook - possible research directions

- High-performance and open-source nonlinear complementarity problem solvers, e.g., for FESD problems.
- Reuse FESD ideas for similar problem classes not treated in this course, e.g. project dynamical systems.
- Generalizing time-freezing to further systems classes with state jumps.
- Good implementations of MPCC methods.
- Model predictive control algorithms based on *inaccurate* solves of the nonsmooth OCPs - nonsmooth real-time iterations?
- What else?