Lecture 7: Time-Freezing for State Jumps
Part I: Elastic Impacts and Hybrid Automata

Moritz Diehl and Armin Nurkanović

Systems Control and Optimization Laboratory (syscop)

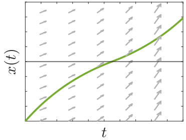
Summer School on Direct Methods for Optimal Control of Nonsmooth Systems

September 11-15, 2023

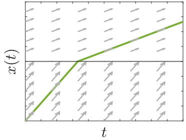
universität freiburg

Outline of the lecture

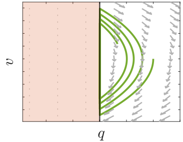
Regard an ordinary differential equation (ODE) with a **nonsmooth** right-hand side (RHS).
Distinguish three cases:



NSD1: nondifferentiable RHS, e.g., $\dot{x} = 1 + |x|$



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



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

Outline of the lecture

1 Time-freezing for mechanical systems with elastic impacts

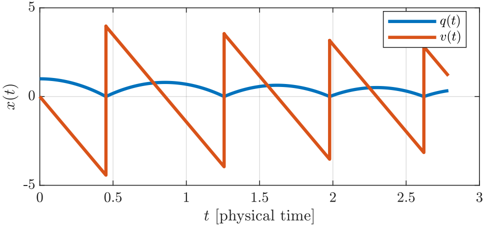
2 Time-freezing for finite automata with hysteresis

NSD3 state jump example: bouncing ball

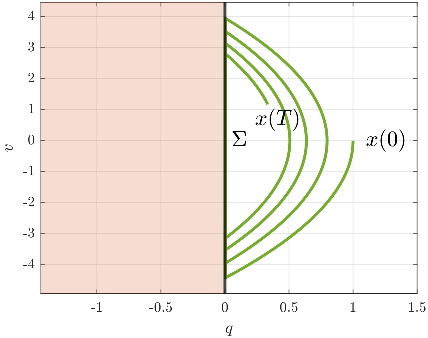
Bouncing ball with state $x = (q, v)$:

$$\dot{q} = v,\ m\dot{v} = -mg, \qquad \text{if } q > 0$$
$$v(t^+) = -0.9\, v(t^-), \qquad \text{if } q(t^-) = 0 \text{ and } v(t^-) < 0$$

Time plot of bouncing ball trajectory:



Phase plot of bouncing ball trajectory:



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

Three ideas:

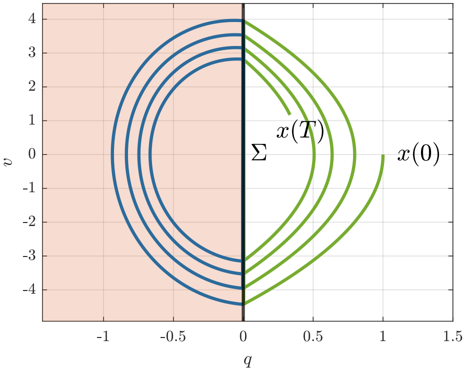
1. mimic state jump by **auxiliary dynamic system** $\dot{x} = f_{\mathrm{aux}}(x)$ on prohibited region
2. introduce a **clock state** $t(\tau)$ that stops counting when the auxiliary system is active
3. adapt speed of time, $\frac{\mathrm{d}t}{\mathrm{d}\tau} = s$ with $s \geq 1$, and **impose terminal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

The time-freezing reformulation

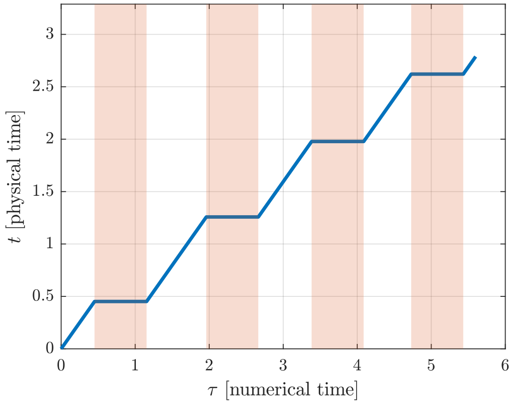
Augmented state $(x, t) \in \mathbb{R}^{n+1}$ evolves in **numerical time** τ. Augmented system is nonsmooth, of NSD2 type:

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

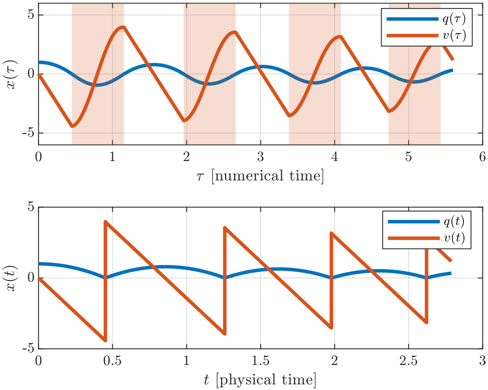
- During normal times, system and clock state evolve with adapted speed $s \geq 1$.
- Auxiliary system $\frac{\mathrm{d}x}{\mathrm{d}\tau} = f_{\mathrm{aux}}(x)$ mimics state jump while time is frozen, $\frac{\mathrm{d}t}{\mathrm{d}\tau} = 0$.



Time-freezing for bouncing ball example



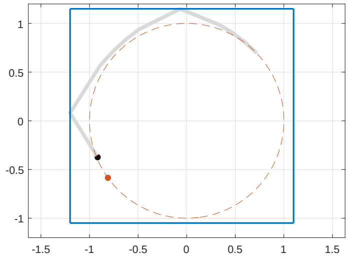
Evolution of physical time (clock state) during augmented system simulation ($s = 1$).



We can recover the true solution by plotting $x(\tau)$ vs. $t(\tau)$ and disregarding "frozen pieces".

A tracking OCP example with Time-Freezing and FESD in NOSNOC

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



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

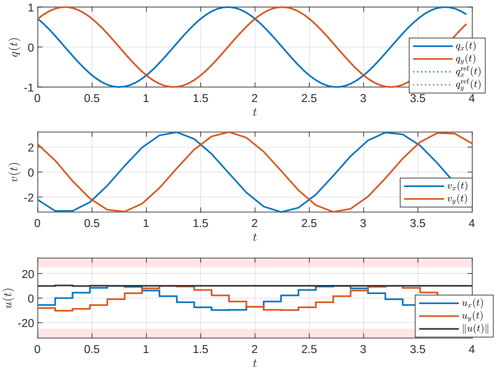
$$
\begin{aligned}
\min_{\substack{x(.),u(.),s(.),\\ \theta(.),\lambda(.),\mu(.)}} \quad & \int_0^T (q - q_{\mathrm{ref}}(\tau))^\top (q - q_{\mathrm{ref}}(\tau))\, s(\tau)\, \mathrm{d}\tau \\
\text{s.t.} \quad & x(0) = x_0, \quad t(T) = T, \\
& x'(\tau) = \sum_{i=1}^{n_f} \theta_i(\tau) f_i(x(\tau),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

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

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 $N_{\mathrm{FE}} = 3$ finite elements with Radau IIA 3 on each control interval
- each FESD interval has one constant control u and one speed of time s
- MPCC solved via ℓ_∞ penalty reformulation and homotopy
- For homotopy convergence: in total 4 NLPs solved with `IPOPT` via `CasADi`



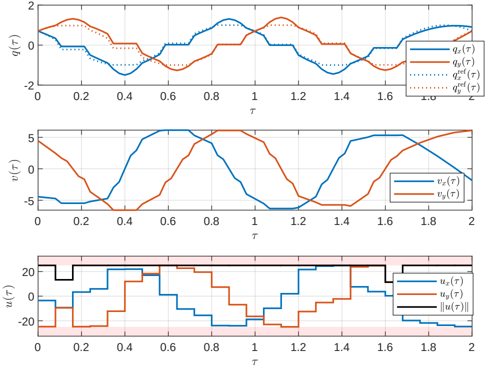
States and controls in physical time.

Results with slowly moving reference - movie

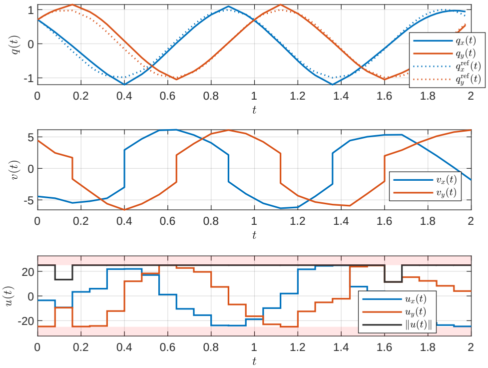
For $\omega = \pi$, tracking is easy: no jumps occur in optimal solution.

Results with fast reference

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



States and controls in numerical time.



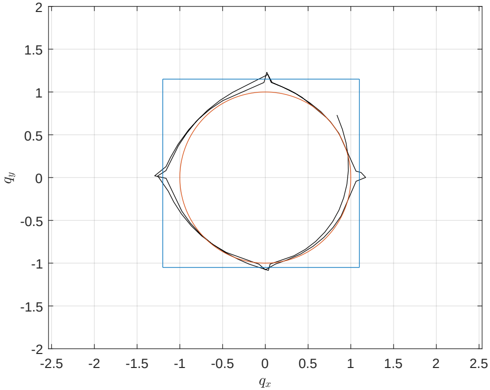
States and controls in physical time.

Results with fast reference - movie

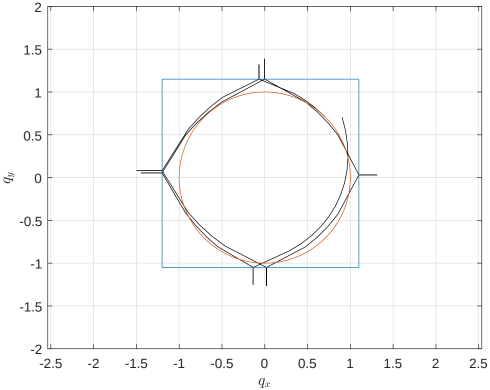
For $\omega = 2\pi$, tracking is only possible if ball bounces against walls.

Homotopy: first iteration vs converged solution

Geometric trajectory



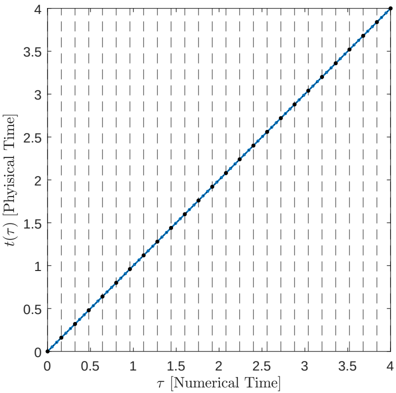
After the first homotopy iteration



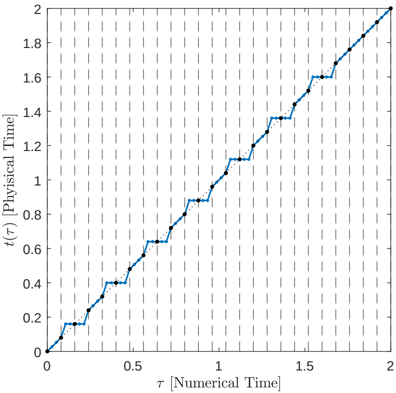
The solution trajectory after convergence

Physical vs. Numerical Time

for $\omega = \pi$



for $\omega = 2\pi$

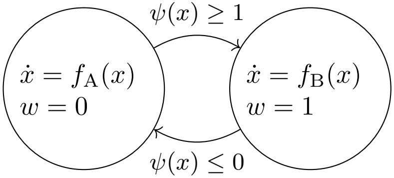


Outline of the lecture

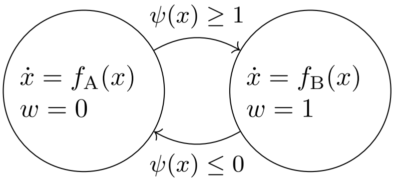
1 Time-freezing for mechanical systems with elastic impacts

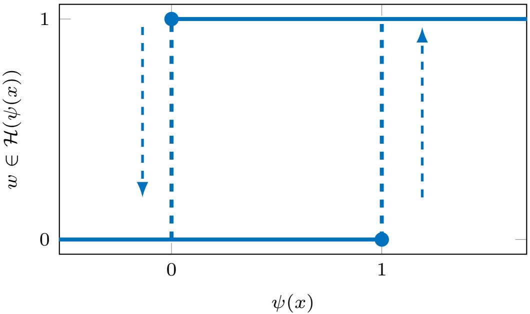
2 Time-freezing for finite automata with hysteresis

Hybrid systems and finite automaton



Hybrid systems and finite automaton

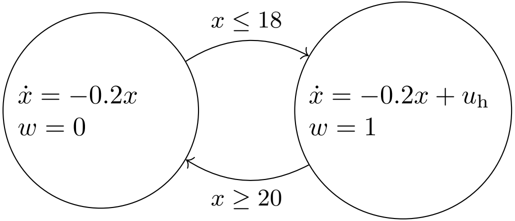




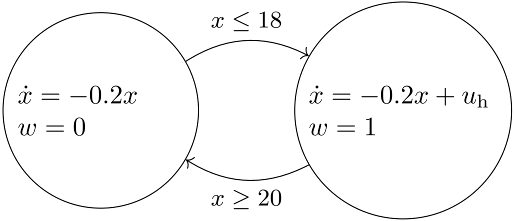
Hybrid system with hysteresis (*incomplete description*)

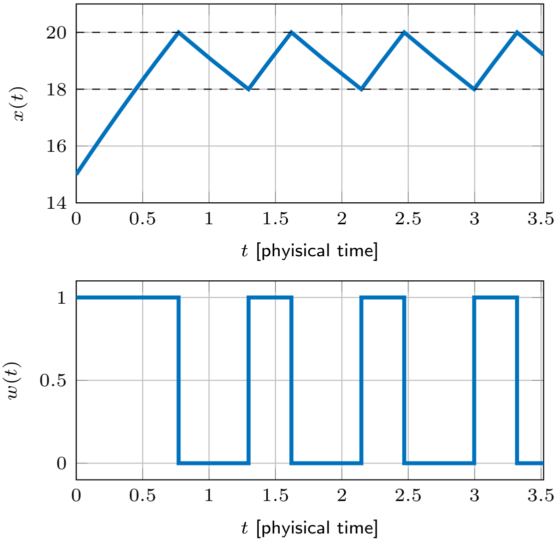
$$\dot{x} = f(x, w) = (1 - w) f_{\mathrm{A}}(x) + w f_{\mathrm{B}}(x)$$

Tutorial example: thermostat with hysteresis



Tutorial example: thermostat with hysteresis

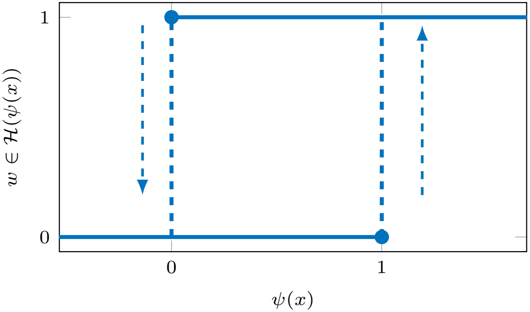




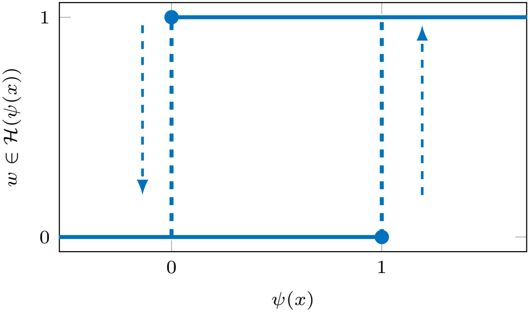
Hysteresis: a system with state jumps

Hybrid system with hysteresis

$$\dot{x} = f(x,w) = (1-w)f_{\mathrm{A}}(x) + w f_{\mathrm{B}}(x)$$
$$\dot{w} = 0$$



Hysteresis: a system with state jumps



Hybrid system with hysteresis

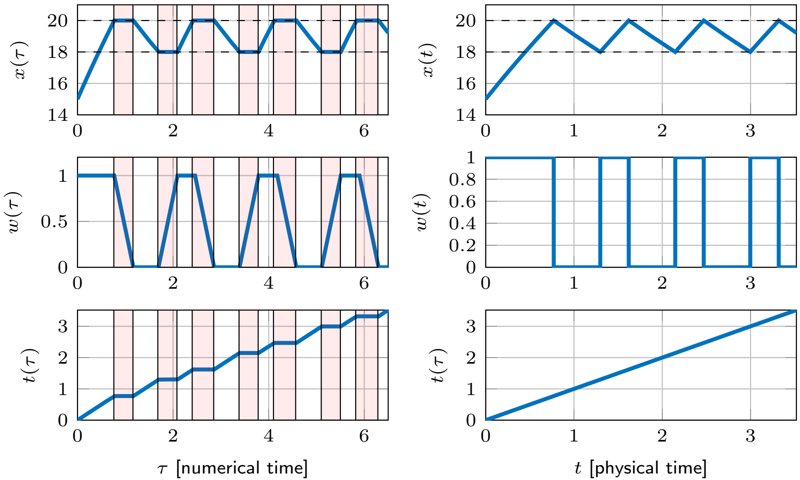
$$\dot{x} = f(x, w) = (1 - w) f_{\mathrm{A}}(x) + w f_{\mathrm{B}}(x)$$
$$\dot{w} = 0$$

The State Jump Law

1. if $w(t_{\mathrm{s}}^-) = 0$ and $\psi(x(t_{\mathrm{s}}^-)) = 1$, then $x(t_{\mathrm{s}}^+) = x(t_{\mathrm{s}}^-)$ and $w(t_{\mathrm{s}}^+) = 1$
2. if $w(t_{\mathrm{s}}^-) = 1$ and $\psi(x(t_{\mathrm{s}}^-)) = 0$, then $x(t_{\mathrm{s}}^+) = x(t_{\mathrm{s}}^-)$ and $w(t_{\mathrm{s}}^+) = 0$

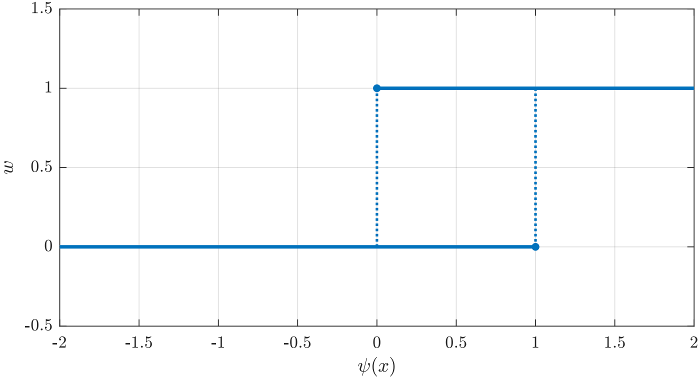
Remember: $w(t)$ is now a discontinuous differential state!

Tutorial example: thermostat and time-freezing



Time-freezing: the state space

A look at the $(\psi(x), w)-$plane



- Everything except the blue solid curve is prohibited in the $(\psi, w)-$ space
- The evolution happens in a lower-dimensional space $\implies$ *sliding mode*

Voronoi regions

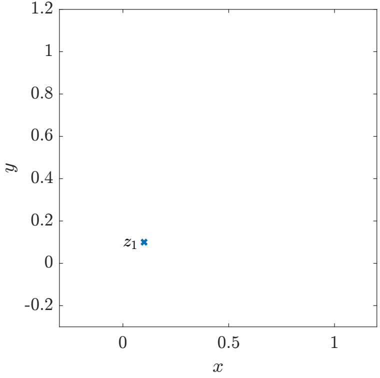
Voronoi regions/cells: each region contains a specific point z_i and all points within that region are closer to that specific point than to any other point in the space

Voronoi regions

Voronoi regions/cells: each region contains a specific point z_i and all points within that region are closer to that specific point than to any other point in the space

- Given a set of points $\mathcal{Z} = \{z_1, z_2, \dots\} \subset \mathbb{R}^n$, the regions R_i are defined as

$$R_i = \{z \in \mathbb{R}^n \mid \|z - z_i\| < \|z - z_j\|, \ \forall z_j \in \mathcal{Z}, \ j \neq i\}$$



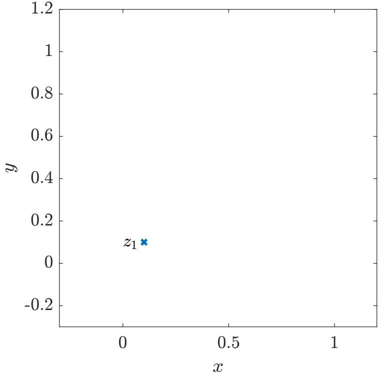
Voronoi regions

Voronoi regions/cells: each region contains a specific point z_i and all points within that region are closer to that specific point than to any other point in the space

- Given a set of points $\mathcal{Z} = \{z_1, z_2, \dots\} \subset \mathbb{R}^n$, the regions R_i are defined as

$$R_i = \{z \in \mathbb{R}^n \mid \underbrace{\|z - z_i\|}_{g_i(z)} < \underbrace{\|z - z_j\|}_{g_j(z)}, \ \forall z_j \in \mathcal{Z}, \ j \neq i\}$$

- Naturally in Stewart's form
  $R_i = \{z \mid g_i(z) < g_j(z), \forall j, \, i \neq j\}$



Voronoi regions

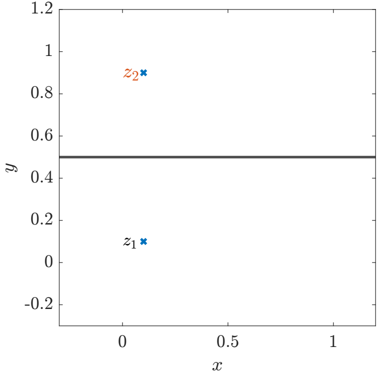
Voronoi regions/cells: each region contains a specific point z_i and all points within that region are closer to that specific point than to any other point in the space

- Given a set of points $\mathcal{Z} = \{z_1, z_2, \dots\} \subset \mathbb{R}^n$, the regions R_i are defined as

$$R_i = \{z \in \mathbb{R}^n \mid \underbrace{\|z - z_i\|}_{g_i(z)} < \underbrace{\|z - z_j\|}_{g_j(z)},\ \forall z_j \in \mathcal{Z},\ j \neq i\}$$

- Naturally in Stewart's form
  $R_i = \{z \mid g_i(z) < g_j(z), \forall j,\ i \neq j\}$
- Using the squared two norm $\implies$ **linear** inequalities:

$$R_i = \{z \in \mathbb{R}^n \mid (z_j - z_i)^\top z < \frac{1}{2}(\|z_j\|_2^2 - \|z_i\|_2^2), \\ \forall z_j \in \mathcal{Z},\ j \neq i\}$$



Voronoi regions

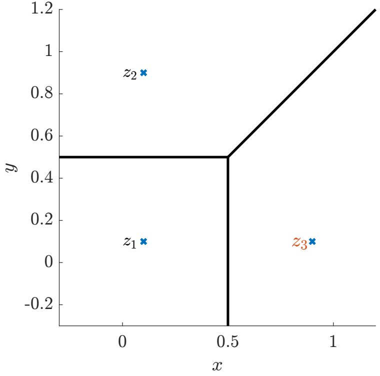
Voronoi regions/cells: each region contains a specific point z_i and all points within that region are closer to that specific point than to any other point in the space

- Given a set of points $\mathcal{Z} = \{z_1, z_2, \dots\} \subset \mathbb{R}^n$, the regions R_i are defined as

$$R_i = \{z \in \mathbb{R}^n \mid \underbrace{\|z - z_i\|}_{g_i(z)} < \underbrace{\|z - z_j\|}_{g_j(z)},\ \forall z_j \in \mathcal{Z},\ j \neq i\}$$

- Naturally in Stewart's form $R_i = \{z \mid g_i(z) < g_j(z), \forall j, i \neq j\}$
- Using the squared two norm $\implies$ **linear** inequalities:

$$R_i = \{z \in \mathbb{R}^n \mid (z_j - z_i)^\top z < \frac{1}{2}(\|z_j\|_2^2 - \|z_i\|_2^2),\ \forall z_j \in \mathcal{Z},\ j \neq i\}$$



Voronoi regions

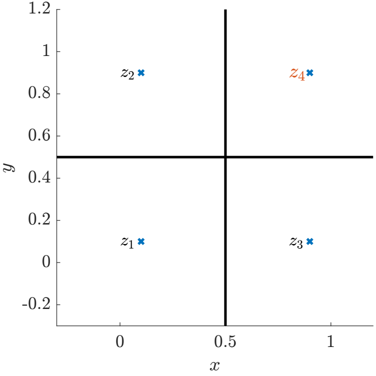
Voronoi regions/cells: each region contains a specific point z_i and all points within that region are closer to that specific point than to any other point in the space

- Given a set of points $\mathcal{Z} = \{z_1, z_2, \ldots\} \subset \mathbb{R}^n$, the regions R_i are defined as

$$R_i = \{z \in \mathbb{R}^n \mid \underbrace{\|z - z_i\|}_{g_i(z)} < \underbrace{\|z - z_j\|}_{g_j(z)},\ \forall z_j \in \mathcal{Z},\ j \neq i\}$$

- Naturally in Stewart's form
  $R_i = \{z \mid g_i(z) < g_j(z), \forall j, i \neq j\}$
- Using the squared two norm $\implies$ **linear** inequalities:

$$R_i = \{z \in \mathbb{R}^n \mid (z_j - z_i)^\top z < \frac{1}{2}(\|z_j\|_2^2 - \|z_i\|_2^2), \quad \forall z_j \in \mathcal{Z},\ j \neq i\}$$



Voronoi regions

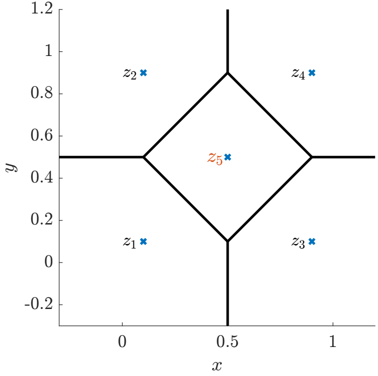
Voronoi regions/cells: each region contains a specific point z_i and all points within that region are closer to that specific point than to any other point in the space

- Given a set of points $\mathcal{Z} = \{z_1, z_2, \dots\} \subset \mathbb{R}^n$, the regions R_i are defined as

$$R_i = \{z \in \mathbb{R}^n \mid \underbrace{\|z - z_i\|}_{g_i(z)} < \underbrace{\|z - z_j\|}_{g_j(z)},\ \forall z_j \in \mathcal{Z},\ j \neq i\}$$

- Naturally in Stewart's form
$R_i = \{z \mid g_i(z) < g_j(z), \forall j, i \neq j\}$
- Using the squared two norm $\implies$ **linear** inequalities:

$$R_i = \{z \in \mathbb{R}^n \mid (z_j - z_i)^\top z < \frac{1}{2}(\|z_j\|_2^2 - \|z_i\|_2^2), \\ \forall z_j \in \mathcal{Z},\ j \neq i\}$$



Voronoi regions

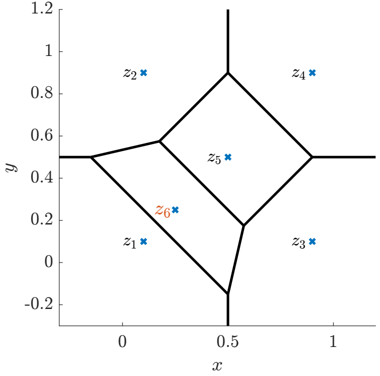
Voronoi regions/cells: each region contains a specific point z_i and all points within that region are closer to that specific point than to any other point in the space

- Given a set of points $\mathcal{Z} = \{z_1, z_2, \ldots\} \subset \mathbb{R}^n$, the regions R_i are defined as

$$R_i = \{z \in \mathbb{R}^n \mid \underbrace{\|z - z_i\|}_{g_i(z)} < \underbrace{\|z - z_j\|}_{g_j(z)},\ \forall z_j \in \mathcal{Z},\ j \neq i\}$$

- Naturally in Stewart's form
$R_i = \{z \mid g_i(z) < g_j(z), \forall j, i \neq j\}$
- Using the squared two norm $\implies$ **linear** inequalities:

$$R_i = \{z \in \mathbb{R}^n \mid (z_j - z_i)^\top z < \frac{1}{2}(\|z_j\|_2^2 - \|z_i\|_2^2),\ \forall z_j \in \mathcal{Z},\ j \neq i\}$$



Voronoi regions

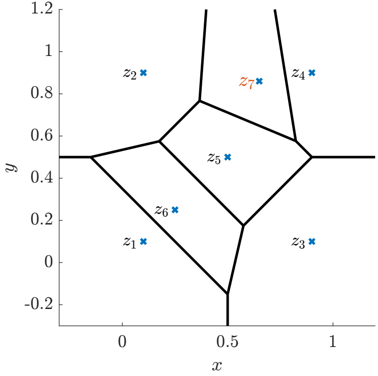
Voronoi regions/cells: each region contains a specific point z_i and all points within that region are closer to that specific point than to any other point in the space

- Given a set of points $\mathcal{Z} = \{z_1, z_2, \dots\} \subset \mathbb{R}^n$, the regions R_i are defined as

$$R_i = \{z \in \mathbb{R}^n \mid \underbrace{\|z - z_i\|}_{g_i(z)} < \underbrace{\|z - z_j\|}_{g_j(z)},\ \forall z_j \in \mathcal{Z},\ j \neq i\}$$

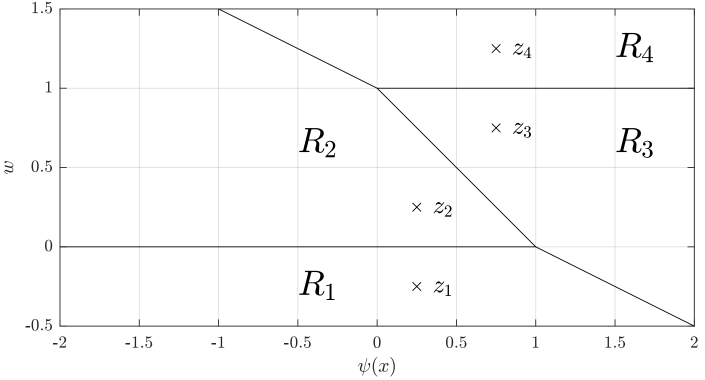
- Naturally in Stewart's form $R_i = \{z \mid g_i(z) < g_j(z), \forall j,\, i \neq j\}$
- Using the squared two norm $\implies$ **linear** inequalities:

$$R_i = \{z \in \mathbb{R}^n \mid (z_j - z_i)^\top z < \frac{1}{2}(\|z_j\|_2^2 - \|z_i\|_2^2),\ \forall z_j \in \mathcal{Z},\ j \neq i\}$$



Time-freezing: partitioning of the space

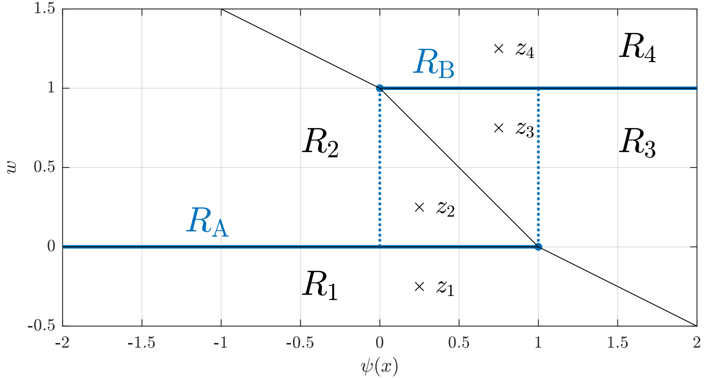
An efficient partition leads to less variables in FESD



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

Time-freezing: partitioning of the space

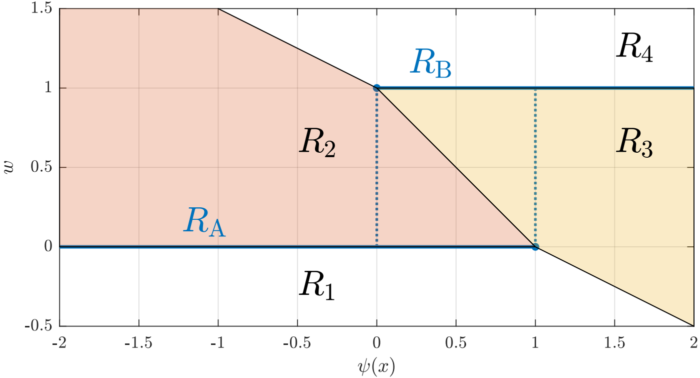
An efficient partition leads to less variables in FESD



- Partition the state space into *Voronoi regions*:
  $R_i = \{z \mid \|z - z_i\|^2 < \|z - z_j\|^2,\ j = 1, \dots, 4, j \neq i\},\ z = (\psi(x), w)$
- Feasible region for initial *hybrid system with hysteresis* on the region boundaries

Time-freezing: auxiliary dynamics

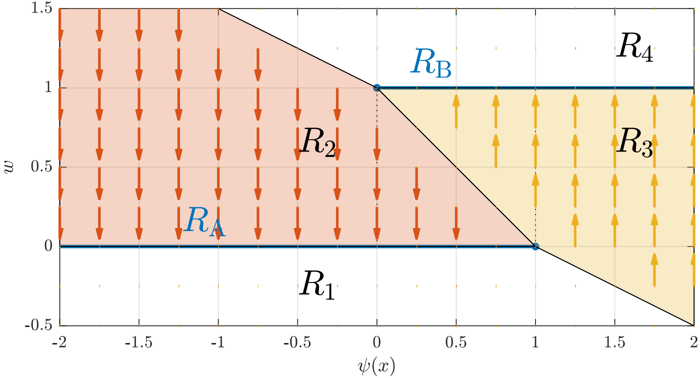
To mimic state jumps in finite numerical time



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

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 $\psi \in \{0, 1\}$

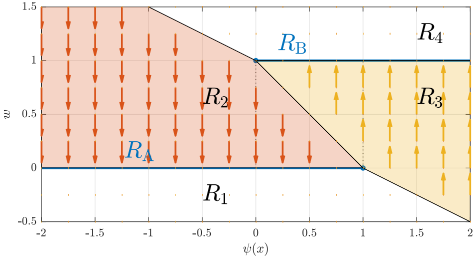
Time-freezing: auxiliary dynamics

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

Auxiliary dynamics

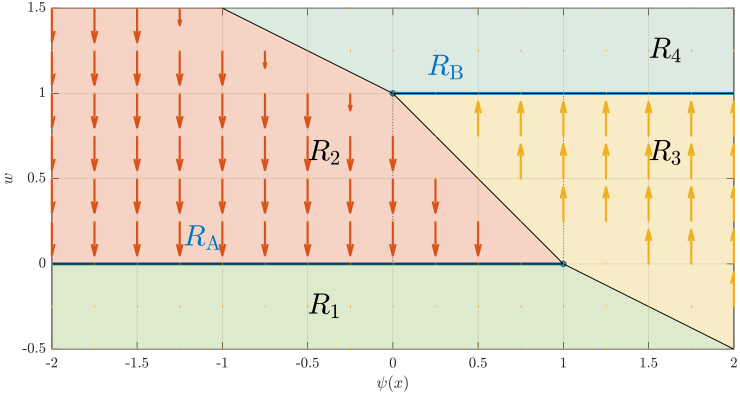
$$f_2(y) = \begin{bmatrix} 0 \\ -a \\ 0 \end{bmatrix}, \quad f_3(y) = \begin{bmatrix} 0 \\ a \\ 0 \end{bmatrix}$$

$a > 0$



Time-freezing: DAE forming dynamics

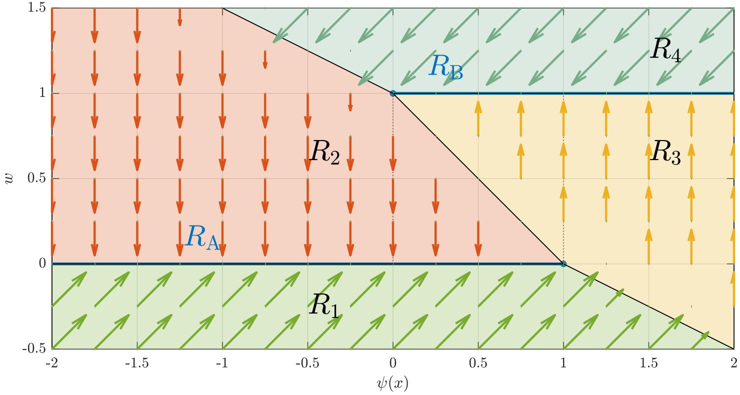
Stop the state jump and construct suitable sliding mode



- 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 $R_\mathrm{A} := \partial R_1 \cap \partial R_2$ and $R_\mathrm{B} := \partial R_3 \cap \partial R_4$ match $f_\mathrm{A}(y)$ and $f_\mathrm{B}(y)$, resp.

Time-freezing: summary

DAE-forming dynamics

$$y = (x, w, t)$$

$$\frac{\mathrm{d}y}{\mathrm{d}\tau} = f_1(y) = \begin{bmatrix} 2f_{\mathrm{A}}(x) \\ a \\ 2 \end{bmatrix}$$

$$\frac{\mathrm{d}y}{\mathrm{d}\tau} = f_4(y) = \begin{bmatrix} 2f_{\mathrm{B}}(x) \\ -a \\ 2 \end{bmatrix}$$

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

Time-freezing: summary

DAE-forming dynamics

$$y = (x, w, t)$$

$$\frac{\mathrm{d}y}{\mathrm{d}\tau} = f_1(y) = \begin{bmatrix} 2f_{\mathrm{A}}(x) \\ a \\ 2 \end{bmatrix}$$

$$\frac{\mathrm{d}y}{\mathrm{d}\tau} = f_4(y) = \begin{bmatrix} 2f_{\mathrm{B}}(x) \\ -a \\ 2 \end{bmatrix}$$

- 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 to mimic state jump

Time-freezing: summary

DAE-forming dynamics

$$y = (x, w, t)$$

$$\frac{\mathrm{d}y}{\mathrm{d}\tau} = f_1(y) = \begin{bmatrix} 2f_{\mathrm{A}}(x) \\ a \\ 2 \end{bmatrix}$$

$$\frac{\mathrm{d}y}{\mathrm{d}\tau} = f_4(y) = \begin{bmatrix} 2f_{\mathrm{B}}(x) \\ -a \\ 2 \end{bmatrix}$$

- 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 to mimic state jump
- Regions R_1 and R_4 equipped with DAE-forming dynamics to recover original dynamics in sliding mode

Time-freezing: summary

DAE-forming dynamics

$$y = (x, w, t)$$

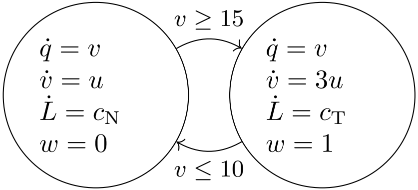
$$\frac{\mathrm{d}y}{\mathrm{d}\tau} = f_1(y) = \begin{bmatrix} 2f_{\mathrm{A}}(x) \\ a \\ 2 \end{bmatrix}$$

$$\frac{\mathrm{d}y}{\mathrm{d}\tau} = f_4(y) = \begin{bmatrix} 2f_{\mathrm{B}}(x) \\ -a \\ 2 \end{bmatrix}$$

- 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 to mimic state jump
- Regions R_1 and R_4 equipped with DAE-forming dynamics to recover original dynamics in sliding mode
- E.g., $w' = 0 \implies \theta_1 f_1(y) + \theta_2 f_2(y) = f_{\mathrm{A}}(y)$ (sliding mode)
- Conclusion: we have a PSS and can treat it with FESD

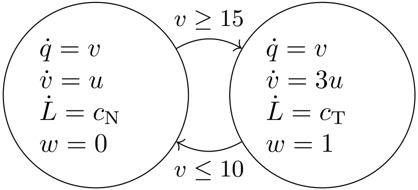
Time optimal control of a car with a turbo accelerator

Example from [Avraam, 2000] solved with NOSNOC



Time optimal control of a car with a turbo accelerator

Example from [Avraam, 2000] solved with NOSNOC

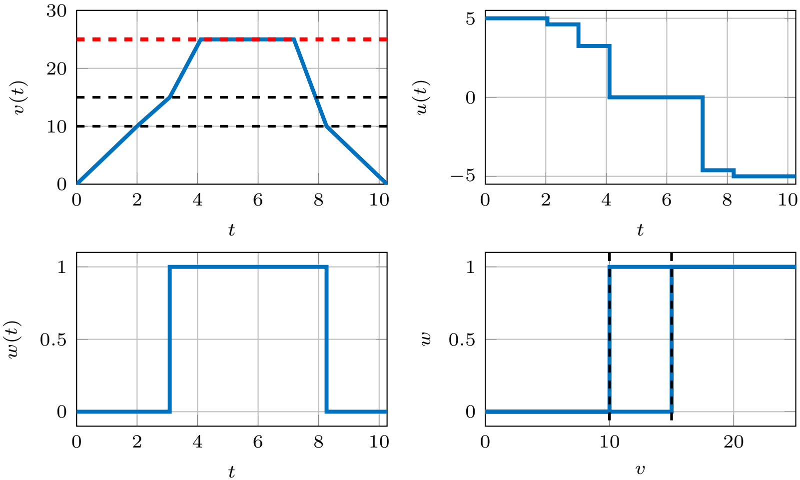


Time optimal control problem

$$
\begin{aligned}
\min_{y(\cdot),u(\cdot),s(\cdot)} \quad & t(\tau_{\mathrm{f}}) + L(\tau_{\mathrm{f}}) \\
\text{s.t.} \quad & y(0) = (z_0, 0) \\
& y'(\tau) \in s(\tau) F_{\mathrm{TF}}(y(\tau), u(\tau)) \\
& -\bar{u} \leq u(\tau) \leq \bar{u} \\
& \bar{s}^{-1} \leq s(\tau) \leq \bar{s} \\
& -\bar{v} \leq v(\tau) \leq \bar{v}, \ \tau \in [0, \tau_{\mathrm{f}}] \\
& (q(\tau_{\mathrm{f}}), v(\tau_{\mathrm{f}})) = (q_{\mathrm{f}}, v_{\mathrm{f}})
\end{aligned}
$$

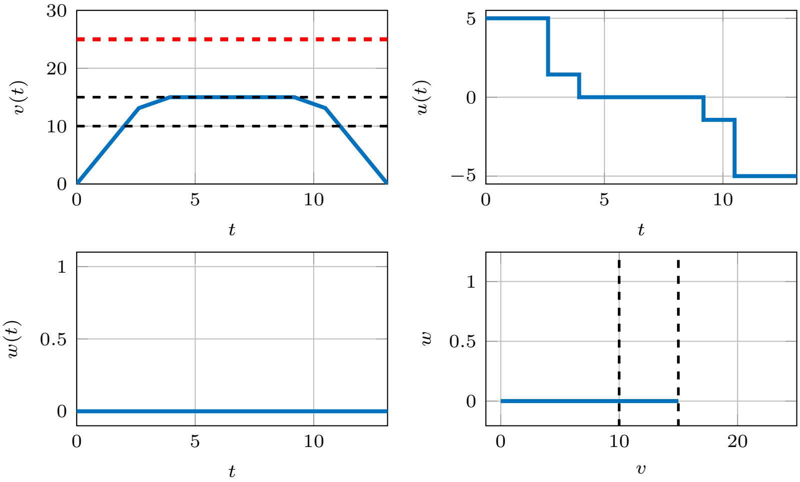
Scenario 1: turbo and nominal cost the same

$c_{\mathrm{N}} = c_{\mathrm{T}}$



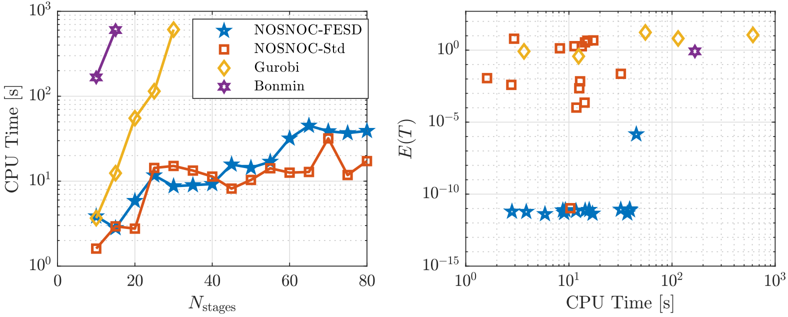
Scenario 2: Turbo is Expensive

$c_{\mathrm{N}} < c_{\mathrm{T}}$



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 infeasibility happens with grid search in T
- MILP/MINLP and NOSNOC-Std no switch detection $=$ low accuracy

Conclusions

- Time-freezing allows us to transform systems with state jumps of level NSD3 to the easier level NSD2
- Finding auxiliary dynamics is in practice often easy

Conclusions

- Time-freezing allows us to transform systems with state jumps of level NSD3 to the easier level NSD2
- Finding auxiliary dynamics is in practice often easy
- Treat systems with state jumps as Filippov systems - provides a unified theoretical and numerical treatment for many NSD2 and NSD3 systems
- Finite Elements with Switch Detection (FESD) allow highly accurate simulation and optimal control for switched systems of level NSD2