Stability analysis of nonsmooth systems for control

Riccardo Bertollo

(Eindhoven University of Technology)

UCLouvain – January 21, 2025

TU/e EINDHOVEN UNIVERSITY OF TECHNOLOGY

UCLouvain

About me

About me

MSc – Mechatronics Engineering
(2017 - 2019)

About me

MSc – Mechatronics Engineering
(2017 - 2019)

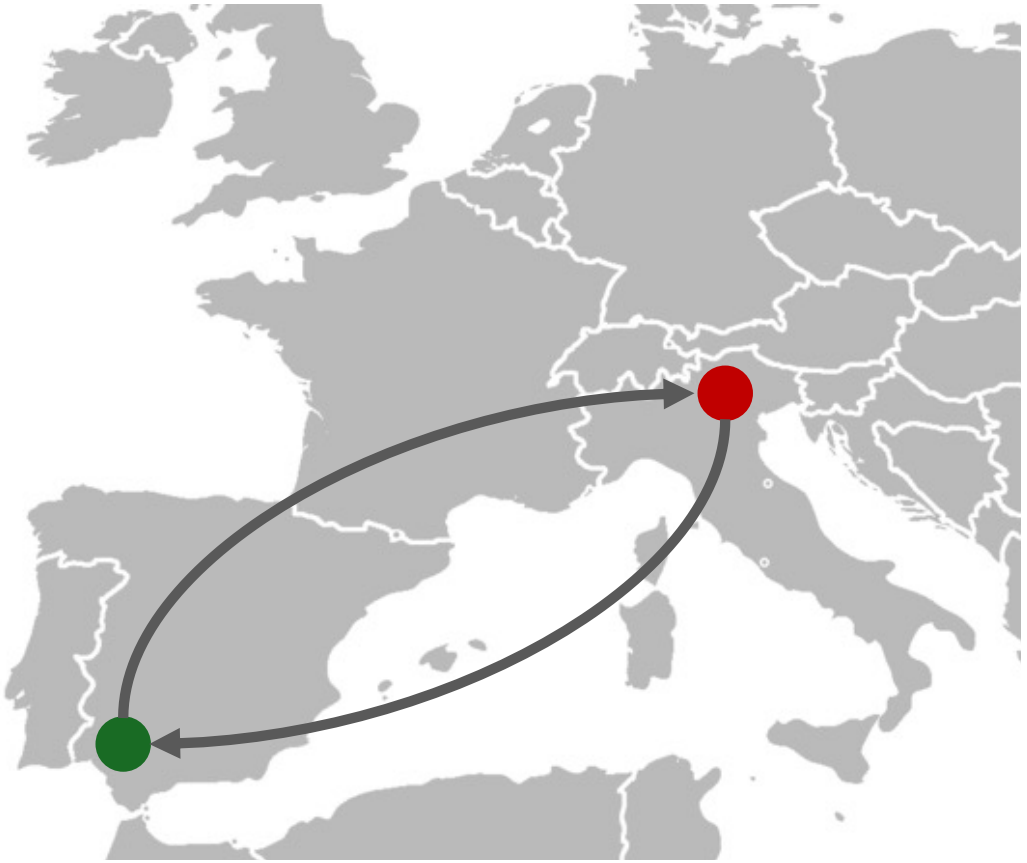
PhD – Mechatronics Engineering
(11/2019 – 03/2023)

About me

MSc – Mechatronics Engineering
(2017 - 2019)

PhD – Mechatronics Engineering
(11/2019 – 03/2023)

Visiting PhD student
(10/2021 – 03/2022)



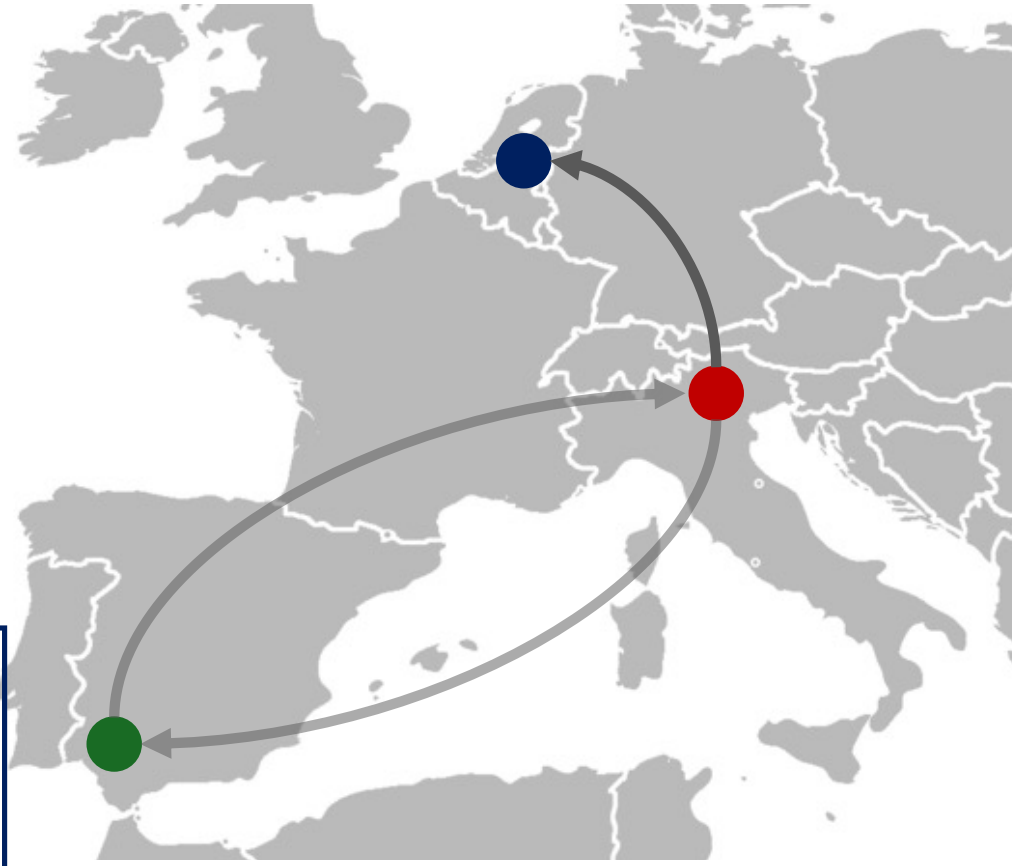
About me

MSc – Mechatronics Engineering
(2017 - 2019)

PhD – Mechatronics Engineering
(11/2019 – 03/2023)

Visiting PhD student
(10/2021 – 03/2022)

Postdoc – CST group @ ME
(03/2023 – present)



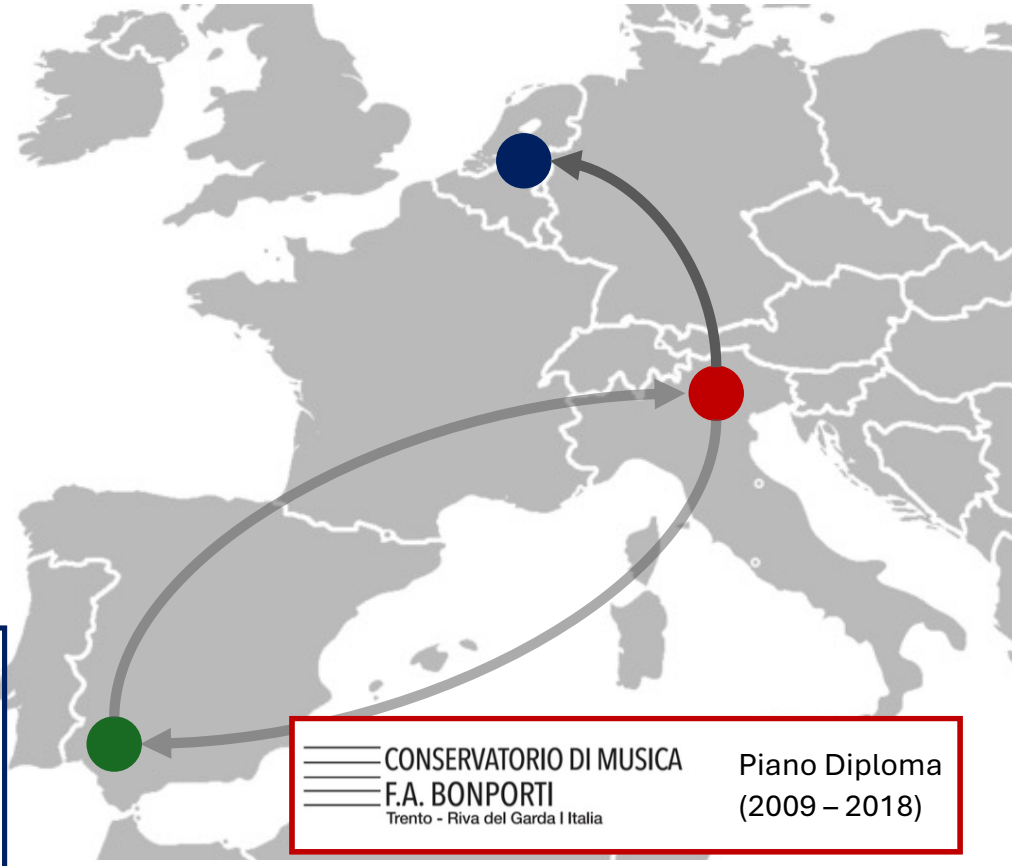
About me

MSc – Mechatronics Engineering
(2017 - 2019)

PhD – Mechatronics Engineering
(11/2019 – 03/2023)

Visiting PhD student
(10/2021 – 03/2022)

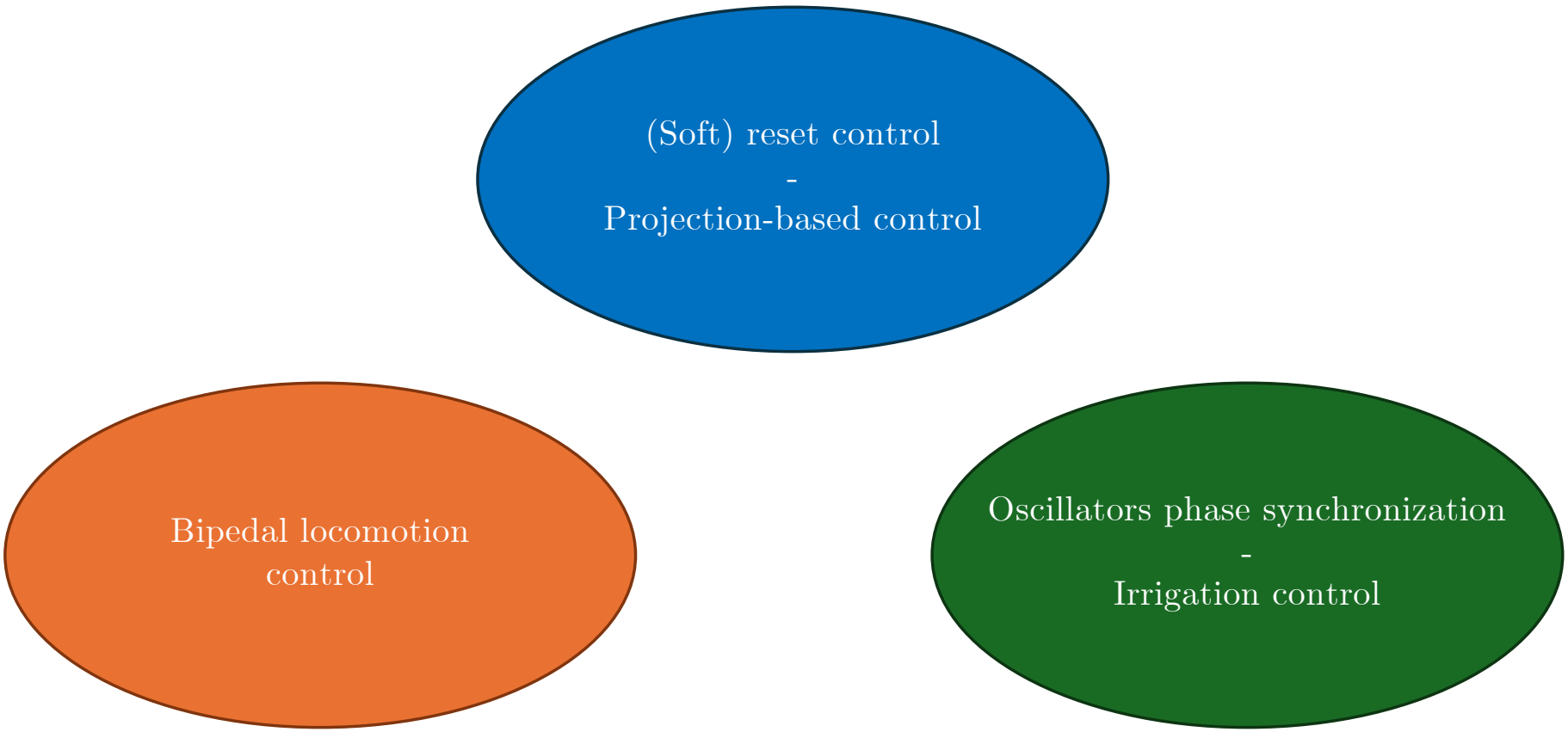
TU/e
Postdoc – CST group @ ME
(03/2023 – present)



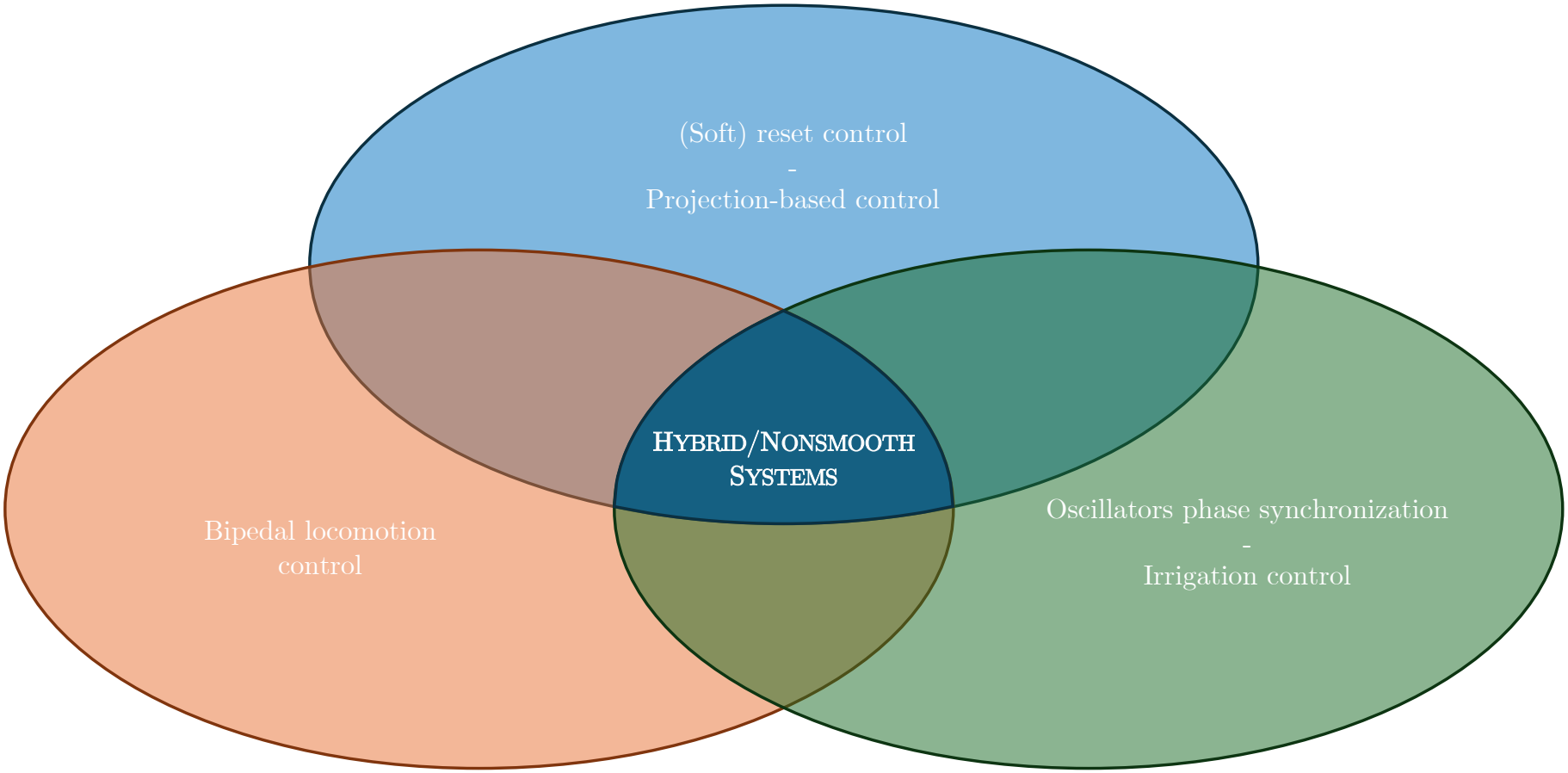
CONSERVATORIO DI MUSICA
F.A. BONPORTI
Trento - Riva del Garda | Italia

Piano Diploma
(2009 – 2018)

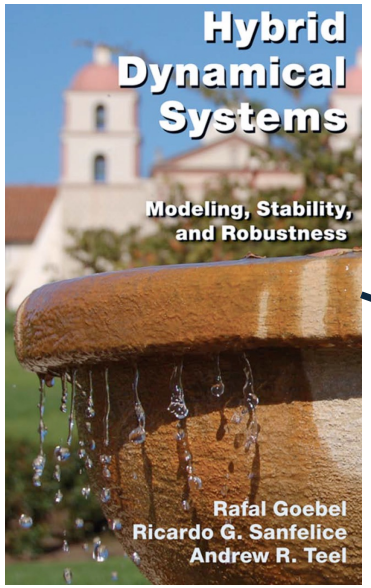
My research so far



My research so far

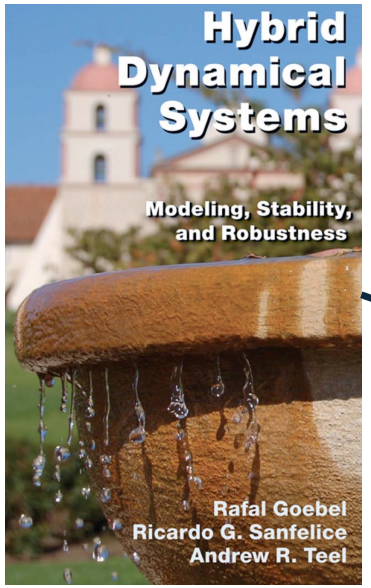


Hybrid dynamical systems



$$\begin{cases} \dot{x} \in F(x,u), & (x,u) \in C \\ x^+ \in G(x,u), & (x,u) \in D \end{cases}$$

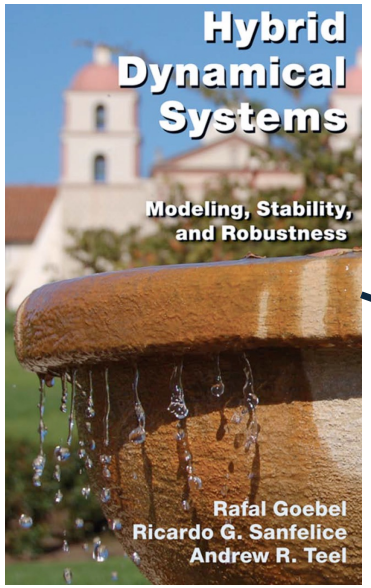
Hybrid dynamical systems

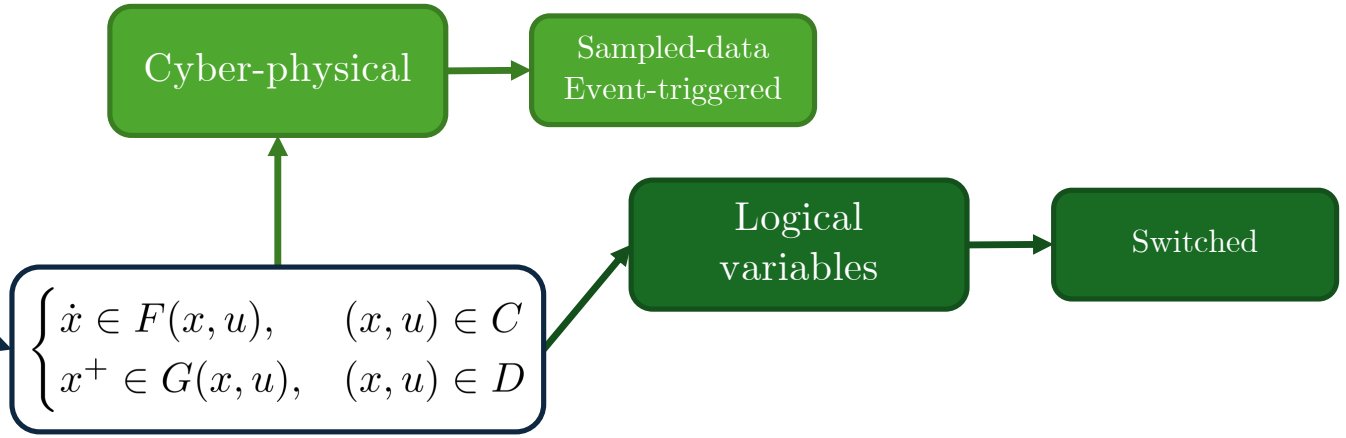


Cyber-physical → Sampled-data
Event-triggered

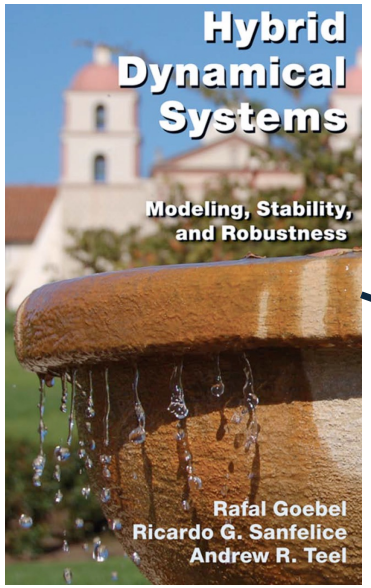
$$\begin{cases} \dot{x} \in F(x,u), & (x,u) \in C \\ x^+ \in G(x,u), & (x,u) \in D \end{cases}$$

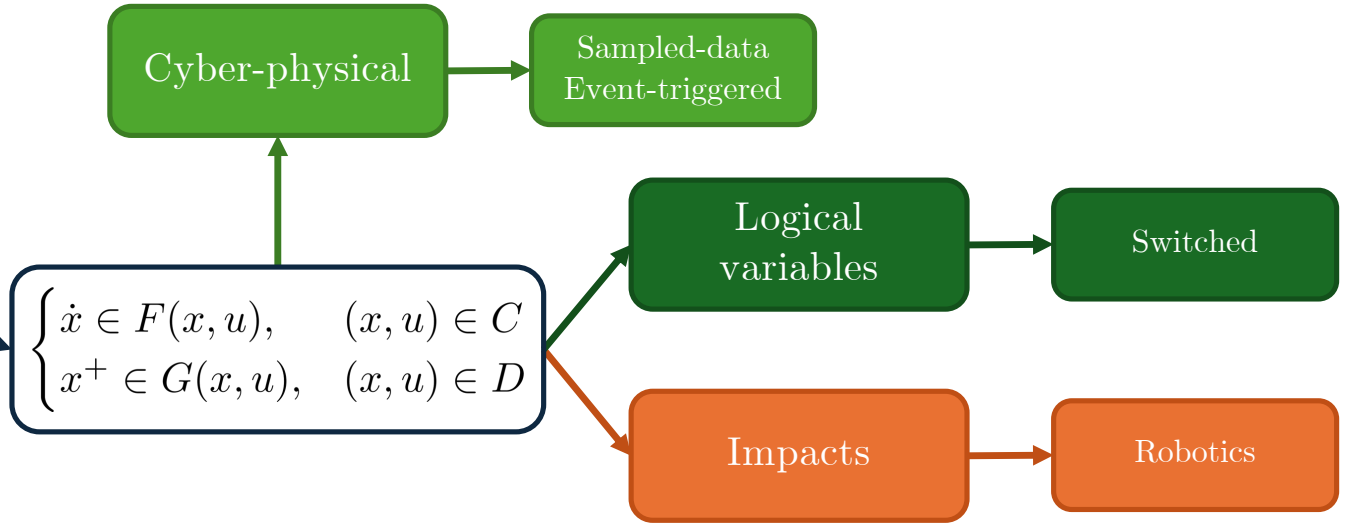
Hybrid dynamical systems



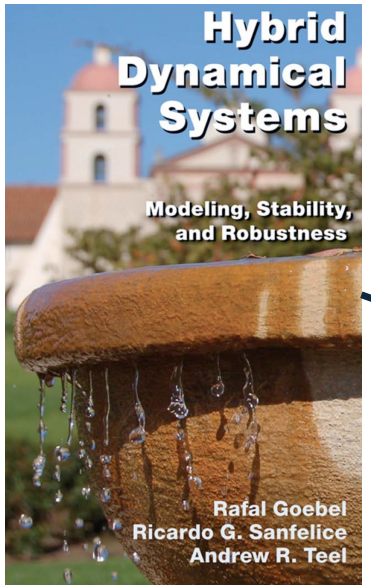


Hybrid dynamical systems





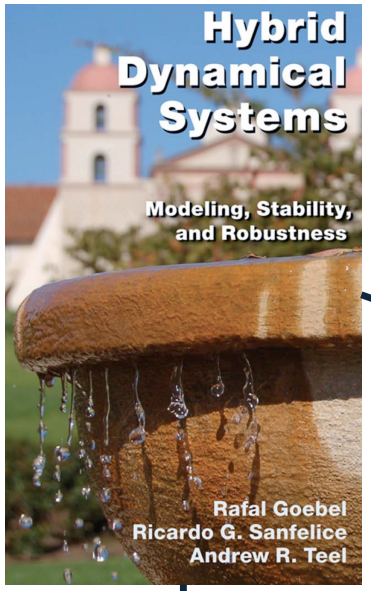
Hybrid dynamical systems



$$\begin{cases} \dot{x} \in F(x,u), & (x,u) \in C \\ x^+ \in G(x,u), & (x,u) \in D \end{cases}$$

- Cyber-physical
  - Sampled-data
  - Event-triggered
- Logical variables
  - Switched
- Impacts
  - Robotics
- Hybrid controller
  - Resets
  - Projections

Hybrid dynamical systems



$$\begin{cases} \dot{x} \in F(x,u), & (x,u) \in C \\ x^+ \in G(x,u), & (x,u) \in D \end{cases}$$

- Cyber-physical → Sampled-data, Event-triggered
- Logical variables → Switched
- Impacts → Robotics
- Hybrid controller → Resets, Projections
- Lyapunov techniques

First Order Reset Elements

Inspired by the Clegg integrator [1958]

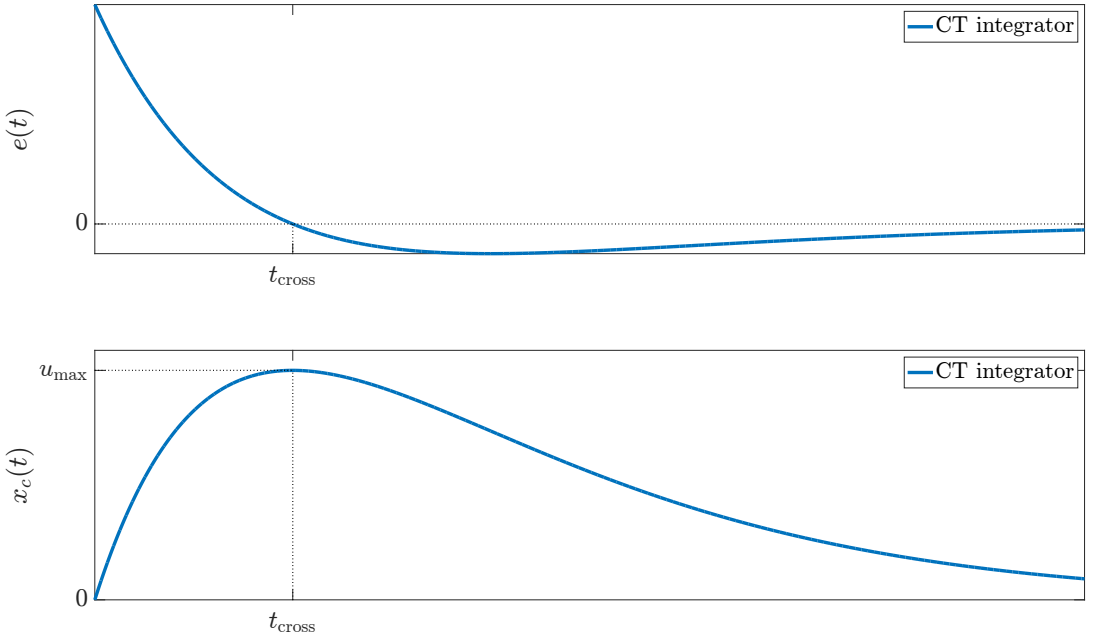
$$\begin{cases} \dot{x}_c = e, & \text{“normally”} \\ x_c^+ = 0, & \text{“at the right moment”} \end{cases}$$

L. Zaccarian, D. Nesic and A. R. Teel. “First Order Reset Elements and the Clegg integrator revisited”. *Proceedings of the 2005, American Control Conference.* 2005.

First Order Reset Elements

Inspired by the Clegg integrator [1958]

$$\begin{cases} \dot{x}_c = e, & \text{“normally”} \\ x_c^+ = 0, & \text{“at the right moment”} \end{cases}$$

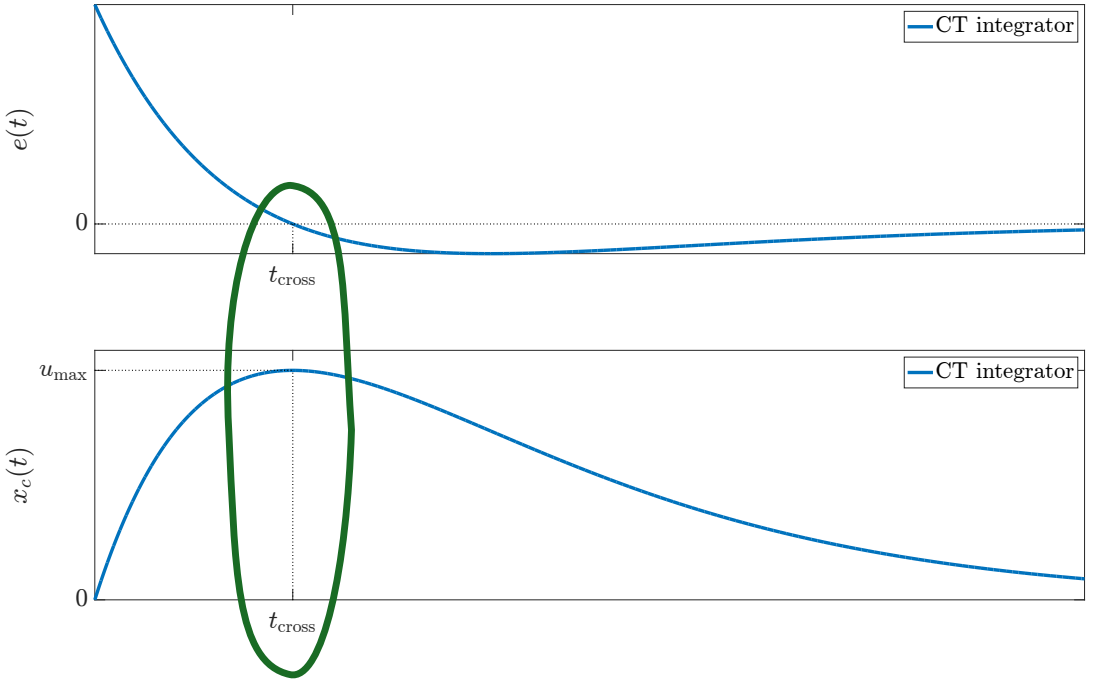


L. Zaccarian, D. Nesic and A. R. Teel. “First Order Reset Elements and the Clegg integrator revisited”. *Proceedings of the 2005, American Control Conference.* 2005.

First Order Reset Elements

Inspired by the Clegg integrator [1958]

$$\begin{cases} \dot{x}_c = e, & \text{“normally”} \\ x_c^+ = 0, & \text{“at the right moment”} \end{cases}$$



L. Zaccarian, D. Nesic and A. R. Teel. “First Order Reset Elements and the Clegg integrator revisited”. *Proceedings of the 2005, American Control Conference.* 2005.

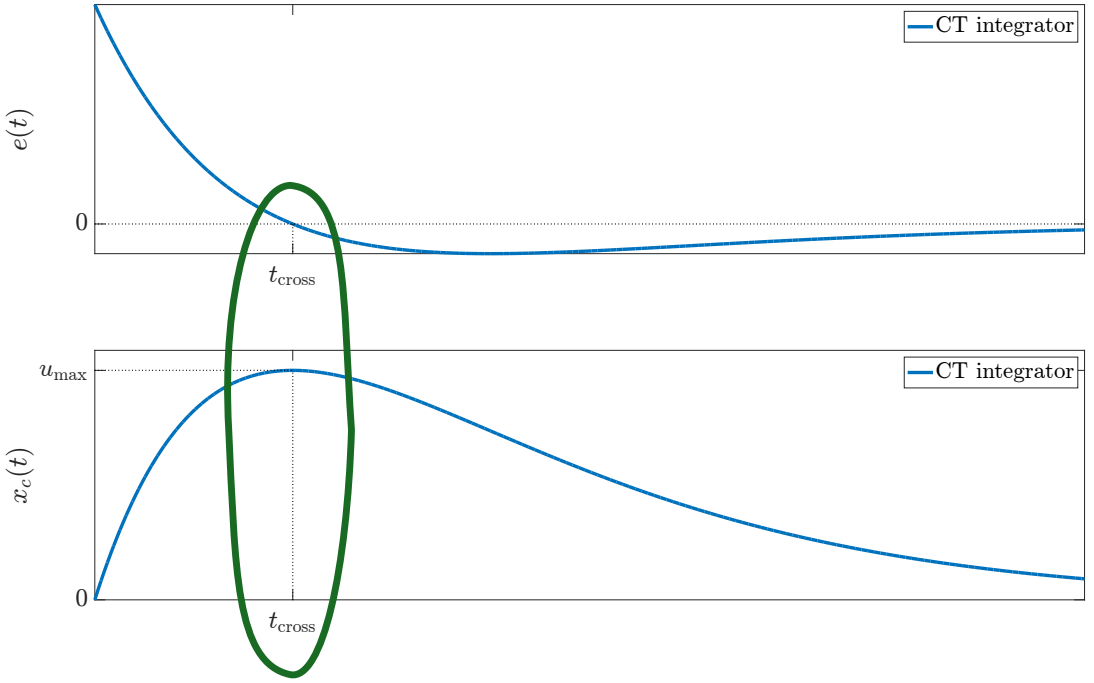
First Order Reset Elements

Inspired by the Clegg integrator [1958]

$$\begin{cases} \dot{x}_c = e, & \text{“normally”} \\ x_c^+ = 0, & \text{“at the right moment”} \end{cases}$$

Clegg integrator dynamics:

$$\begin{cases} \dot{x}_c = e, & x_c e \geq 0 \\ x_c^+ = 0, & x_c e \leq 0 \end{cases}$$



L. Zaccarian, D. Nesic and A. R. Teel. “First Order Reset Elements and the Clegg integrator revisited”. *Proceedings of the 2005, American Control Conference.* 2005.

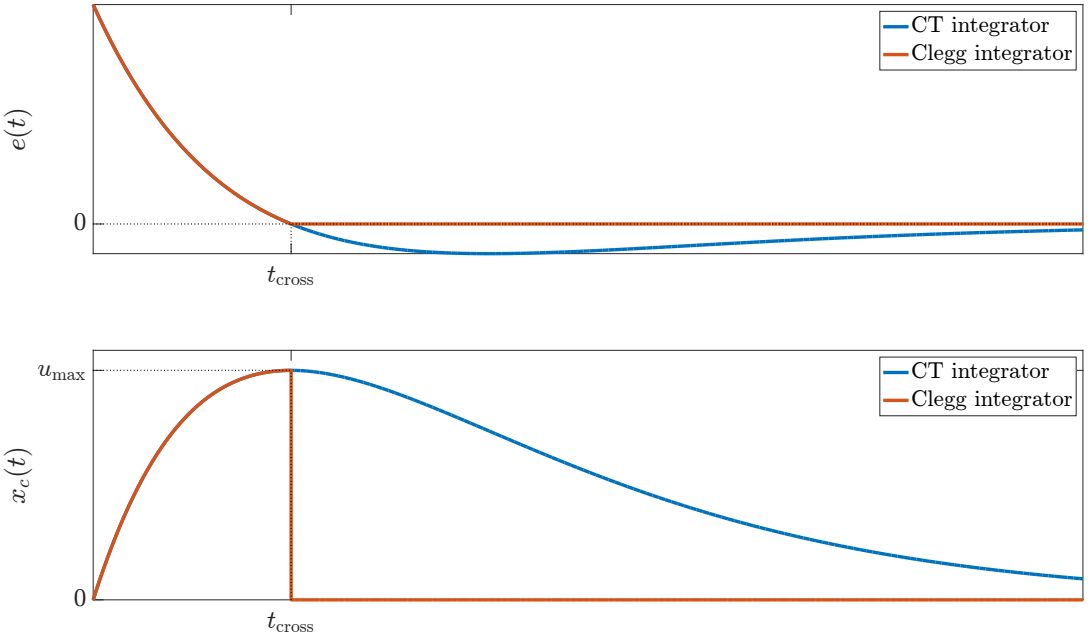
First Order Reset Elements

Inspired by the Clegg integrator [1958]

$$\begin{cases} \dot{x}_c = e, & \text{“normally”} \\ x_c^+ = 0, & \text{“at the right moment”} \end{cases}$$

Clegg integrator dynamics:

$$\begin{cases} \dot{x}_c = e, & x_c e \geq 0 \\ x_c^+ = 0, & x_c e \leq 0 \end{cases}$$



L. Zaccarian, D. Nesic and A. R. Teel. “First Order Reset Elements and the Clegg integrator revisited”. *Proceedings of the 2005, American Control Conference.* 2005.

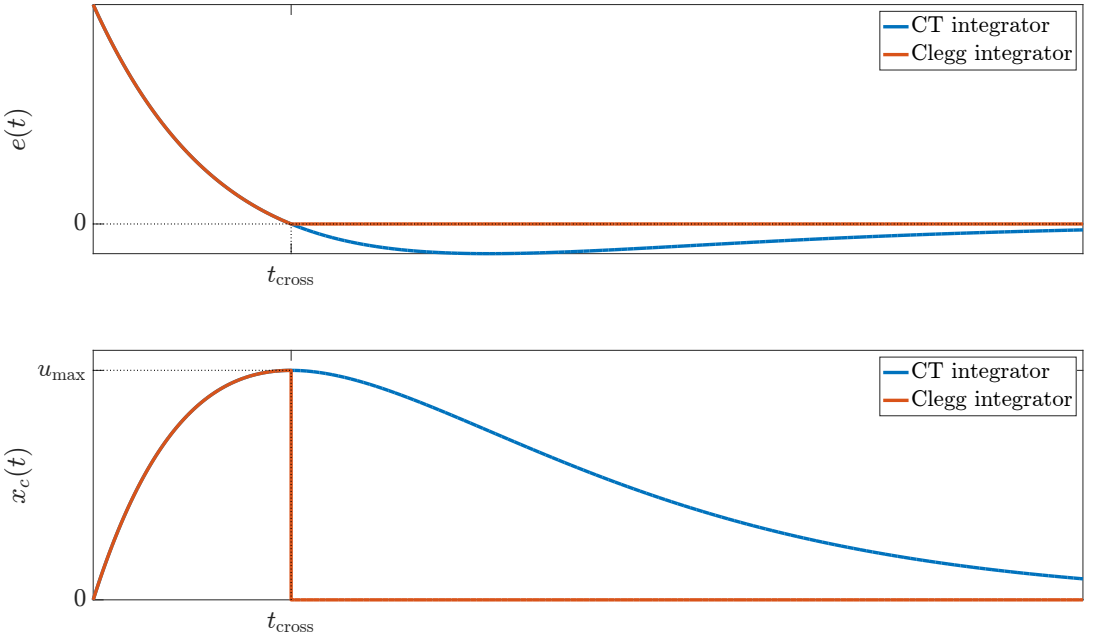
First Order Reset Elements

Inspired by the Clegg integrator [1958]

$$\begin{cases} \dot{x}_c = e, & \text{“normally”} \\ x_c^+ = 0, & \text{“at the right moment”} \end{cases}$$

FORE dynamics:

$$\begin{cases} \dot{x}_c = a_c x_c + b_c e, & x_c e \geq 0 \\ x_c^+ = 0, & x_c e \leq 0 \end{cases}$$



L. Zaccarian, D. Nesic and A. R. Teel. “First Order Reset Elements and the Clegg integrator revisited”. *Proceedings of the 2005, American Control Conference.* 2005.

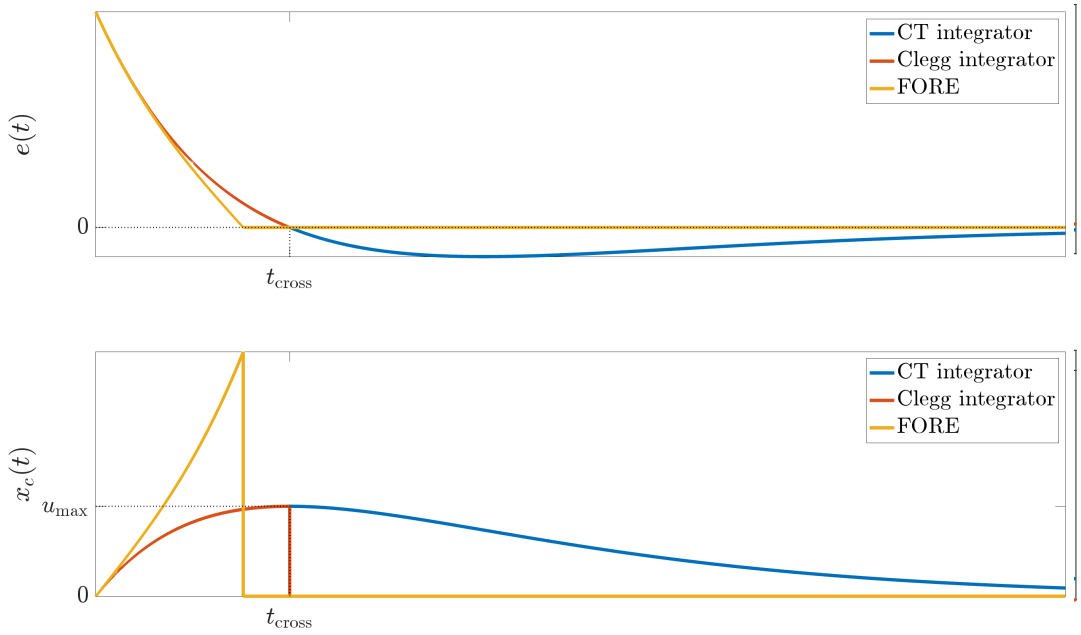
First Order Reset Elements

Inspired by the Clegg integrator [1958]

$$\begin{cases} \dot{x}_c = e, & \text{“normally”} \\ x_c^+ = 0, & \text{“at the right moment”} \end{cases}$$

FORE dynamics:

$$\begin{cases} \dot{x}_c = a_c x_c + b_c e, & x_c e \geq 0 \\ x_c^+ = 0, & x_c e \leq 0 \end{cases}$$

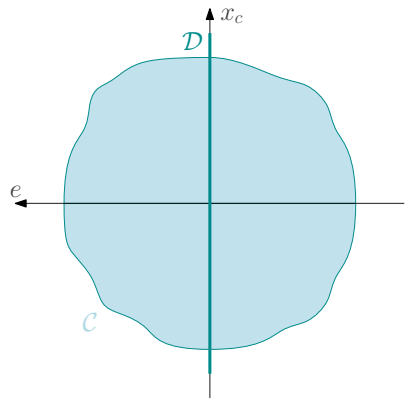


L. Zaccarian, D. Nesic and A. R. Teel. “First Order Reset Elements and the Clegg integrator revisited”. *Proceedings of the 2005, American Control Conference.* 2005.

FORE: modelling nuances

Initial model [1]

$$\begin{cases} \dot{x}_c = e, & e \neq 0 \\ x_c^+ = 0, & e = 0 \end{cases}$$



[1] I. Horowitz and P. Rosenbaum. "Non-linear design for cost of feedback reduction in systems with large parameter uncertainty". *International Journal of Control.* 1975.

[2] L. Zaccarian, D. Nesic and A. R. Teel. "First Order Reset Elements and the Clegg integrator revisited". *Proceedings of the 2005, American Control Conference.* 2005.

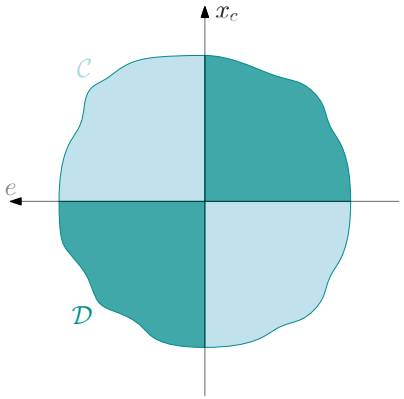
FORE: modelling nuances

Initial model [1]

$$\begin{cases} \dot{x}_c = e, & e \neq 0 \\ x_c^+ = 0, & e = 0 \end{cases}$$

Inaccurate, it was replaced by a robust version [2]

$$\begin{cases} \dot{x}_c = e, & x_c e \geq 0 \\ x_c^+ = 0, & x_c e \leq 0 \end{cases}$$



[1] I. Horowitz and P. Rosenbaum. "Non-linear design for cost of feedback reduction in systems with large parameter uncertainty". *International Journal of Control.* 1975.

[2] L. Zaccarian, D. Nesic and A. R. Teel. "First Order Reset Elements and the Clegg integrator revisited". *Proceedings of the 2005, American Control Conference.* 2005.

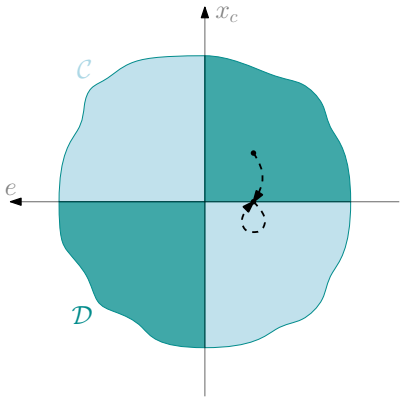
FORE: modelling nuances

Initial model [1]

$$\begin{cases} \dot{x}_c = e, & e \neq 0 \\ x_c^+ = 0, & e = 0 \end{cases}$$

Inaccurate, it was replaced by a robust version [2]

$$\begin{cases} \dot{x}_c = e, & x_c e \geq 0 \\ x_c^+ = 0, & x_c e \leq 0 \end{cases}$$



[1] I. Horowitz and P. Rosenbaum. "Non-linear design for cost of feedback reduction in systems with large parameter uncertainty". *International Journal of Control.* 1975.

[2] L. Zaccarian, D. Nesic and A. R. Teel. "First Order Reset Elements and the Clegg integrator revisited". *Proceedings of the 2005, American Control Conference.* 2005.

FORE: modelling nuances

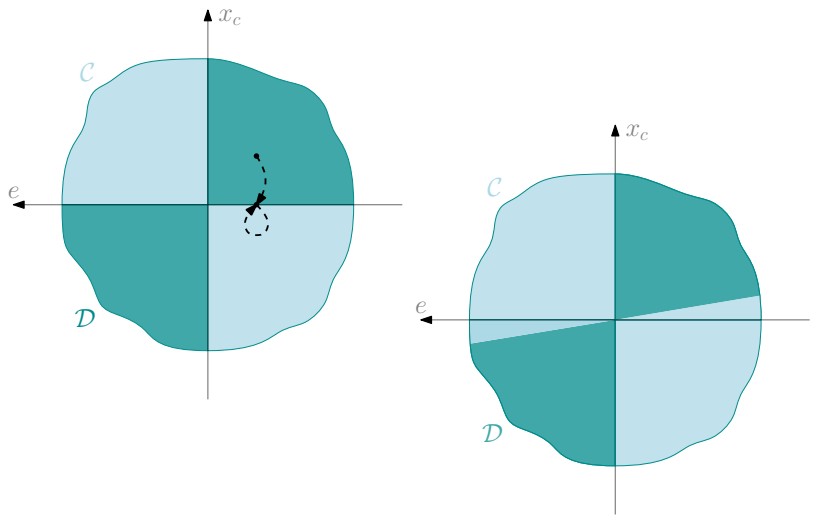
Initial model [1]

$$\begin{cases} \dot{x}_c = e, & e \neq 0 \\ x_c^+ = 0, & e = 0 \end{cases}$$

Inaccurate, it was replaced by a robust version [2]

$$\begin{cases} \dot{x}_c = e, & x_c e \geq 0 \\ x_c^+ = 0, & x_c e \leq 0 \end{cases}$$

Still, there are "bad" discrete-time solutions: solved introducing time/space regularization



[1] I. Horowitz and P. Rosenbaum. "Non-linear design for cost of feedback reduction in systems with large parameter uncertainty". *International Journal of Control.* 1975.

[2] L. Zaccarian, D. Nesic and A. R. Teel. "First Order Reset Elements and the Clegg integrator revisited". *Proceedings of the 2005, American Control Conference.* 2005.

FORE: modelling nuances

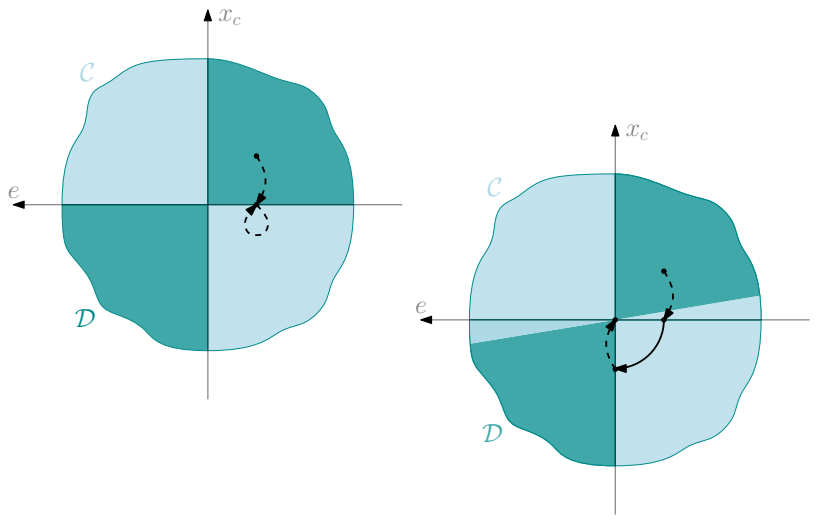
Initial model [1]

$$\begin{cases} \dot{x}_c = e, & e \neq 0 \\ x_c^+ = 0, & e = 0 \end{cases}$$

Inaccurate, it was replaced by a robust version [2]

$$\begin{cases} \dot{x}_c = e, & x_c e \geq 0 \\ x_c^+ = 0, & x_c e \leq 0 \end{cases}$$

Still, there are "bad" discrete-time solutions: solved introducing time/space regularization



[1] I. Horowitz and P. Rosenbaum. "Non-linear design for cost of feedback reduction in systems with large parameter uncertainty". *International Journal of Control.* 1975.

[2] L. Zaccarian, D. Nesic and A. R. Teel. "First Order Reset Elements and the Clegg integrator revisited". *Proceedings of the 2005, American Control Conference.* 2005.

Soft-reset control [1]

Start from generic reset-based closed loop...

$$\begin{cases} \dot{x} = Ax, & x^\top M x \leq 0 \\ x^+ = Rx, & x^\top M x \geq 0 \end{cases}$$

[1] A.R. Teel. "Continuous-time implementation of reset control systems". *Trends in Nonlinear and Adaptive Control (Z.P. Jiang, C. Prieur, and A. Astolfi, editors)*. 2021.

Soft-reset control [1]

Start from generic reset-based closed loop...

$$\begin{cases} \dot{x} = Ax, & x^\top M x \leq 0 \\ x^+ = Rx, & x^\top M x \geq 0 \end{cases}$$

...and define a differential inclusion

$$\dot{x} \in F(x) := Ax + \gamma(\mathrm{SGN}(x^\top M x) + 1)(Rx - x)$$

[1] A.R. Teel. "Continuous-time implementation of reset control systems". *Trends in Nonlinear and Adaptive Control (Z.P. Jiang, C. Prieur, and A. Astolfi, editors)*. 2021.

Soft-reset control [1]

Start from generic reset-based closed loop...

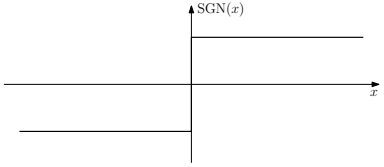
$$\begin{cases} \dot{x} = Ax, & x^\top M x \leq 0 \\ x^+ = Rx, & x^\top M x \geq 0 \end{cases}$$

...and define a differential inclusion

$$\dot{x} \in F(x) := Ax + \gamma(\mathrm{SGN}(x^\top M x) + 1)(Rx - x)$$

Krasovskii regularization:

$$\mathrm{SGN}(x) := \begin{cases} \mathrm{sgn}(x), & \text{if } x \neq 0 \\ [-1, 1], & \text{if } x = 0 \end{cases}$$



[1] A.R. Teel. "Continuous-time implementation of reset control systems". *Trends in Nonlinear and Adaptive Control (Z.P. Jiang, C. Prieur, and A. Astolfi, editors)*. 2021.

Soft-reset control [1]

Start from generic reset-based closed loop...

$$\begin{cases} \dot{x} = Ax, & x^\top M x \leq 0 \\ x^+ = Rx, & x^\top M x \geq 0 \end{cases}$$

...and define a differential inclusion

$$\dot{x} \in F(x) := Ax + \gamma(\mathrm{SGN}(x^\top M x) + 1)(Rx - x)$$

[1] A.R. Teel. "Continuous-time implementation of reset control systems". *Trends in Nonlinear and Adaptive Control (Z.P. Jiang, C. Prieur, and A. Astolfi, editors)*. 2021.

Soft-reset control [1]

Start from generic reset-based closed loop...

$$\begin{cases} \dot{x} = Ax, & x^\top M x \leq 0 \\ x^+ = Rx, & x^\top M x \geq 0 \end{cases}$$

...and define a differential inclusion

$$\dot{x} \in F(x) := Ax + \gamma(\mathrm{SGN}(x^\top M x) + 1)(Rx - x)$$

[1] A.R. Teel. "Continuous-time implementation of reset control systems". *Trends in Nonlinear and Adaptive Control (Z.P. Jiang, C. Prieur, and A. Astolfi, editors)*. 2021.

Soft-reset control [1]

Start from generic reset-based closed loop...

$$\begin{cases} \dot{x} = Ax, & x^\top M x \leq 0 \\ x^+ = Rx, & x^\top M x \geq 0 \end{cases}$$

...and define a differential inclusion

$$\dot{x} \in F(x) := Ax + \gamma(\mathrm{SGN}(x^\top M x) + 1)(Rx - x)$$

[1] A.R. Teel. "Continuous-time implementation of reset control systems". *Trends in Nonlinear and Adaptive Control (Z.P. Jiang, C. Prieur, and A. Astolfi, editors)*. 2021.

Soft-reset control [1]

Start from generic reset-based closed loop...

$$\begin{cases} \dot{x} = Ax, & x^\top Mx \leq 0 \\ x^+ = Rx, & x^\top Mx \geq 0 \end{cases}$$

...and define a differential inclusion

$$\dot{x} \in F(x) := Ax + \gamma(\mathrm{SGN}(x^\top Mx) + 1)(Rx - x)$$

[1] A.R. Teel. "Continuous-time implementation of reset control systems". *Trends in Nonlinear and Adaptive Control (Z.P. Jiang, C. Prieur, and A. Astolfi, editors)*. 2021.

Soft-reset control [1]

Start from generic reset-based closed loop...

$$\begin{cases} \dot{x} = Ax, & x^\top Mx \leq 0 \\ x^+ = Rx, & x^\top Mx \geq 0 \end{cases}$$

...and define a differential inclusion

$$\dot{x} \in F(x) := Ax + \gamma(\mathrm{SGN}(x^\top Mx) + 1)(Rx - x)$$

???

[1] A.R. Teel. "Continuous-time implementation of reset control systems". *Trends in Nonlinear and Adaptive Control (Z.P. Jiang, C. Prieur, and A. Astolfi, editors)*. 2021.

Soft-reset control [1]

Start from generic reset-based closed loop...

$$\begin{cases} \dot{x} = Ax, & x^\top Mx \leq 0 \\ x^+ = Rx, & x^\top Mx \geq 0 \end{cases}$$

...and define a differential inclusion

$$\dot{x} \in F(x) := Ax + \gamma(\mathrm{SGN}(x^\top Mx) + 1)(Rx - x)$$

$$x^\top Mx \leq 0 \rightarrow \mathrm{SGN}(x^\top Mx) + 1 = 0$$

$$F(x) = Ax$$

[1] A.R. Teel. "Continuous-time implementation of reset control systems". *Trends in Nonlinear and Adaptive Control (Z.P. Jiang, C. Prieur, and A. Astolfi, editors)*. 2021.

Soft-reset control [1]

Start from generic reset-based closed loop...

$$\begin{cases} \dot{x} = Ax, & x^\top Mx \leq 0 \\ x^+ = Rx, & x^\top Mx \geq 0 \end{cases}$$

...and define a differential inclusion

$$\dot{x} \in F(x) := Ax + \gamma(\mathrm{SGN}(x^\top Mx) + 1)(Rx - x)$$

$$x^\top Mx \leq 0 \rightarrow \mathrm{SGN}(x^\top Mx) + 1 = 0$$
$$F(x) = Ax$$

$$x^\top Mx \geq 0 \rightarrow \mathrm{SGN}(x^\top Mx) + 1 = 2$$
$$\gamma \gg 1 \rightarrow \dot{x} = -\gamma(x - Rx)$$

[1] A.R. Teel. "Continuous-time implementation of reset control systems". *Trends in Nonlinear and Adaptive Control (Z.P. Jiang, C. Prieur, and A. Astolfi, editors)*. 2021.

Soft-reset control [1]

Start from generic reset-based closed loop...

$$\begin{cases} \dot{x} = Ax, & x^\top Mx \leq 0 \\ x^+ = Rx, & x^\top Mx \geq 0 \end{cases}$$

...and define a differential inclusion

$$\dot{x} \in F(x) := Ax + \gamma(\mathrm{SGN}(x^\top Mx) + 1)(Rx - x)$$

Regulates how closely the CT solutions resemble the ones of the original HS!

$$x^\top Mx \leq 0 \rightarrow \mathrm{SGN}(x^\top Mx) + 1 = 0$$
$$F(x) = Ax$$

$$x^\top Mx \geq 0 \rightarrow \mathrm{SGN}(x^\top Mx) + 1 = 2$$
$$\gamma \gg 1 \rightarrow \dot{x} = -\gamma(x - Rx)$$

[1] A.R. Teel. "Continuous-time implementation of reset control systems". *Trends in Nonlinear and Adaptive Control (Z.P. Jiang, C. Prieur, and A. Astolfi, editors)*. 2021.

Soft-reset control [1]

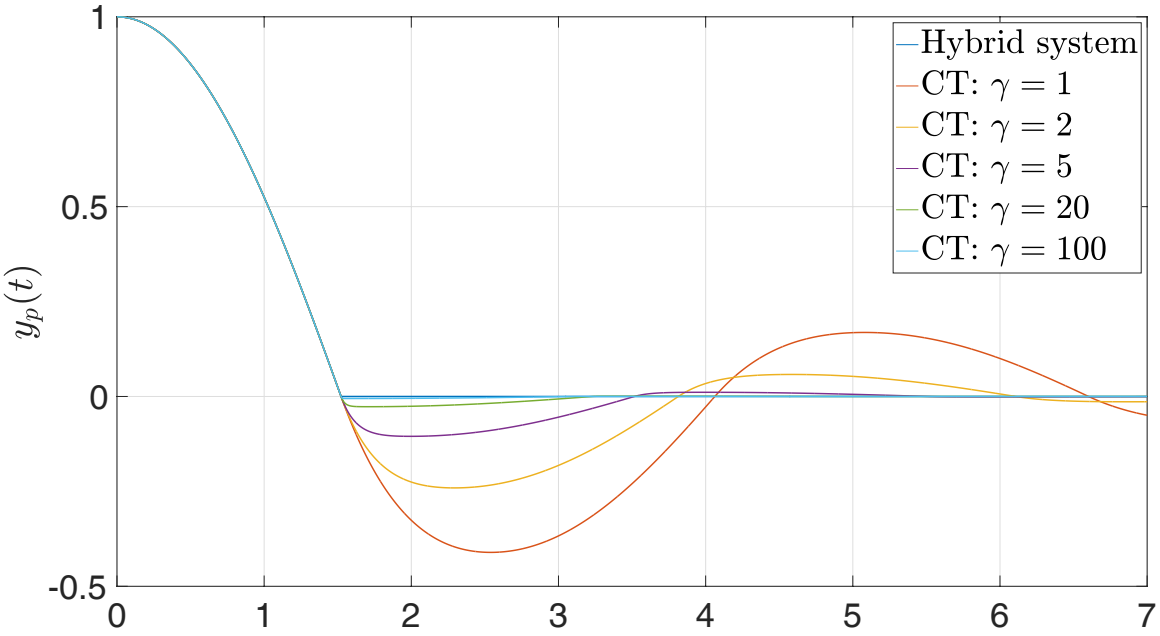
Start from generic reset-based closed loop...

$$\begin{cases} \dot{x} = Ax, & x^\top Mx \leq 0 \\ x^+ = Rx, & x^\top Mx \geq 0 \end{cases}$$

...and define a differential inclusion

$$\dot{x} \in F(x) := Ax + \gamma(\mathrm{SGN}(x^\top Mx) + 1)(Rx - x)$$

Regulates how closely the CT solutions resemble the ones of the original HS!



[1] A.R. Teel. "Continuous-time implementation of reset control systems". *Trends in Nonlinear and Adaptive Control (Z.P. Jiang, C. Prieur, and A. Astolfi, editors)*. 2021.

Stability with soft-reset control

$$\dot{x} \in F(x) := Ax + \gamma(\mathrm{SGN}(x^\top Mx) + 1)(Rx - x)$$

Theorem

If $\exists \varepsilon > 0$ and a Lyapunov function V such that

1. V is **Lipschitz continuous**, homogeneous of degree 2, positive definite, strongly convex;
2. defined $C_\varepsilon := \{x \in \mathbb{R}^b : x^\top Mx \leq \varepsilon x^\top x\}$, we have

$$\begin{aligned} &\langle \nabla V, Ax \rangle \leq 0, && \textbf{for almost all } x \in C_\varepsilon, \\ &V(Rx) - V(x) \leq 0, && \text{for all } x \in D; \end{aligned}$$

3. $x \in D$ implies $Rx \in C$;
4. no continuous solution of the hybrid system keeps V constant and nonzero;

then the origin is GES for $\dot{x} \in F(x)$, for $\gamma > 0$ large enough.

Why do we need nonsmoothness?

FORE + integrator closed loop:

$$\dot{x} = \begin{bmatrix} \dot{x}_c \\ \dot{e} \end{bmatrix} = \begin{bmatrix} 0.1 & 1 \\ -1 & 0 \end{bmatrix} \begin{bmatrix} x_c \\ e \end{bmatrix} = Ax$$

Why do we need nonsmoothness?

FORE + integrator closed loop:

$$\dot{x} = \begin{bmatrix} \dot{x}_c \\ \dot{e} \end{bmatrix} = \begin{bmatrix} 0.1 & 1 \\ -1 & 0 \end{bmatrix} \begin{bmatrix} x_c \\ e \end{bmatrix} = Ax$$

Quadratic Lyapunov function:

$$V = x^\top P x = x^\top \begin{bmatrix} p_{11} & p_{12} \\ p_{12} & p_{22} \end{bmatrix} x$$

Why do we need nonsmoothness?

FORE + integrator closed loop:

$$\dot{x} = \begin{bmatrix} \dot{x}_c \\ \dot{e} \end{bmatrix} = \begin{bmatrix} 0.1 & 1 \\ -1 & 0 \end{bmatrix} \begin{bmatrix} x_c \\ e \end{bmatrix} = Ax$$

Quadratic Lyapunov function:

$$V = x^\top P x = x^\top \begin{bmatrix} p_{11} & p_{12} \\ p_{12} & p_{22} \end{bmatrix} x$$

Flow condition:

$$\dot{V}(x) := \langle \nabla V, Ax \rangle = 2x^\top APx = 2x^\top \begin{bmatrix} 0.1p_{11} - p_{12} & p_{11} \\ 0.1p_{12} - p_{22} & p_{12} \end{bmatrix} x$$

Why do we need nonsmoothness?

FORE + integrator closed loop:

$$\dot{x} = \begin{bmatrix} \dot{x}_c \\ \dot{e} \end{bmatrix} = \begin{bmatrix} 0.1 & 1 \\ -1 & 0 \end{bmatrix} \begin{bmatrix} x_c \\ e \end{bmatrix} = Ax$$

Quadratic Lyapunov function:

$$V = x^\top P x = x^\top \begin{bmatrix} p_{11} & p_{12} \\ p_{12} & p_{22} \end{bmatrix} x$$

Flow condition:

$$\dot{V}(x) := \langle \nabla V, Ax \rangle = 2x^\top APx = 2x^\top \begin{bmatrix} 0.1p_{11} - p_{12} & p_{11} \\ 0.1p_{12} - p_{22} & p_{12} \end{bmatrix} x$$

$$\dot{V}\left([0,1]^\top\right) \leq 0 \quad \dot{V}\left([1,0]^\top\right) \leq 0$$

$$\downarrow \qquad \downarrow$$

$$p_{12} \leq 0 \qquad p_{12} - 0.1p_{11} \geq 0$$

Why do we need nonsmoothness?

FORE + integrator closed loop:

$$\dot{x} = \begin{bmatrix} \dot{x}_c \\ \dot{e} \end{bmatrix} = \begin{bmatrix} 0.1 & 1 \\ -1 & 0 \end{bmatrix} \begin{bmatrix} x_c \\ e \end{bmatrix} = Ax$$

Quadratic Lyapunov function:

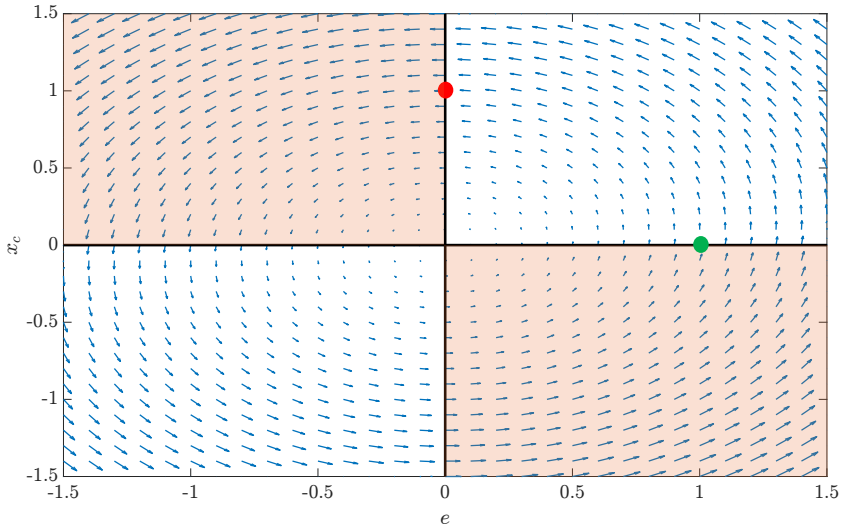
$$V = x^\top P x = x^\top \begin{bmatrix} p_{11} & p_{12} \\ p_{12} & p_{22} \end{bmatrix} x$$

Flow condition:

$$\dot{V}(x) := \langle \nabla V, Ax \rangle = 2x^\top APx = 2x^\top \begin{bmatrix} 0.1p_{11} - p_{12} & p_{11} \\ 0.1p_{12} - p_{22} & p_{12} \end{bmatrix} x$$

$$\dot{V}\left([0,1]^\top\right) \leq 0 \quad \Rightarrow \quad p_{12} \leq 0$$

$$\dot{V}\left([1,0]^\top\right) \leq 0 \quad \Rightarrow \quad p_{12} - 0.1p_{11} \geq 0$$



Why do we need nonsmoothness?

FORE + integrator closed loop:

$$\dot{x} = \begin{bmatrix} \dot{x}_c \\ \dot{e} \end{bmatrix} = \begin{bmatrix} 0.1 & 1 \\ -1 & 0 \end{bmatrix} \begin{bmatrix} x_c \\ e \end{bmatrix} = Ax$$

Quadratic Lyapunov function:

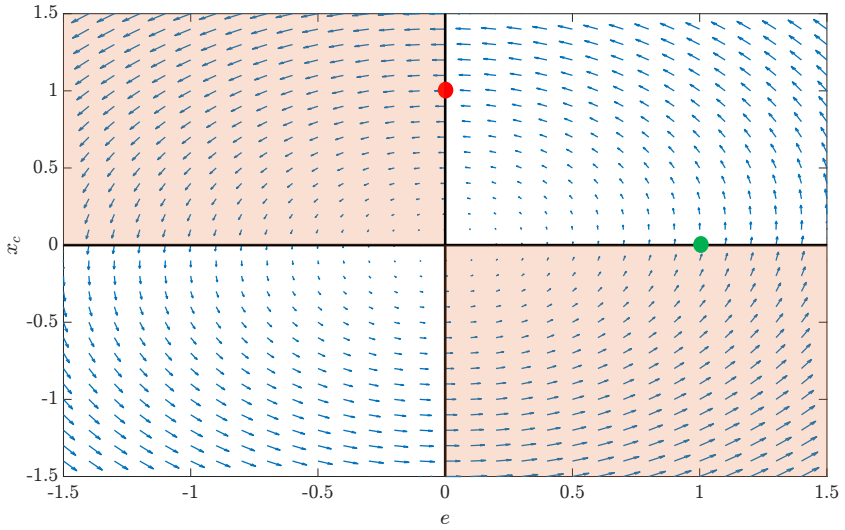
$$V = x^\top P x = x^\top \begin{bmatrix} p_{11} & p_{12} \\ p_{12} & p_{22} \end{bmatrix} x$$

Flow condition:

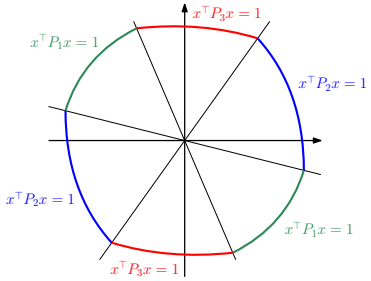
$$\dot{V}(x) := \langle \nabla V, Ax \rangle = 2x^\top APx = 2x^\top \begin{bmatrix} 0.1p_{11} - p_{12} & p_{11} \\ 0.1p_{12} - p_{22} & p_{12} \end{bmatrix} x$$

$$\dot{V}\left([0,1]^\top\right) \leq 0 \quad \Rightarrow \quad p_{12} \leq 0$$

$$\dot{V}\left([1,0]^\top\right) \leq 0 \quad \Rightarrow \quad p_{12} - 0.1p_{11} \geq 0$$

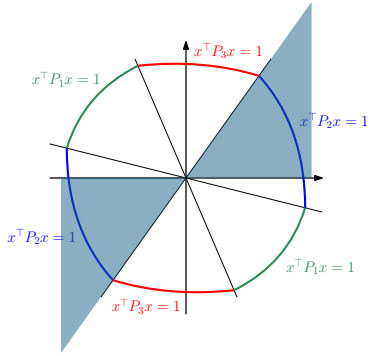


Piecewise-quadratic (max-of-quadratics) functions!



Numerical sufficient conditions

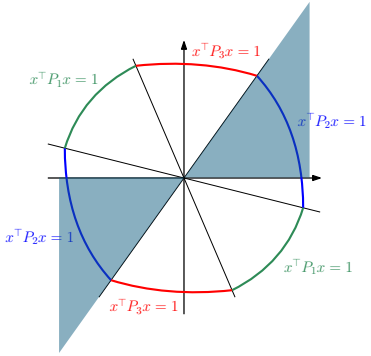
$$V(x) = \max_{i \in \{1,\dots,q\}} x^\top P_i x, \quad P_i \succ 0$$



Numerical sufficient conditions

$$V(x) = \max_{i \in \{1,\dots,q\}} x^\top P_i x, \quad P_i \succ 0$$

How do we check the Lyapunov conditions numerically?



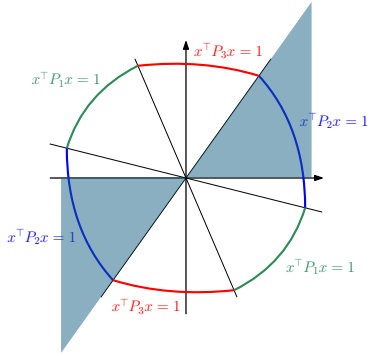
Numerical sufficient conditions

$$V(x) = \max_{i \in \{1,\dots,q\}} x^\top P_i x, \quad P_i \succ 0$$

How do we check the Lyapunov conditions numerically?

$A^\top P_i + P_i A \preceq 0, \quad \forall i \in \{1, \dots, q\}$?

Still conservative!



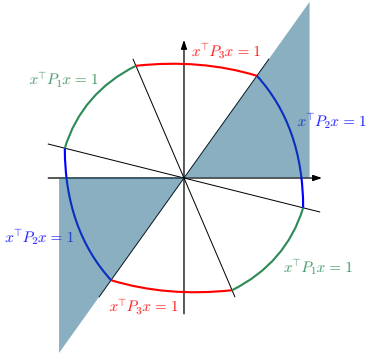
Numerical sufficient conditions

$$V(x) = \max_{i \in \{1,\dots,q\}} x^\top P_i x, \quad P_i \succ 0$$

How do we check the Lyapunov conditions numerically?

$$A^\top P_i + P_i A \preceq 0, \quad \forall i \in \{1,\dots,q\}?$$

Still conservative!



We need to check

$$\begin{cases} x \in C_\varepsilon \\ x^\top (P_i - P_j) x \geq 0, \ \forall j \end{cases} \implies \langle P_i x, Ax \rangle \leq 0$$

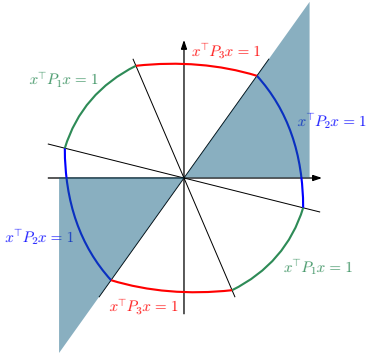
Numerical sufficient conditions

$$V(x) = \max_{i \in \{1,\dots,q\}} x^\top P_i x, \quad P_i \succ 0$$

How do we check the Lyapunov conditions numerically?

$$A^\top P_i + P_i A \preceq 0, \quad \forall i \in \{1,\dots,q\}?$$

Still conservative!



S-procedure

We need to check

$$\begin{cases} x \in C_\varepsilon \\ x^\top (P_i - P_j) x \geq 0, \ \forall j \end{cases} \implies \langle P_i x, Ax \rangle \leq 0$$

Numerical sufficient conditions

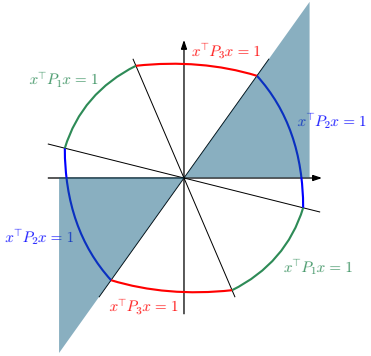
$$V(x) = \max_{i \in \{1,\dots,q\}} x^\top P_i x, \quad P_i \succ 0$$

How do we check the Lyapunov conditions numerically?

$A^\top P_i + P_i A \preceq 0, \quad \forall i \in \{1,\dots,q\}$?
Still conservative!

We need to check

$$\begin{cases} x \in C_\varepsilon \\ x^\top (P_i - P_j) x \geq 0, \ \forall j \end{cases} \implies \langle P_i x, Ax \rangle \leq 0$$



S-procedure

Applied once: necessary and sufficient

$$x^\top Q x \geq 0 \implies x^\top S x \geq 0$$

iff

$$\exists \mu > 0 : S - \mu Q \succeq 0$$

Numerical sufficient conditions

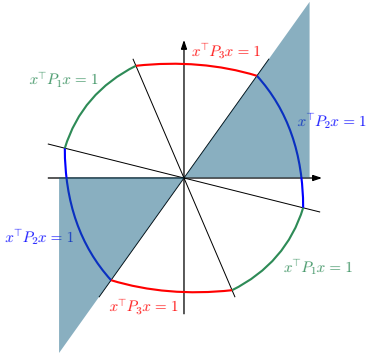
$$V(x) = \max_{i \in \{1,\dots,q\}} x^\top P_i x, \quad P_i \succ 0$$

How do we check the Lyapunov conditions numerically?

$A^\top P_i + P_i A \preceq 0, \quad \forall i \in \{1,\dots,q\}$?
Still conservative!

We need to check

$$\begin{cases} x \in C_\varepsilon \\ x^\top (P_i - P_j) x \geq 0, \ \forall j \end{cases} \implies \langle P_i x, Ax \rangle \leq 0$$



S-procedure

Applied once: necessary and sufficient

$$x^\top Q x \geq 0 \implies x^\top S x \geq 0$$

iff

$$\exists \mu > 0 : S - \mu Q \succeq 0$$

Applied multiple times: only sufficient

$$x^\top Q_1 x \geq 0, \dots, x^\top Q_q x \geq 0 \implies x^\top S x \geq 0$$

if

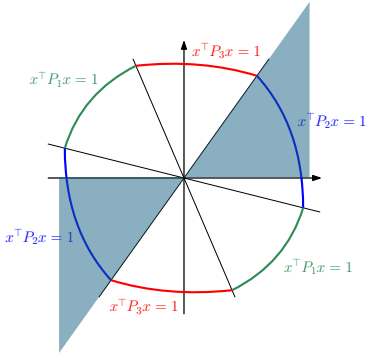
$$\exists \mu_1, \dots, \mu_q > 0 : S - \sum_{i=1}^{q} \mu_i Q_i \succeq 0$$

Numerical sufficient conditions

$$V(x) = \max_{i \in \{1,\dots,q\}} x^\top P_i x, \quad P_i \succ 0$$

How do we check the Lyapunov conditions numerically?

$A^\top P_i + P_i A \preceq 0, \quad \forall i \in \{1,\dots,q\}$?
Still conservative!



S-procedure

Applied once: necessary and sufficient

$$x^\top Q x \geq 0 \implies x^\top S x \geq 0$$

iff

$$\exists \mu > 0 : S - \mu Q \succeq 0$$

We need to check

$$\begin{cases} x \in C_\varepsilon \\ x^\top (P_i - P_j) x \geq 0, \ \forall j \end{cases} \implies \langle P_i x, Ax \rangle \leq 0$$

Many numerical tests, none feasible

Applied multiple times: only sufficient

$$x^\top Q_1 x \geq 0, \dots, x^\top Q_q x \geq 0 \implies x^\top S x \geq 0$$

if

$$\exists \mu_1, \dots, \mu_q > 0 : S - \sum_{i=1}^{q} \mu_i Q_i \succeq 0$$

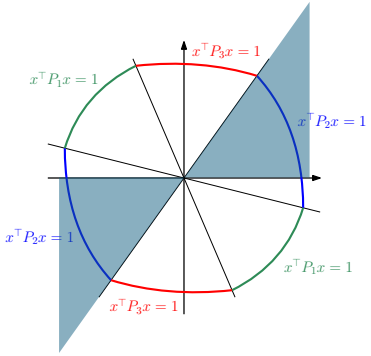
Numerical sufficient conditions

$$V(x) = \max_{i \in \{1,\dots,q\}} x^\top P_i x, \quad P_i \succ 0$$

How do we check the Lyapunov conditions numerically?

$$A^\top P_i + P_i A \preceq 0, \quad \forall i \in \{1,\dots,q\}?$$

Still conservative!



Sum Of Squares (SOS) programming

We need to check

$$\begin{cases} x \in C_\varepsilon \\ x^\top (P_i - P_j) x \geq 0, \ \forall j \end{cases} \implies \langle P_i x, Ax \rangle \leq 0$$

Numerical sufficient conditions

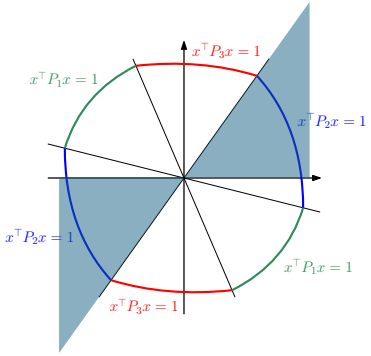
$$V(x) = \max_{i \in \{1,\dots,q\}} x^\top P_i x, \quad P_i \succ 0$$

How do we check the Lyapunov conditions numerically?

$A^\top P_i + P_i A \preceq 0, \quad \forall i \in \{1,\dots,q\}$?
Still conservative!

We need to check

$$\begin{cases} x \in C_\varepsilon \\ x^\top (P_i - P_j) x \geq 0, \ \forall j \end{cases} \implies \langle P_i x, Ax \rangle \leq 0$$



Sum Of Squares (SOS) programming

Same concept, but with $\mu(x)$

$$x^\top Q x \geq 0 \implies x^\top S x \geq 0$$

iff

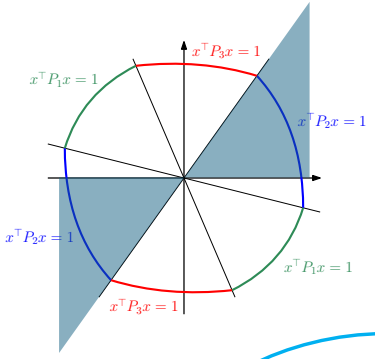
$$\exists \mu(x) \in SOS : x^\top S x - \mu(x) x^\top Q x \in SOS$$

Numerical sufficient conditions

$$V(x) = \max_{i \in \{1,\dots,q\}} x^\top P_i x, \quad P_i \succ 0$$

How do we check the Lyapunov conditions numerically?

$A^\top P_i + P_i A \preceq 0, \quad \forall i \in \{1,\dots,q\}?$
Still conservative!



Sum Of Squares (SOS) programming

Same concept, but with $\mu(x)$

$$x^\top Q x \geq 0 \implies x^\top S x \geq 0$$

iff

$$\exists \mu(x) \in SOS : x^\top S x - \mu(x) x^\top Q x \in SOS$$

We need to check

$$\begin{cases} x \in C_\varepsilon \\ x^\top (P_i - P_j) x \geq 0, \ \forall j \end{cases} \implies \langle P_i x, Ax \rangle \leq 0$$

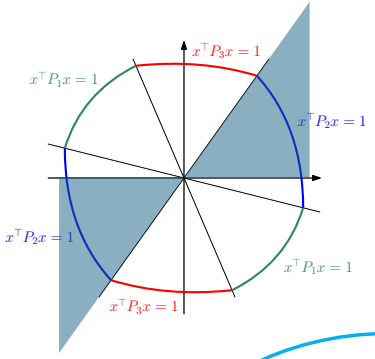
$$-x^\top (A^\top P_i + P_i A) x - 2\alpha x^\top P_i x - r_{i,F}(x)(1 - x^\top x) - \sum_{i=1}^{q} \mu_{ij}(x) x^\top (P_i - P_j) x - \mu_{i0}(x) x^\top (M - \varepsilon I) x \in SOS$$

Numerical sufficient conditions

$$V(x) = \max_{i \in \{1,\dots,q\}} x^\top P_i x, \quad P_i \succ 0$$

How do we check the Lyapunov conditions numerically?

$A^\top P_i + P_i A \preceq 0, \quad \forall i \in \{1,\dots,q\}?$
Still conservative!



Sum Of Squares (SOS) programming

Same concept, but with $\mu(x)$

$$x^\top Q x \geq 0 \implies x^\top S x \geq 0$$

iff

$$\exists \mu(x) \in SOS : x^\top S x - \mu(x) x^\top Q x \in SOS$$

We need to check

$$\begin{cases} x \in C_\varepsilon \\ x^\top (P_i - P_j) x \geq 0, \ \forall j \end{cases} \implies \langle P_i x, Ax \rangle \leq 0$$

$\dot{V}(x) \leq -2\alpha V(x)$

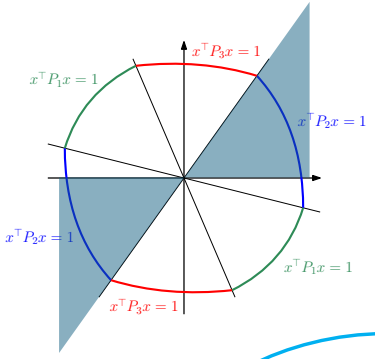
$$-x^\top (A^\top P_i + P_i A) x - 2\alpha x^\top P_i x - r_{i,F}(x)(1 - x^\top x) - \sum_{i=1}^{q} \mu_{ij}(x) x^\top (P_i - P_j) x - \mu_{i0}(x) x^\top (M - \varepsilon I) x \in SOS$$

Numerical sufficient conditions

$$V(x) = \max_{i \in \{1,\dots,q\}} x^\top P_i x, \quad P_i \succ 0$$

How do we check the Lyapunov conditions numerically?

$A^\top P_i + P_i A \preceq 0, \quad \forall i \in \{1,\dots,q\}?$
Still conservative!



Sum Of Squares (SOS) programming

Same concept, but with $\mu(x)$

$$x^\top Q x \geq 0 \implies x^\top S x \geq 0$$

iff

$$\exists \mu(x) \in SOS : x^\top S x - \mu(x) x^\top Q x \in SOS$$

We need to check

$$\begin{cases} x \in C_\varepsilon \\ x^\top (P_i - P_j) x \geq 0, \ \forall j \end{cases} \implies \langle P_i x, Ax \rangle \leq 0$$

$\dot{V}(x) \leq -2\alpha V(x)$ $V(x) = x^\top P_i x$

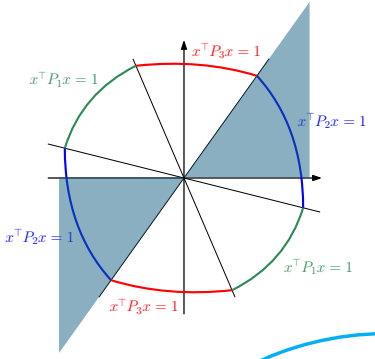
$$-x^\top (A^\top P_i + P_i A) x - 2\alpha x^\top P_i x - r_{i,F}(x)(1 - x^\top x) - \sum_{i=1}^{q} \mu_{ij}(x) x^\top (P_i - P_j) x - \mu_{i0}(x) x^\top (M - \varepsilon I) x \in SOS$$

Numerical sufficient conditions

$$V(x) = \max_{i \in \{1,\dots,q\}} x^\top P_i x, \quad P_i \succ 0$$

How do we check the Lyapunov conditions numerically?

$A^\top P_i + P_i A \preceq 0, \quad \forall i \in \{1,\dots,q\}?$
Still conservative!



Sum Of Squares (SOS) programming

Same concept, but with $\mu(x)$

$$x^\top Q x \geq 0 \implies x^\top S x \geq 0$$

iff

$$\exists \mu(x) \in SOS : x^\top S x - \mu(x) x^\top Q x \in SOS$$

We need to check

$$\begin{cases} x \in C_\varepsilon \\ x^\top (P_i - P_j) x \geq 0, \ \forall j \end{cases} \implies \langle P_i x, Ax \rangle \leq 0$$

$\dot{V}(x) \leq -2\alpha V(x)$ $V(x) = x^\top P_i x$ $x \in C_\varepsilon$

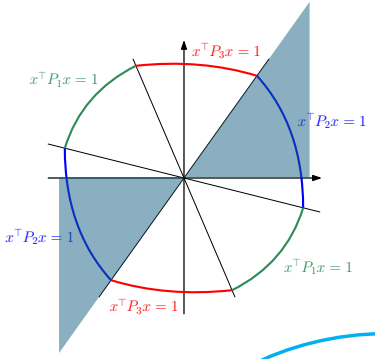
$$-x^\top (A^\top P_i + P_i A) x - 2\alpha x^\top P_i x - r_{i,F}(x)(1 - x^\top x) - \sum_{i=1}^{q} \mu_{ij}(x) x^\top (P_i - P_j) x - \mu_{i0}(x) x^\top (M - \varepsilon I) x \in SOS$$

Numerical sufficient conditions

$$V(x) = \max_{i \in \{1,\dots,q\}} x^\top P_i x, \quad P_i \succ 0$$

How do we check the Lyapunov conditions numerically?

$A^\top P_i + P_i A \preceq 0, \quad \forall i \in \{1,\dots,q\}?$
Still conservative!



Sum Of Squares (SOS) programming

Same concept, but with $\mu(x)$

$$x^\top Q x \geq 0 \implies x^\top S x \geq 0$$

iff

$$\exists \mu(x) \in SOS : x^\top S x - \mu(x) x^\top Q x \in SOS$$

We need to check

$$\begin{cases} x \in C_\varepsilon \\ x^\top (P_i - P_j) x \geq 0, \ \forall j \end{cases} \implies \langle P_i x, Ax \rangle \leq 0$$

$\dot{V}(x) \leq -2\alpha V(x)$ $V(x) = x^\top P_i x$ $x \in C_\varepsilon$

Check only on the unit sphere

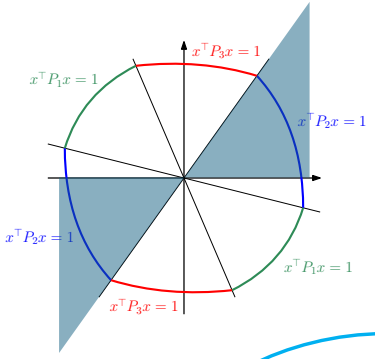
$$-x^\top (A^\top P_i + P_i A) x - 2\alpha x^\top P_i x - r_{i,F}(x)(1 - x^\top x) - \sum_{i=1}^{q} \mu_{ij}(x) x^\top (P_i - P_j) x - \mu_{i0}(x) x^\top (M - \varepsilon I) x \in SOS$$

Numerical sufficient conditions

$$V(x) = \max_{i \in \{1,\dots,q\}} x^\top P_i x, \quad P_i \succ 0$$

How do we check the Lyapunov conditions numerically?

$A^\top P_i + P_i A \preceq 0, \quad \forall i \in \{1,\dots,q\}?$
Still conservative!



Sum Of Squares (SOS) programming

Same concept, but with $\mu(x)$

$$x^\top Q x \geq 0 \implies x^\top S x \geq 0$$

iff

$$\exists \mu(x) \in SOS : x^\top S x - \mu(x) x^\top Q x \in SOS$$

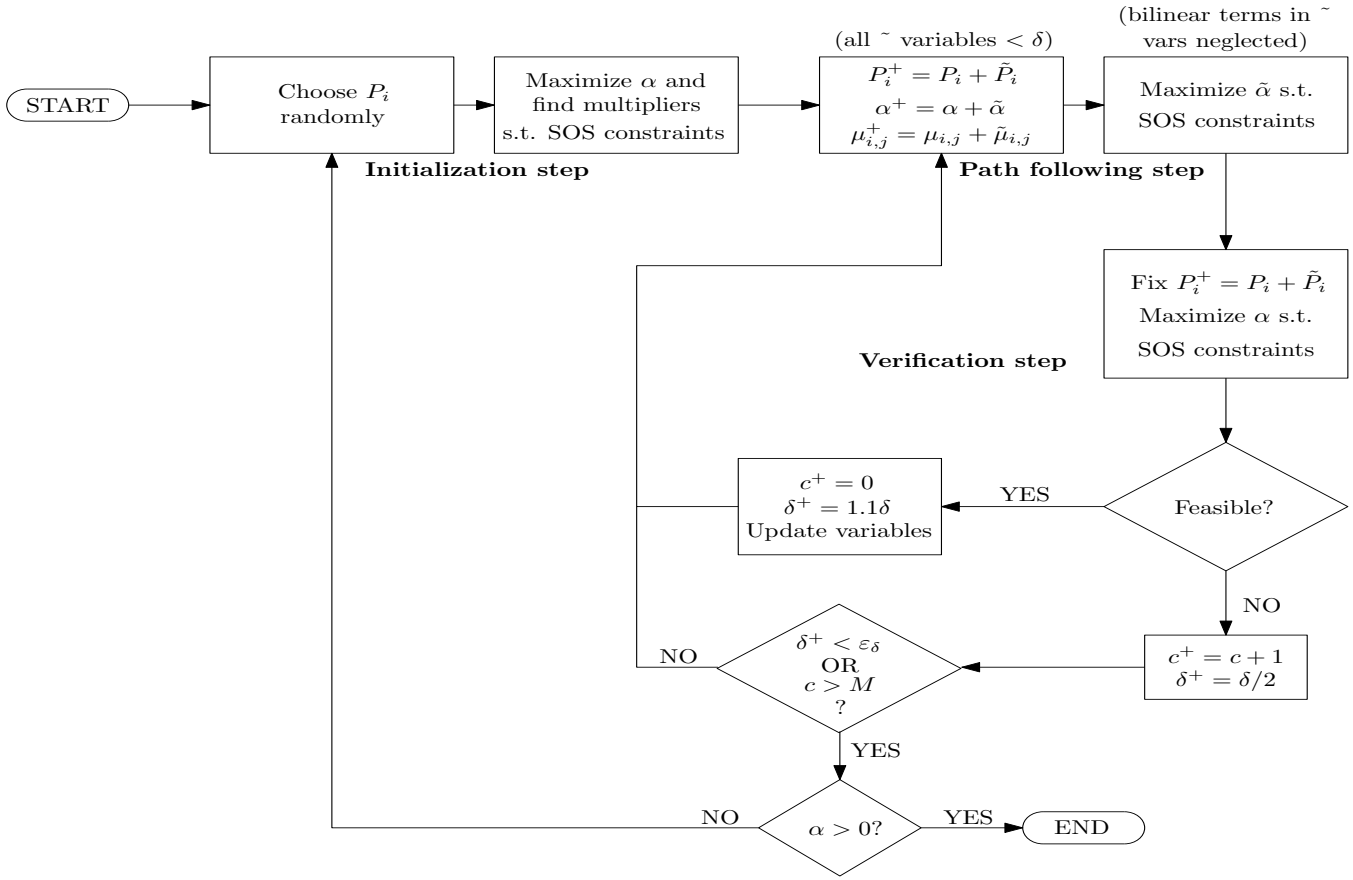
We need to check

$$\begin{cases} x \in C_\varepsilon \\ x^\top (P_i - P_j) x \geq 0, \ \forall j \end{cases} \implies \langle P_i x, Ax \rangle \leq 0$$

We cannot optimize over P_i because this term would be bilinear!

$$-x^\top (A^\top P_i + P_i A) x - 2\alpha x^\top P_i x - r_{i,F}(x)(1 - x^\top x) - \sum_{i=1}^{q} \mu_{ij}(x) x^\top (P_i - P_j) x - \mu_{i0} x^\top (M - \varepsilon I) x \in SOS$$

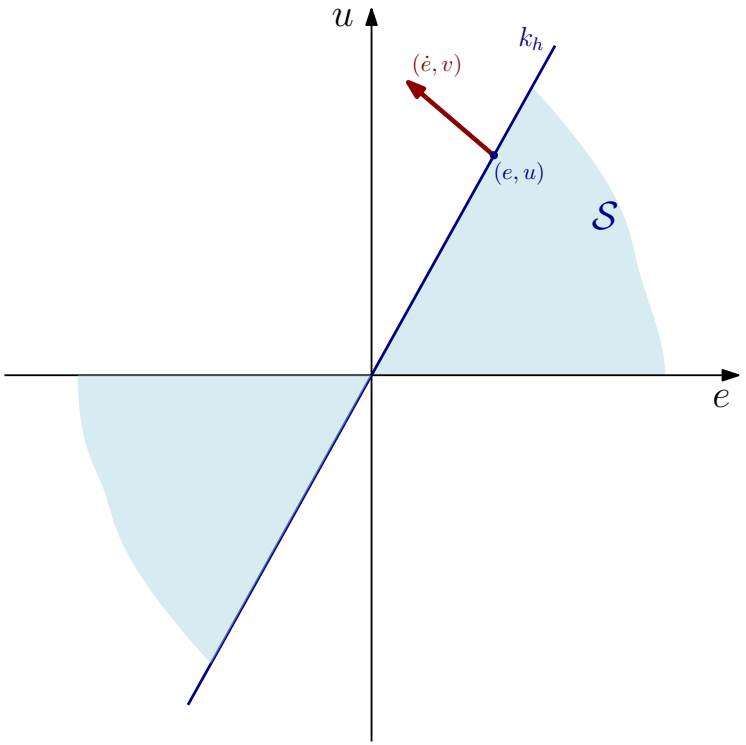
Recursive algorithm for synthesis



[1] A. Hassibi, J. How, S. Boyd, "A path-following method for solving BMI problems in control". *Proceedings of the American Control Conference.* 1999.

To (soft-)reset or to project?

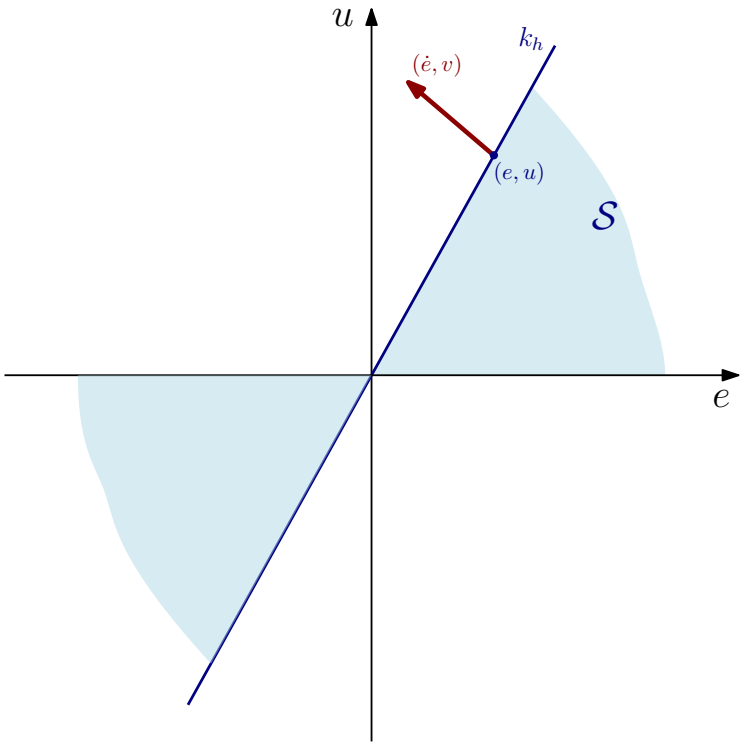
Reset control works well because it keeps $ex_c \geq 0$



To (soft-)reset or to project?

Reset control works well because it keeps $ex_c \geq 0$

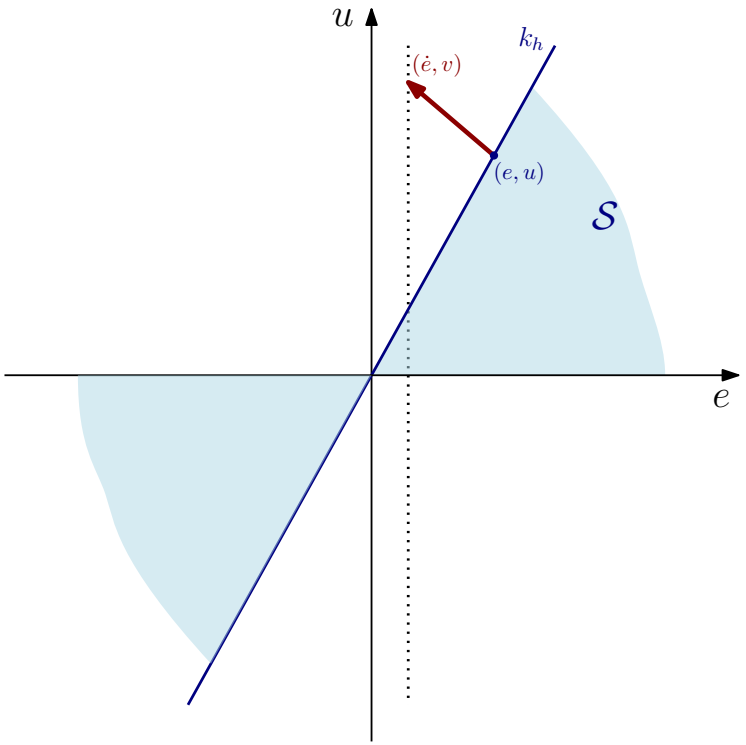
Another way to bound trajectories is through **projections**



To (soft-)reset or to project?

Reset control works well because it keeps $ex_c \geq 0$

Another way to bound trajectories is through **projections**



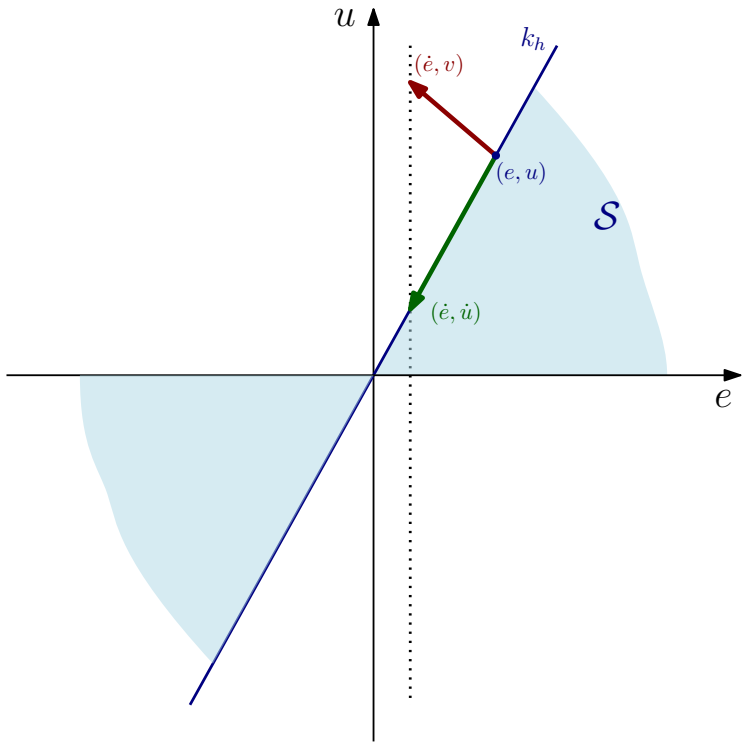
To (soft-)reset or to project?

Reset control works well because it keeps $ex_c \geq 0$

Another way to bound trajectories is through **projections**

Hybrid Integrator Gain Systems (HIGS) [1]

$$\dot{u} = \begin{cases} \omega_h e, & (\omega_h e, \dot{e}) \in T_{\mathcal{S}}(e,u) \\ k_h \dot{e}, & (\omega_h e, \dot{e}) \notin T_{\mathcal{S}}(e,u) \end{cases}$$



[1] D.A. Deenen et al., "Projection-based integrators for improved motion control: formalization, well-posedness and stability of Hybrid Integrator-Gain Systems". *Automatica.* 2021.

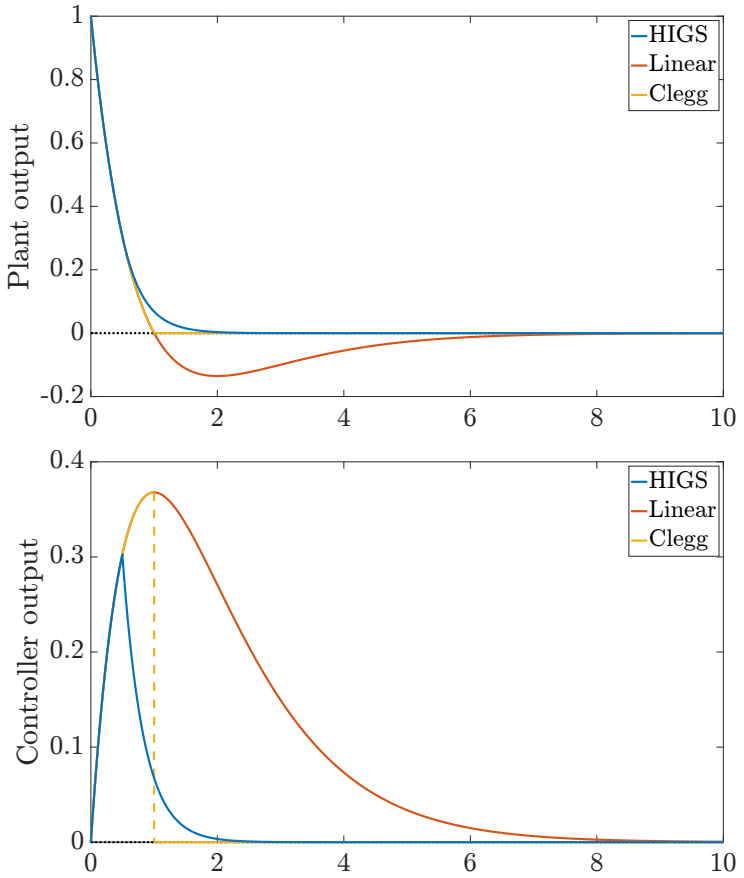
To (soft-)reset or to project?

Reset control works well because it keeps $ex_c \geq 0$

Another way to bound trajectories is through **projections**

Hybrid Integrator Gain Systems (HIGS) [1]

$$\dot{u} = \begin{cases} \omega_h e, & (\omega_h e, \dot{e}) \in T_{\mathcal{S}}(e, u) \\ k_h \dot{e}, & (\omega_h e, \dot{e}) \notin T_{\mathcal{S}}(e, u) \end{cases}$$



[1] D.A. Deenen et al., "Projection-based integrators for improved motion control: formalization, well-posedness and stability of Hybrid Integrator-Gain Systems". *Automatica.* 2021.

To (soft-)reset or to project?

Reset control works well because it keeps $ex_c \geq 0$

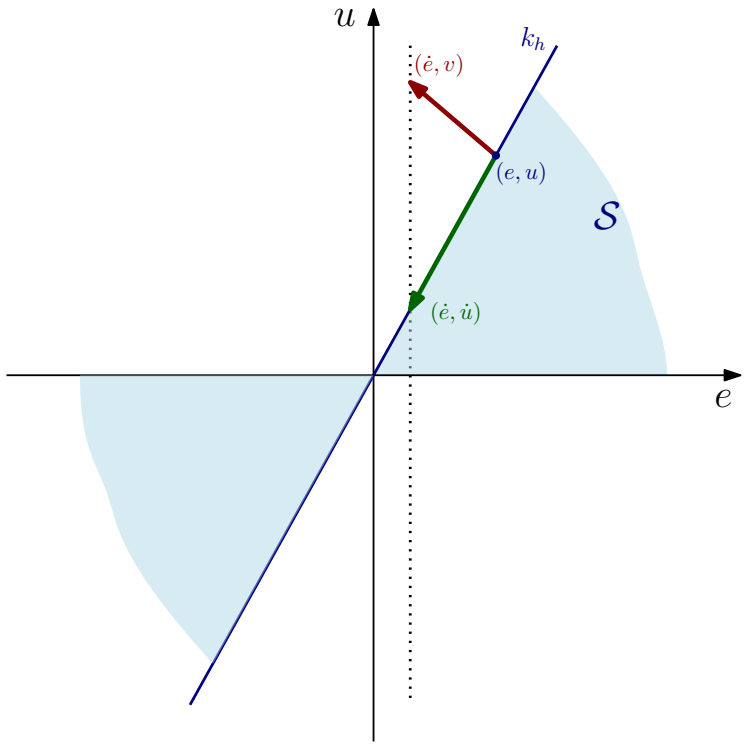
Another way to bound trajectories is through **projections**

Hybrid Integrator Gain Systems (HIGS) [1]

$$\dot{u} = \begin{cases} \omega_h e, & (\omega_h e, \dot{e}) \in T_{\mathcal{S}}(e, u) \\ k_h \dot{e}, & (\omega_h e, \dot{e}) \notin T_{\mathcal{S}}(e, u) \end{cases}$$

Can be any "nominal" dynamics

$$\dot{u} = \begin{cases} \boldsymbol{v}, & (\boldsymbol{v}, \dot{e}) \in T_{\mathcal{S}}(e, u) \\ k_h \dot{e}, & (\boldsymbol{v}, \dot{e}) \notin T_{\mathcal{S}}(e, u) \end{cases}$$



[1] D.A. Deenen et al., "Projection-based integrators for improved motion control: formalization, well-posedness and stability of Hybrid Integrator-Gain Systems". *Automatica.* 2021.

To (soft-)reset or to project?

Reset control works well because it keeps $ex_c \geq 0$

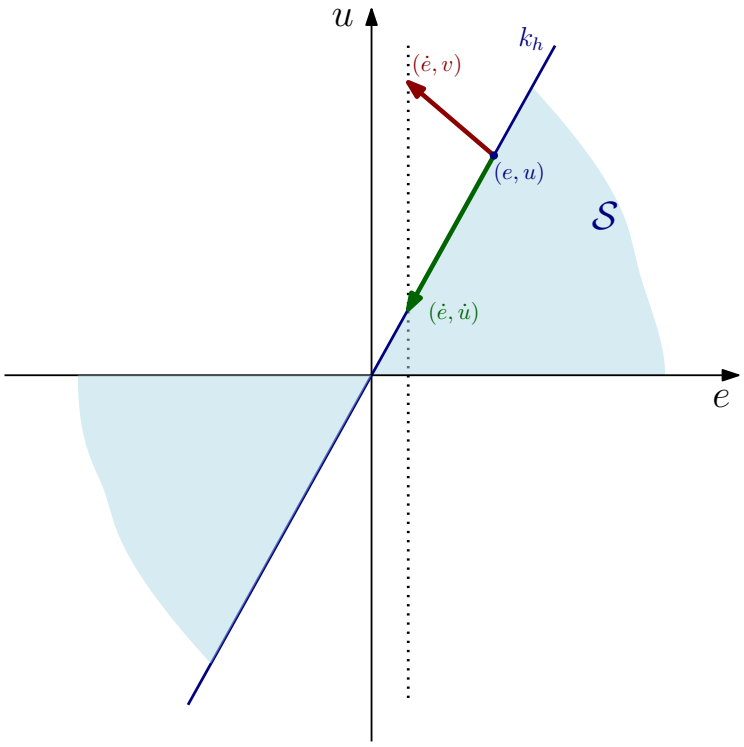
Another way to bound trajectories is through **projections**

Hybrid Integrator Gain Systems (HIGS) [1]

$$\dot{u} = \begin{cases} \omega_h e, & (\omega_h e, \dot{e}) \in T_{\mathcal{S}}(e,u) \\ k_h \dot{e}, & (\omega_h e, \dot{e}) \notin T_{\mathcal{S}}(e,u) \end{cases}$$

First-Order Projection Elements (FOPE)

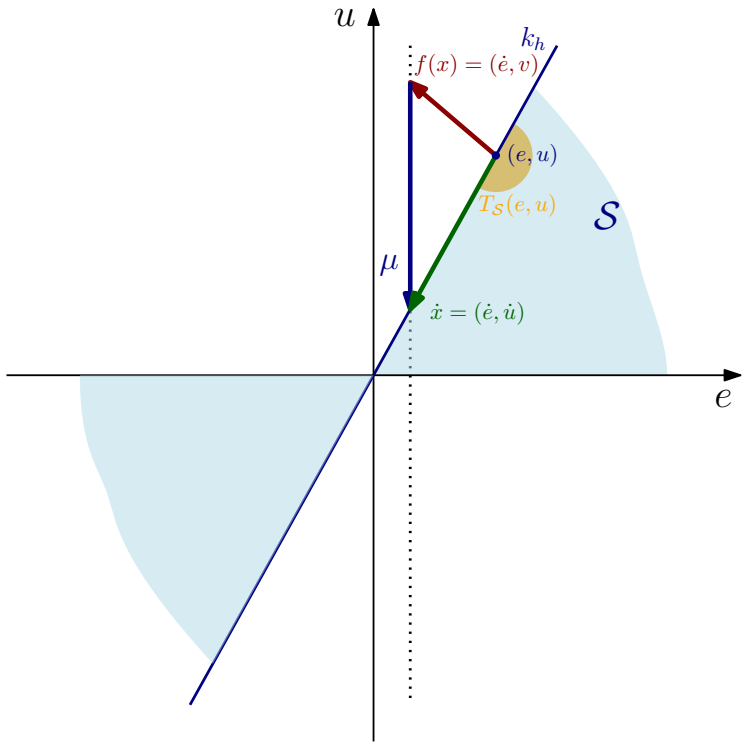
$$\dot{u} = \begin{cases} au + be, & (au + be, \dot{e}) \in T_{\mathcal{S}}(e,u) \\ k_h \dot{e}, & (au + be, \dot{e}) \notin T_{\mathcal{S}}(e,u) \end{cases}$$



[1] D.A. Deenen et al., "Projection-based integrators for improved motion control: formalization, well-posedness and stability of Hybrid Integrator-Gain Systems". *Automatica.* 2021.

Analysis is not trivial

(1) The set $\mathcal{S}$ is **not convex!**

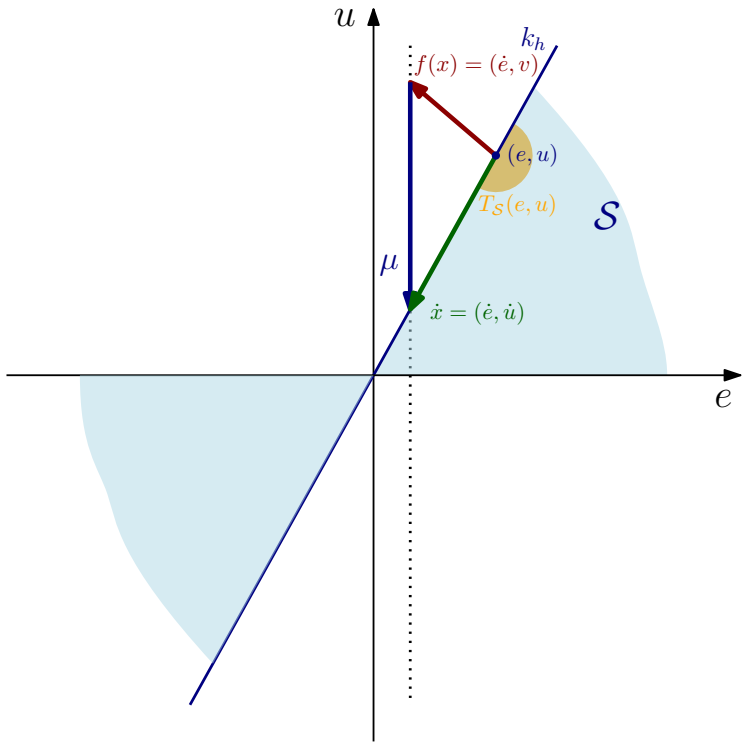


[1] W.P.M.H. Heemels, A. Tanwani, "Existence and completeness of solutions to Extended Projected Dynamical Systems and sector-bounded projection-based controllers". *IEEE L-CSS*. 2023

Analysis is not trivial

(1) The set $\mathcal{S}$ is **not convex!**

(2) **Cannot modify $\dot{e}$** $\rightarrow$ not an actual projection



[1] W.P.M.H. Heemels, A. Tanwani, "Existence and completeness of solutions to Extended Projected Dynamical Systems and sector-bounded projection-based controllers". *IEEE L-CSS*. 2023

Analysis is not trivial

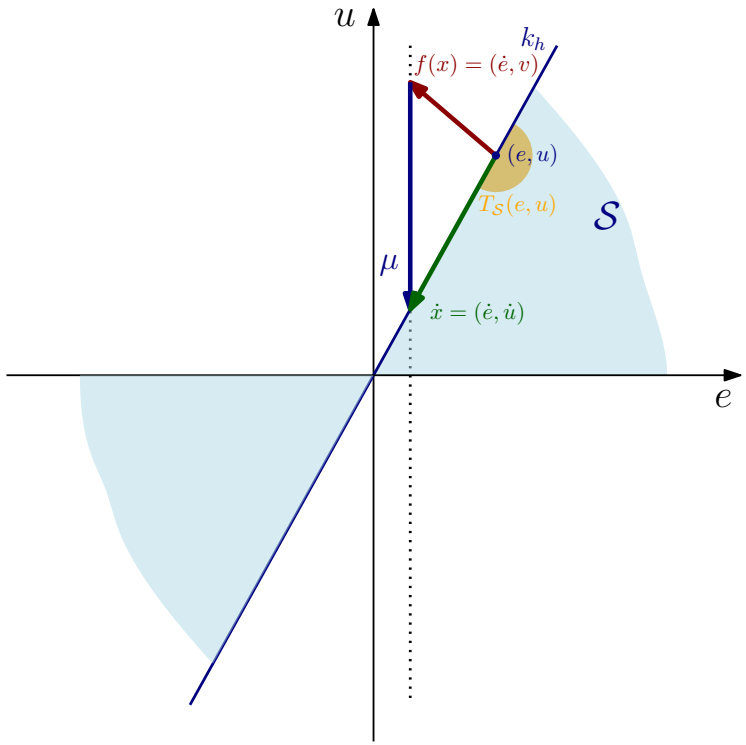
(1) The set $\mathcal{S}$ is **not convex!**

(2) **Cannot modify** $\dot{e} \rightarrow$ not an actual projection

Extended projected dynamical systems (ePDS) [1]

$$\dot{x} = \Pi_{\mathcal{S},\mathcal{E}} f(x) := f(x) + \arg\min \|\mu\|^2$$

$$\text{s.t.} \begin{cases} f(x) + \mu \in T_{\mathcal{S}}(x) \\ \boldsymbol{\mu \in \mathcal{E}} \end{cases}$$



[1] W.P.M.H. Heemels, A. Tanwani, "Existence and completeness of solutions to Extended Projected Dynamical Systems and sector-bounded projection-based controllers". *IEEE L-CSS*. 2023

A bunch of cool properties

Solutions to HIGS exist and **are complete** [1]

[1] W.P.M.H. Heemels, A. Tanwani, "Existence and completeness of solutions to Extended Projected Dynamical Systems and sector-bounded projection-based controllers". *IEEE L-CSS*. 2023

[2] S.J.A.M. van den Eijnden et al., "Hybrid Integrator-Gain Systems: a remedy for overshoot limitations in linear control?". *IEEE L-CSS*. 2020.

A bunch of cool properties

Solutions to HIGS exist and **are complete** [1]

HIGS is **incrementally stable** [2]

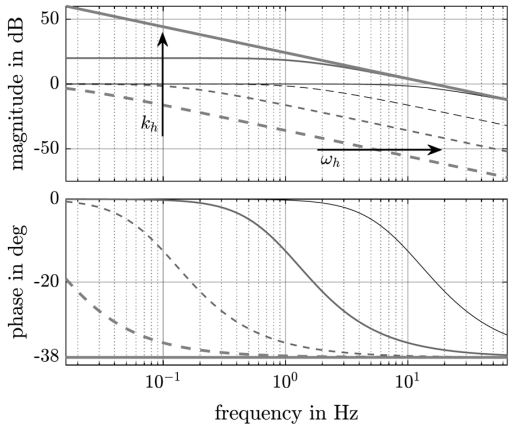
[1] W.P.M.H. Heemels, A. Tanwani, "Existence and completeness of solutions to Extended Projected Dynamical Systems and sector-bounded projection-based controllers". *IEEE L-CSS*. 2023

[2] S.J.A.M. van den Eijnden et al., "Hybrid Integrator-Gain Systems: a remedy for overshoot limitations in linear control?". *IEEE L-CSS*. 2020.

A bunch of cool properties

Solutions to HIGS exist and **are complete** [1]

HIGS is **incrementally stable** [2]



[1] W.P.M.H. Heemels, A. Tanwani, "Existence and completeness of solutions to Extended Projected Dynamical Systems and sector-bounded projection-based controllers". *IEEE L-CSS*. 2023

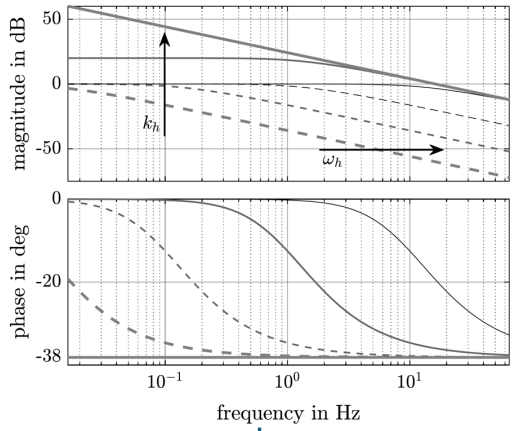
[2] S.J.A.M. van den Eijnden et al., "Hybrid Integrator-Gain Systems: a remedy for overshoot limitations in linear control?". *IEEE L-CSS*. 2020.

A bunch of cool properties

Solutions to HIGS exist and **are complete** [1]

HIGS is **incrementally stable** [2]

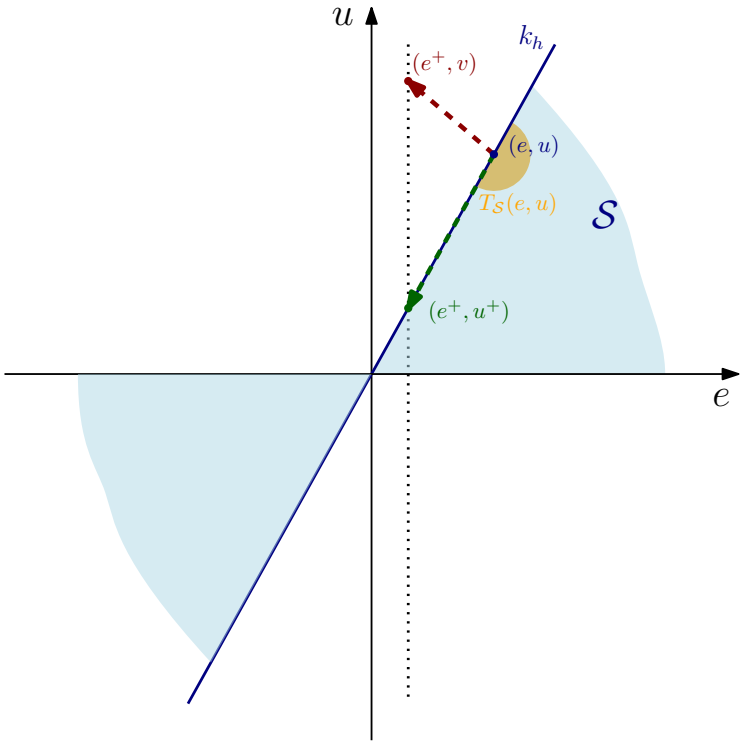
HIGS **improves performance** w.r.t. linear integrators [2]
(phase lag reduced to 38.15°)



[1] W.P.M.H. Heemels, A. Tanwani, "Existence and completeness of solutions to Extended Projected Dynamical Systems and sector-bounded projection-based controllers". *IEEE L-CSS*. 2023

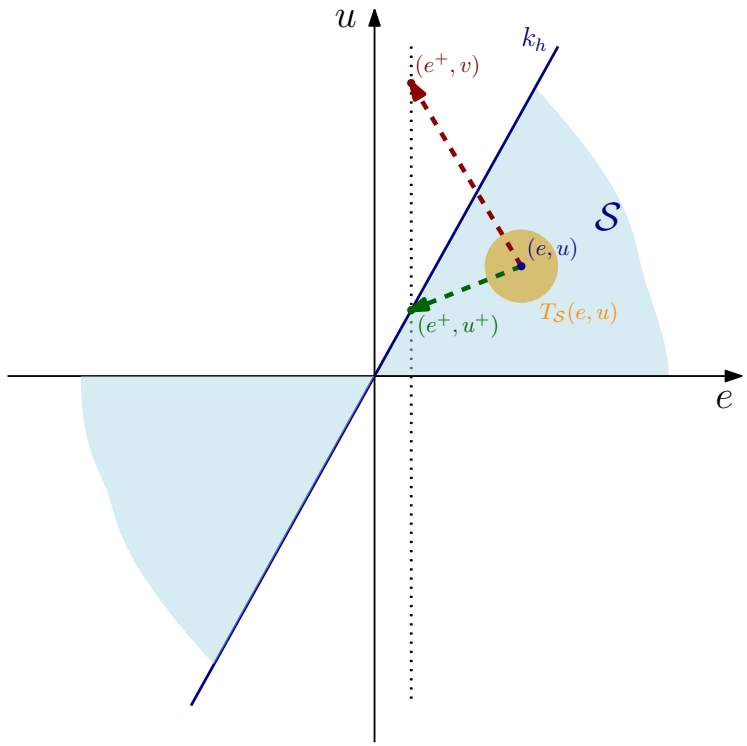
[2] S.J.A.M. van den Eijnden et al., "Hybrid Integrator-Gain Systems: a remedy for overshoot limitations in linear control?". *IEEE L-CSS*. 2020.

What happens in discrete time?



What happens in discrete time?

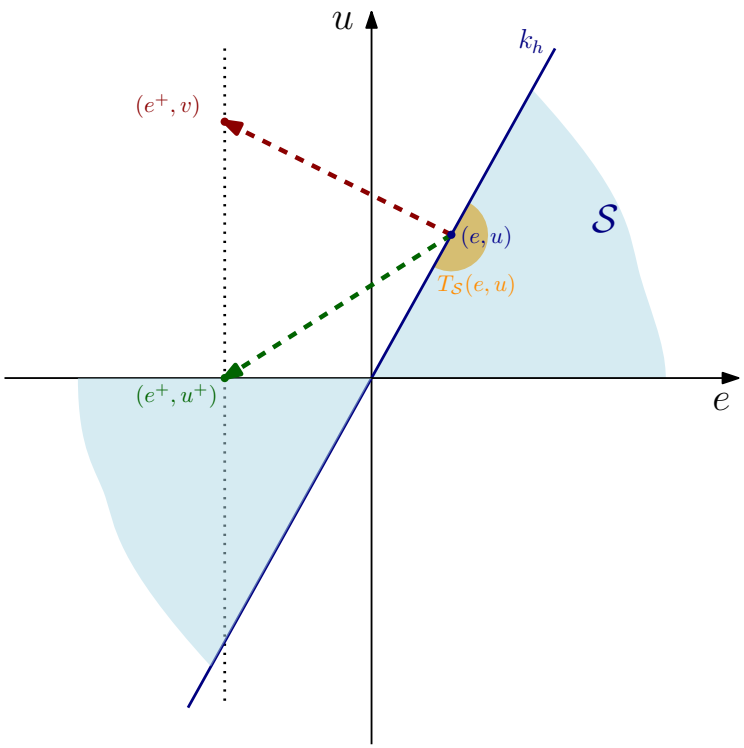
(1) Projection active even if $(e, x_c) \in \mathbf{int}(\mathcal{S})$



What happens in discrete time?

(1) Projection active even if $(e, x_c) \in \mathbf{int}(\mathcal{S})$

(2) e could **change sign**



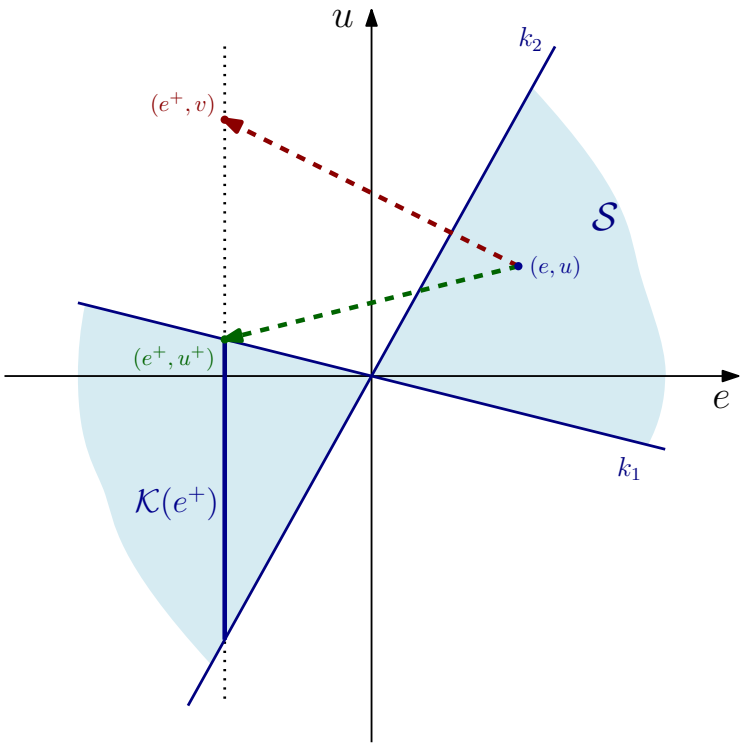
What happens in discrete time?

(1) Projection active even if $(e, x_c) \in \mathbf{int}(\mathcal{S})$

(2) e could **change sign**

Projection on an **input-dependent segment** [1]

$$u^+ = \Pi_{\mathcal{K}(e^+)}(v)$$
$$= \Pi_{\mathcal{K}(e^+)}(au + be^+)$$



What happens in discrete time?

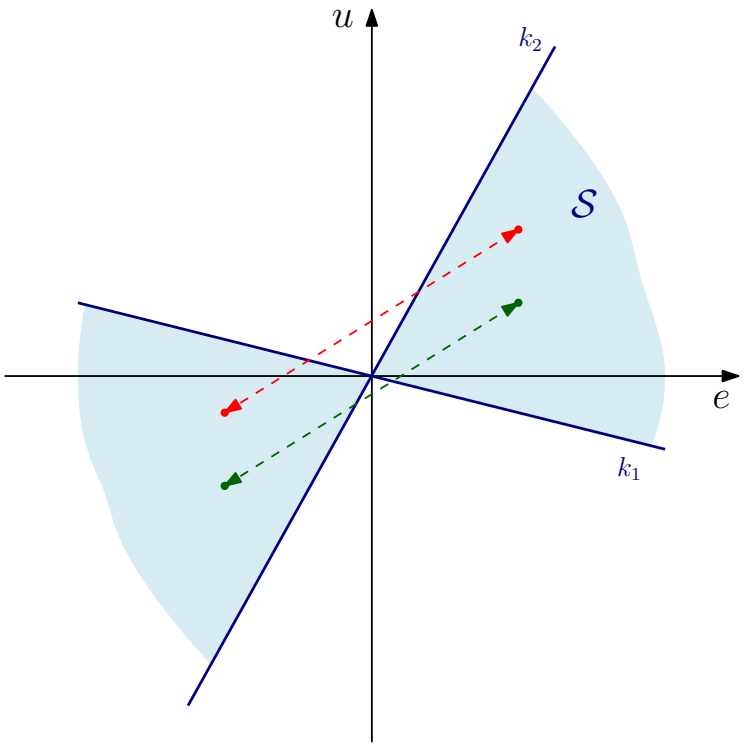
(1) Projection active even if $(e, x_c) \in \mathbf{int}(\mathcal{S})$

(2) e could **change sign**

Projection on an **input-dependent segment** [1]

$$u^+ = \Pi_{\mathcal{K}(e^+)}(v)$$
$$= \Pi_{\mathcal{K}(e^+)}(au + be^+)$$

(3) We **lose incremental stability**



What happens in discrete time?

(1) Projection active even if $(e, x_c) \in \mathbf{int}(\mathcal{S})$

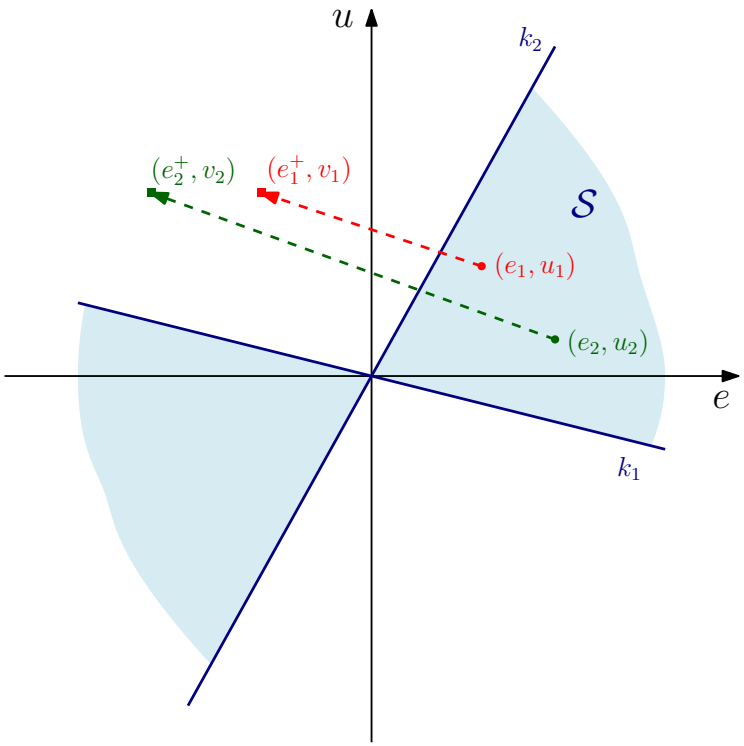
(2) e could **change sign**

Projection on an **input-dependent segment** [1]

$$u^+ = \Pi_{\mathcal{K}(e^+)}(v)$$
$$= \Pi_{\mathcal{K}(e^+)}(au + be^+)$$

(3) We **lose incremental stability**

(4) **Incremental ISS** introduces more conservatism



What happens in discrete time?

(1) Projection active even if $(e, x_c) \in \mathbf{int}(\mathcal{S})$

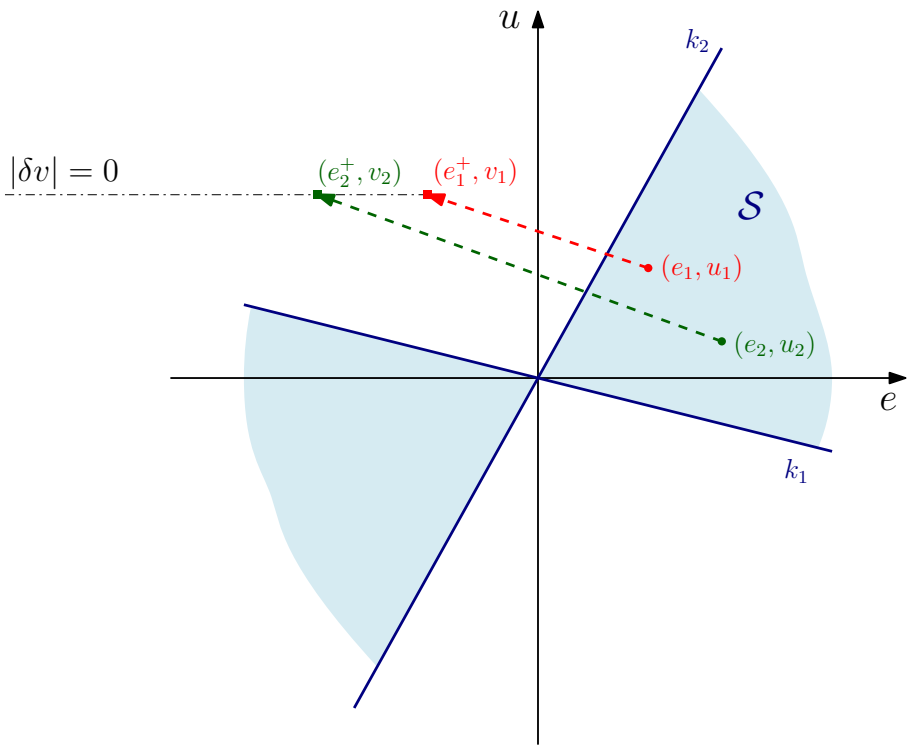
(2) e could **change sign**

Projection on an **input-dependent segment** [1]

$$u^+ = \Pi_{\mathcal{K}(e^+)}(v)$$
$$= \Pi_{\mathcal{K}(e^+)}(au + be^+)$$

(3) We **lose incremental stability**

(4) **Incremental ISS** introduces more conservatism



What happens in discrete time?

(1) Projection active even if $(e, x_c) \in \mathbf{int}(\mathcal{S})$

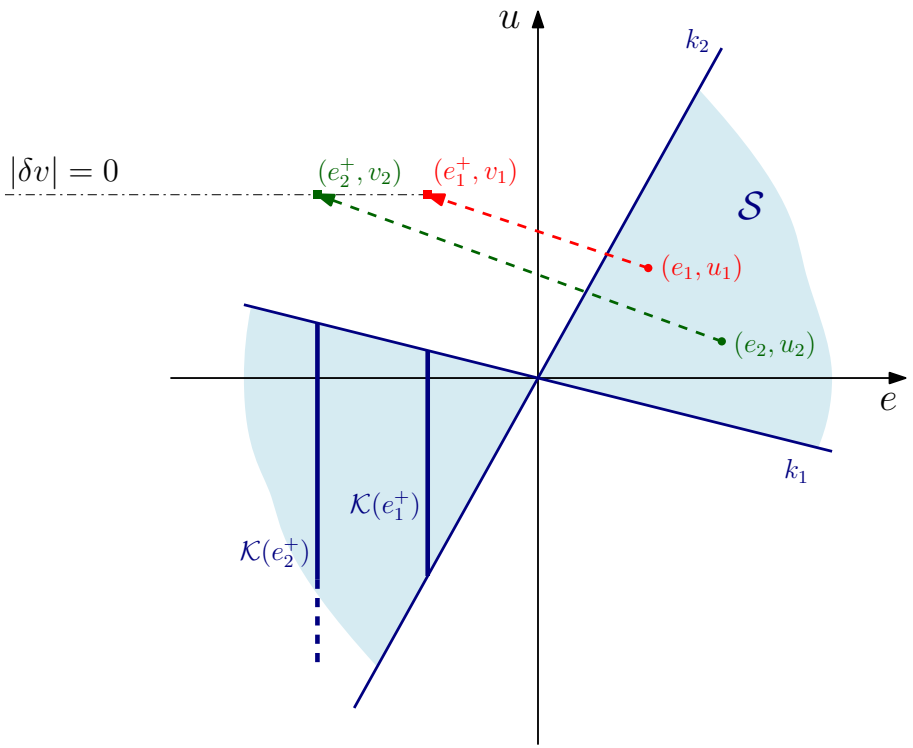
(2) e could **change sign**

Projection on an **input-dependent segment** [1]

$$u^+ = \Pi_{\mathcal{K}(e^+)}(v)$$
$$= \Pi_{\mathcal{K}(e^+)}(au + be^+)$$

(3) We **lose incremental stability**

(4) **Incremental ISS** introduces more conservatism



What happens in discrete time?

(1) Projection active even if $(e, x_c) \in \mathbf{int}(\mathcal{S})$

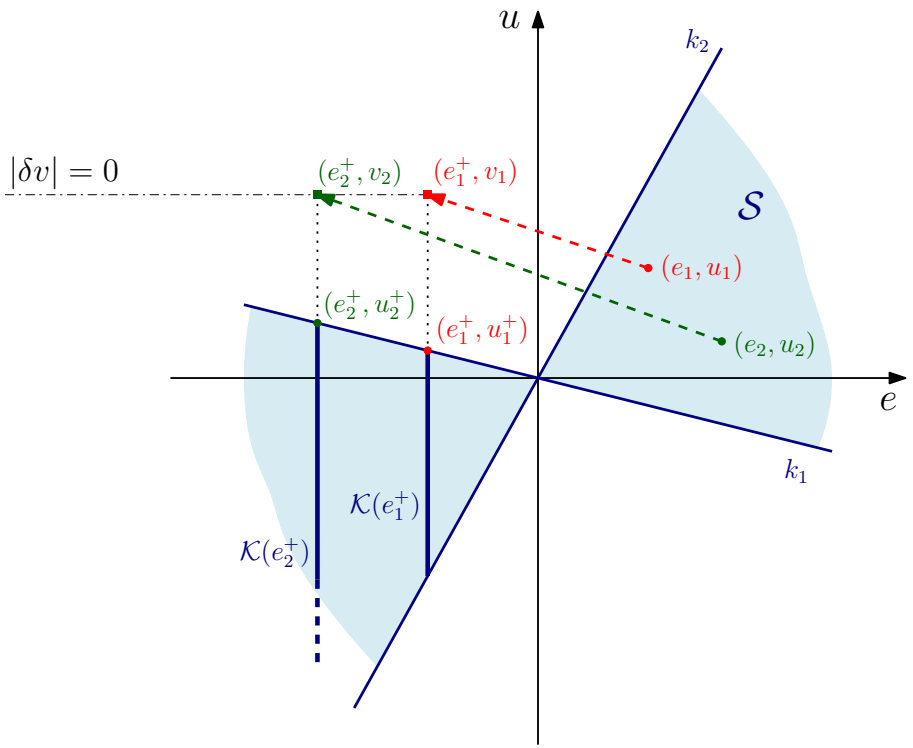
(2) e could **change sign**

Projection on an **input-dependent segment** [1]

$$u^+ = \Pi_{\mathcal{K}(e^+)}(v)$$
$$= \Pi_{\mathcal{K}(e^+)}(au + be^+)$$

(3) We **lose incremental stability**

(4) **Incremental ISS** introduces more conservatism



What happens in discrete time?

(1) Projection active even if $(e, x_c) \in \mathbf{int}(\mathcal{S})$

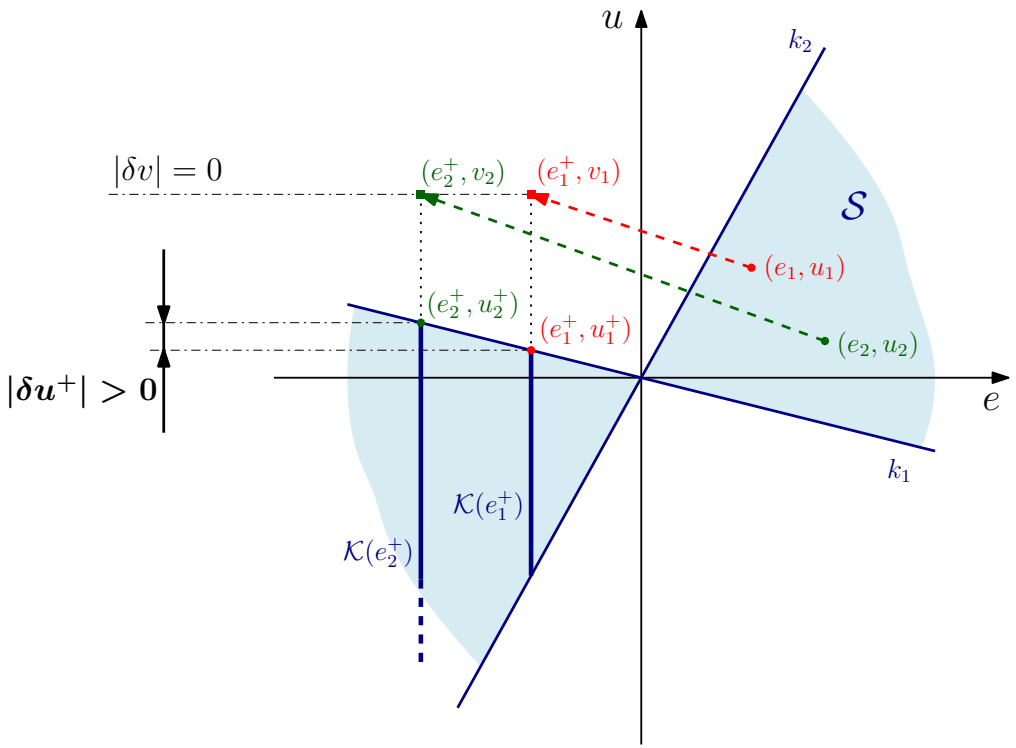
(2) e could **change sign**

Projection on an **input-dependent segment** [1]

$$u^+ = \Pi_{\mathcal{K}(e^+)}(v)$$
$$= \Pi_{\mathcal{K}(e^+)}(au + be^+)$$

(3) We **lose incremental stability**

(4) **Incremental ISS** introduces more conservatism



Some results in discrete time

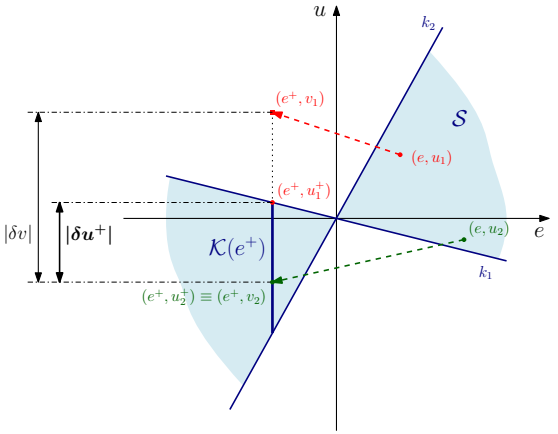
Theorem 1

If $|a| < 1$, the FOPE is incrementally exp. stable

Some results in discrete time

Theorem 1

If $|a| < 1$, the FOPE is incrementally exp. stable



Some results in discrete time

Theorem 1

If $|a| < 1$, the FOPE is incrementally exp. stable

Theorem 2

If $a = 1$ and $b \notin (2k_1, 2k_2)$, the FOPE is incrementally stable

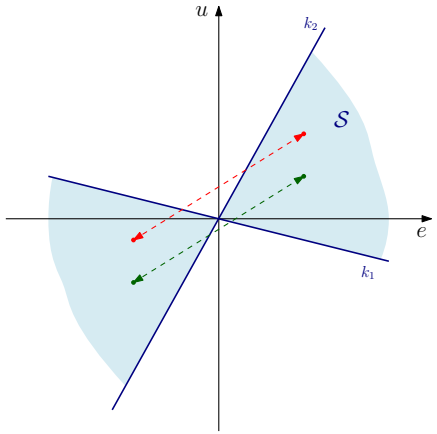
Some results in discrete time

Theorem 1

If $|a| < 1$, the FOPE is incrementally exp. stable

Theorem 2

If $a = 1$ and $b \notin (2k_1, 2k_2)$, the FOPE is incrementally stable



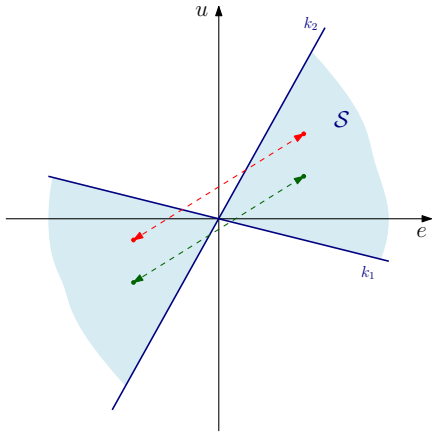
Some results in discrete time

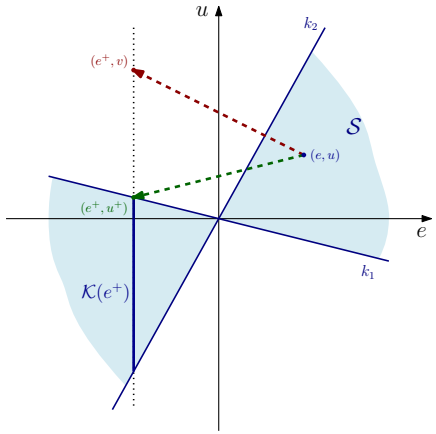
Theorem 1

If $|a| < 1$, the FOPE is incrementally exp. stable

Theorem 2

If $a = 1$ and $b \notin (2k_1, 2k_2)$, the FOPE is incrementally stable





Some results in discrete time

Theorem 1

If $|a| < 1$, the FOPE is incrementally exp. stable

Theorem 2

If $a = 1$ and $b \notin (2k_1, 2k_2)$, the FOPE is incrementally stable

Theorem 3

If $|a| < 1$, the FOPE is incrementally ISS

Some results in discrete time

Theorem 1

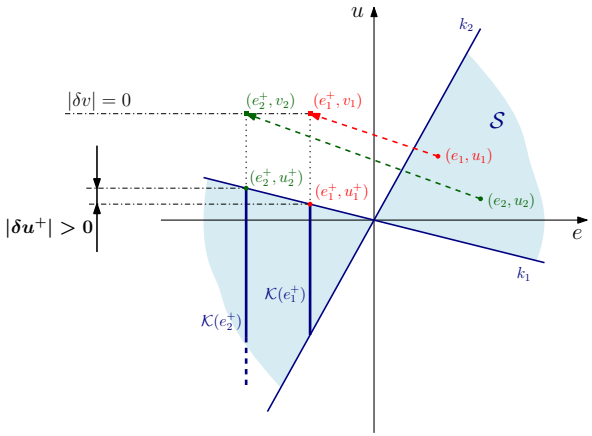
If $|a| < 1$, the FOPE is incrementally exp. stable

Theorem 2

If $a = 1$ and $b \notin (2k_1, 2k_2)$, the FOPE is incrementally stable

Theorem 3

If $|a| < 1$, the FOPE is incrementally ISS



Some results in discrete time

Theorem 1

If $|a| < 1$, the FOPE is incrementally exp. stable

Theorem 2

If $a = 1$ and $b \notin (2k_1, 2k_2)$, the FOPE is incrementally stable

Theorem 3

If $|a| < 1$, the FOPE is incrementally ISS

Theorem 4

If the FOPE is connected to a stable plant and

$$\frac{b + \max\{|k_1|, |k_2|\}}{1 - |a|} < \frac{1}{\|\mathcal{P}\|_{\ell_1}},$$

the closed loop is incrementally ISS

Some results in discrete time

Theorem 1

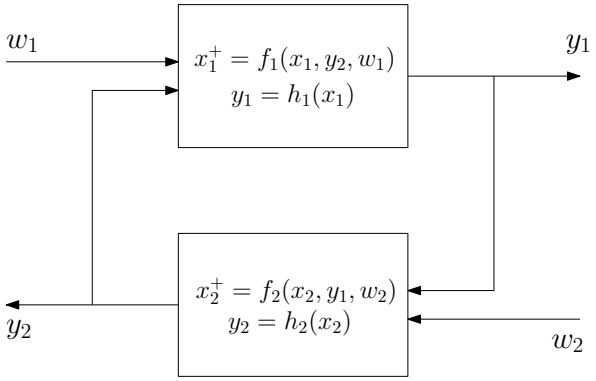
If $|a| < 1$, the FOPE is incrementally exp. stable

Theorem 2

If $a = 1$ and $b \notin (2k_1, 2k_2)$, the FOPE is incrementally stable

Theorem 3

If $|a| < 1$, the FOPE is incrementally ISS



Theorem 4

If the FOPE is connected to a stable plant and

$$\frac{b + \max\{|k_1|, |k_2|\}}{1 - |a|} < \frac{1}{\|\mathcal{P}\|_{\ell_1}},$$

the closed loop is incrementally ISS

More general analysis: ePDS, and connection to regular PDS

Extended projected dynamical systems (ePDS) [1]

$$\dot{x} = \Pi_{\mathcal{S},\mathcal{E}} f(x) := f(x) + \arg\min \|\mu\|^2$$
$$\text{s.t.} \begin{cases} f(x) + \mu \in T_{\mathcal{S}}(x) \\ \boldsymbol{\mu \in \mathcal{E}} \end{cases}$$

Oblique projected dynamical systems (oPDS) [2]

$$\dot{x} = \Pi_{\mathcal{S}}^{P(x)} f(x) := f(x) + \arg\min \|\mu\|^2_{P(x)}$$
$$\text{s.t. } f(x) + \mu \in T_{\mathcal{S}}(x)$$

$P(x)$

[1] A. Hauswirth, S. Bolognani, F. Dörfler, "Projected dynamical systems on irregular, non-Euclidean domains for nonlinear optimization". *SIAM Journal on Control and Optimization.* 2021.

More general analysis: ePDS, and connection to regular PDS

Extended projected dynamical systems (ePDS) [1]

$$\dot{x} = \Pi_{\mathcal{S},\mathcal{E}} f(x) := f(x) + \arg\min \|\mu\|^2$$
$$\text{s.t.} \begin{cases} f(x) + \mu \in T_{\mathcal{S}}(x) \\ \boldsymbol{\mu \in \mathcal{E}} \end{cases}$$

?

Oblique projected dynamical systems (oPDS) [2]

$$\dot{x} = \Pi_{\mathcal{S}}^{P(x)} f(x) := f(x) + \arg\min \|\mu\|^2_{P(x)}$$
$$\text{s.t. } f(x) + \mu \in T_{\mathcal{S}}(x)$$

$P(x)$

[1] A. Hauswirth, S. Bolognani, F. Dörfler, "Projected dynamical systems on irregular, non-Euclidean domains for nonlinear optimization". *SIAM Journal on Control and Optimization.* 2021.

More general analysis: ePDS, and connection to regular PDS

Extended projected dynamical systems (ePDS) [1]

$$\dot{x} = \Pi_{\mathcal{S},\mathcal{E}} f(x) := f(x) + \arg\min \|\mu\|^2$$
$$\text{s.t.} \begin{cases} f(x) + \mu \in T_{\mathcal{S}}(x) \\ \boldsymbol{\mu \in \mathcal{E}} \end{cases}$$

?

Oblique projected dynamical systems (oPDS) [2]

$$\dot{x} = \Pi_{\mathcal{S}}^{P(x)} f(x) := f(x) + \arg\min \|\mu\|^2_{P(x)}$$
$$\text{s.t. } f(x) + \mu \in T_{\mathcal{S}}(x)$$

- First result: sufficient conditions for existence and completeness of solutions
- Second result: a norm matrix $P(x)$ for equivalence can always be defined
  - For nonsmooth sets, even for relatively simple examples, the norm is discontinuous
  - [1] proves completeness of solutions for continuous norms, but we proved the result in this particular discontinuous case → possible extension of the result

[1] A. Hauswirth, S. Bolognani, F. Dörfler, "Projected dynamical systems on irregular, non-Euclidean domains for nonlinear optimization". *SIAM Journal on Control and Optimization.* 2021.

More general analysis: ePDS, and connection to regular PDS

Extended projected dynamical systems (ePDS) [1]

$$\dot{x} = \Pi_{\mathcal{S},\mathcal{E}} f(x) := f(x) + \arg\min \|\mu\|^2$$
$$\text{s.t.} \begin{cases} f(x) + \mu \in T_{\mathcal{S}}(x) \\ \boldsymbol{\mu \in \mathcal{E}} \end{cases}$$

?

Oblique projected dynamical systems (oPDS) [2]

$$\dot{x} = \Pi_{\mathcal{S}}^{P(x)} f(x) := f(x) + \arg\min \|\mu\|^2_{P(x)}$$
$$\text{s.t. } f(x) + \mu \in T_{\mathcal{S}}(x)$$

- First result: sufficient conditions for existence and completeness of solutions
- Second result: a norm matrix $P(x)$ for equivalence can always be defined
  - For nonsmooth sets, even for relatively simple examples, the norm is discontinuous
  - [1] proves completeness of solutions for continuous norms, but we proved the result in this particular discontinuous case → possible extension of the result

Application to feedback optimization with state constraints

[1] A. Hauswirth, S. Bolognani, F. Dörfler, "Projected dynamical systems on irregular, non-Euclidean domains for nonlinear optimization". *SIAM Journal on Control and Optimization.* 2021.

Other (current) projects

Other (current) projects

- Equivalence between ePDS and higher order sweeping processes?

Other (current) projects

- Equivalence between ePDS and higher order sweeping processes?
- Connections between PDS and anti-windup control?

Other (current) projects

- Equivalence between ePDS and higher order sweeping processes?
- Connections between PDS and anti-windup control?
- In general: what is the relation among the many nonsmooth, safety-critical controllers?

Other (current) projects

- Equivalence between ePDS and higher order sweeping processes?
- Connections between PDS and anti-windup control?
- In general: what is the relation among the many nonsmooth, safety-critical controllers?

- Modelling of partially elastic impacts as a “hybrid singular perturbation” of inelastic impacts, for control purposes?

Other (older) projects

Other (older) projects

Presented next:

Other (older) projects

Presented next:

- Stabilization of bipedal gait using hybrid dynamics and reference spreading

Other (older) projects

Presented next:

- Stabilization of bipedal gait using hybrid dynamics and reference spreading

Not included:

Other (older) projects

Presented next:

- Stabilization of bipedal gait using hybrid dynamics and reference spreading

Not included:

- Global uniform asymptotic/finite-time synchronization of nonlinear oscillators using hybrid coupling

Other (older) projects

Presented next:

- Stabilization of bipedal gait using hybrid dynamics and reference spreading

Not included:

- Global uniform asymptotic/finite-time synchronization of nonlinear oscillators using hybrid coupling
- Use of reset control in the adaptive control framework

Other (older) projects

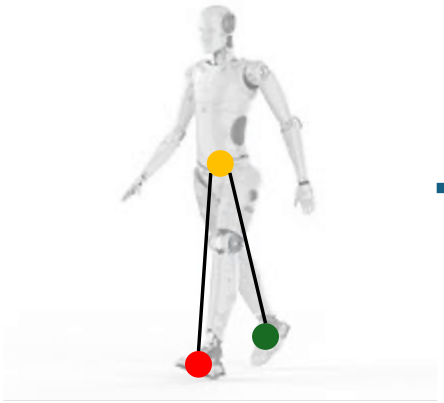
Presented next:

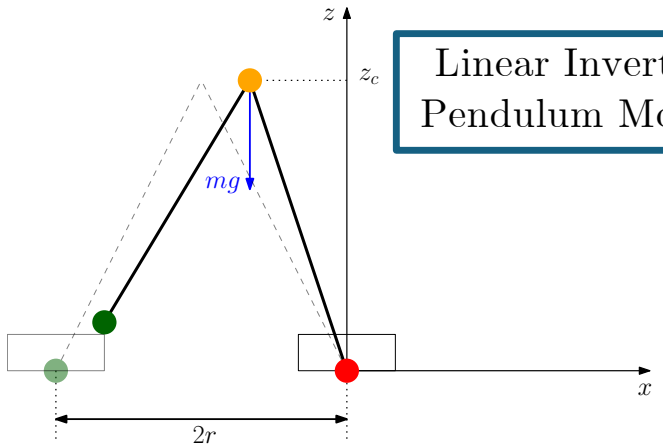
- Stabilization of bipedal gait using hybrid dynamics and reference spreading

Not included:

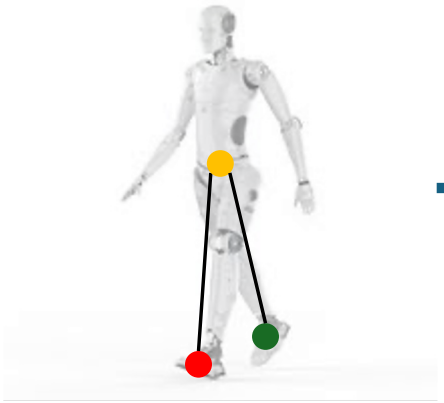
- Global uniform asymptotic/finite-time synchronization of nonlinear oscillators using hybrid coupling
- Use of reset control in the adaptive control framework
- Distributed observers with different timescales (sampled-data systems with hybrid systems formalism)

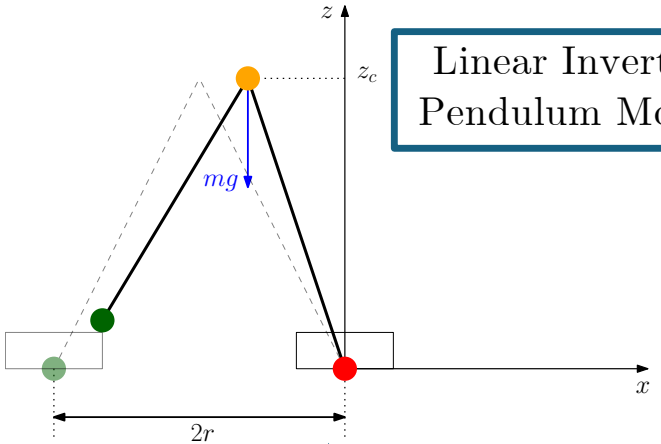
Bipedal gait stabilization using reference spreading



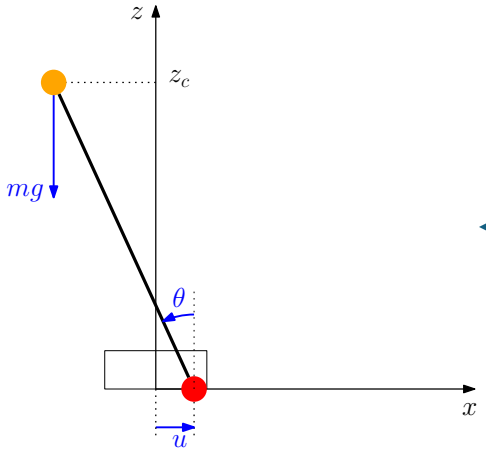


Bipedal gait stabilization using reference spreading



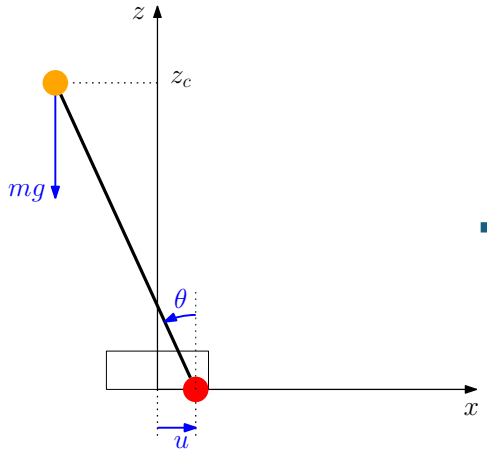


Linear Inverted Pendulum Model



Control action: moving the pivot

Bipedal gait stabilization using reference spreading



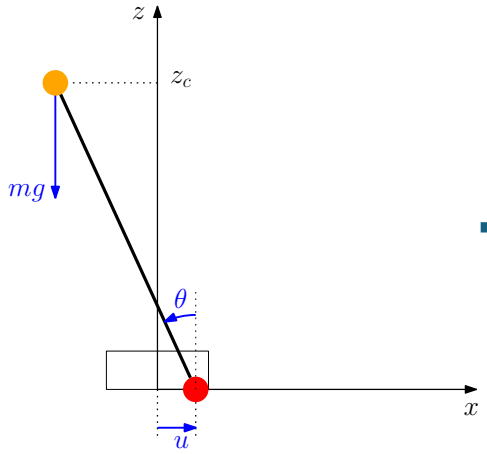
Equations of motion:

$$\ddot{\theta} = -\frac{g}{z_c}\sin\theta = -\omega^2 \sin\theta$$

Small angles linearization:

$$\ddot{x}_p = -\omega^2(x_p - u)$$

Bipedal gait stabilization using reference spreading



Equations of motion:

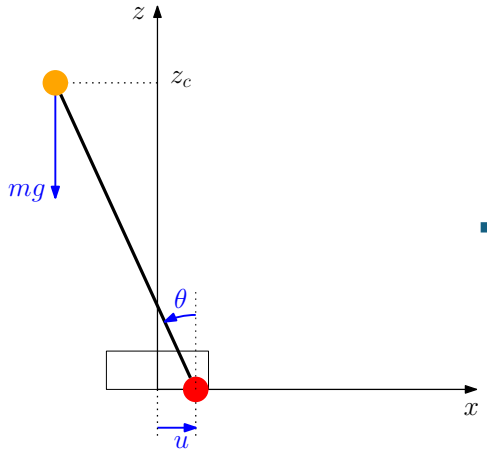
$$\ddot{\theta} = -\frac{g}{z_c}\sin\theta = -\omega^2 \sin\theta$$

Small angles linearization:

$$\ddot{x}_p = -\omega^2(x_p - u)$$

However: when the grounded foot changes, the pivot "jumps" forwards

Bipedal gait stabilization using reference spreading



Equations of motion:

$$\ddot{\theta} = -\frac{g}{z_c}\sin\theta = -\omega^2 \sin\theta$$

Small angles linearization:

$$\ddot{x}_p = -\omega^2(x_p - u)$$

However: when the grounded foot changes, the pivot "jumps" forwards

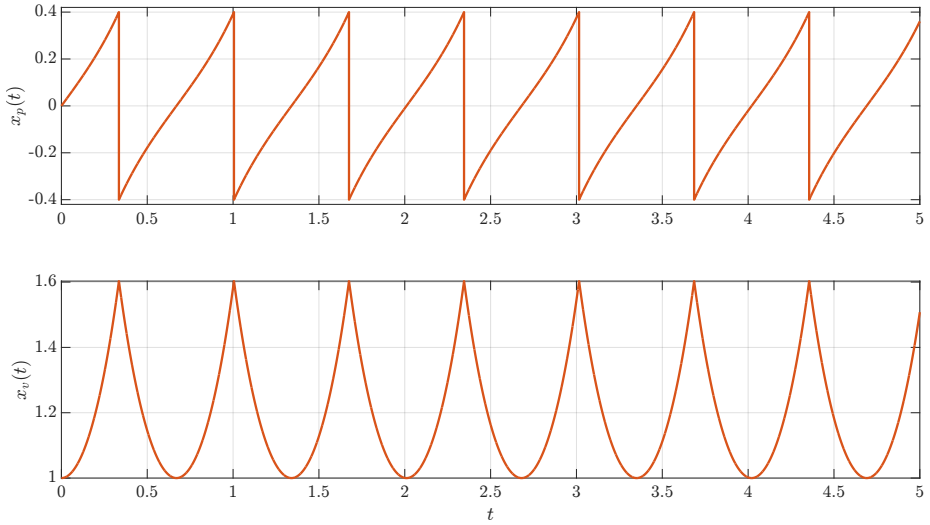
Hybrid dynamical system!

$$x = \begin{bmatrix} x_p \\ x_v \end{bmatrix} \quad \rightarrow \quad \begin{cases} \dot{x} = \begin{bmatrix} 0 & 1 \\ \omega^2 & 0 \end{bmatrix} x + \begin{bmatrix} 0 \\ -\omega^2 \end{bmatrix} u, & x_p \in [-r, r] \\ x^+ = x - \begin{bmatrix} 2r \\ 0 \end{bmatrix}, & x_p = r \end{cases}$$

Bipedal gait stabilization using reference spreading

The hybrid LIPM has a periodic trajectory (no inputs)

Goal: stabilize the gait on the periodic reference



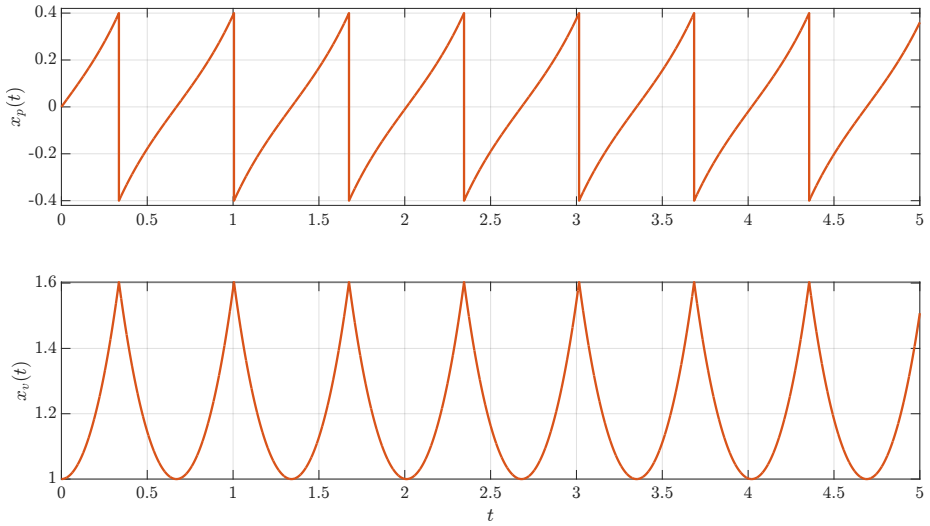
Bipedal gait stabilization using reference spreading

The hybrid LIPM has a periodic trajectory (no inputs)

Goal: stabilize the gait on the periodic reference

First idea:
$e = x - x_{\text{ref}}$

Issues when robot and reference are **standing on different feet**



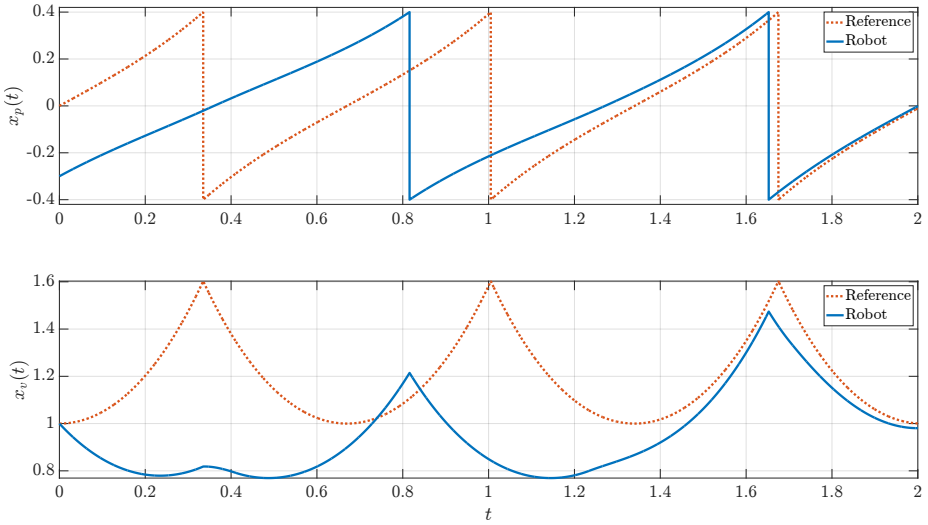
Bipedal gait stabilization using reference spreading

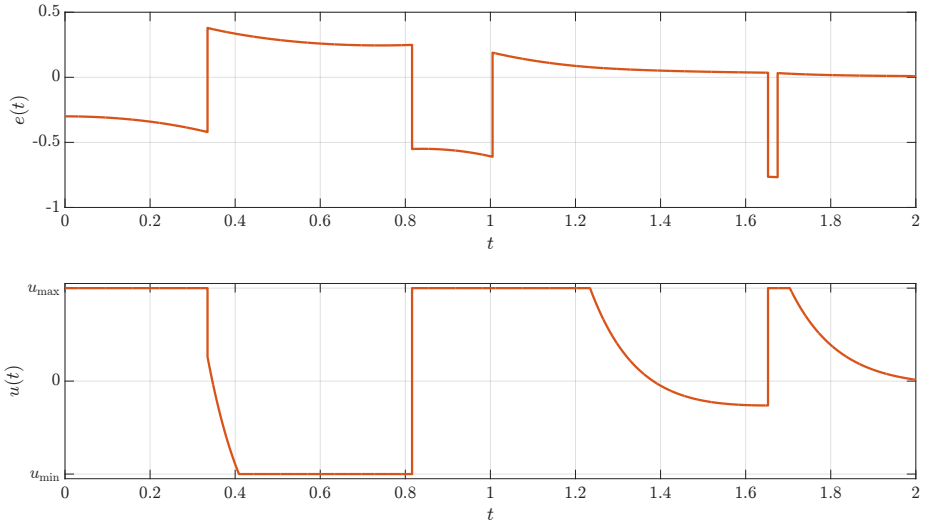
The hybrid LIPM has a periodic trajectory (no inputs)

Goal: stabilize the gait on the periodic reference

First idea:
$e = x - x_{\text{ref}}$

Issues when robot and reference are **standing on different feet**





Bipedal gait stabilization using reference spreading

New idea:

- Introduce a timer state to parametrize the reference
  - Reset the timer when the robot changes foot

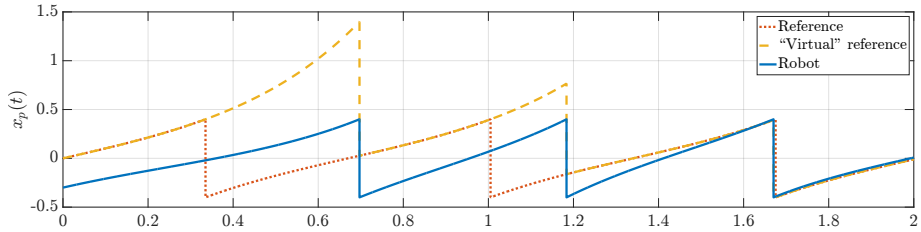
$$\begin{cases} \dot{\tau} = 1, & x_p \in [-r, r] \\ \tau^+ = \tau - T, & x_p = r \end{cases}$$

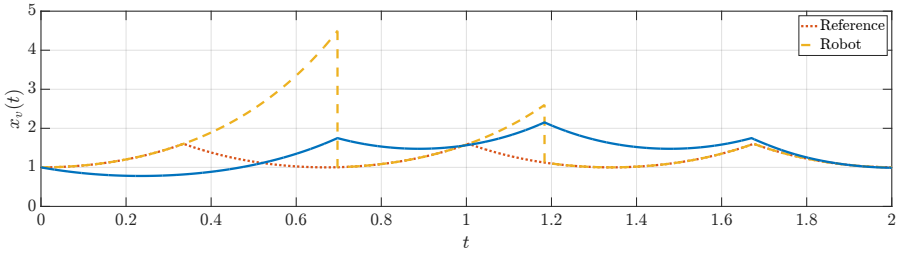
Bipedal gait stabilization using reference spreading

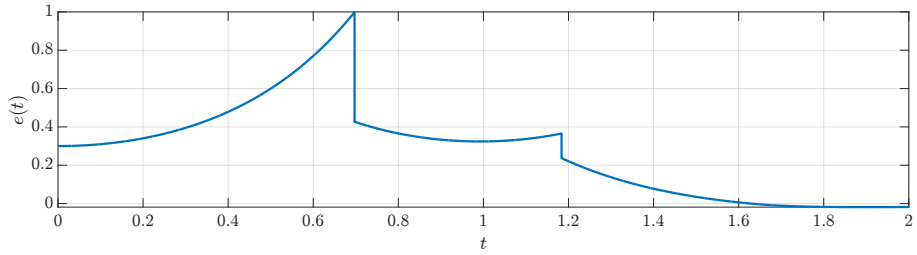
New idea:

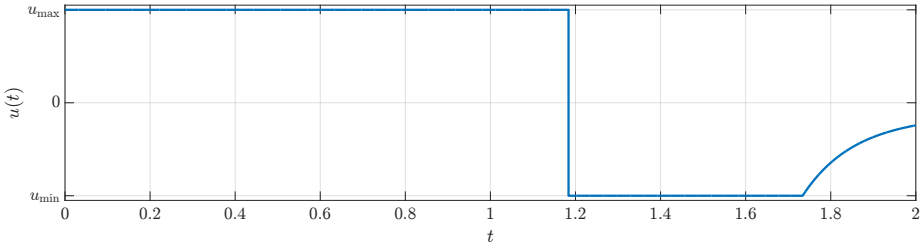
- Introduce a timer state to parametrize the reference
  - Reset the timer when the robot changes foot

$$\begin{cases} \dot{\tau} = 1, & x_p \in [-r, r] \\ \tau^+ = \tau - T, & x_p = r \end{cases}$$







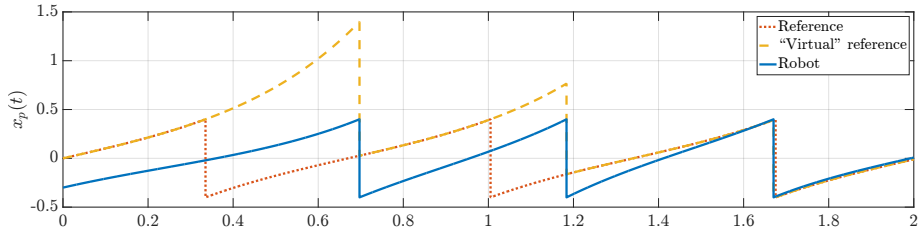


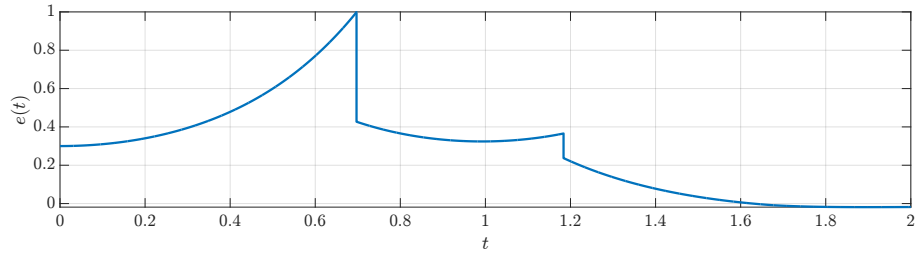
Bipedal gait stabilization using reference spreading

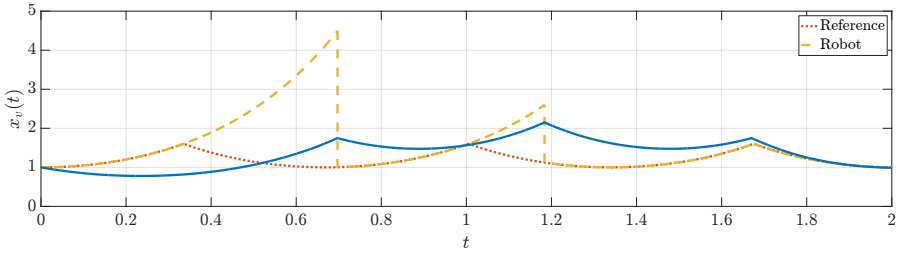
New idea:

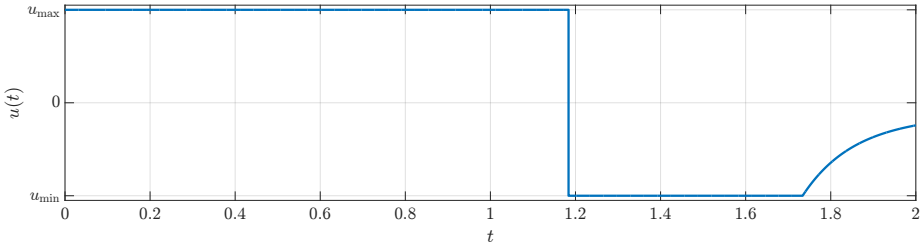
- Introduce a timer state to parametrize the reference
  - Reset the timer when the robot changes foot

$$\begin{cases} \dot{\tau} = 1, & x_p \in [-r, r] \\ \tau^+ = \tau - T, & x_p = r \end{cases}$$









With such a "regular" error trajectory, the control synthesis problem becomes easier!
We defined **LMIs** to synthesize a stabilizing saturated full-state feedback, and we identified **necessary and sufficient conditions for feasibility**