Control Barrier Functions for Nonlinear System Safety Control

Jason Jangho Choi

Berkeley
Mechanical Engineering

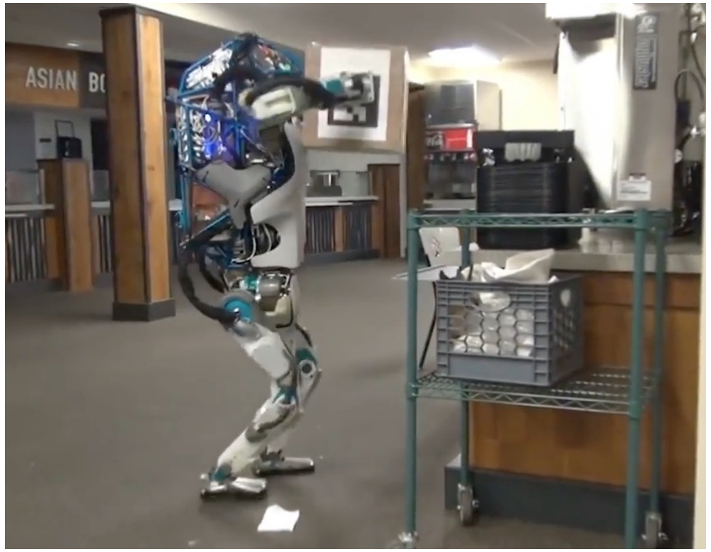
HYBRID ROBOTICS
GROUP @ BERKELEY

Hybrid Systems
Laboratory

Mar. 22, 2023
@EECS 206B

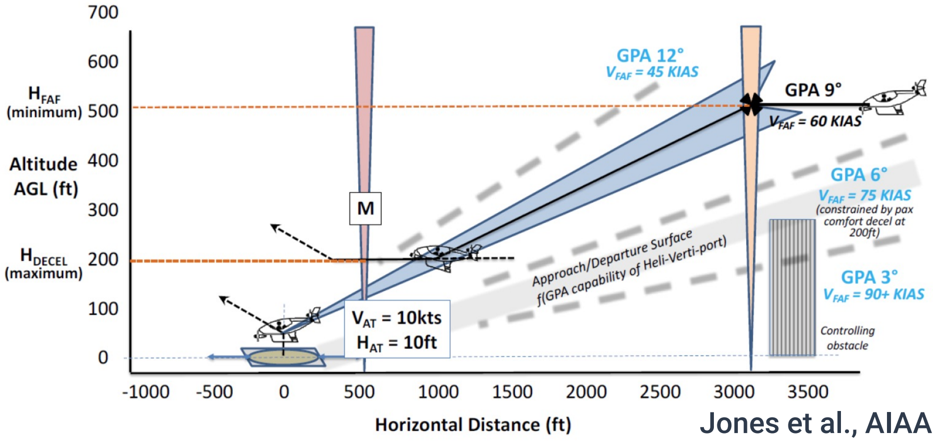
Big gap between what state-of-the-art controllers

can achieve and what they ***guarantee***…



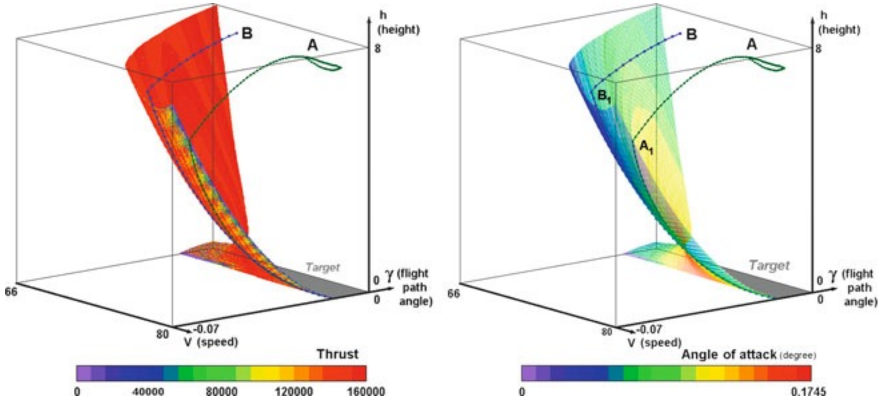
Many real-world applications are *safety-critical!*

Aircraft control

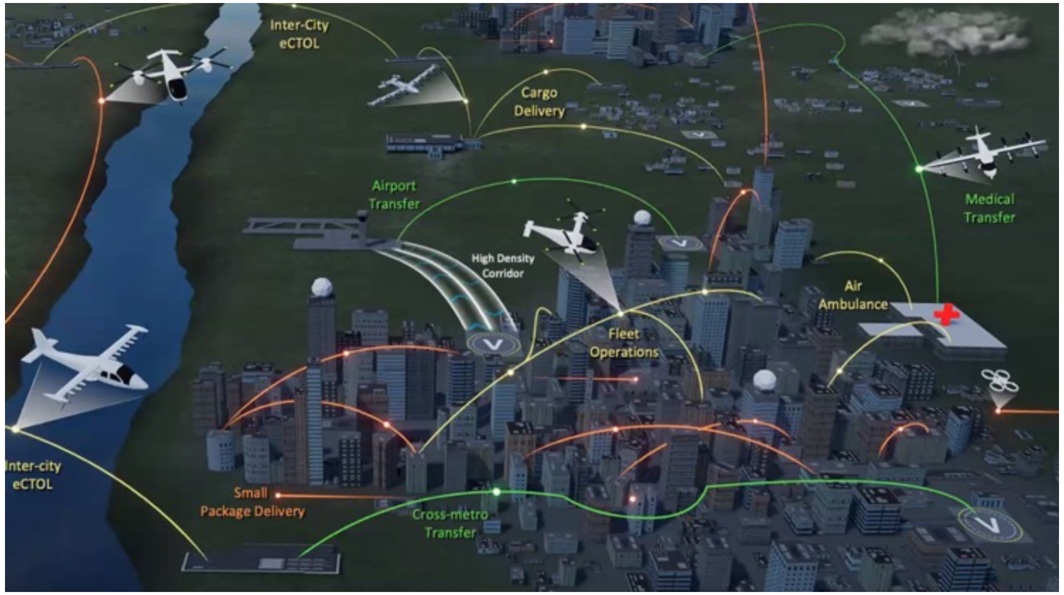


Jones et al., AIAA'22

Fig. 13 Heliport Approach Maneuver



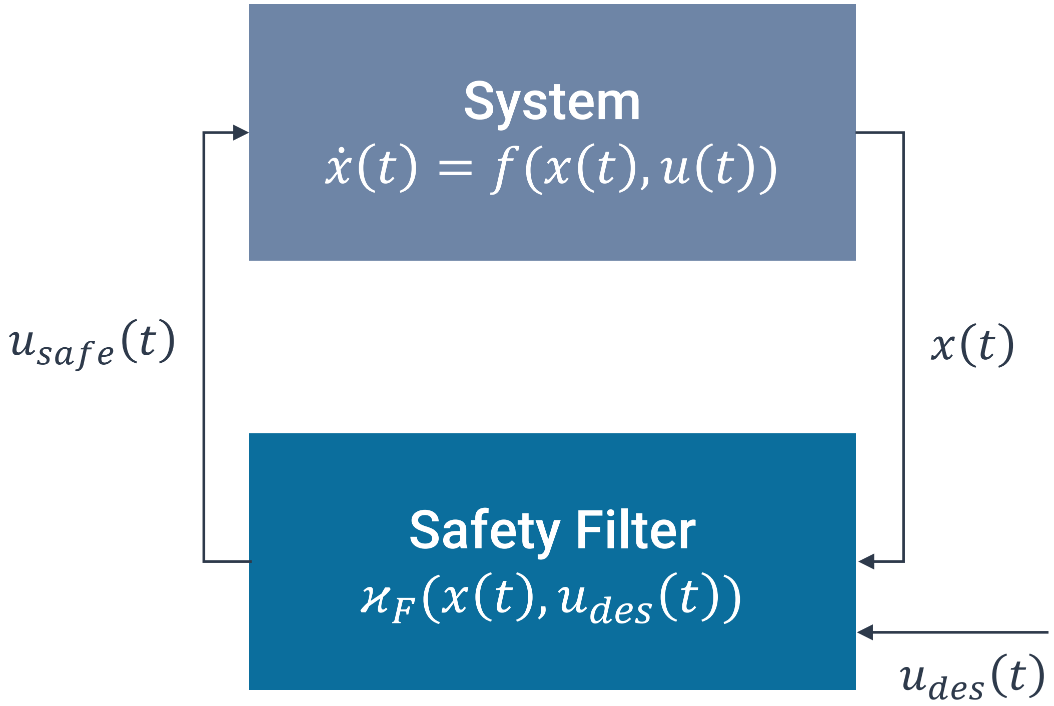
Aubin et al., 2011



Source: NASA

Advanced Air Mobility in
safety-critical and highly congested environments

Safety Filter – Basic concept



System

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

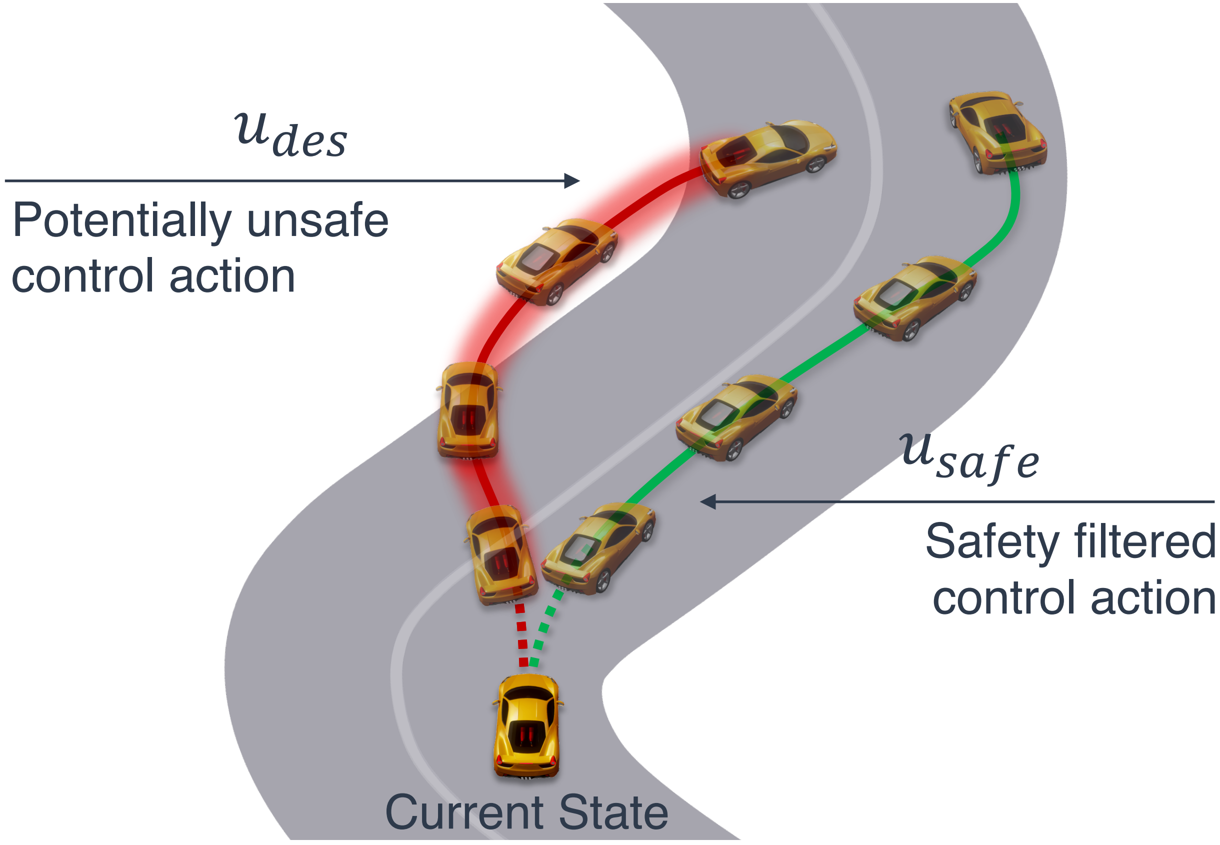
Safety Filter

$\varkappa_F(x(t), u_{des}(t))$

$u_{safe}(t)$

$x(t)$

$u_{des}(t)$



u_{des}

Potentially unsafe control action

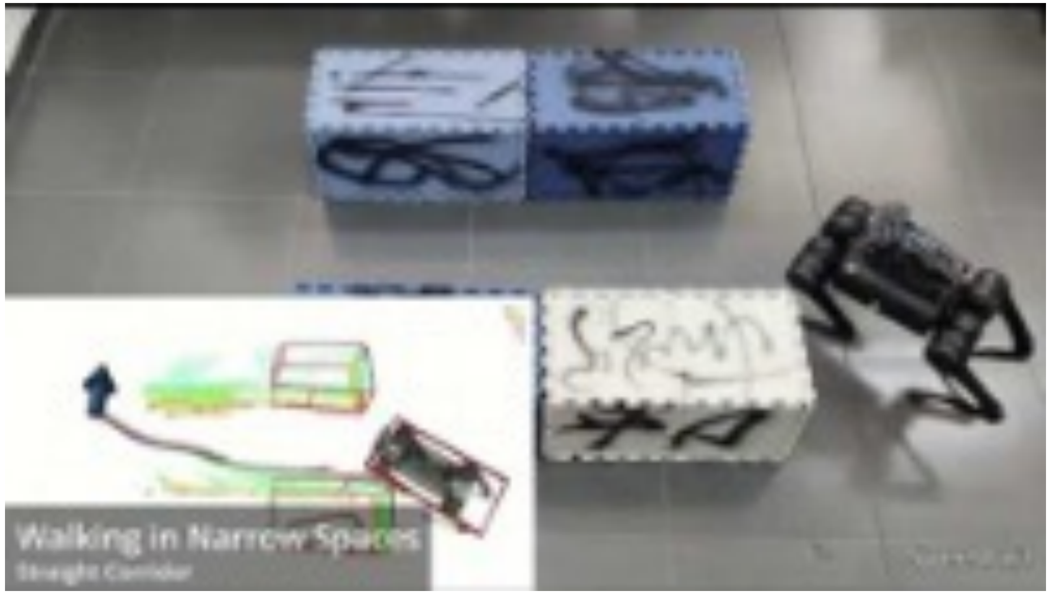
u_{safe}

Safety filtered control action

Current State

Wabersich*, Taylor*, Choi* et al., Data-Driven Safety Filters, Under review

Control Barrier Functions for safety control

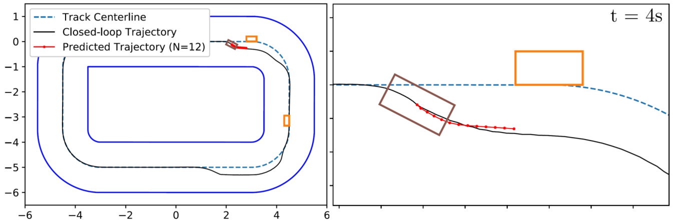


Liao et al., Arxiv’22

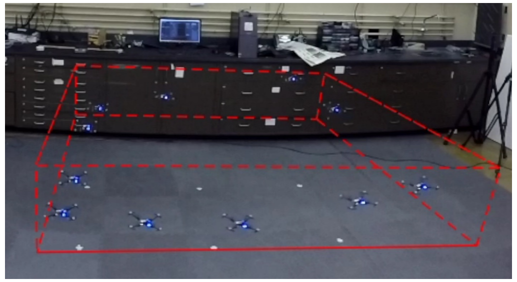
Grandia et al., ICRA’21

Control Barrier Functions for safety control

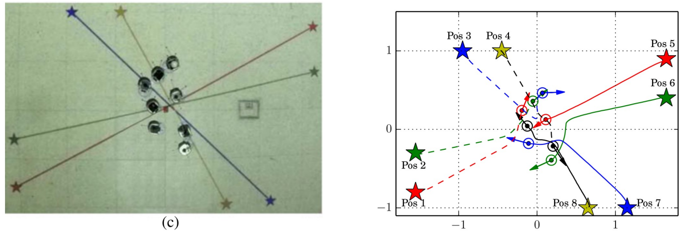
Autonomous mobile robots



Zeng et al., ACC’21[1]



Xu et al., ICRA’18[2]

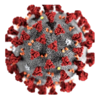


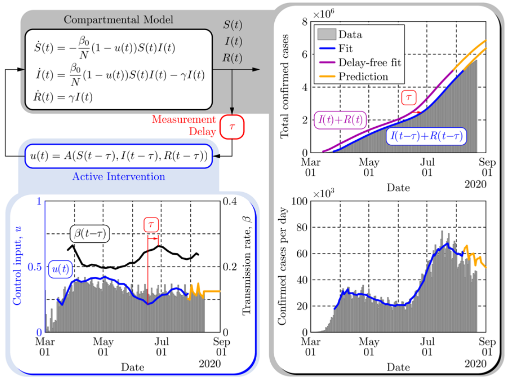
Wang et al., TRo’17, ICRA’17

Manipulators

Singletary et al., Arxiv 2022

Infection control





Moln´ar et al., L-CSS’21

Contents

Part 1. Background

- Safety Filter
- Control Invariance

Part 2. Introduction to CBF

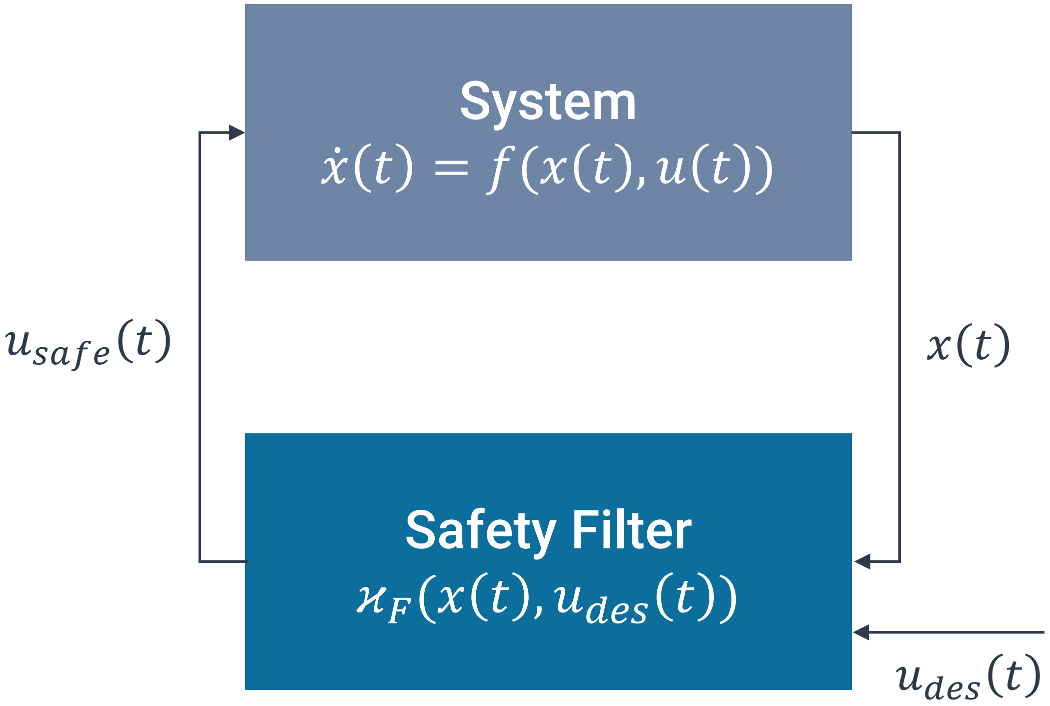
- Control Barrier Function
- Comparison principle perspective.
- Nagumo’s theorem perspective.
- CBF-based safety filter for control-affine systems: CBF-QP

Part 3. Alternative methods for safety control

- Hamilton-Jacobi Reachability
- Model Predictive Control

1. Background

Safety Filter – Basic concept



System

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

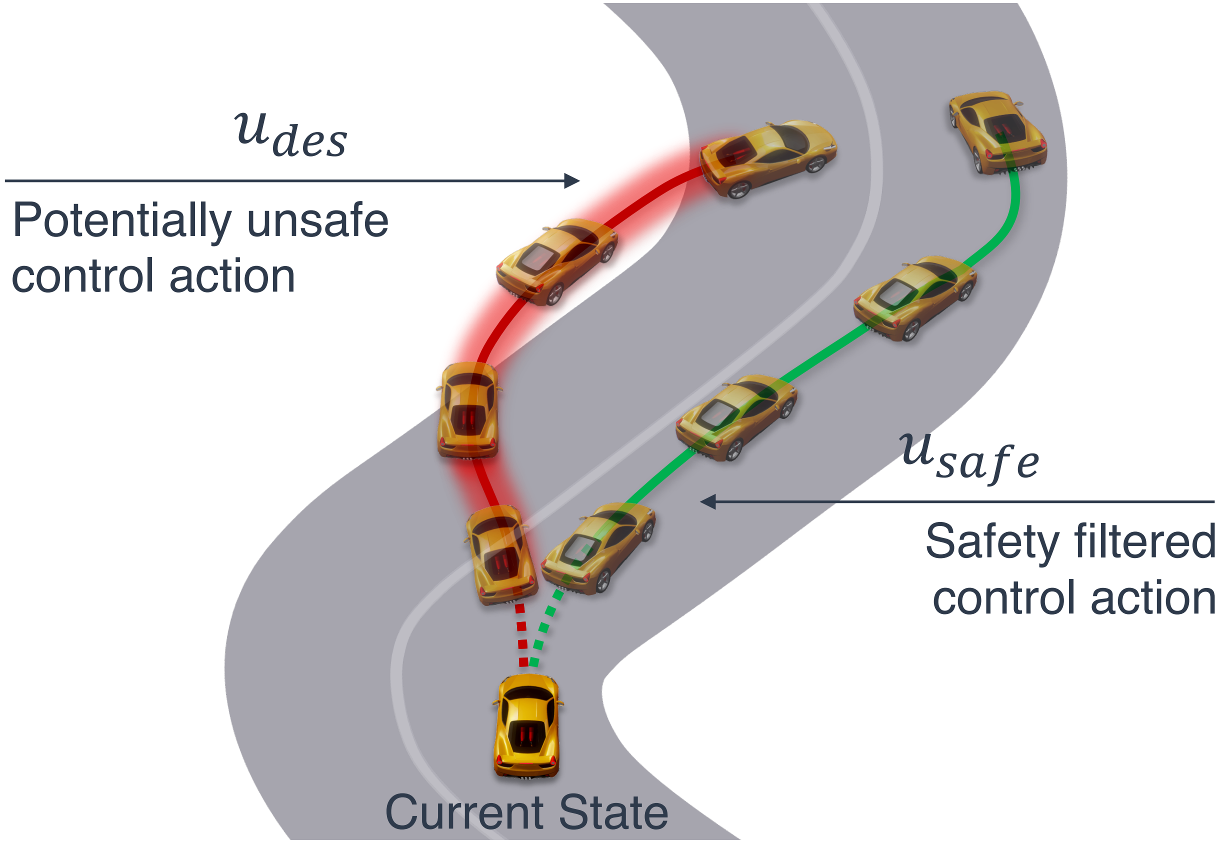
$x(t)$

$u_{safe}(t)$

Safety Filter

$\varkappa_F(x(t), u_{des}(t))$

$u_{des}(t)$



u_{des}

Potentially unsafe control action

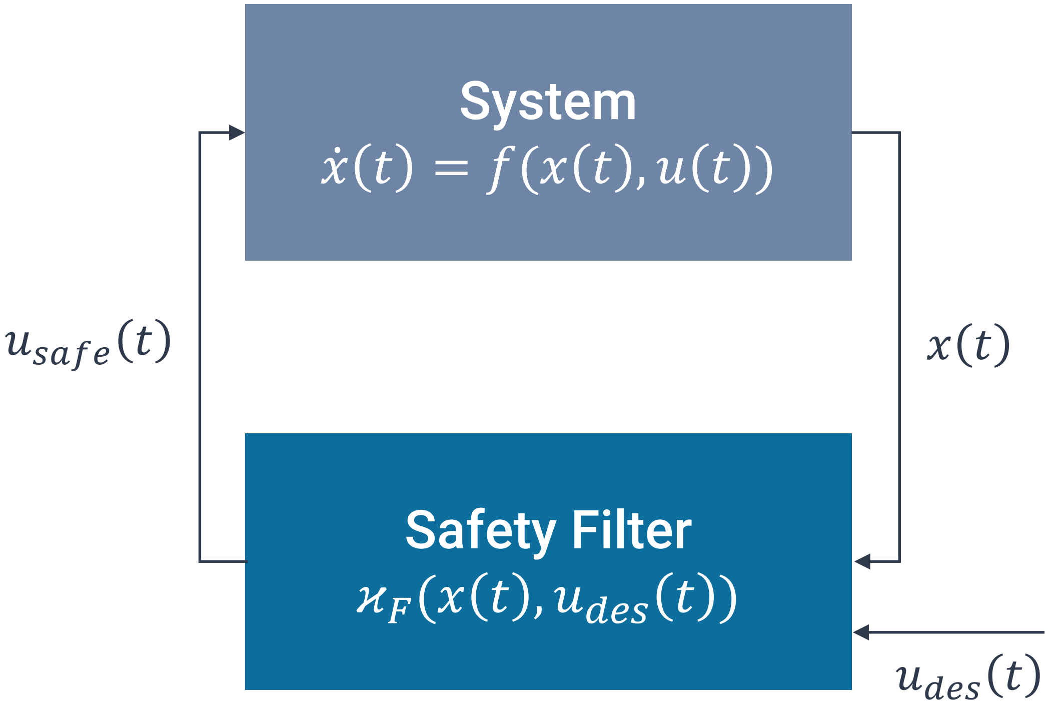
u_{safe}

Safety filtered control action

Current State

Wabersich*, Taylor*, Choi* et al., Data-Driven Safety Filters, Under review

Safety Filter – Basic concept



$$u(\cdot) = \underset{v(\cdot)}{\operatorname{argmin}} \quad \underbrace{\int_{t=0}^{\infty} \|v(t) - u_{\mathrm{des}}(t)\| \; \mathrm{d}t}_{\textbf{Minimum deviation from desired control}}$$

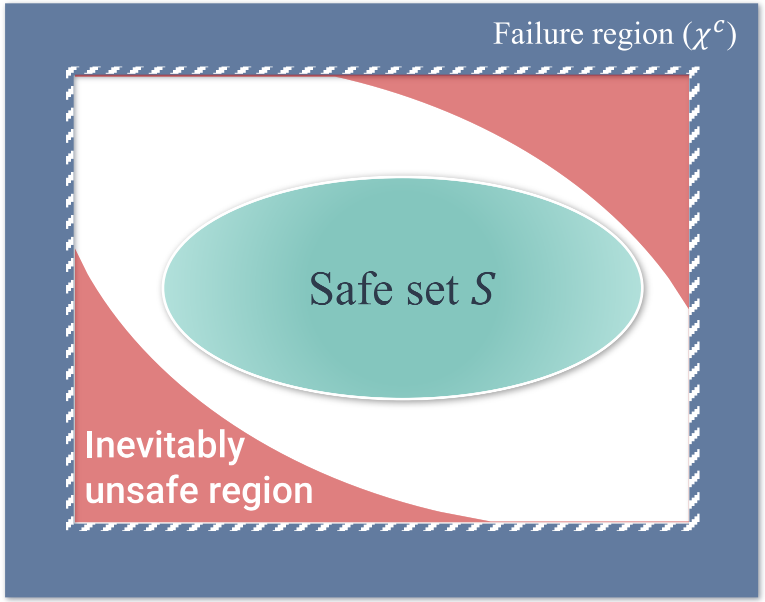
$$\begin{aligned}
\text{s.t.} \quad & v(\cdot) \in \mathcal{PC}(\mathbb{R}_{\geq 0}, \mathcal{U}) \longleftarrow \textbf{admissible control signal} \\
& x(0) = x_0, \\
& \text{for all } t \in \mathbb{R}_{\geq 0} : \\
& \quad \dot{x}(t) = f(x(t), v(t)), \\
& \quad x(t) \in \mathcal{X}. \qquad \longleftarrow \textbf{target safety constraint}
\end{aligned}$$

Wabersich*, Taylor*, Choi* et al., Data-Driven Safety Filters, Under review

Two main problems in designing safety filters

- **Safe set synthesis**

Not all states in the target safe set χ is safe.



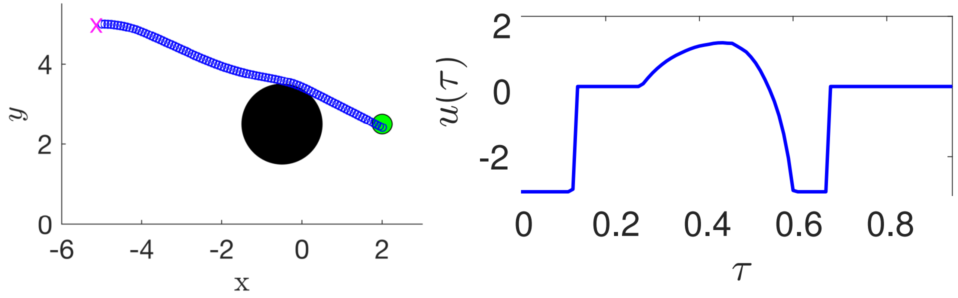
Failure region (χ^c)

Safe set S

Inevitably unsafe region

- **Safe control signal synthesis**

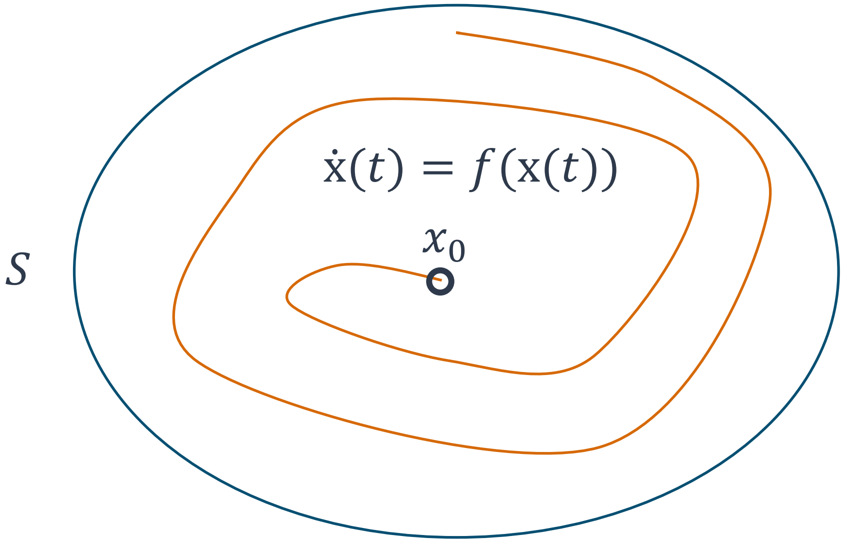
Regulating the control signal such that we remain in the safe set S.



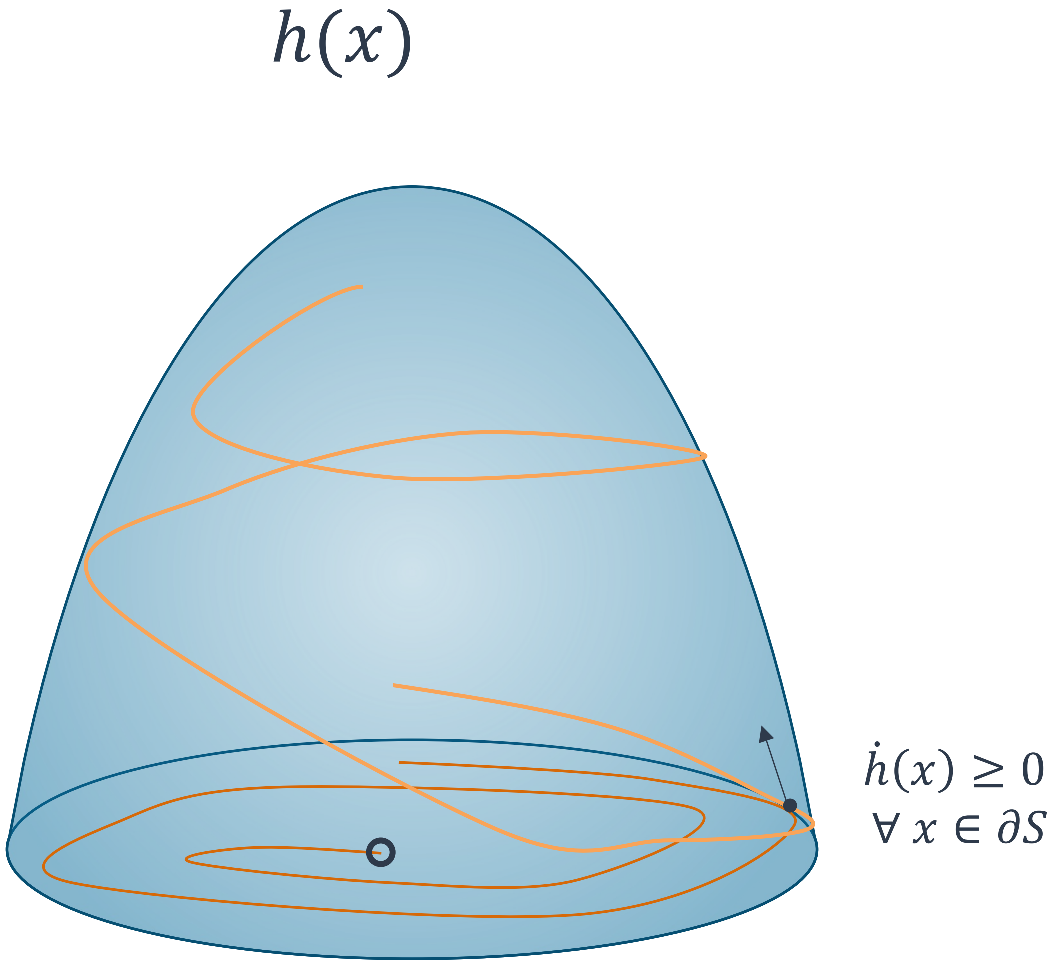
Forward Invariance for autonomous systems & Nagumo's theorem

A set S is forward invariant for the system $\dot{\mathrm{x}}(t) = f(\mathrm{x}(t))$ if for any initial state $\mathrm{x}(0) \in S$,

$$\mathrm{x}(t) \in S \text{ for all } t \geq 0.$$



How do we check if S is forward invariant?



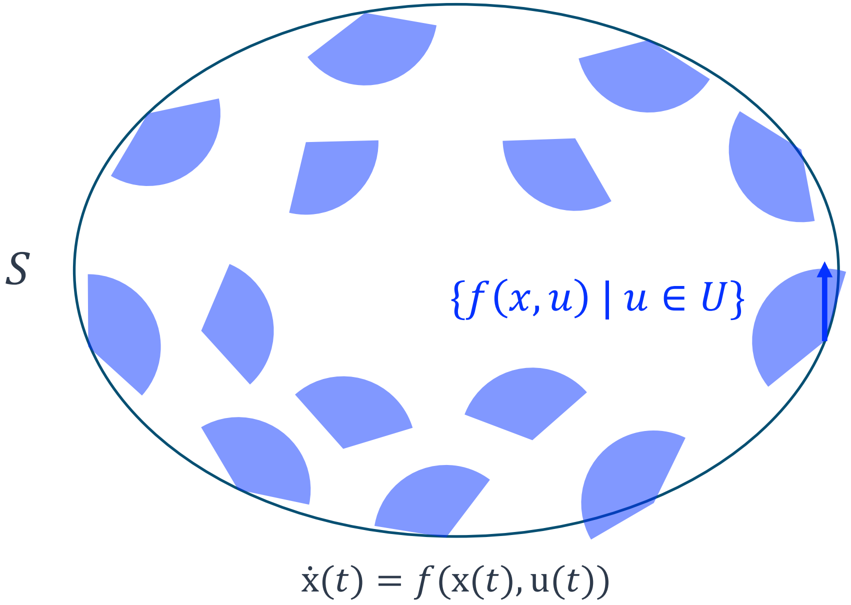
$S = \{x \mid h(x) \geq \mathbf{0}\}$

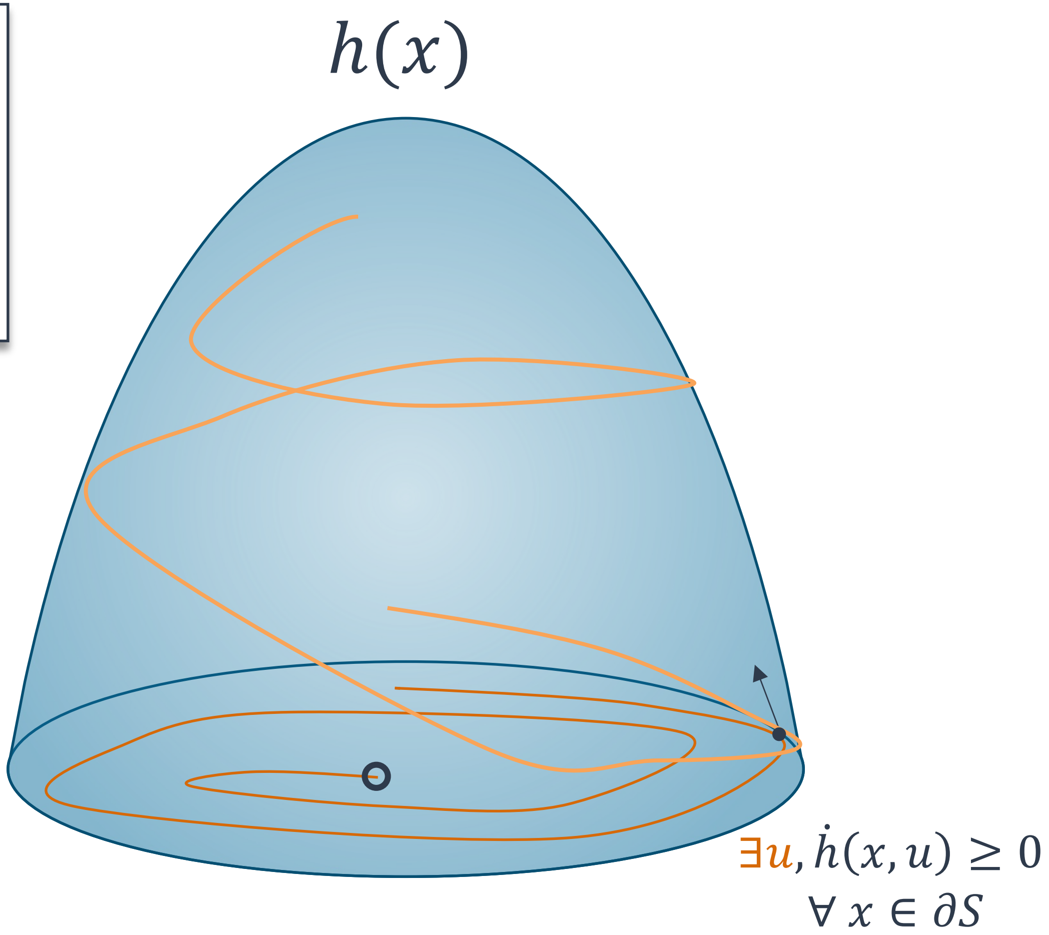
is (forward) invariant!

Control Invariance for control systems & Tangential characterization of control invariant sets

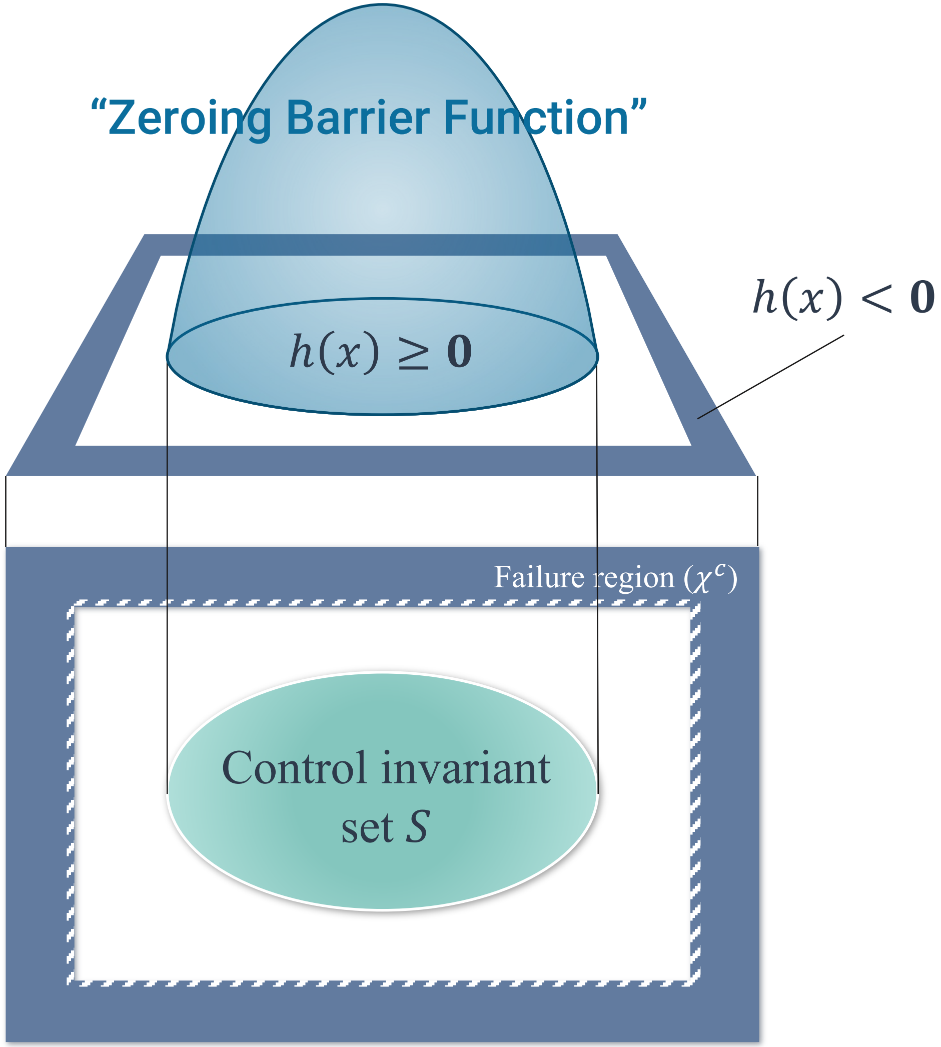
A set S is control invariant for the system $\dot{\mathrm{x}}(t) = f(\mathrm{x}(t), u(t))$ if for any initial state $\mathrm{x}(0) \in S$, there exists a control input signal $u(\cdot) \in \mathcal{U}$ under which

$$\mathrm{x}(t) \in S \text{ for all } t \geq 0.$$


S
$\{f(x,u) \mid u \in U\}$
$\dot{\mathrm{x}}(t) = f(\mathrm{x}(t), \mathrm{u}(t))$


$h(x)$
$\exists u, \dot{h}(x,u) \geq 0$
$\forall\, x \in \partial S$

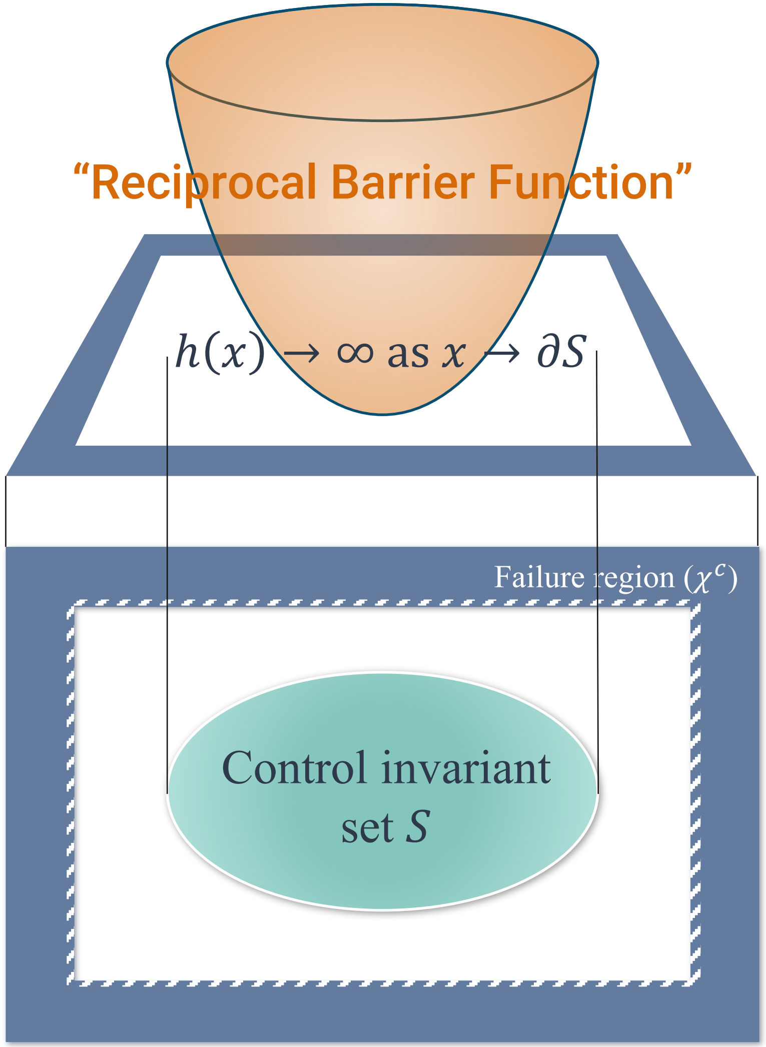
Control invariant set and Barrier Function / Certificate



"Zeroing Barrier Function"
$h(x) \geq \mathbf{0}$
$h(x) < \mathbf{0}$
Failure region (χ^c)
Control invariant set S

Weiland, Allgöwer, IFAC'07, Prajna, Automatica'06

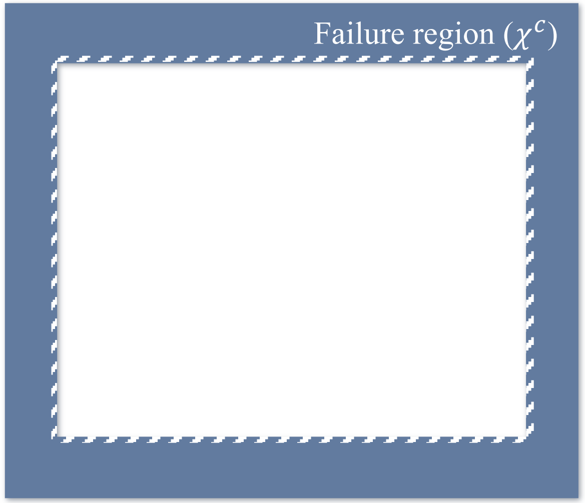
Control invariant set and Barrier Function / Certificate


"Reciprocal Barrier Function"
$h(x) \rightarrow \infty$ as $x \rightarrow \partial S$
Failure region (χ^c)
Control invariant set S

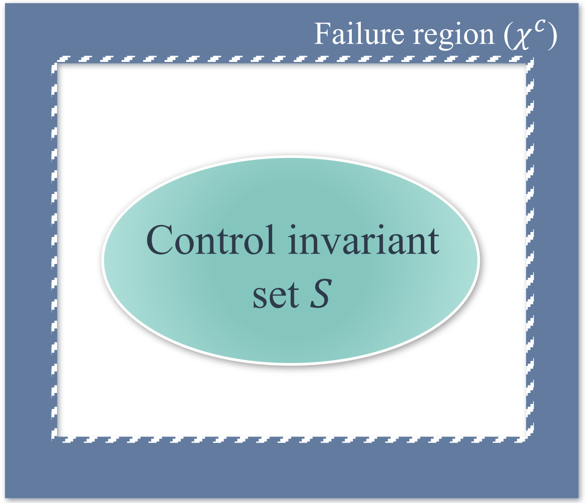
Ames et al, Control Barrier Function Based Quadratic Programs for Safety Critical Systems, TAC'17

General design procedure of safe controllers

1. Define the target safe set $\mathcal{X}$ based on the safety specifications.

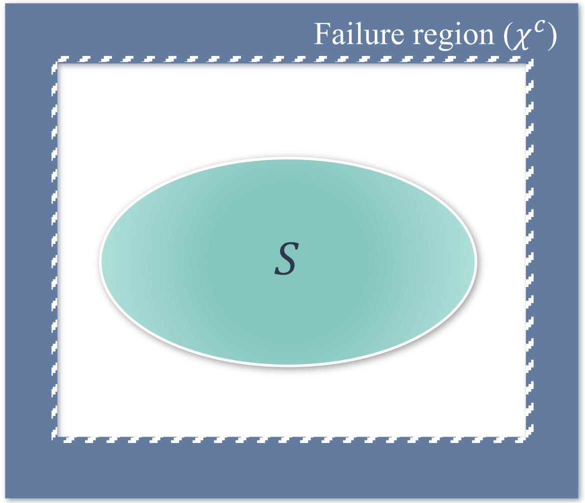


2. Verify a control invariant set S contained in $\mathcal{X}$.



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

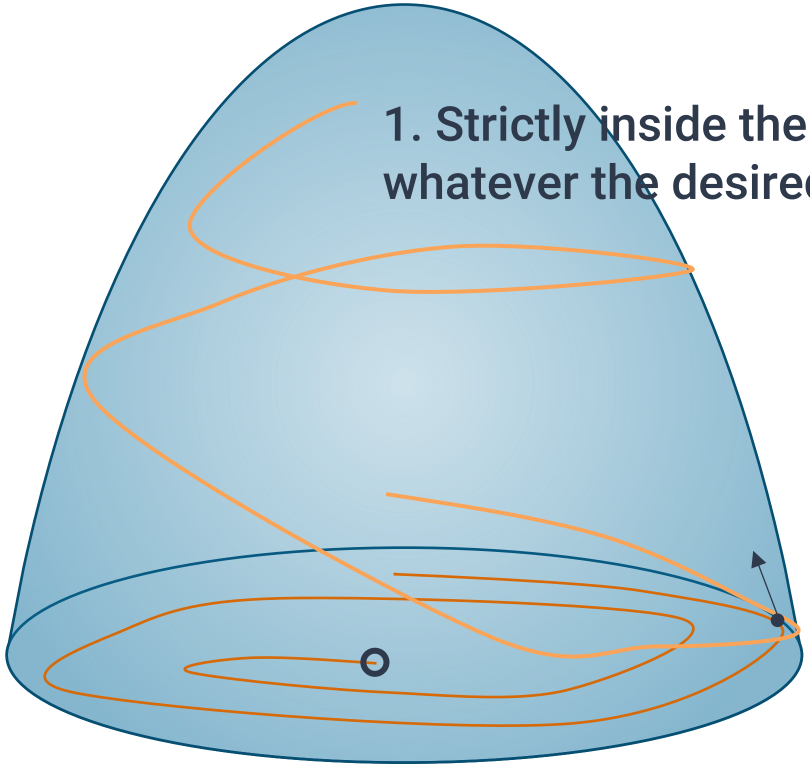
3. Design a safe controller $\varkappa_s(x)$ and verify that S is forward invariant for the closed-loop dynamics under $\varkappa_s$.



$$\dot{\mathrm{x}}(t) = f(\mathrm{x}(t), \varkappa_s(\mathrm{x}(t)))$$

Most basic safety filter

$$\kappa_F(x, u_{\text{des}}(t)) = \begin{cases} \kappa_{\mathcal{S}}(x), & x \in \partial\mathcal{S} \text{ or } u_{\text{des}}(t) \notin \mathcal{U}, \\ u_{\text{des}}(t), & \text{else.} \end{cases}$$

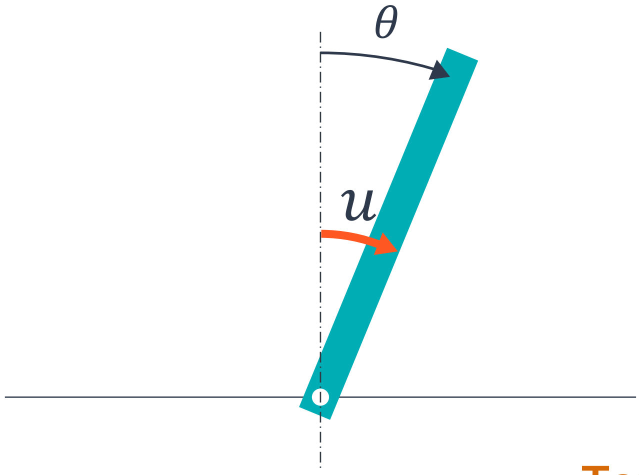


1. Strictly inside the set, the trajectory is allowed to do whatever the desired control signal wants.

2. On the boundary, you have to ensure that the trajectory is pushed back into the set. The backup controller $\varkappa_s(x)$ renders S invariant.

$$\dot{h}(x, \varkappa_s(x)) \geq 0\,, x \in \partial S$$

Example: Inverted Pendulum



$$\underbrace{\frac{\mathrm{d}}{\mathrm{dt}}\begin{bmatrix}\theta\\ \dot{\theta}\end{bmatrix}}_{\dot{x}} = \underbrace{\begin{bmatrix}\dot{\theta}\\ \frac{g}{\ell}\sin\theta\end{bmatrix}}_{f(x)} + \underbrace{\begin{bmatrix}0\\ \frac{1}{m\ell^2}\end{bmatrix}}_{g(x)} u$$

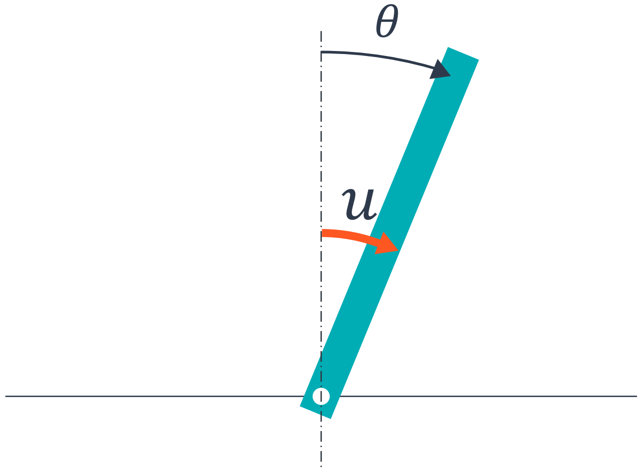
Input constraint: $\mathcal{U} = \{u \in \mathbb{R} || u | \leq 3\}$

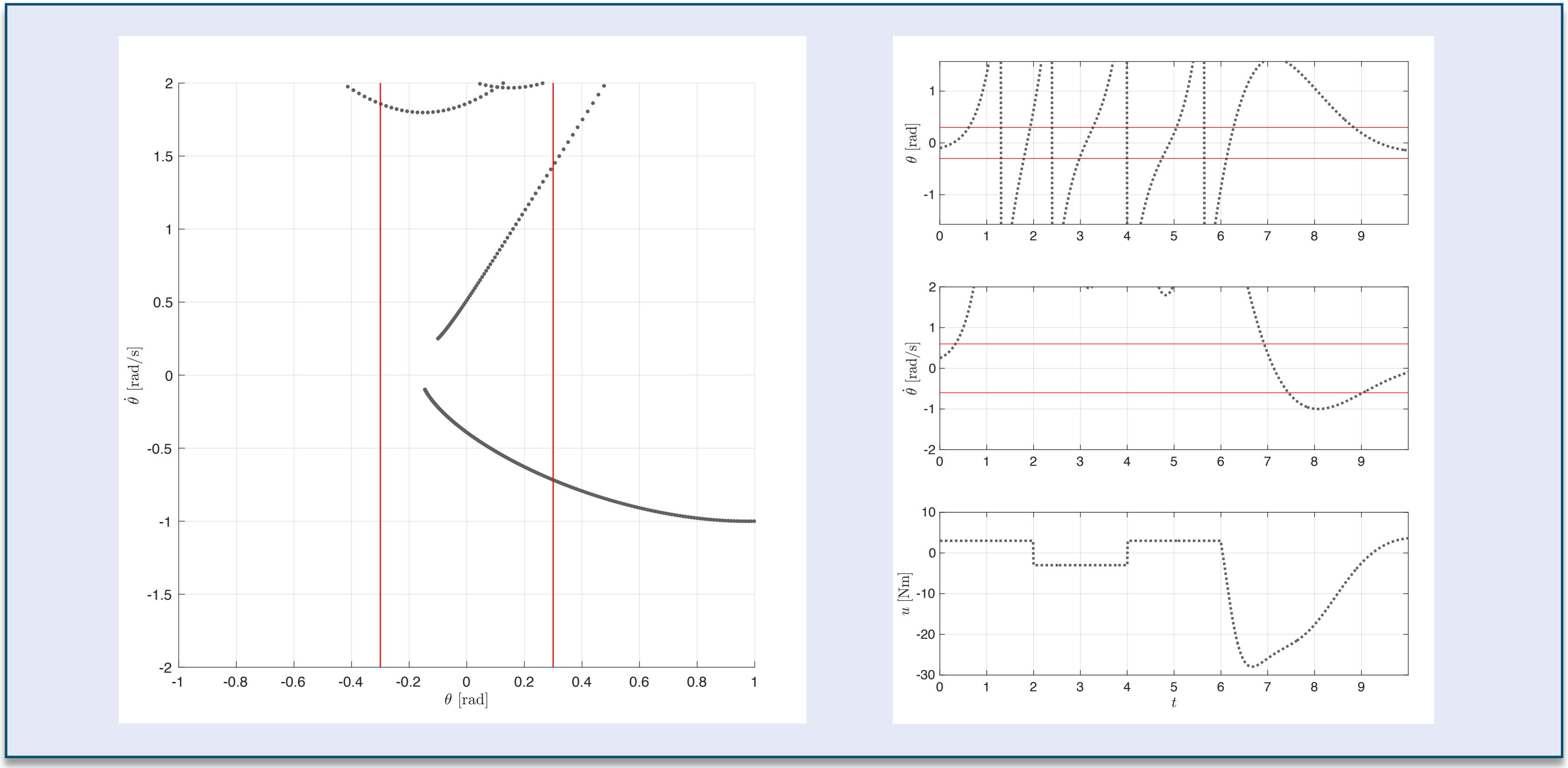
Target safety constraint: $\mathcal{X} = \{x \in \mathbb{R}^2 || x_1| \leq 0.3\}$

Desired (unsafe) control signal: $u_{\text{des}}(t) = \begin{cases} 3, & t \in [0,2) \\ -3, & t \in [2,4) \\ 3, & t \in [4,6) \\ m\ell^2\left(-\frac{g}{\ell}\sin x_1 - [1.5, 1.5]x\right), & \text{else} \end{cases}$

Wabersich*, Taylor*, Choi* et al., Data-Driven Safety Filters, Under review

Example: Unsafe desired control signal

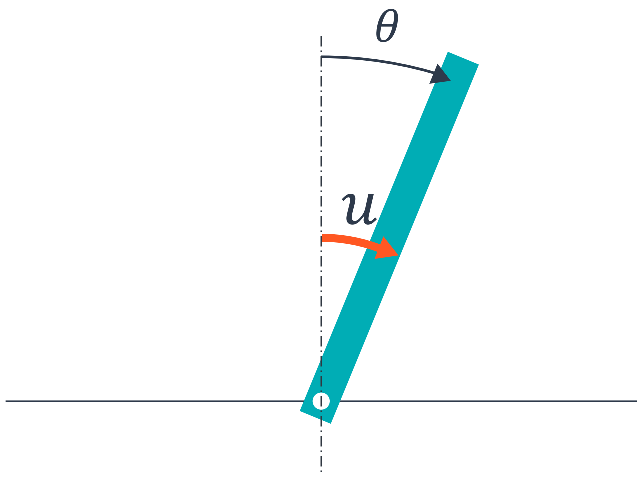




Wabersich*, Taylor*, Choi* et al., Data-Driven Safety Filters, Under review

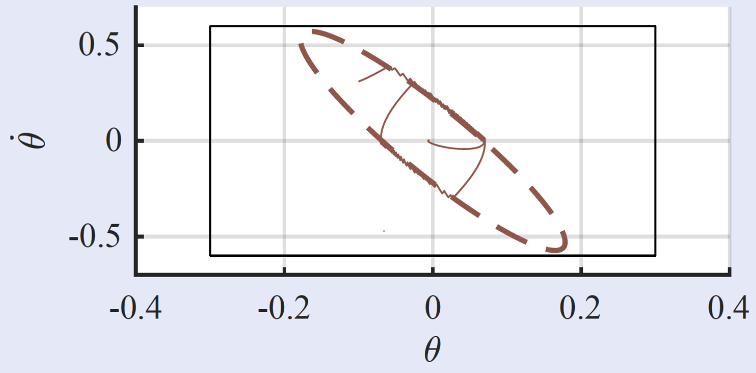
Example: Basic safety filter

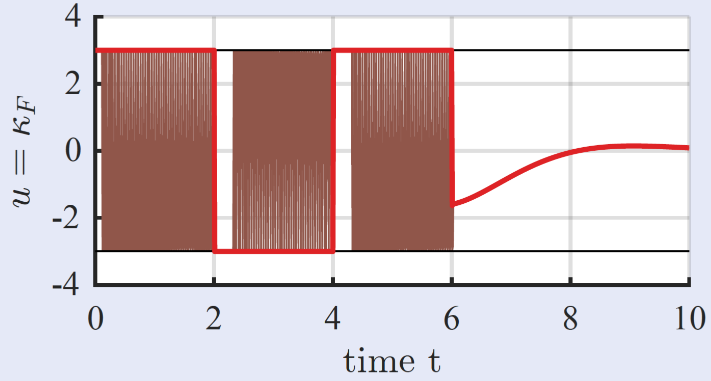
$$\kappa_F(x, u_{\text{des}}(t)) = \begin{cases} \kappa_{\mathcal{S}}(x), & x \in \partial\mathcal{S} \text{ or } u_{\text{des}}(t) \notin \mathcal{U}, \\ u_{\text{des}}(t), & \text{else.} \end{cases}$$



The backup controller $\varkappa_s(x)$ and the safe set S is designed by the LQR and its Lyapunov function.

$$\mathcal{S} = \{x \in \mathbb{R}^2 \mid \gamma - x^\top P x \geq 0\}$$

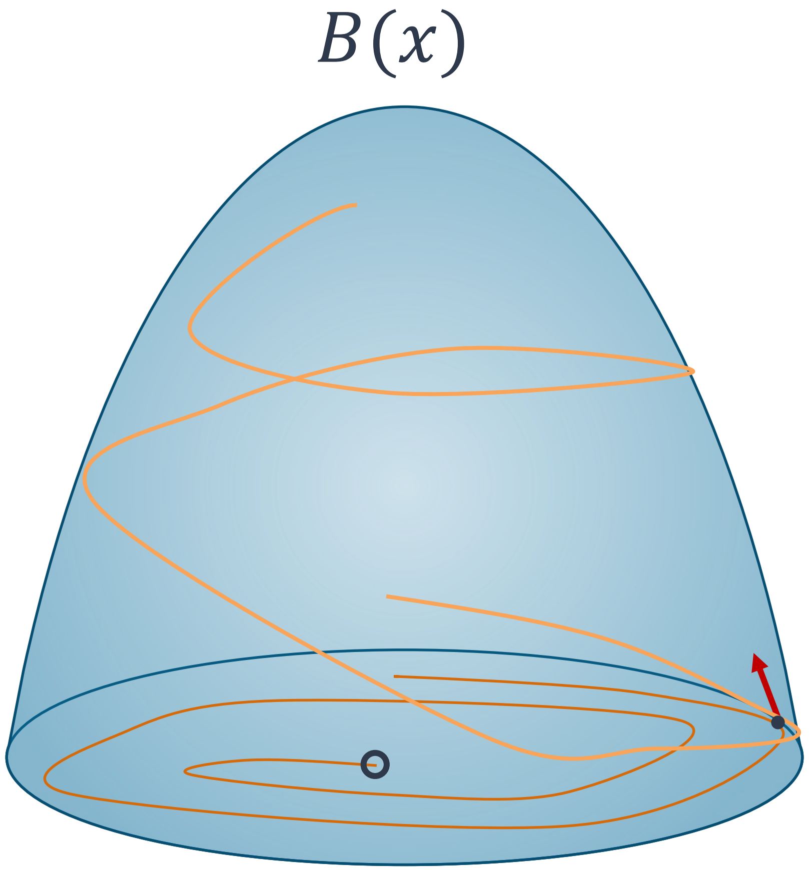




Wabersich*, Taylor*, Choi* et al., Data-Driven Safety Filters, Under review

2. Introduction to CBF

Main Idea: Smooth Braking



Rather than "hard stop" at the boundary, why not "smoothly brake" the trajectory as it approaches the boundary?

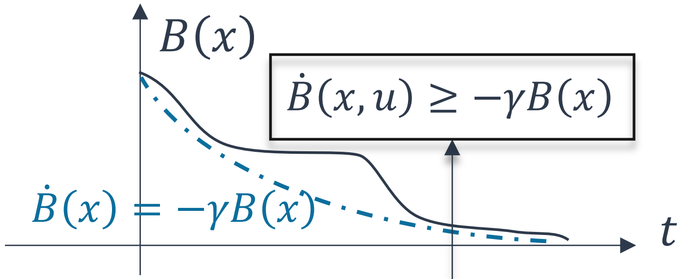
Control Barrier Function

$B(x): \mathbb{R}^n \to \mathbb{R}$, a continuously differentiable function.

$S = \{x \mid B(x) \geq 0\}$, $\nabla B(x) \neq 0$ **for all** $x \in \partial S$.

$B(x)$ **is a Control Barrier Function if** $\exists\, \gamma > 0$ **s.t. for all** $x \in S$

$$\sup_{u \in U} \dot{B}(x,u) + \gamma B(x) \geq 0.$$

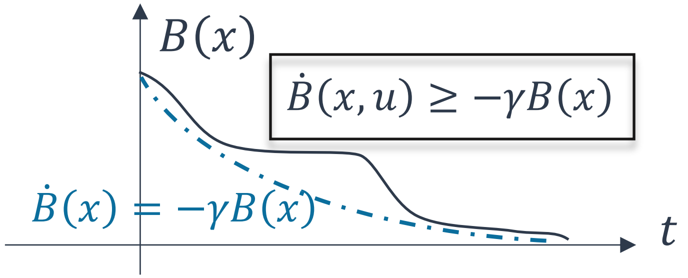


I will call this “smooth braking constraint”.

Original Definition: Ames et al, Control Barrier Function Based Quadratic Programs for Safety Critical Systems, TAC’17

Safety Guarantee

Main Theorem: Given a set S, if the CBF B exist, under $\mathrm{u}(t)$ that satisfies the smooth braking constraint, the set S is forward invariant.



Formal statements and proofs: Ames et al, TAC'17, Jankovic, Automatica'18

Proof: Comparison principle perspective

Comparison Lemma:
Let $a(\cdot)$ be the solution of the ODE

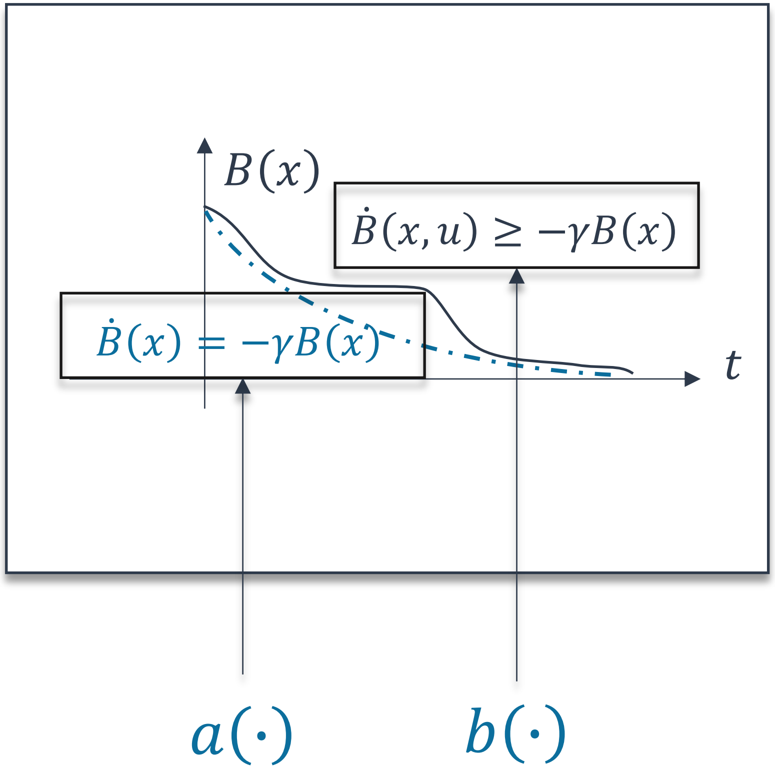
$$\dot{a} = f(t, a), \qquad a(0) = a_0$$

and $b(\cdot)$ be a almost-everywhere differentiable function that satisfies

$$\dot{b}(t) \geq f\big(t, b(t)\big), \qquad b(0) = b_0 \geq a_0$$

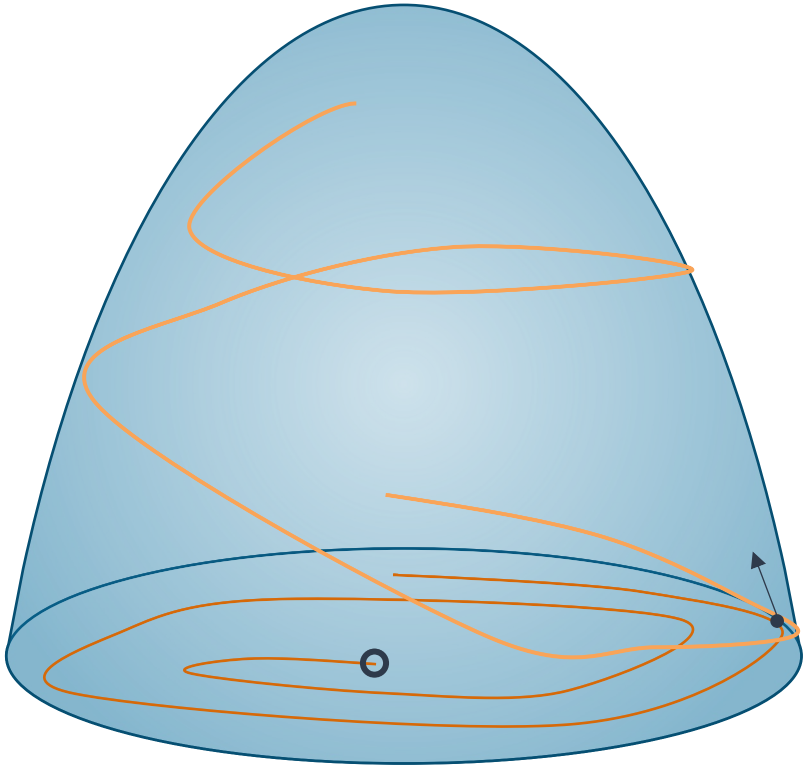
Then,

$$b(t) \geq a(t) \;\; \forall t \geq 0.$$



Khalil, Nonlinear systems, Ch3.4

Proof: Nagumo’s theorem perspective



$\exists u \in U \text{ s.t.}$
$\dot{B}(x,u) \geq 0$

- **Smooth braking constraint at the boundary:**

$$\dot{B}(x,u) \geq -\gamma B(x) = 0$$

- **Therefore, according to the definition of the CBF, at the boundary,**

$$\exists u \in U \text{ s.t. } \dot{B}(x,u) \geq 0$$

- **From Nagumo’s theorem, the set is forward invariant.**

Dynamics – Control Affine System

We will mainly deal with a specific type of a nonlinear system, a **control affine system:**

$$\dot{x} = \underbrace{f(x)}_{\substack{\text{“drift term”} \\ \text{“autonomous vector field”}}} + \underbrace{g(x)}_{\substack{\text{“actuation effect”} \\ \text{“control vector field”}}} u$$

where $f: \mathbb{R}^n \to \mathbb{R}^n$ and $g: \mathbb{R}^n \to \mathbb{R}^{n \times m}$ are Lipschitz continuous in x.

Many mechanical systems are control affine systems:

$$\begin{bmatrix} \dot{q} \\ \ddot{q} \end{bmatrix} = \begin{bmatrix} \dot{q} \\ D^{-1}(-C\dot{q} - G) \end{bmatrix} + \begin{bmatrix} 0 \\ D^{-1}B \end{bmatrix} u$$

CBF-QP: Min-norm safety filter

- QP for control-affine systems ($\dot{x} = f(x) + g(x)u$)

$$\underset{u}{\operatorname{argmin}} \quad \left\| u - u_{ref} \right\|^2$$

u: control input

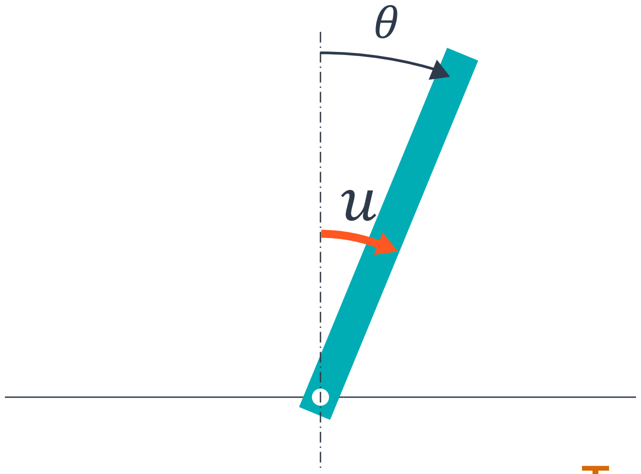
$$\text{subject to: } \overset{\dot{B}(x,u)}{\boxed{L_f B(x) + L_g B(x) u}} + \gamma B(x) \geq 0$$

$$u \in U$$

- CBF constraint can be relaxed if infeasibility is concerned.
- Closed-form solution exists for single-constraint unbounded CBF-QP * .

***Alan et al.**, Control Barrier Functions and Input-to-State Safety with Application to Automated Vehicles, Arxiv'22

Example: Inverted Pendulum



$$\underbrace{\frac{\mathrm{d}}{\mathrm{dt}}\begin{bmatrix}\theta\\ \dot{\theta}\end{bmatrix}}_{\dot{x}} = \underbrace{\begin{bmatrix}\dot{\theta}\\ \frac{g}{\ell}\sin\theta\end{bmatrix}}_{f(x)} + \underbrace{\begin{bmatrix}0\\ \frac{1}{m\ell^2}\end{bmatrix}}_{g(x)} u$$

Input constraint: $\mathcal{U} = \{u \in \mathbb{R} || u | \leq 3\}$

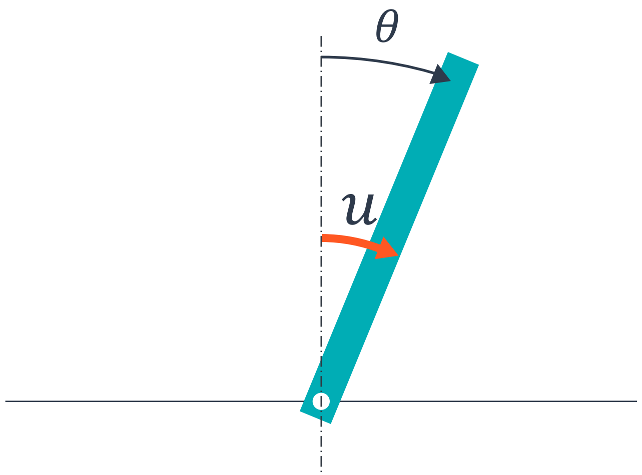
Target safety constraint: $\mathcal{X} = \{x \in \mathbb{R}^2 || x_1 | \leq 0.3\}$

Desired (unsafe) control signal: $u_{\text{des}}(t) = \begin{cases} 3, & t \in [0,2) \\ -3, & t \in [2,4) \\ 3, & t \in [4,6) \\ m\ell^2\left(-\frac{g}{\ell}\sin x_1 - [1.5, 1.5]x\right), & \text{else} \end{cases}$

Wabersich[*], Taylor[*], Choi[*] et al., Data-Driven Safety Filters, Under review

Example: Quadratic form (Lyapunov-based design)

$$h_{\mathcal{S}}(x) = 1 - x^\top \begin{pmatrix} 1/a^2 & 0.5/ab \\ 0.5/ab & 1/b^2 \end{pmatrix} x$$



Evaluate the CBF constraint feasibility:

$$\nabla h_{\mathcal{S}}(x) g(x) = 0 \Rightarrow \nabla h_{\mathcal{S}}(x) f(x) + \alpha(h_{\mathcal{S}}(x)) > 0,$$

Determine the valid parameters:

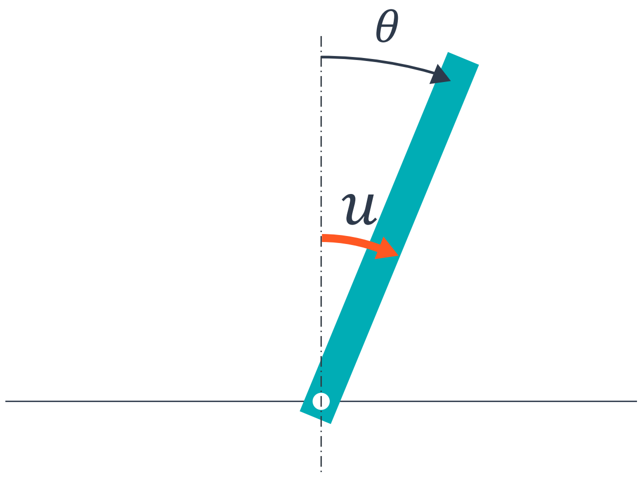
$$\gamma \geq 0.2, \qquad a = 5/6, \qquad b = 5/3$$

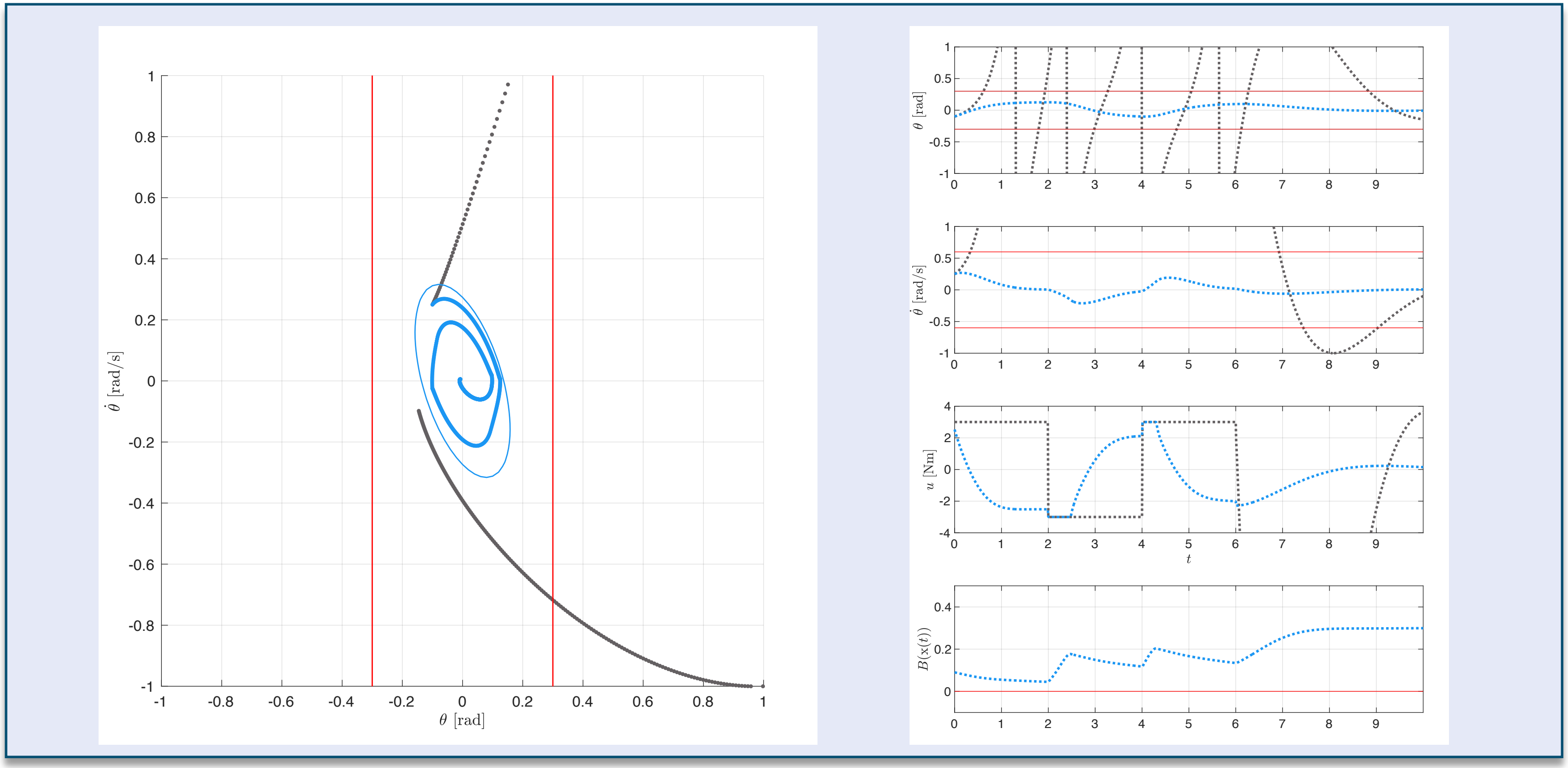
This procedure can be done more generally by constructing an optimization problem[1].

[1] 1. Wang et al., Permissive barrier certificates for safe stabilization using sum-of-squares, ACC 2018

Example: Quadratic form (Lyapunov-based design)

$$h_{\mathcal{S}}(x) = 1 - x^\top \begin{pmatrix} 1/a^2 & 0.5/ab \\ 0.5/ab & 1/b^2 \end{pmatrix} x\,, \qquad \gamma = 0.2$$

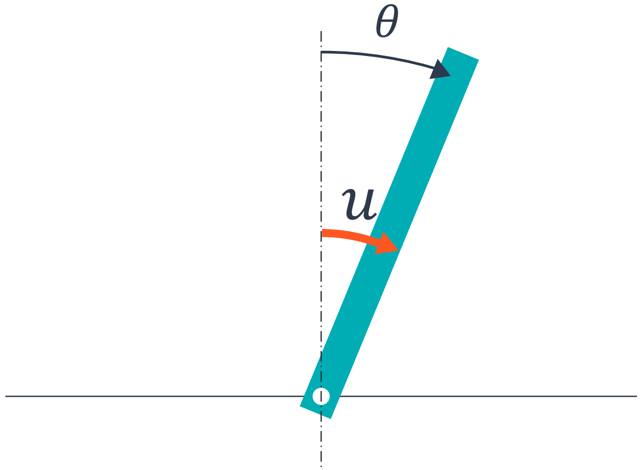


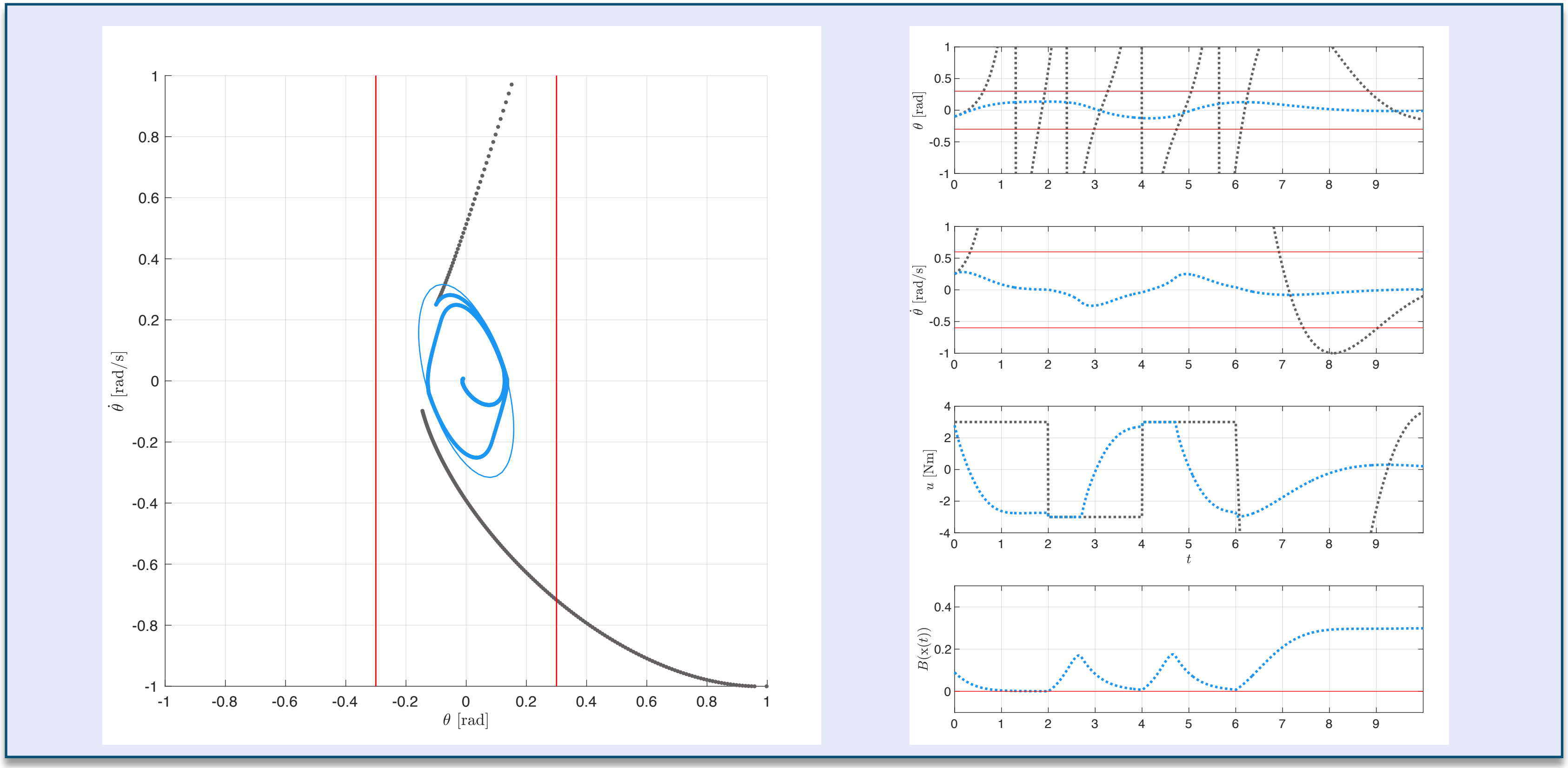


Wabersich*, Taylor*, Choi* et al., Data-Driven Safety Filters, Under review

Example: Quadratic form (Lyapunov-based design)

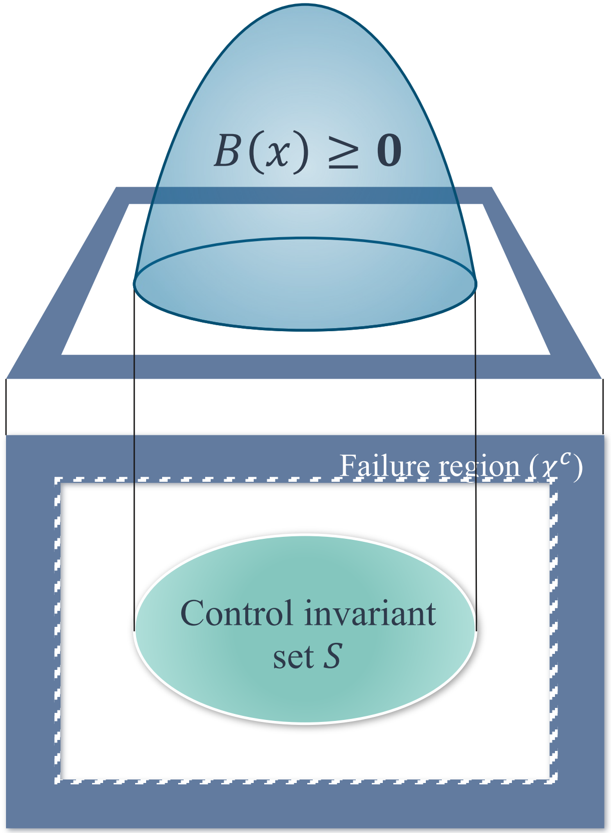
$$h_{\mathcal{S}}(x) = 1 - x^\top \begin{pmatrix} 1/a^2 & 0.5/ab \\ 0.5/ab & 1/b^2 \end{pmatrix} x\,, \qquad \gamma = 2.0$$





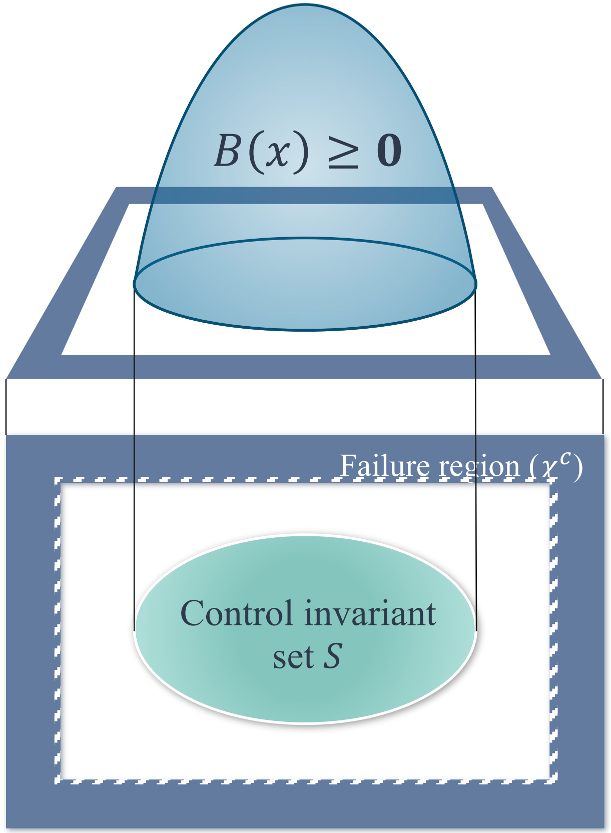
Wabersich*, Taylor*, Choi* et al., Data-Driven Safety Filters, Under review

Crucial design step 1: Choice of CBF $B(x)$



1. CBF zero-superlevel set has to be contained in $\mathcal{X}$.
2. The zero-superlevel set has to be control invariant.

Crucial design step 2: Choice of γ



γ decides the profile of smooth braking.

- Small γ is more anticipative, but more restrictive. Crucially, the smooth braking constraint can be infeasible if γ is not large enough.
- Large γ is less restrictive, but more myopic.

CBF-CLF Helper

- Library: https://github.com/HybridRobotics/CBF-CLF-Helper
- Designed to let users easily implement safety-controller based on CBFs and CLFs with Matlab.
    - An easy interface for construction and simulation of a control-affine system.
    - Safety controller including CLF-QP, CBF-QP, and CBF-CLF-QP as built-in functions.
- New version releasing soon
    - https://github.com/ChoiJangho/CBF-CLF-Helper/tree/feedback_linearization
    - Supports feedback linearization, numerical CBFs, more demos.

Main Research Challenges

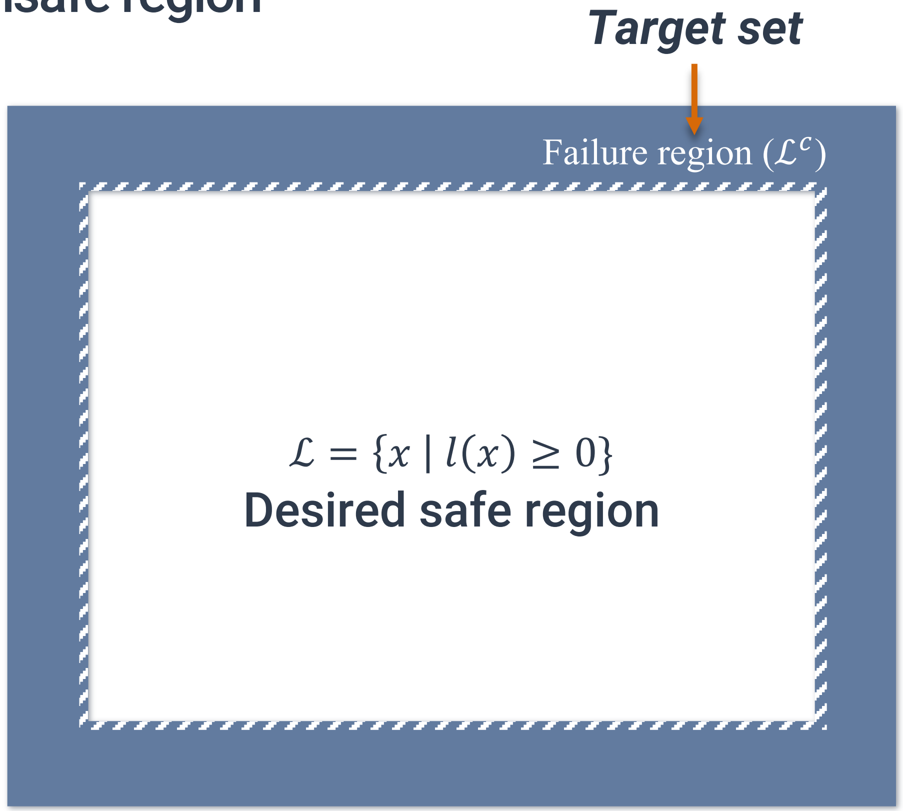
- Finding / designing a valid CBF
  - Finding “good” control invariant sets
  - Guaranteeing CBF constraints under control input bounds
  - Sum-of-squares programming
    - Dai, Permenter, Convex synthesis and verification of control-Lyapunov and barrier functions with input constraints, ACC’23
  - Deep Learning
    - Dawson et al., Safe Control With Learned Certificates: A Survey of Neural Lyapunov, Barrier, and Contraction Methods for Robotics and Control, TRO’23
    - Castaneda et al., In-Distribution Barrier Functions: Self-Supervised Policy Filters that Avoid Out-of-Distribution States, L4DC’23

- Combining with other methods
  - Hamilton-Jacobi Reachability
  - MPC

3. Alternative methods:
Hamilton-Jacobi Reachability& MPC

HJ Reachability for safety control

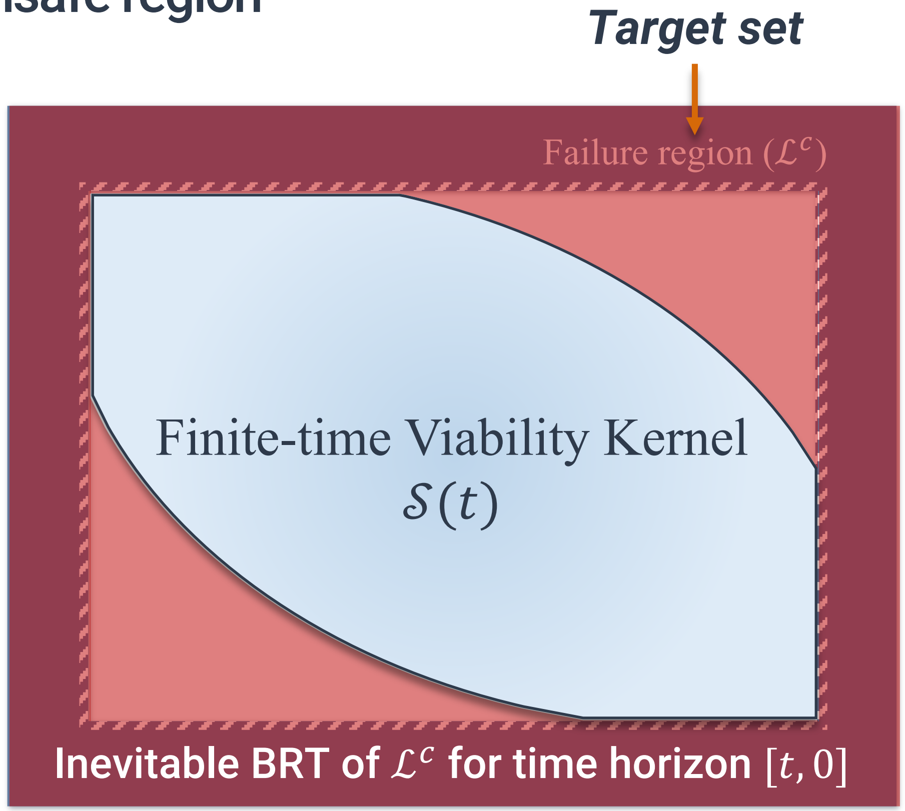
- Control Objective: to avoid the unsafe region during all prescribed future time horizon.
- (Inevitable) Backward Reachable Tube (BRT) of the unsafe region



Target set
Failure region ($\mathcal{L}^c$)
$\mathcal{L} = \{x \mid l(x) \geq 0\}$
Desired safe region

HJ Reachability for safety control

- Control Objective: to avoid the unsafe region during all prescribed future time horizon.
- (Inevitable) Backward Reachable Tube (BRT) of the unsafe region



Target set
Failure region ($\mathcal{L}^c$)
Finite-time Viability Kernel
$\mathcal{S}(t)$
Inevitable BRT of $\mathcal{L}^c$ for time horizon $[t, 0]$

HJ Reachability for safety control

Value Function

$$V(\mathrm{x}(t), t) = \sup_{u(\cdot)\in\mathcal{U}} \min_{\tau\in[t,0]} l(\mathrm{x}(\tau))$$

Dynamic Programming Principle

$$V(\mathrm{x}(t), t) = \sup_{u(\cdot)\in\mathcal{U}} \min \left\{ l(\mathrm{x}(t)), V(\mathrm{x}(t+\delta), t+\delta) \right\}$$

HJ-VI:

$$0 = \min \left\{ l(x) - V(x,t), D_t V(x,t) + \max_{u\in U} D_x V(x,t) \cdot f(x,u) \right\}$$

HJ Reachability for safety control

Value Function

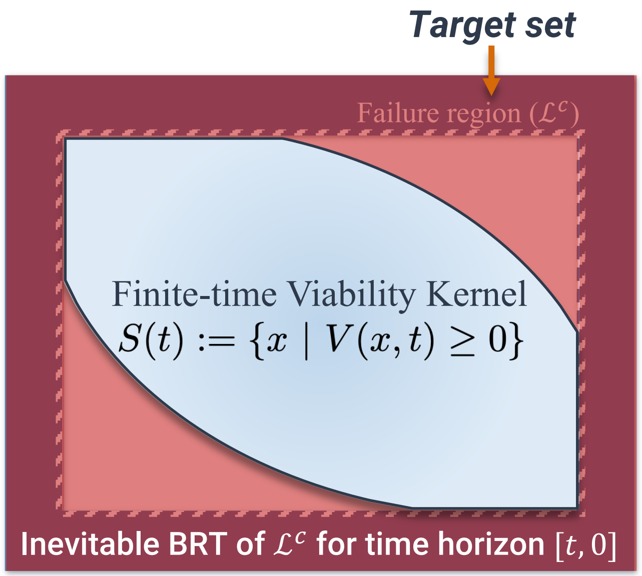
$$V(\mathrm{x}(t), t) = \sup_{u(\cdot)\in\mathcal{U}} \min_{\tau\in[t,0]} l(\mathrm{x}(\tau))$$

Dynamic Programming Principle

$$V(\mathrm{x}(t), t) = \sup_{u(\cdot)\in\mathcal{U}} \min\left\{l(\mathrm{x}(t)), V(\mathrm{x}(t+\delta), t+\delta)\right\}$$

HJ-VI:

$$0 = \min\left\{l(x) - V(x,t), D_t V(x,t) + \max_{u\in U} D_x V(x,t)\cdot f(x,u)\right\}$$


Target set
Failure region ($\mathcal{L}^c$)
Finite-time Viability Kernel
$S(t) := \{x \mid V(x,t) \geq 0\}$
Inevitable BRT of $\mathcal{L}^c$ for time horizon $[t, 0]$

$$0 = \min\left\{l(x) - V(x,t), D_t V(x,t) + \min_{u\in U} D_x V(x,t)\cdot f(x,u)\right\}$$

$$\begin{aligned}\pi_V^*(x,t) &\in K_V(x,t)\\ &:= \{u \in U : D_t V(x,t) + D_x V(x,t)\cdot f(x,u) \geq 0\}\end{aligned}$$

$$\begin{aligned}\dot{V}(\mathrm{x}(t), t) = {}& D_t V(\mathrm{x}(t), t)\\ &+ D_x V(\mathrm{x}(t), t)\cdot f(\mathrm{x}(t), \pi_V^*(\mathrm{x}(t), t), d) \geq 0,\end{aligned}$$

HJ Reachability for safety control

Value Function

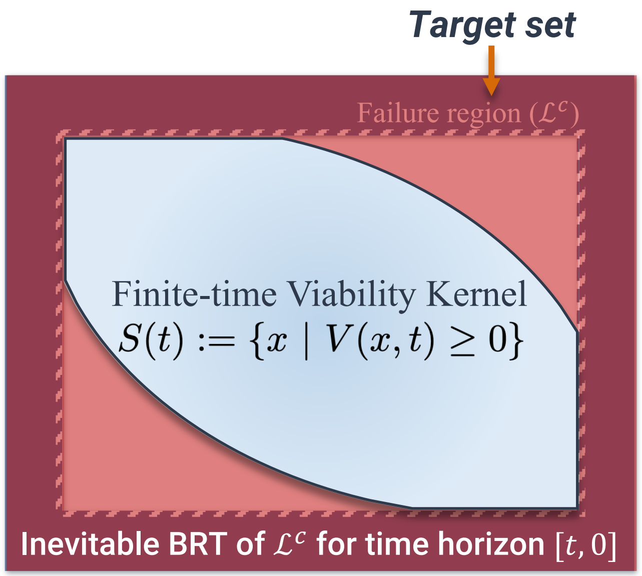
$$V(\mathrm{x}(t),t) = \sup_{u(\cdot)\in\mathcal{U}} \min_{\tau\in[t,0]} l(\mathrm{x}(\tau),\tau)$$

Dynamic Programming Principle

$$V(\mathrm{x}(t),t) = \sup_{u(\cdot)\in\mathcal{U}} \min\{l(\mathrm{x}(t)), V(\mathrm{x}(t+\delta),t+\delta)\}$$

HJ-VI:

$$0 = \min\left\{ l(x) - V(x,t), D_t V(x,t) + \max_{u\in U} D_x V(x,t)\cdot f(x,u) \right\}$$


Target set
Failure region ($\mathcal{L}^c$)
Finite-time Viability Kernel
$S(t) := \{x \mid V(x,t) \geq 0\}$
Inevitable BRT of $\mathcal{L}^c$ for time horizon $[t,0]$

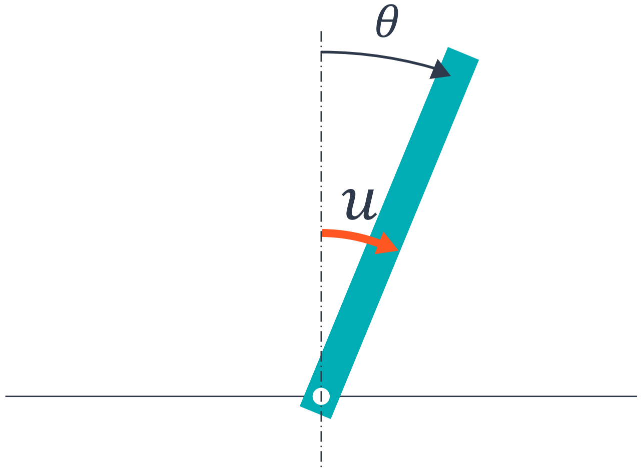
$$\begin{aligned}\pi_V^*(x,t) &\in K_V(x,t)\\ &:= \{u \in U : D_t V(x,t) + D_x V(x,t)\cdot f(x,u) \geq 0\}\end{aligned}$$

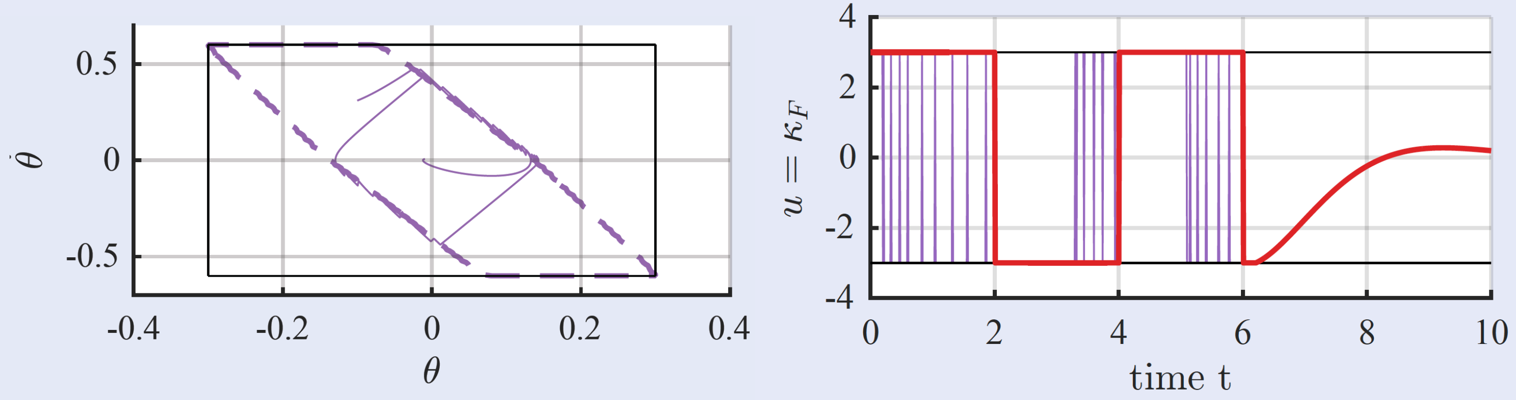
Least-restrictive reachability safety filter

$$\kappa_F(x, u_{\text{des}}(t)) = \begin{cases} \pi_V^*(x), & V(x) \leq \epsilon \\ u_{\text{des}}(t), & \text{else.} \end{cases}$$

Example: Least-restrictive reachability safety filter

$$\kappa_F(x, u_{\text{des}}(t)) = \begin{cases} \pi_V^*(x), & V(x) \le \epsilon \\ u_{\text{des}}(t), & \text{else.} \end{cases}$$





Predictive Safety Filter (MPC)

- **Discrete-time formulation:**

$$x(k+1) = x(k) + \Delta T f(x(k), u(k)) = \mathrm{f}(x(k), u(k)).$$

$$\begin{aligned}
\min_{u_{i|k}} \quad & \|u_{\text{des}}(k) - u_{0|k}\| \\
\text{s.t.} \quad & x_{i+1|k} = \mathrm{f}(x_{i|k}, u_{i|k}), \\
& x_{0|k} = x(k), \\
\text{Target safety constraint:} \quad & x_{i|k} \in \mathcal{X}, && \text{for } i = 0, ..., N-1, \\
\text{Input constraint:} \quad & u_{i|k} \in \mathcal{U}, && \text{for } i = 0, ..., N-1, \\
\boxed{\text{Terminal set constraint:} \; x_{N|k} \in \mathcal{S}^t,} &
\end{aligned}$$

Predictive Safety Filter (MPC)

$$\min_{u_{i|k}} \quad \|u_{\text{des}}(k) - u_{0|k}\|$$

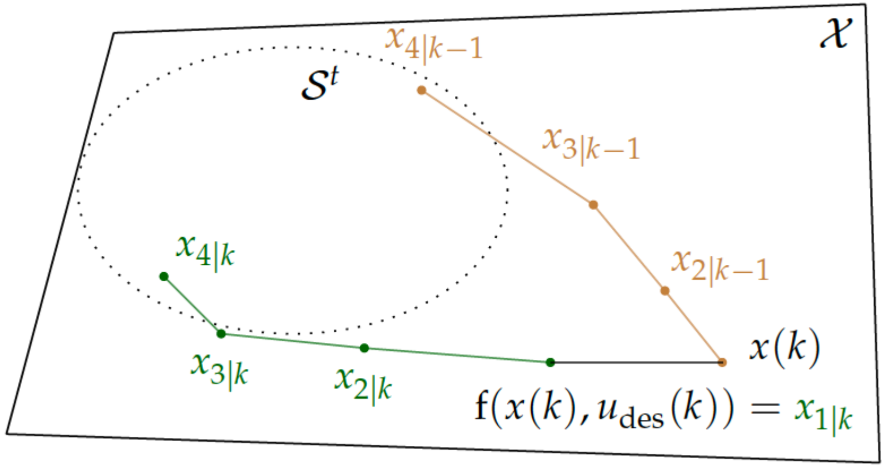
$$\text{s.t.} \quad x_{i+1|k} = \mathrm{f}(x_{i|k}, u_{i|k}),$$

$$x_{0|k} = x(k),$$

Target safety constraint: $x_{i|k} \in \mathcal{X},$

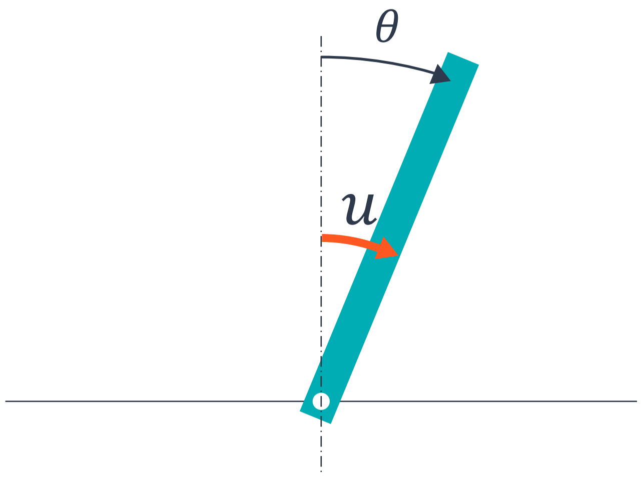
Input constraint: $u_{i|k} \in \mathcal{U},$

Terminal set constraint: $x_{N|k} \in \mathcal{S}^t,$



- In order to guarantee recursive feasibility, the terminal set has to be control invariant.
- A feasible sequence of $u_{i|k}$ serves as the “backup” plan.

Example: Predictive Safety Filter



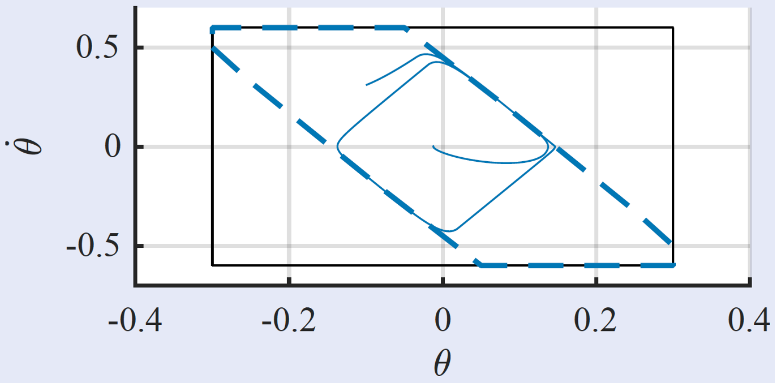
Terminal invariant set designed based on robust Lyapunov function:

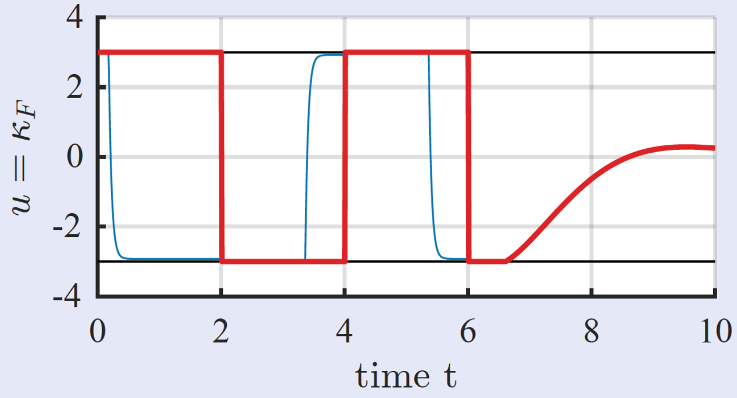
$$\mathcal{S}_\gamma^t = \left\{x \in \mathbb{R}^{n_x} \mid \gamma - x^\top P x \geq 0\right\} \text{ with } \gamma \in (0,1]$$

Lyapunov stability condition:

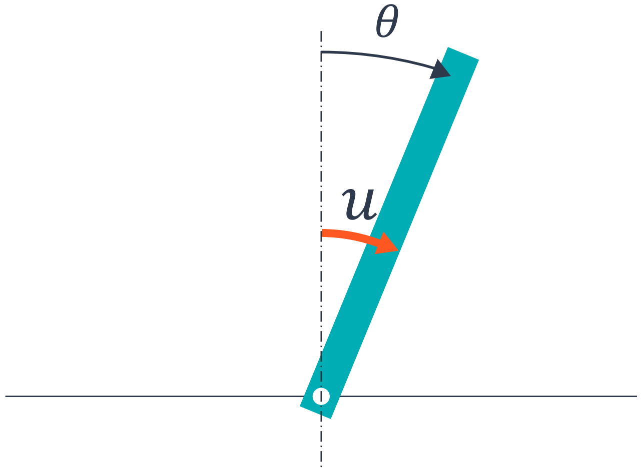
$$(A_K x + r_K(x))^\top P \left(A_k + r_K(x)\right) - x^\top P x \leq 0$$

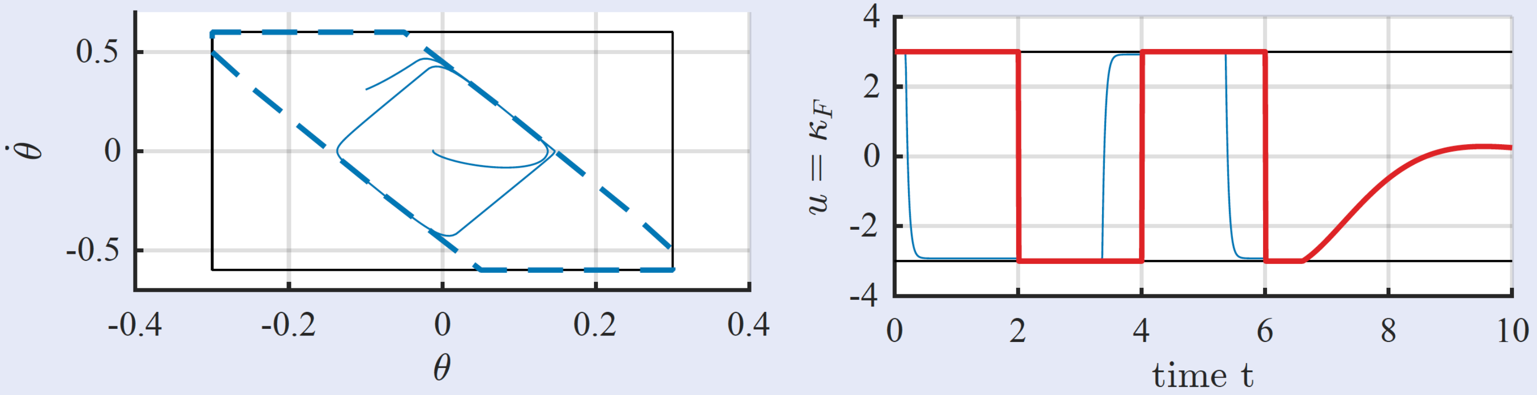
Solve for the maximal γ such that the above condition is satisfied.



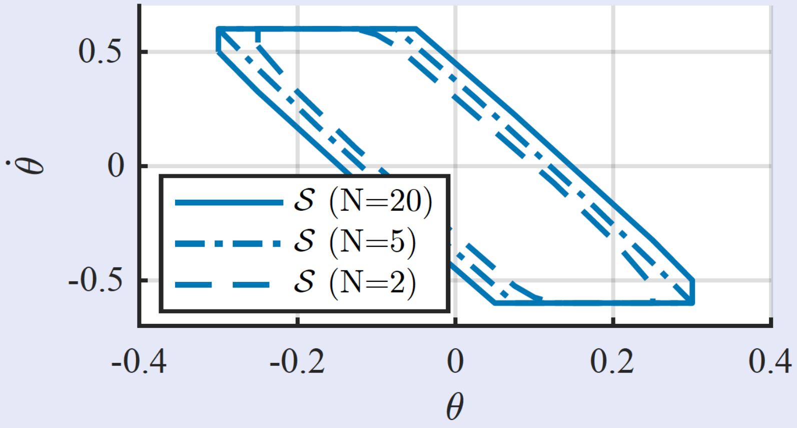


Example: Predictive Safety Filter

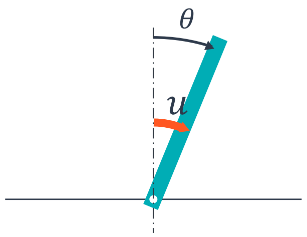




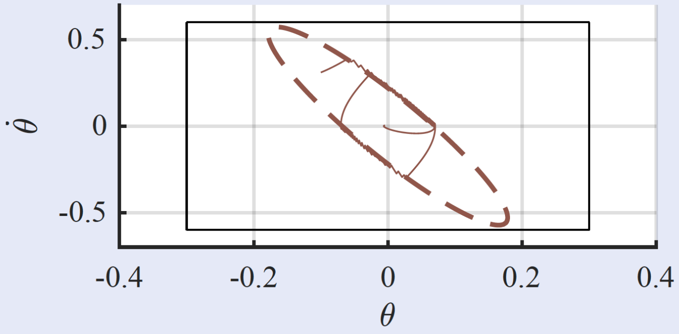
Implicit safe sets w.r.t. prediction horizon:



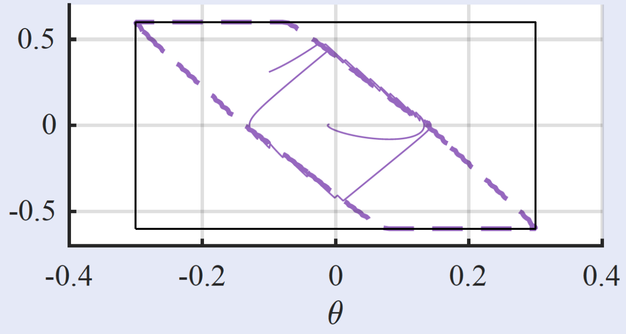
Example: Summary



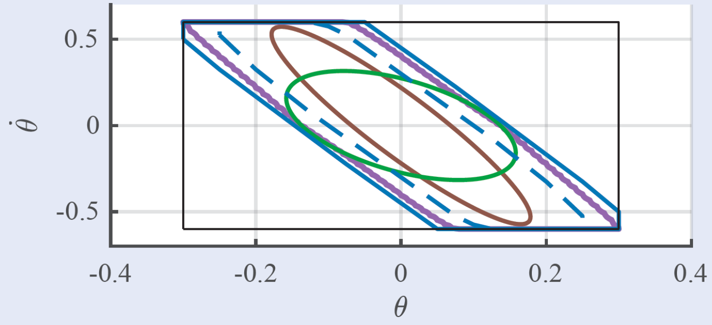
Basic Safety Filter



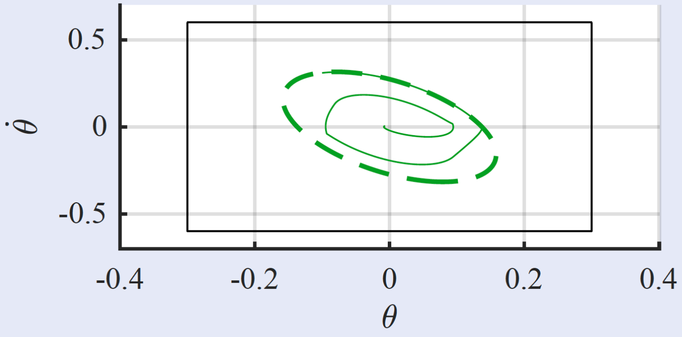
HJ Reachability



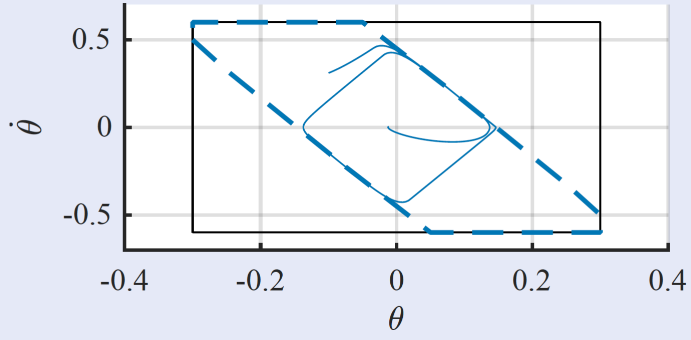
Safe set comparison



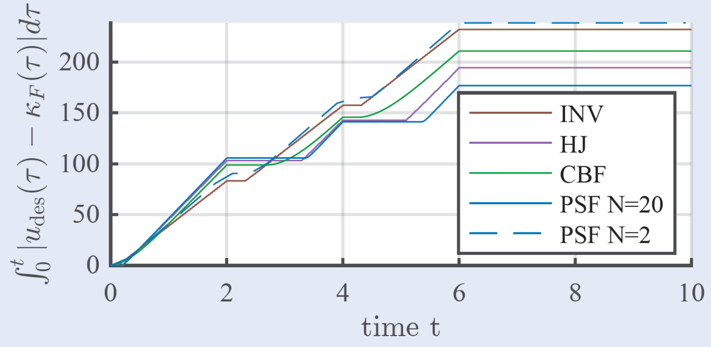
CBF-QP



Predictive safety filter

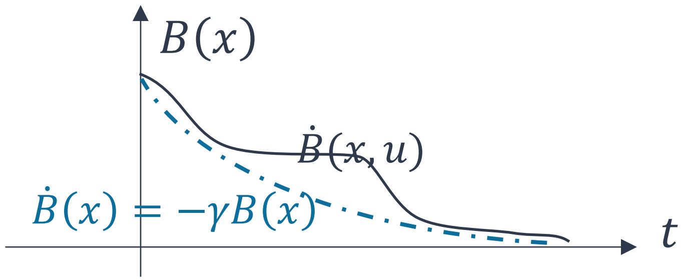


Deviation from desired control

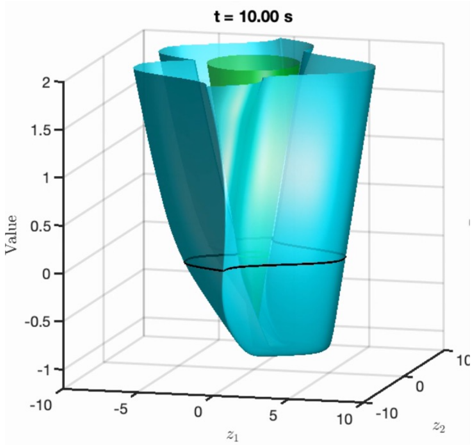


Comparison with HJ reachability and Predictive filter

Control Barrier Functions

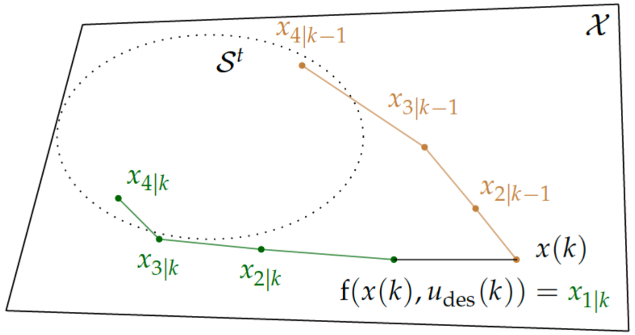


HJ Reachability



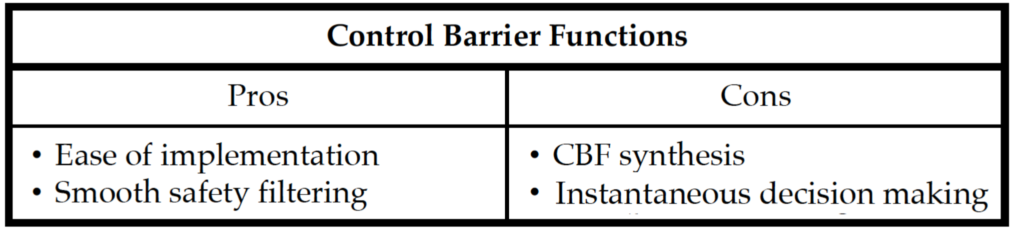
Source: Herbert, Bansal et al. CDC’17

Predictive Filter



Wabersich*, Taylor*, Choi* et al., Data-Driven Safety Filters, Under review

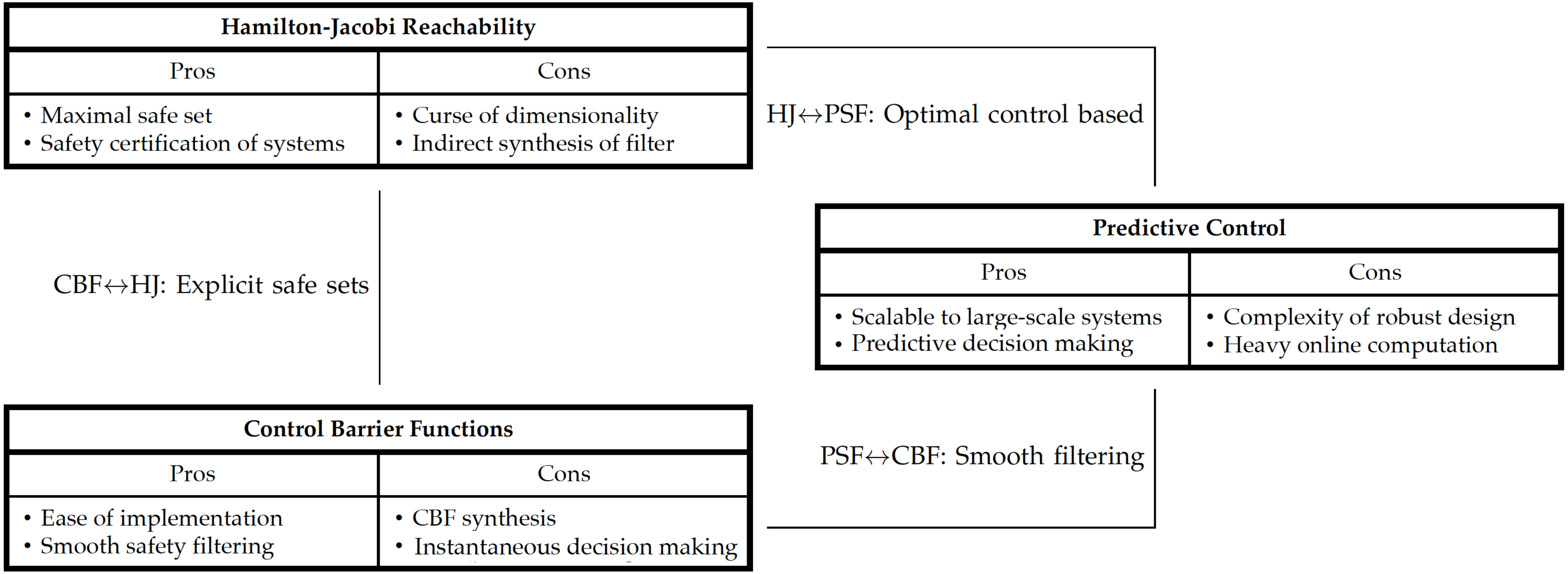
Comparison with HJ reachability and Predictive filter



Control Barrier Functions	
Pros	Cons
• Ease of implementation • Smooth safety filtering	• CBF synthesis • Instantaneous decision making

Wabersich*, Taylor*, Choi* et al., Data-Driven Safety Filters, Under review

Comparison with HJ reachability and Predictive filter



Hamilton-Jacobi Reachability	
Pros	Cons
• Maximal safe set • Safety certification of systems	• Curse of dimensionality • Indirect synthesis of filter

HJ↔PSF: Optimal control based

CBF↔HJ: Explicit safe sets

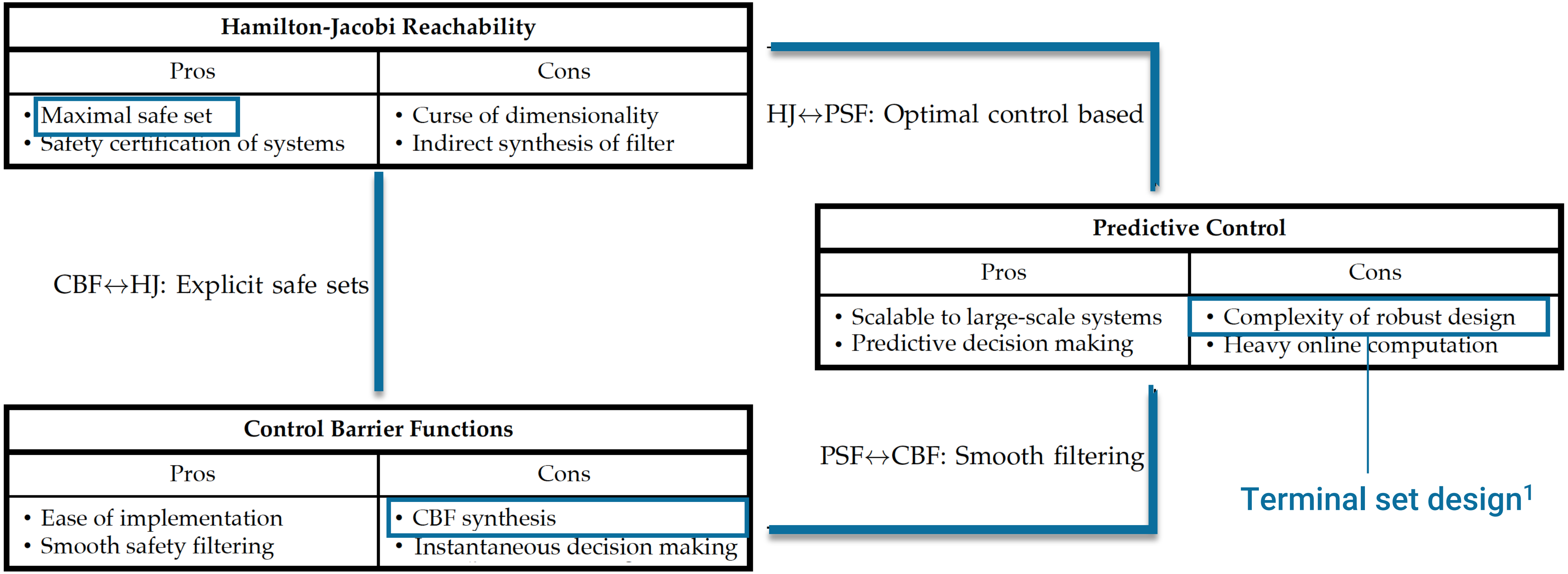
Predictive Control	
Pros	Cons
• Scalable to large-scale systems • Predictive decision making	• Complexity of robust design • Heavy online computation

PSF↔CBF: Smooth filtering

Control Barrier Functions	
Pros	Cons
• Ease of implementation • Smooth safety filtering	• CBF synthesis • Instantaneous decision making

Wabersich*, Taylor*, Choi* et al., Data-Driven Safety Filters, Under review

The core of all three methods – Finding control invariant sets!



Hamilton-Jacobi Reachability	
Pros	Cons
• Maximal safe set • Safety certification of systems	• Curse of dimensionality • Indirect synthesis of filter

HJ↔PSF: Optimal control based

CBF↔HJ: Explicit safe sets

Predictive Control	
Pros	Cons
• Scalable to large-scale systems • Predictive decision making	• Complexity of robust design • Heavy online computation

PSF↔CBF: Smooth filtering

Terminal set design[1]

Control Barrier Functions	
Pros	Cons
• Ease of implementation • Smooth safety filtering	• CBF synthesis • Instantaneous decision making

Rosolia, Borrelli, et al., Learning model predictive control for iterative tasks. A Data-Driven Control Framework, TAC'17

Thank you, Questions are welcomed!

jason.choi@berkeley.edu

CBF-CLF-Helper: https://github.com/HybridRobotics/CBF-CLF-Helper

Appendix—Robustness of CBF

Robustness of CBF- 1. Attractivity of the zero-superlevel set

$$\boxed{\dot{B}(x,u) \geq -\gamma B(x)}$$

- If $B(x) < 0$, Define $V(x) \coloneqq -B(x)$. Then

$$\dot{V}(x,u) \leq -\gamma V(x).$$

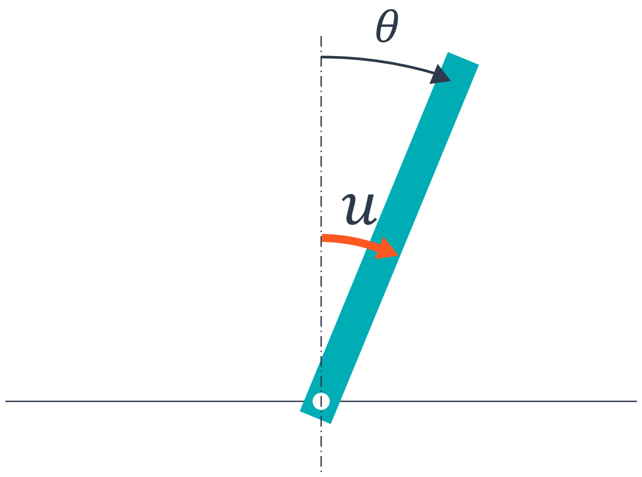
- This is the condition of exponential stability!
- Even if the trajectory exits the safe set accidently, it can promptly recover to the set.

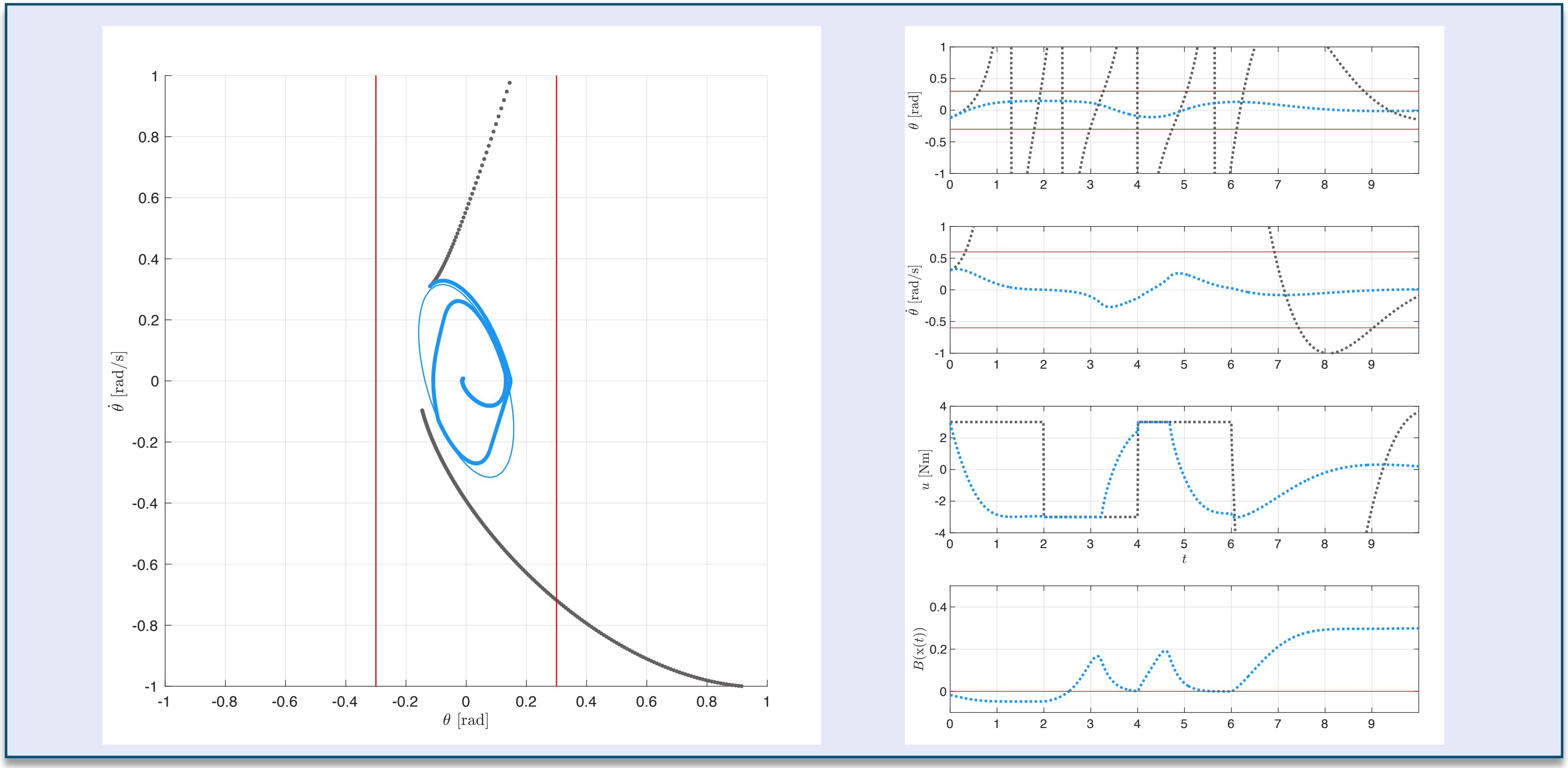
Berkeley UNIVERSITY OF CALIFORNIA

Ames et al, Control Barrier Function Based Quadratic Programs for Safety Critical Systems, TAC'17

Example: Recovery of CBF-QP to the safe set

$$h_S(x) = 1 - x^\top \begin{pmatrix} 1/a^2 & 0.5/ab \\ 0.5/ab & 1/b^2 \end{pmatrix} x \,, \qquad x_0 \notin S$$





Robustness of CBF- 2. Input-to-state safety (ISSf)

- System with bounded disturbance:

$$\dot{\mathrm{x}}(t) = f(\mathrm{x}(t), \mathrm{u}(t))\} + d(t), \ \ |\nabla B(\mathrm{x}(t)) \cdot d(t)| \leq \bar{d}$$

- Let the CBF B satisfy the following property:

$$\sup_{u \in U} \dot{B}(x, u) + \gamma B(x) \geq -\bar{d} \ \text{ for all } \ x \in \mathcal{X}$$

- For all $x \in \mathcal{X}$ such that $B(x) \leq -\bar{d}/\gamma$

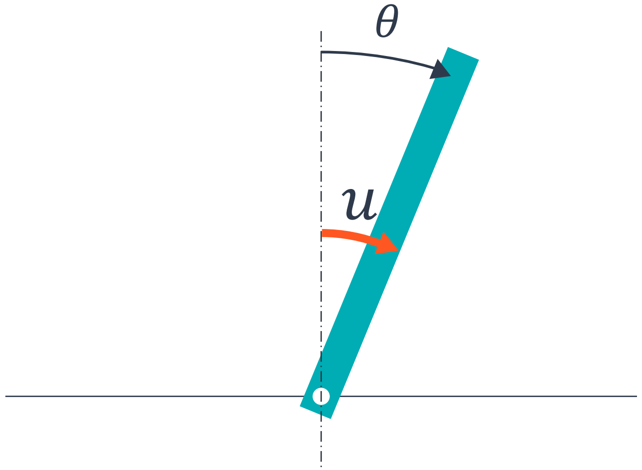
$$\sup_{u \in U} \dot{B}(x, u) \geq -\gamma B(x) - \bar{d} \geq 0.$$

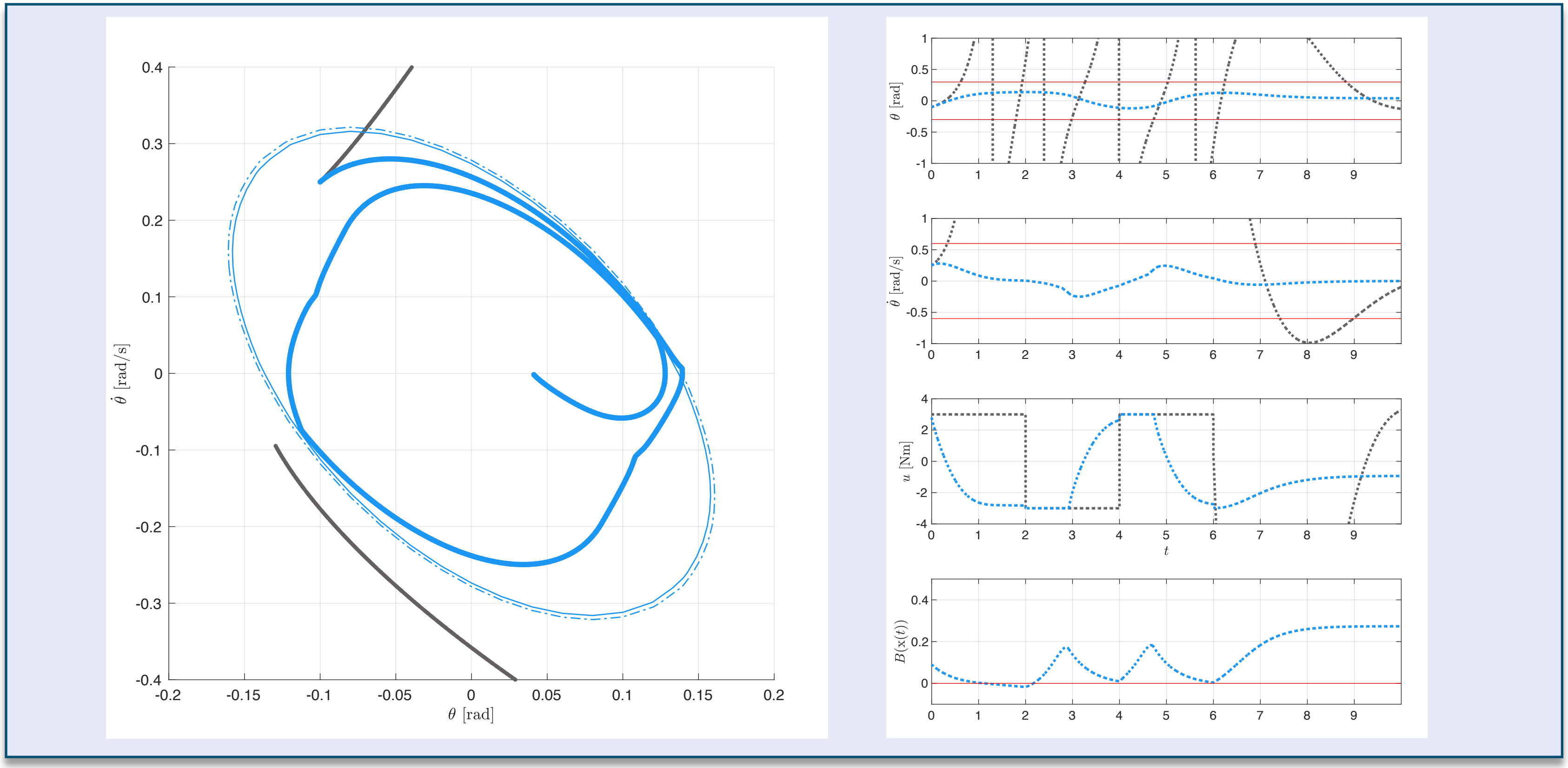
- Thus, according to Nagumo's theorem $\{x \mid B(x) \geq -\bar{d}/\gamma\}$ is control invariant.

Alan et al., Control Barrier Functions and Input-to-State Safety with Application to Automated Vehicles, Arxiv'22

Example: Input-to-state safety of CBF

$$\left|\nabla B\big(\mathrm{x}(t)\big)\cdot d(t)\right| \le \bar{d} = 0.02,\ \ \gamma = 2.0,\ \ -\bar{d}/\gamma = -0.01$$




$\dot{\theta}$ [rad/s]
θ [rad]
θ [rad]
$\dot{\theta}$ [rad/s]
u [Nm]
t
$B(\mathrm{x}(t))$