Advanced Model Predictive Control

Chapter 9
Safety Filter - Part II

Dr. Kim Wabersich / Prof. Melanie Zeilinger

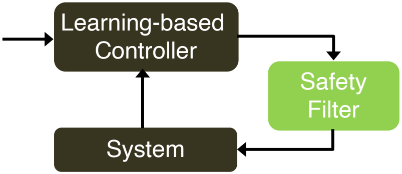
ETH Zurich

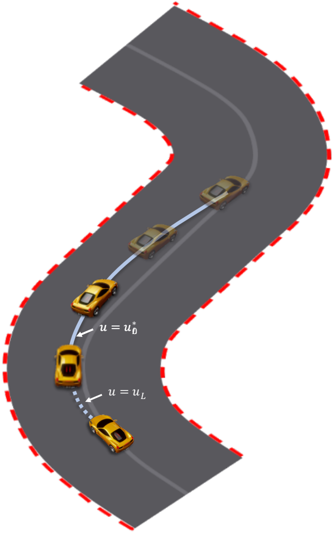
Fall 2024

Coauthors: Dr. Lukas Hewing
Dr. Andrea Carron
Dr. Johannes Köhler

Safety Filter - Main Idea

Add safety verification module to any controller

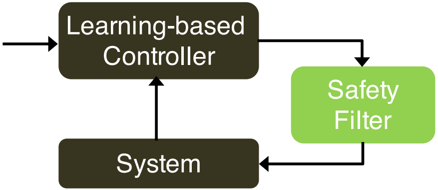


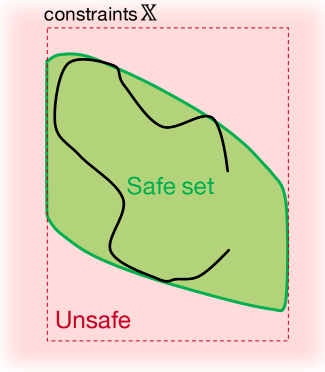


Previous Lecture: Verify control input and modify if required to ensure safety at all future times

Refine Safety Filter Behavior

Consider the controlled system $x(k+1) = f(x(k), u(k))$





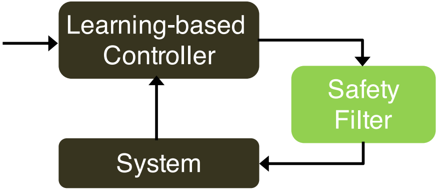
Safety filters from Lecture 8 with switching control law lead to harsh interventions

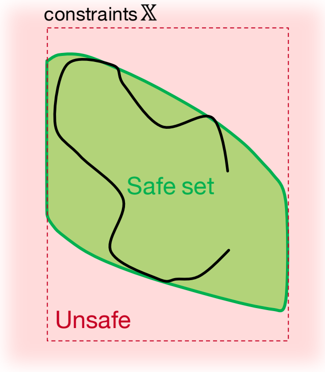
Goal I: Refine safety filter $\pi_{SF}(u_L, x) \in \mathcal{U}$ to

- Interfere as little as necessary: $\min \|\pi_{SF}(u_L, x) - u_L\|$
- Design intervention dynamics close to safe set boundary

Avoid Explicit Design Computation

Consider the controlled system $x(k+1) = f(x(k), u(k))$





Safety filters from Lecture 8 rely on explicit safe set and controller computations, which are often conservative

Goal II: Use model predictive control for safety filters to

- Reduce conservativeness
- Provide principled robust and stochastic design procedures

Learning Objectives

- Familiarization with practical limitations of switching safety filters
- Goal I (Refinements of invariance-based safety filters):
  - Ensure minimal interventions using optimization-based safety filters
  - Design intervention dynamics using control barrier functions
- Goal II (Reduce conservativeness using MPC):
  - Understand benefits of predictive safety filter
  - Understand how to make predictive safety filter idea work in the context of uncertain system models
  - Modify predictive safety filter formulations for different uncertainty assumptions
  - Analyze how to successively enlarge the terminal safe set based on data
- Summary: Overview of different methodologies

Outline

1. Refinements of invariance-based Safety Filters (Goal I)

2. Model Predictive Safety Filter (Goal II)

3. Predictive Safety Filter based on Robust Techniques (Goal II)

4. Examples

5. Successive Improvements of Safety Filter based on Data

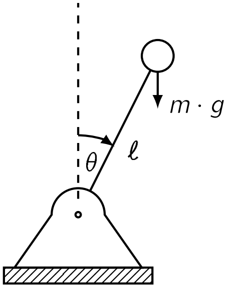
Outline

1. Refinements of invariance-based Safety Filters (Goal I)
2. Model Predictive Safety Filter (Goal II)
3. Predictive Safety Filter based on Robust Techniques (Goal II)
4. Examples
5. Successive Improvements of Safety Filter based on Data

Practical Limitations of Switching Safety Filters

Linearized dynamics of an inverted pendulum at its upward position:

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



- Constraints: $|\theta| \leq 0.3$, $|\dot{\theta}| \leq 0.6$ & input torque constraints $|u| \leq 3$

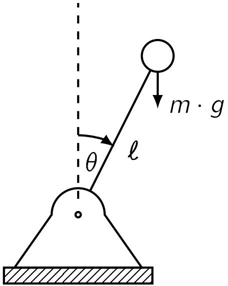
Safety filter for discretization of linearized model (1) at $x = 0$ using LQR:

- $Q = 25I_2$, $R = 1$ yields $J^*(x) = x^\top Px$ under $u = K_{LQR}x$ (Lec 5, p.19)

Practical Limitations of Switching Safety Filters

Linearized dynamics of an inverted pendulum at its upward position:

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



- Constraints: $|\theta| \leq 0.3$, $|\dot{\theta}| \leq 0.6$ & input torque constraints $|u| \leq 3$

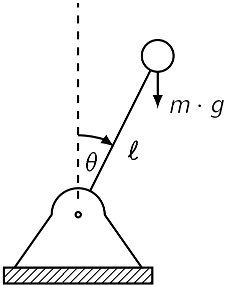
Safety filter for discretization of linearized model (1) at $x = 0$ using LQR:

- $Q = 25I_2$, $R = 1$ yields $J^*(x) = x^\top P x$ under $u = K_{LQR}x$ (Lec 5, p.19)
- Any sublevel set $\mathcal{S}_\alpha = \{x \in \mathbb{R}^2 | x^\top P x \leq \alpha\}$ is invariant (Lec. 8, p.18)

Practical Limitations of Switching Safety Filters

Linearized dynamics of an inverted pendulum at its upward position:

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



- Constraints: $|\theta| \leq 0.3$, $|\dot{\theta}| \leq 0.6$ & input torque constraints $|u| \leq 3$

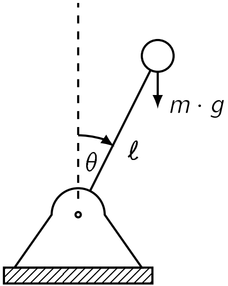
Safety filter for discretization of linearized model (1) at $x = 0$ using LQR:

- $Q = 25I_2$, $R = 1$ yields $J^*(x) = x^\top Px$ under $u = K_{LQR}x$ (Lec 5, p.19)
- Any sublevel set $\mathcal{S}_\alpha = \{x \in \mathbb{R}^2 | x^\top Px \leq \alpha\}$ is invariant (Lec. 8, p.18)
- Select α such that $\mathcal{S}_\alpha \subseteq \mathcal{X}$ and $K_{LQR}\mathcal{S}_\alpha \subseteq \mathcal{U}$ (Lec. 8, p.25)

Practical Limitations of Switching Safety Filters

Linearized dynamics of an inverted pendulum at its upward position:

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



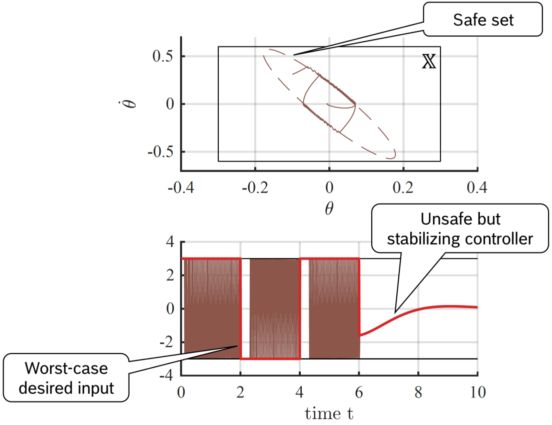
- Constraints: $|\theta| \le 0.3$, $|\dot{\theta}| \le 0.6$ & input torque constraints $|u| \le 3$

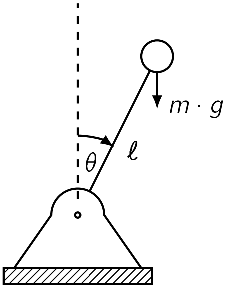
Safety filter for discretization of linearized model (1) at $x = 0$ using LQR:

- $Q = 25 I_2$, $R = 1$ yields $J^*(x) = x^\top P x$ under $u = K_{LQR} x$ (Lec 5, p.19)
- Any sublevel set $\mathcal{S}_\alpha = \{x \in \mathbb{R}^2 | x^\top P x \le \alpha\}$ is invariant (Lec. 8, p.18)
- Select α such that $\mathcal{S}_\alpha \subseteq \mathcal{X}$ and $K_{LQR}\mathcal{S}_\alpha \subseteq \mathcal{U}$ (Lec. 8, p.25)

$$\pi_{SF}(u_L, x) = \begin{cases} u_L & \text{if } (Ax + Bu_L)^\top P (Ax + Bu_L) \le \alpha,\ |u_L| \le 3 \\ K_{LQR} x & \text{otherwise} \end{cases}$$

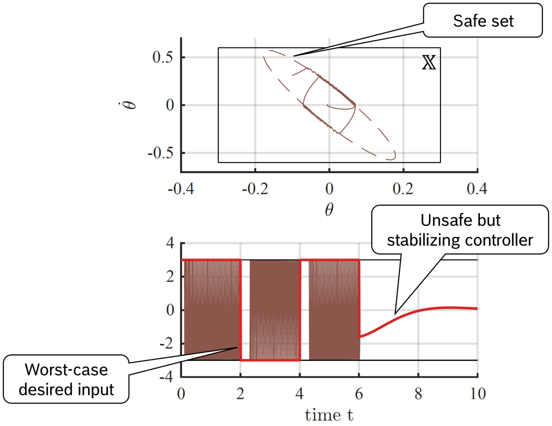
Practical Limitations of Switching Safety Filters

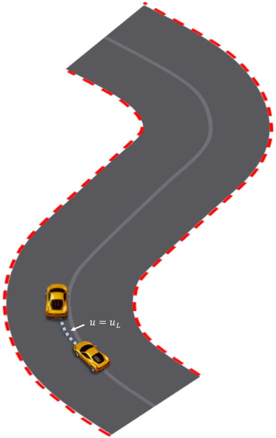




- (Top) Solid brown line: $x(k)$, Dashed brown line: $\partial \mathcal{S}_\alpha$
- (Bottom) Brown line: Input $\pi_{SF}(u_L, x)$, Red line: u_L

Practical Limitations of Switching Safety Filters



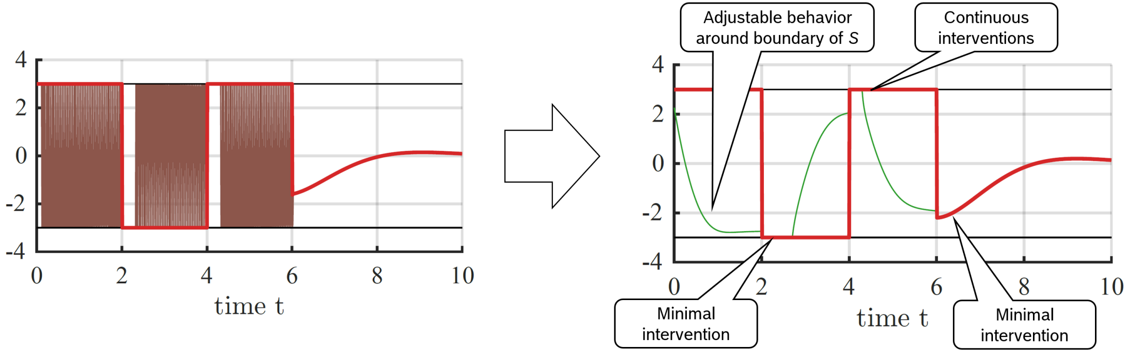


- High-frequency bang-bang input interventions cause stress on actuators and system
- Vehicle example: Would you like to sit in a car with bang-bang interventions?

Refinement goals for Switching Safety Filter

Refine the switching safety filter to provide the following properties:

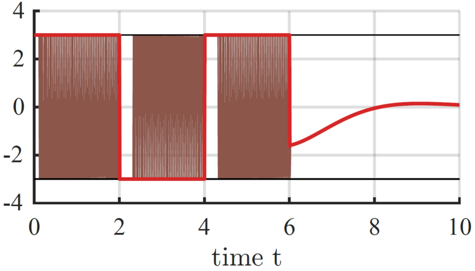
1. Minimal and continuous interventions with respect to u_L
2. Adjustable behavior when reaching the safe set boundary and safety interventions begin

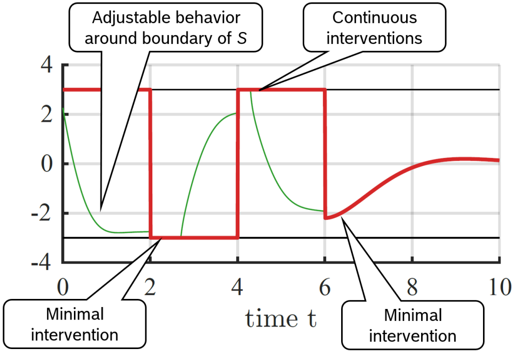


Refinement goals for Switching Safety Filter

Two additional mechanisms

1. Change desired input only as little as necessary
2. Introduce damping close to constraints using 'Control Barrier Function' concepts





Refine Switching Filter: Minimal interventions

Switching Safety Filter

$$\pi_{SF}(u_L(k), x) = \begin{cases} \text{learning input } u_L(k) & \text{if } f(x, u_L(k)) \in \mathcal{S}, u_L(k) \in \mathcal{U} \\ \text{safe input } \pi_s(x) & \text{otherwise} \end{cases}$$

Pro: Low computation
Con: Harsh intervention

Minimally Invasive Safety Filter

$\pi_{SF}(u_L(k), x) = u_0^*$ with u_0^* being the solution to:

$$\begin{aligned} \min_{u_0} \quad & \|u_0 - u_L(k)\| && \text{(2a)} \\ \text{s.t.} \quad & f(x, u_0) \in \mathcal{S} && \text{(2b)} \\ & u_0 \in \mathcal{U} && \text{(2c)} \end{aligned}$$

Pro: Minimal modifications
Con: Optimization problem

Refine Switching Filter: Minimal interventions

Switching Safety Filter

$$\pi_{SF}(u_L(k), x) = \begin{cases} \text{learning input } u_L(k) & \text{if } f(x, u_L(k)) \in \mathcal{S}, u_L(k) \in \mathcal{U} \\ \text{safe input } \pi_s(x) & \text{otherwise} \end{cases}$$

Pro: Low computation
Con: Harsh intervention

Minimally Invasive Safety Filter

$\pi_{SF}(u_L(k), x) = u_0^*$ with u_0^* being the solution to:

$$\begin{aligned} \min_{u_0} \quad & \|u_0 - u_L(k)\| && \text{(2a)} \\ \text{s.t.} \quad & f(x, u_0) \in \mathcal{S} && \text{(2b)} \\ & u_0 \in \mathcal{U} && \text{(2c)} \end{aligned}$$

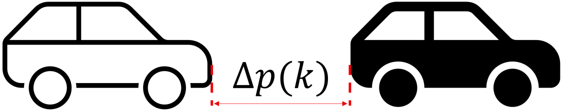
Pro: Minimal modifications
Con: Optimization problem

Reduce (2) to line search by convex parametrization $u_0 = \lambda u_L(k) + (1 - \lambda)\pi_S(x)$, $\lambda \in [0, 1]$

- Feasible candidate solution for all $x \in \mathcal{S}$: $\lambda^* = 0$
- Use gridding or line-search algorithms to determine optimal λ

Refine Switching Filter: Damping at Safe Set Boundary

'Braking assistant': Unsafe desired control input $u_L(k) > 0$



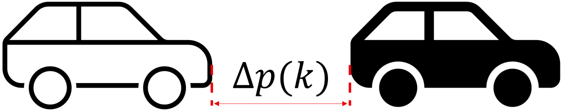
Model dynamics:

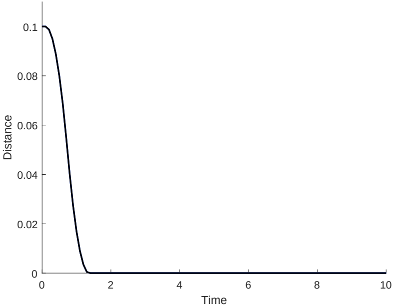
$$\underbrace{\begin{bmatrix} \Delta p(k+1) \\ \Delta v(k+1) \end{bmatrix}}_{x(k+1)} = \underbrace{\begin{bmatrix} 1 & T_s \\ 0 & 1 \end{bmatrix}}_{A} \underbrace{\begin{bmatrix} \Delta p(k) \\ \Delta v(k) \end{bmatrix}}_{x(k)} + \underbrace{\begin{bmatrix} 0 \\ T_s \end{bmatrix}}_{B} \underbrace{\Delta a(k)}_{u(k)}, \tag{3}$$

- Relative position, velocity, and acceleration to leading vehicle $\Delta p(k)$, $\Delta v(k)$, and $\Delta a(k)$
- Constraints: $\Delta p(k) \geq 0$, $|\Delta a(k)| \leq 1$

Refine Switching Filter: Damping at Safe Set Boundary

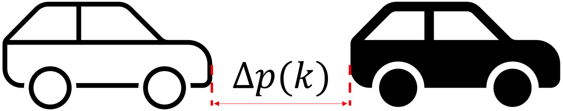
'Braking assistant': Unsafe desired control input $u_L(k) > 0$

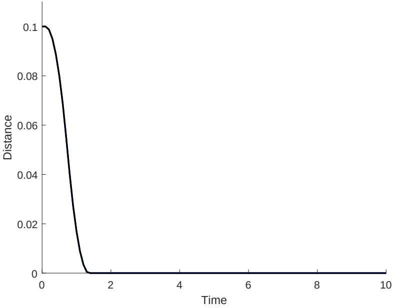


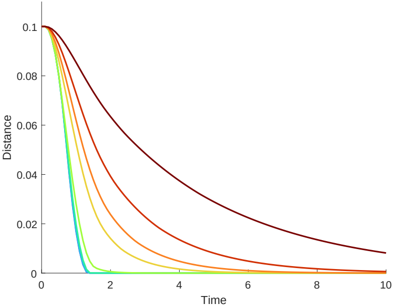


Refine Switching Filter: Damping at Safe Set Boundary

'Braking assistant': Unsafe desired control input $u_L(k) > 0$







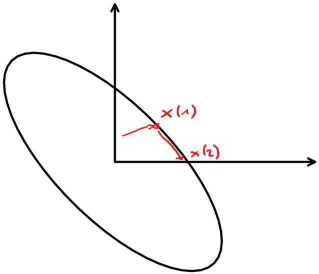
How to allow for tuning the intervention characteristics?

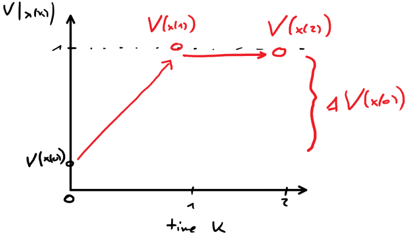
Refine Switching Filter: Damping at Safe Set Boundary

Let $\mathcal{S} = \{x | V(x) \leq 1\}$ be a safe set with $V(x) = x^\top P x \leq 1$ (E.g. LQR sublevel set)

Safety filter using level-set function:

$$
\begin{aligned}
\min_{u_0} \quad & \|u_0 - u_L(k)\| && \text{(3a)}\\
\text{s.t.} \quad & V(f(x, u_0)) \leq 1 && \text{(3b)}\\
& u_0 \in \mathcal{U} && \text{(3c)}
\end{aligned}
$$



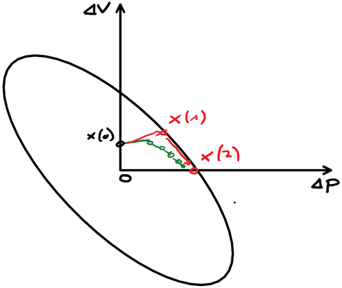


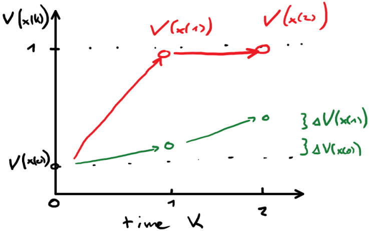
Refine Switching Filter: Damping at Safe Set Boundary

Let $\mathcal{S} = \{x | V(x) \leq 1\}$ be a safe set with $V(x) = x^\top P x \leq 1$ (E.g. LQR sublevel set)

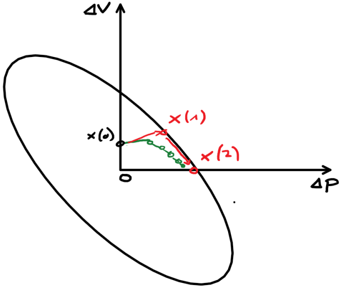
Safety filter using level-set function:

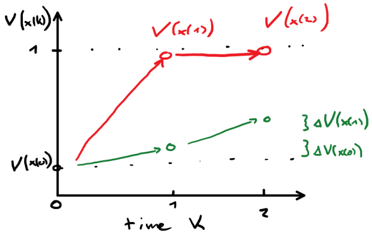
$$\begin{aligned} \min_{u_0} \quad & \|u_0 - u_L(k)\| && \text{(3a)} \\ \text{s.t.} \quad & V(f(x, u_0)) \leq 1 && \text{(3b)} \\ & u_0 \in \mathcal{U} && \text{(3c)} \end{aligned}$$





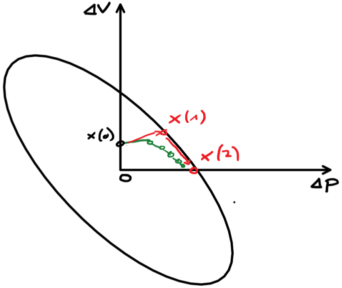
Refine Switching Filter: Damping at Safe Set Boundary

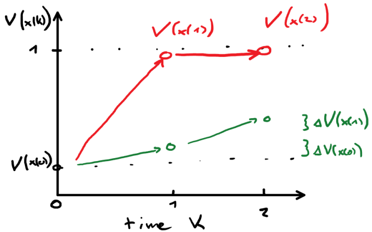




Restrict (damp) $\Delta V(x(k)) = V(x(k+1)) - V(x(k))$ by modifying (3b):

Refine Switching Filter: Damping at Safe Set Boundary

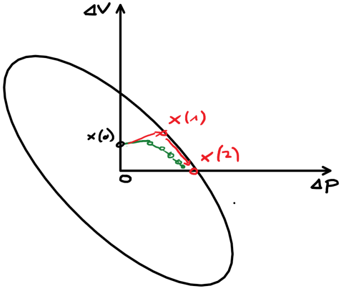


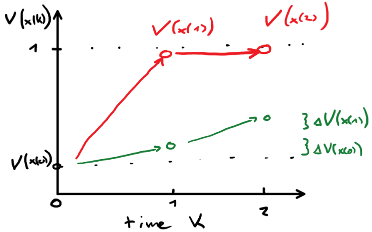


Restrict (damp) $\Delta V(x(k)) = V(x(k+1)) - V(x(k))$ by modifying (3b):

$$V(f(x,u_0)) \leq \color{red}{1} \Leftrightarrow \underbrace{V(f(x,u_0)) - V(x)}_{\Delta V(x,u_0)} \leq \underbrace{\color{red}{1 - V(x)}}_{\text{can be reduced}}$$

Refine Switching Filter: Damping at Safe Set Boundary





Restrict (damp) $\Delta V(x(k)) = V(x(k+1)) - V(x(k))$ by modifying (3b):

$$V(f(x,u_0)) \le 1 \Leftrightarrow \underbrace{V(f(x,u_0)) - V(x)}_{\Delta V(x,u_0)} \le \underbrace{1 - V(x)}_{\text{can be reduced}} \Leftarrow \Delta V(x,u_0) \le \underbrace{\gamma}_{\gamma\in(0,1]} (1 - V(x))$$

Example: $\gamma = 0.5$, $V(x(k)) = 0.5$: $V(x(k+1)) - V(x(k)) \le 0.25$ versus ≤ 0.5 for $\gamma = 1$ in (3b).

Refine Switching Filter: Damping at Safe Set Boundary

Discrete-Time Exponential Control Barrier Function (Ayush and Sreenath 2017)

A function V is a discrete-time exponential control barrier function if for every $x \in \mathcal{S}$ with $\mathcal{S} = \{x \in \mathbb{R}^n | V(x) \leq 1\}$ there exists a control input $u \in \mathcal{U}$ such that

$$V(f(x,u)) - V(x) \leq \gamma(1 - V(x)) \text{ for all } \gamma \in (0,1]. \tag{4}$$

- Since for all $x \in \mathcal{S}$ we have $\gamma(1 - V(x)) \geq 0$ any invariant set from a Lyapunov function satisfies (4), see Lec. 8, p.18. For example an LQR cost level set.

Control Barrier Function Safety Filter

Let V be a discrete-time exponential control barrier function. The corresponding safety filter is given by $\pi_{SF}(u_L(k), x) = u_0^*$ with u_0^* being an optimal solution to

$$\begin{aligned} \min_{u_0} \quad & \|u_0 - u_L(k)\| && \text{(5a)} \\ \text{s.t.} \quad & V(f(x,u_0)) - V(x) \leq \gamma(1 - V(x)) && \text{(5b)} \\ & u_0 \in \mathcal{U} && \text{(5c)} \end{aligned}$$

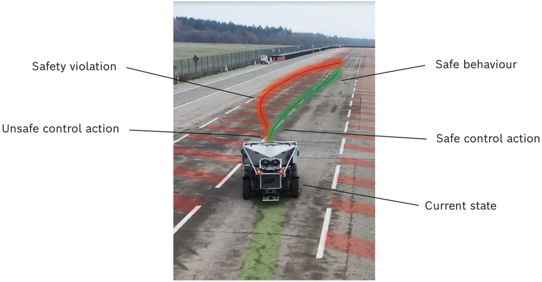
Summary of Refinements for Safety Filters

Pros:

- Efficient implementation possible
- Safety intervention can be tuned using control barrier functions
- Applications in robotics and autonomous driving

Cons:

- Difficult to compute safe set or control barrier function for nonlinear systems
- Robustness by design is difficult



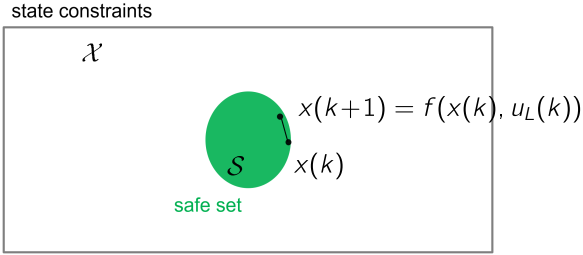
`[LINK] Lane-keep using refined safety filter`

Outline

1. Refinements of invariance-based Safety Filters (Goal I)
2. Model Predictive Safety Filter (Goal II)
3. Predictive Safety Filter based on Robust Techniques (Goal II)
4. Examples
5. Successive Improvements of Safety Filter based on Data

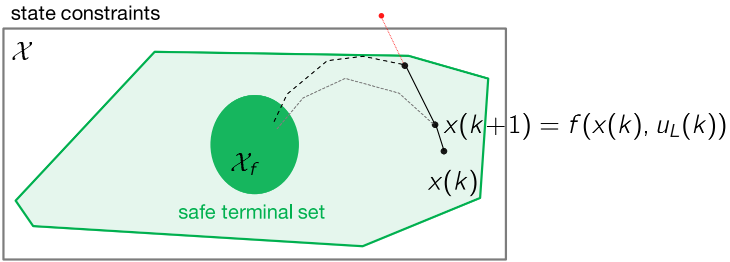
From Invariant sets to Predictive Safety Filters

$$\begin{aligned} \min \quad & \|u_0 - u_L(k)\| \\ \text{s.t.} \quad & f(x(k), u_0) \in \mathcal{S} \\ & u_0 \in \mathcal{U} \end{aligned}$$



Idea: Verify safety by planning safe forward trajectory.

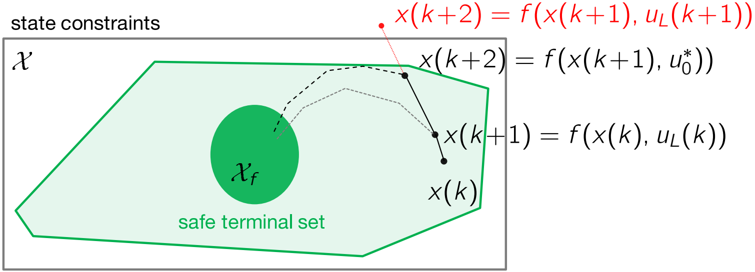
Main Idea of Predictive Safety Filter



Idea: Verify safety by planning safe forward trajectory.

- Use conservative safe set $\mathcal{S}$ as a target set $\mathcal{X}_f$

Main Idea of Predictive Safety Filter

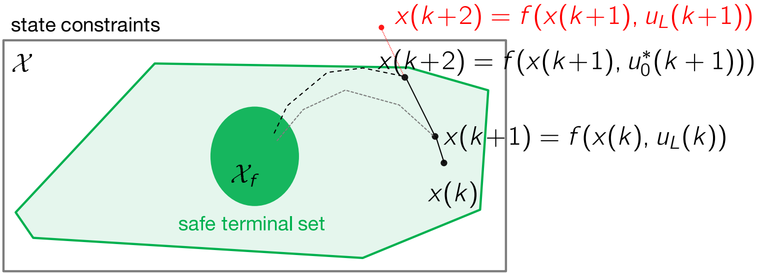


Idea: Verify safety by planning safe forward trajectory.

- Use conservative safe set $\mathcal{S}$ as a target set $\mathcal{X}_f$
- Existence of safety trajectory at time step k ensures safety at time step $k+1$: Could follow safety trajectory computed at time k if no new safety trajectory can be found.

Combined Verification and Safety Controller

$$\begin{aligned} \min \quad & \|u_0 - u_L(k)\| \\ \text{s.t.} \quad & x_{i+1} = f(x_i, u_i) \\ & (x_i, u_i) \in \mathcal{X} \times \mathcal{U} \\ & x_N \in \mathcal{X}_f \\ & x_0 = x(k) \end{aligned}$$



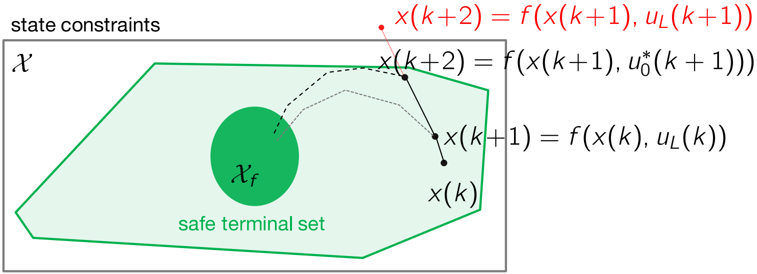
Predictive Safety Filter:

- If learning input is safe: $u_0^*(k) = u_L(k)$
- If learning input is not safe: $u_0^*(k) =$ 'closest' safe input

Assumption: $\mathcal{X}_f$ satisfies regular assumptions on MPC terminal set (invariance under local control law, state and input constraint satisfaction).

Combined Verification and Safety Controller

$$
\begin{aligned}
\min \quad & \|u_0 - u_L(k)\| \\
\text{s.t.} \quad & x_{i+1} = f(x_i, u_i) \\
& (x_i, u_i) \in \mathcal{X} \times \mathcal{U} \\
& x_N \in \mathcal{X}_f \\
& x_0 = x(k)
\end{aligned}
$$



Predictive Safety Filter:

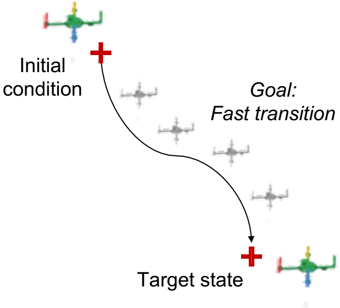
- If learning input is safe: $u_0^*(k) = u_L(k)$
- If learning input is not safe: $u_0^*(k) =$ 'closest' safe input

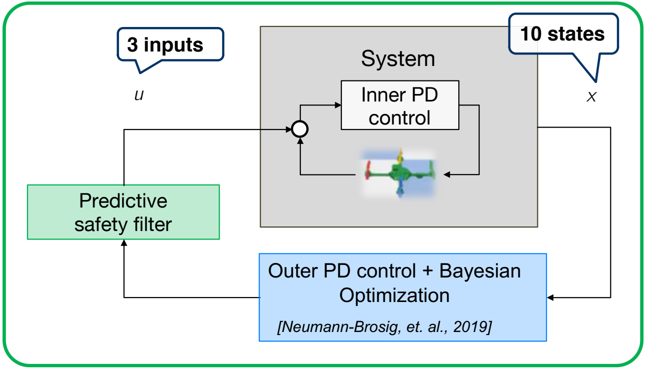
$\rightarrow$ Feasible set is a **safe set**.

$\rightarrow$ Implicit safe set & controller

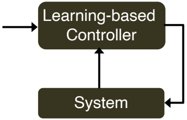
$\rightarrow$ Efficient solvers available

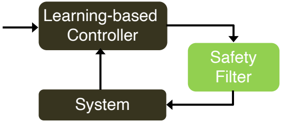
Application Example: Learning to Fly a Quadrotor

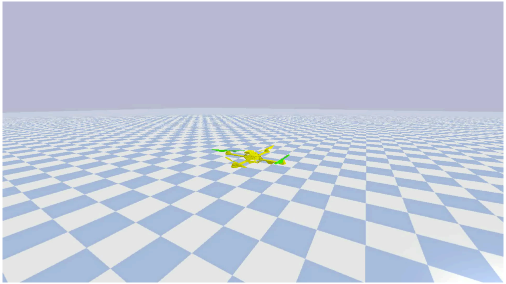




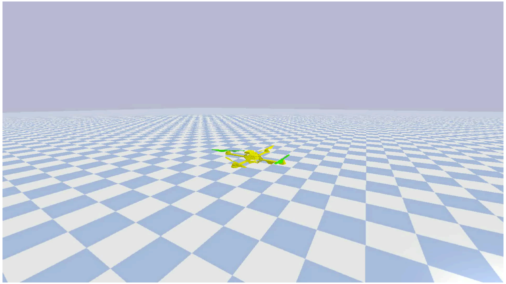
Application Example: Learning to Fly a Quadrotor





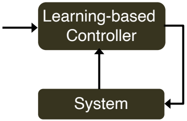


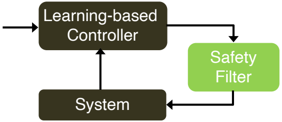
Video: 02_QUADROTOR_UNSAFE

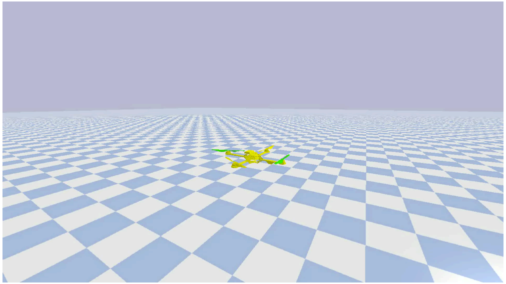


Video: 03_QUADROTOR_SAFE

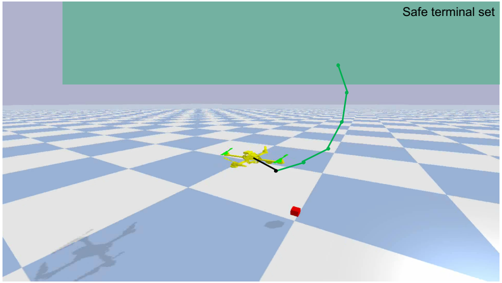
Application Example: Learning to Fly a Quadrotor





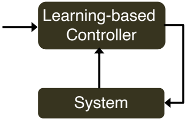


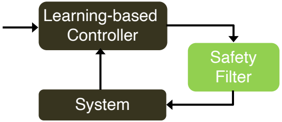
Video: 02_QUADROTOR_UNSAFE

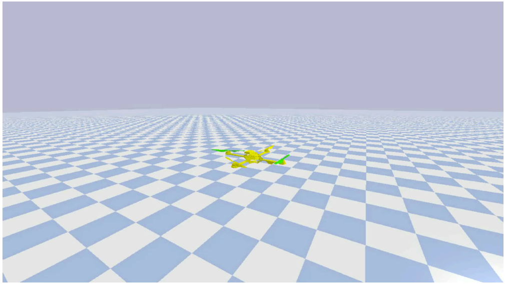


Video: 03_QUADROTOR_SAFE

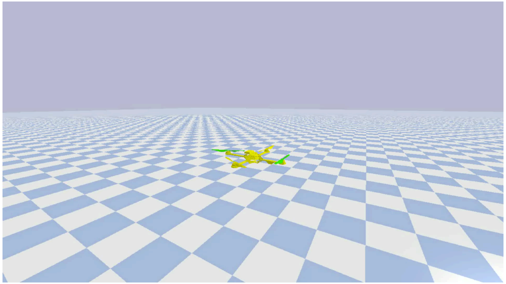
Application Example: Learning to Fly a Quadrotor





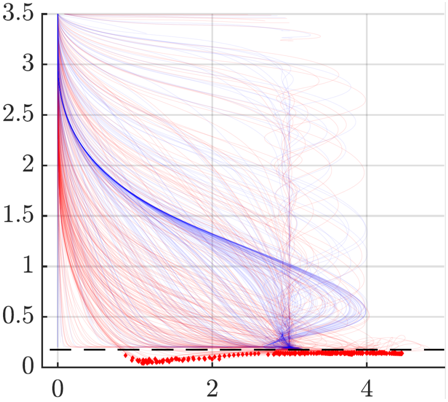


Video: 02_QUADROTOR_UNSAFE

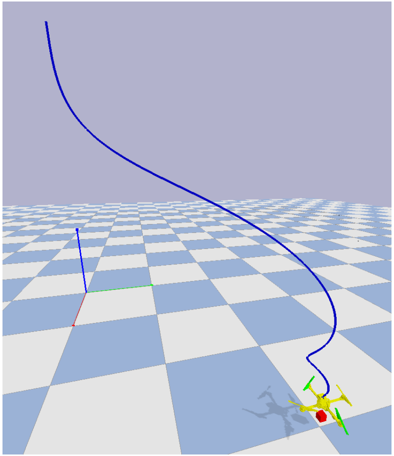


Video: 03_QUADROTOR_SAFE

Application Example: Learning to Fly a Quadrotor



Blue: With safety filter
Red: Without safety filter



Remark: Effects of safety filter on performance

- Filter interventions can negatively affect performance of learning (or other) algorithm
- Common approach: Inclusion of safety metric in learning controller
  - Based on safety value function $V(x)$, defining safe set $S = \{x \in \mathbb{R}^n \mid V(x) \leq 1\}$
    Example: Add barrier term to cost used in learning algorithm

    $$C_s(x, u) = C(x, u) - \gamma log(1 - V(x))$$

    (barrier goes to infinity at boundary of safe set)
  - Based on distance to learning input (e.g. for predictive filter)

    $$C_s(x, u) = C(x, u) - \gamma \|u(k) - u_L(k))\|^2$$

Comparison Predictive vs. Invariance-based Safety Filter

Invariance-based and control barrier function safety filters

+ Scales to high-dimensional systems
+ Cheap online verification of safety and computation of safety controller
+ If safe set is described by a control barrier function, intervention characteristics can be tuned
- Conservative safe set and interventions

Predictive safety filter

+ Scales to high-dimensional systems
+ Reduced conservatism w.r.t. maximum control invariant set

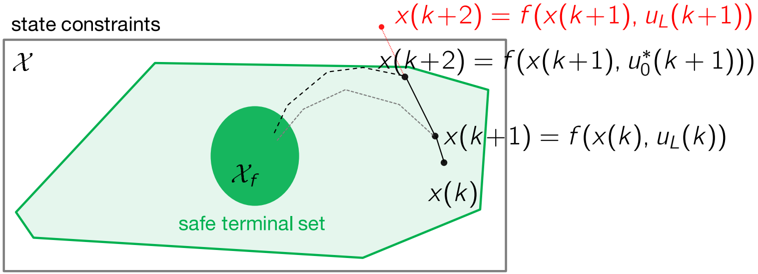
(+) (Conceptually) easily extended to nonlinear systems

- Requires solution of an optimization problem

Model Predictive Safety Filter

Safety filter

$$\begin{aligned} \min \quad & \|u_0 - u_L\| \\ \text{s.t.} \quad & x_{i+1} = f(x_i, u_i) \\ & (x_i, u_i) \in \mathcal{X} \times \mathcal{U} \\ & x_N \in \mathcal{X}_f \\ & x_0 = x(k) \end{aligned}$$



- If learning input is safe: $u_0^*(k) = u_L(k)$
- If learning input is not safe: $u_0^*(k) =$ 'closest' safe input

Main challenge: Safety depends on perfect model without disturbances.

Outline

1. Refinements of invariance-based Safety Filters (Goal I)
2. Model Predictive Safety Filter (Goal II)
3. Predictive Safety Filter based on Robust Techniques (Goal II)
4. Examples
5. Successive Improvements of Safety Filter based on Data

Predictive Safety Filter and Robust/Stochastic MPC

Key insight: Different objective and terminal set without consideration of the cost function

Safety filter

$$
\begin{aligned}
\min \quad & \underbrace{\|u_0 - u_L(k)\|}_{J_{PSF}} \\
\text{s.t.} \quad & x_{i+1} = f(x_i, u_i) \\
& (x_i, u_i) \in \mathcal{X} \times \mathcal{U} \\
& x_N \in \mathcal{X}_f \\
& x_0 = x(k)
\end{aligned}
$$

Model predictive controller

$$
\begin{aligned}
\min \quad & \underbrace{V_f(x_N) + \Sigma_{i=0} l(x_i, u_i)}_{J_{MPC}} \\
\text{s.t.} \quad & x_{i+1} = f(x_i, u_i) \\
& (x_i, u_i) \in \mathcal{X} \times \mathcal{U} \\
& x_N \in \mathcal{X}_f \\
& x_0 = x(k)
\end{aligned}
$$

- Robust MPC considering bounded additive disturbance: (Lec. 3, p.30, Lec. 3, p.43),
- Robust MPC supporting state and input dependent disturbances (Lec. 4)

safety does not rely on J_{MPC}.

Predictive Safety Filter and Robust/Stochastic MPC

Key insight: Different objective and terminal set without consideration of the cost function

Safety filter	Model predictive controller
$$\begin{aligned} \min \quad & \underbrace{\lVert u_0 - u_L(k) \rVert}_{J_{PSF}} \\ \text{s.t.} \quad & x_{i+1} = f(x_i, u_i) \\ & (x_i, u_i) \in \mathcal{X} \times \mathcal{U} \\ & x_N \in \mathcal{X}_f \\ & x_0 = x(k) \end{aligned}$$	$$\begin{aligned} \min \quad & \underbrace{V_f(x_N) + \Sigma_{i=0} l(x_i, u_i)}_{J_{MPC}} \\ \text{s.t.} \quad & x_{i+1} = f(x_i, u_i) \\ & (x_i, u_i) \in \mathcal{X} \times \mathcal{U} \\ & x_N \in \mathcal{X}_f \\ & x_0 = x(k) \end{aligned}$$

- Robust MPC considering bounded additive disturbance: (Lec. 3, p.30, Lec. 3, p.43),
- Robust MPC supporting state and input dependent disturbances (Lec. 4)

safety does not rely on J_{MPC}. Similarly, this holds for the stochastic MPC schemes Lec. 7, p. 34/43.

Exchange the objectives with J_{PSF} to obtain corresponding safety filters - How?

Insert Safety Filter Objective into Robust/Stochastic MPC schemes

Robust MPC with constraint tightening/Stochastic MPC with recovery mechanism:

- Applied input in original MPC scheme: $u(k) = u_0^*$
- Keep the original predictive safety filter objective $J_{PSF} = \|u_0 - u_L(k)\|$

Insert Safety Filter Objective into Robust/Stochastic MPC schemes

Robust MPC with constraint tightening/Stochastic MPC with recovery mechanism:

- Applied input in original MPC scheme: $u(k) = u_0^*$
- Keep the original predictive safety filter objective $J_{PSF} = \|u_0 - u_L(k)\|$

Tube-based MPC schemes and indirect stochastic MPC:

- Applied input in original MPC scheme: $u(k) = v_0^* + K(x(k) - z_0^*)$
- Select $J_{PSF} = \|v_0^* + K(x(k) - z_0^*) - u_L(k)\|$ to match desired input while maintaining guarantees

Insert Safety Filter Objective into Robust/Stochastic MPC schemes

Robust MPC with constraint tightening/Stochastic MPC with recovery mechanism:

- Applied input in original MPC scheme: $u(k) = u_0^*$
- Keep the original predictive safety filter objective $J_{PSF} = \|u_0 - u_L(k)\|$

Tube-based MPC schemes and indirect stochastic MPC:

- Applied input in original MPC scheme: $u(k) = v_0^* + K(x(k) - z_0^*)$
- Select $J_{PSF} = \|v_0^* + K(x(k) - z_0^*) - u_L(k)\|$ to match desired input while maintaining guarantees

This can be applied to various MPC techniques for considering uncertain (data-driven) models. The concepts can equally be applied to distributed systems.

Let's have a closer look at tube-based MPC as one example.

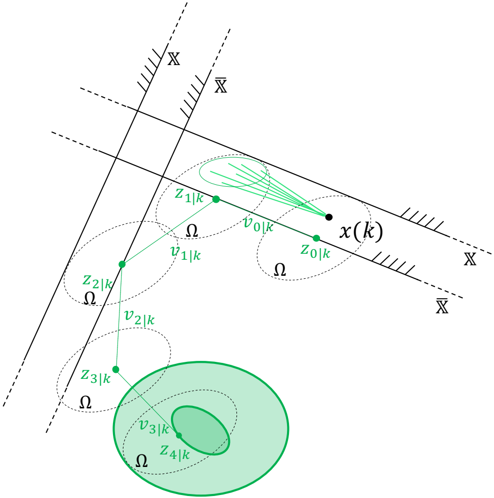
Safety Filter based on Robust MPC

- Uncertain model with uniform error bound:

$$x(k+1) = f(x(k), u(k)) + w(k), w(k) \in \mathcal{W}$$

- Plan nominal trajectory and apply tracking controller
- Tighten constraints on nominal system

$\rightarrow$ Apply robust MPC methods, such as linear or nonlinear tube-based MPC (Lecture 3/4)



Safety Filter based on Linear Tube-Based MPC

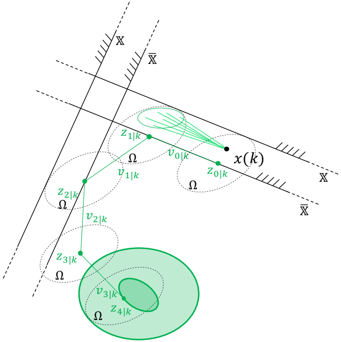
Uncertain system: $x(k+1) = Ax(k) + Bu(k) + w(k),\ w(k) \in \mathcal{W}\ \forall k \geq 0$

Nominal system: $z(k+1) = Az(k) + Bv(k)$

Tracking controller: $u(k) = v(k) + K(x(k) - z(k))$

Error system: $e(k+1) = (A + BK)e(k) + w(k),\ w(k) \in \mathcal{W}\ \forall k \geq 0$

$$
\begin{aligned}
\min \quad & \|u_{0|k} - u_L(k)\| \\
s.t. \quad & u_{0|k} = v_{0|k} + K(x(k) - z_{0|k}) \\
& z_{i+1|k} = Az_{i|k} + Bv_{i|k} \\
& z_{i|k} \in \mathcal{X} \ominus \Omega,\ v_{i|k} \in \mathcal{U} \ominus K\Omega \\
& z_{N|k} \in \mathcal{X}_f \\
& x(k) \in z_{0|k} \oplus \Omega
\end{aligned}
$$



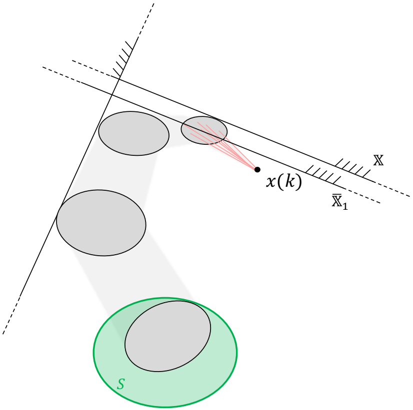
The closed-loop system with $u(k) = v^*_{0|k} + K(x(k) - z^*_{0|k})$ satisfies constraints robustly.

Safety Filter based on Linear Constraint Tightening MPC

Uncertain system: $x(k+1) = Ax(k) + Bu(k) + w(k),\ w(k) \in \mathcal{W}\ \forall k \geq 0$

Nominal system: $z(k+1) = Az(k) + Bv(k)$

$$
\begin{aligned}
\min \quad & \|v_{0|k} - u_L(k)\| \\
s.t. \quad & z_{i+1|k} = Az_{i|k} + Bv_{i|k} \\
& z_{i|k} \in \mathcal{X} \ominus \mathcal{F}_i,\ v_{i|k} \in \mathcal{U} \ominus K\mathcal{F}_i \\
& z_{N|k} \in \mathcal{X}_f \ominus \mathcal{F}_N \\
& z_{0|k} = x(k)
\end{aligned}
$$



The closed-loop system with $u(k) = v^*_{0|k}$ satisfies constraints robustly.

Outline

1. Refinements of invariance-based Safety Filters (Goal I)

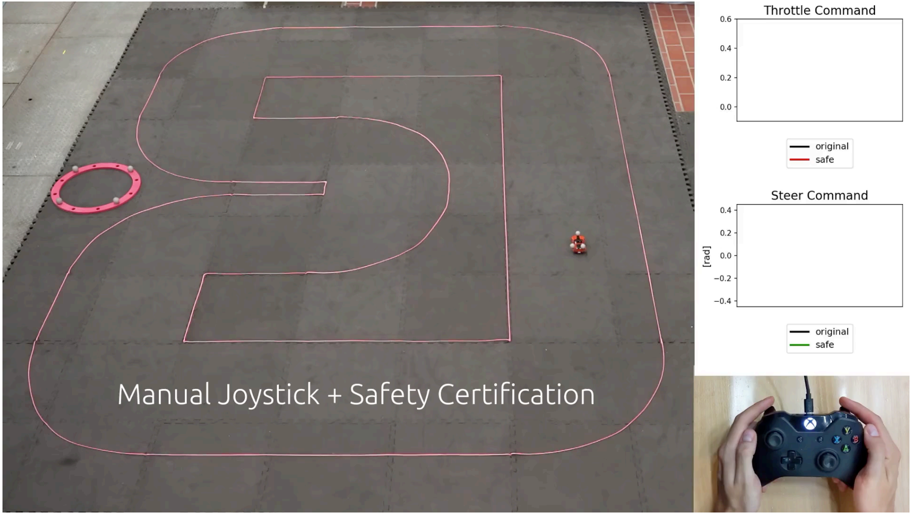
2. Model Predictive Safety Filter (Goal II)

3. Predictive Safety Filter based on Robust Techniques (Goal II)

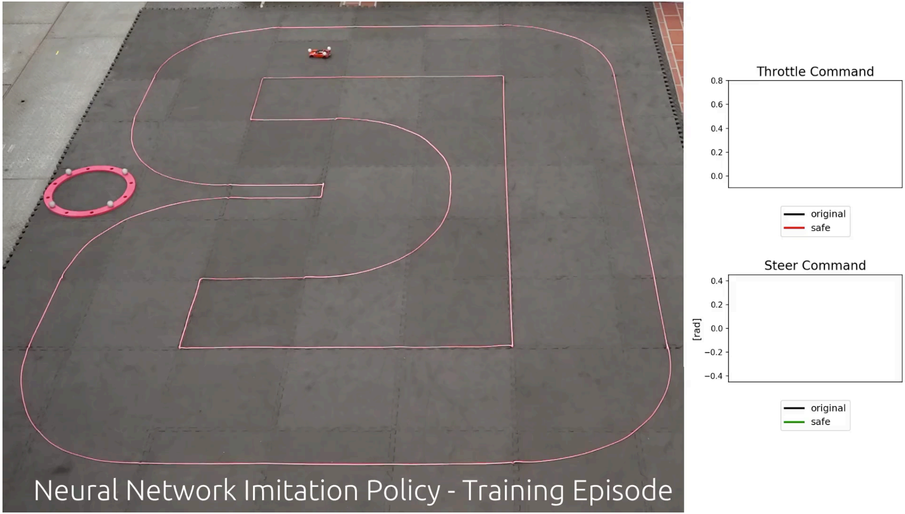
4. Examples

5. Successive Improvements of Safety Filter based on Data

Safety filter for a Human Learner



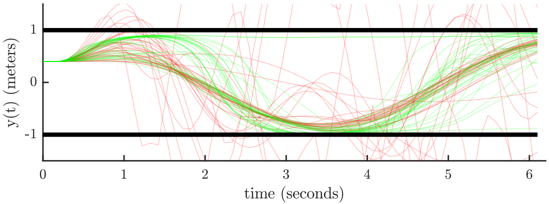
Safety filter for Imitation Learning

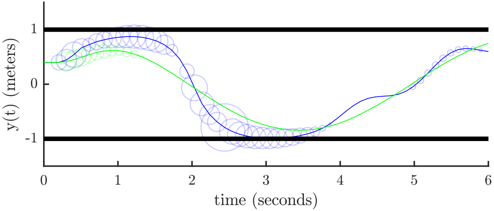


Safely Learning to Track a Trajectory with a Car

Goal: Learn to drive a simulated autonomous car along a desired trajectory without leaving a road.

Learning controller: Learn linear control law (via Bayesian optimization)





- First 30 learning episodes without (red) and with (green) the stochastic safety filter.
- Safety constraints shown in black.

- Safe closed-loop trajectories during learning with initial policy parameters (blue) and final policy parameters (green).
- Circle radii indicate the relative magnitude of safety ensuring modifications.

Outline

1. Refinements of invariance-based Safety Filters (Goal I)
2. Model Predictive Safety Filter (Goal II)
3. Predictive Safety Filter based on Robust Techniques (Goal II)
4. Examples
5. Successive Improvements of Safety Filter based on Data

Outline

5. Successive Improvements of Safety Filter based on Data

   Improvement of Terminal Safe Set

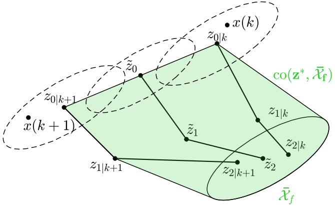
   Improvement of Model and Uncertainty Bounds

Online Enlargement based on Safe States

A larger terminal safe set leads to a larger implicit safe set.

Idea: Leverage previously encountered safe states to enlarge terminal safe set

Consider the set of nominal predicted states obtained from successfully solved instances of the safety filter problem: $\mathbf{z}^*(k) = \{z^*_{j|i}, i \in [1, \ldots, k], j \in [0, \ldots, N]\}$

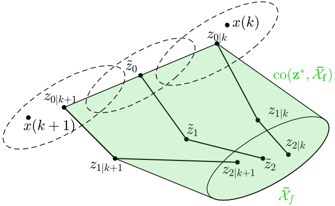


Online Enlargement based on Safe States

A larger terminal safe set leads to a larger implicit safe set.

Idea: Leverage previously encountered safe states to enlarge terminal safe set

Consider the set of nominal predicted states obtained from successfully solved instances of the safety filter problem: $\mathbf{z}^*(k) = \{z^*_{j|i}, i \in [1, \ldots, k], j \in [0, \ldots, N]\}$



If the predictive safety filter problem is convex and $\bar{\mathcal{X}}_f$ is a safe terminal set, then the set

$$\bar{\mathcal{X}}_f^k = \mathrm{co}(\mathbf{z}^*(k), \bar{\mathcal{X}}_f) \quad \text{(where co}(\cdot)\text{ denotes the convex hull)}$$

is again a safe terminal set, i.e. it is invariant for the nominal system under a terminal safe control law and all tightened state and input constraints are satisfied.

Outline

5. Successive Improvements of Safety Filter based on Data

    Improvement of Terminal Safe Set

    Improvement of Model and Uncertainty Bounds

Learning-based Model and Uncertainty Bounds

Safety filter for learning-based controllers (e.g. RL):

- If learning happens in episodes
  - Improve filter design after every episode to reduce conservatism
  - Safety filters for uncertain systems usually rely on nominal model

    $\rightarrow$ improve nominal model based on collected data, update uncertainty bound
  - Data-based computation methods for constraint tightening e.g. based on scenario optimization
- If learning happens online
  Any learning-based MPC method can be employed that ensures recursive feasibility and constraint satisfaction (see e.g. [5])
- See upcoming lectures 12 and 14

Summary: Predictive Safety Filter

Idea: Approximate maximum control invariant set with a predictive optimal control problem.

- Implicit optimization-based formulation
- Controller verification and backup controller computation in one problem
- Scalable to high-dimensional systems
- Leverage stochastic and robust MPC formulations to deal with model uncertainty
- Leverage efficient MPC solvers

$\Rightarrow$ Provide safety for any controller by passing proposed input through safety filter

Note: Extensions to reduce conservatism of robust MPC in [3] and to distributed systems in [4].

Not covered: Soft constraints, control barrier functions inherently stabilize the safe set

References

[1] K.P. Wabersich and M.N. Zeilinger, “Linear model predictive safety certification for learning-based control”, CDC 2018.

[2] K.P. Wabersich, L. Hewing, A. Carron and M.N. Zeilinger, “Probabilistic model predictive safety certification for learning-based control”, IEEE Transactions on Automatic Control, 2021.

[3] K.P. Wabersich and M. N. Zeilinger, “A predictive safety filter for learning-based control of constrained nonlinear dynamical systems,” Automatica, vol. 129, p. 109 597, 2021.

[4] S. Muntwiler, K.P. Wabersich, A. Carron and M.N. Zeilinger, “Distributed Model Predictive Safety Certification for Learning-based Control”, IFAC World Congress 2020.

[5] A. Didier, K.P. Wabersich and M.N. Zeilinger,“Adaptive Model Predictive Safety Certification for Learning-based Control” , IEEE Conf. on Decision and Control, 2021.

[6] Brunner et al., “Stabilizing linear model predictive control: On the enlargement of the terminal set,” in European Control Conference (ECC), 2013.

[7] U. Rosolia and F. Borrelli, “Learning model predictive control for iterative tasks: a computationally efficient approach for linear system,” IFACPapersOnLine, vol. 50, no. 1, 2017.

[8] B. Tearle, K.P. Wabersich, A. Carron and M.N. Zeilinger, “A predictive safety filter for learning-based racing control”, IEEE Robotics and Automation Letters 2021.

[9] A, Ayush, and K. Sreenath. “Discrete control barrier functions for safety-critical control of discrete systems with application to bipedal robot navigation.” Robotics: Science and Systems. Vol. 13. 2017.