Safety Filters and Model Predictive Control

Reading Group | Controls

Allen Emmanuel Binny

April 3, 2025

Autonomous Ground Vehicles, IIT Kharagpur

Outline

Stability

Equilibrium Point

We consider nonlinear time-invariant system $\dot{x} = f(x)$, where $f : \mathbb{R}^n \to \mathbb{R}^n$. A point $x_e \in \mathbb{R}^n$ is an *equilibrium point* of the system if $f(x_e) = 0$.

$$x_e \text{ is an equilibrium point} \iff x(t) = x_e \text{ is a trajectory.}$$

Suppose x_e is an equilibrium point:

- The system is **globally asymptotically stable (G.A.S.)** if for every trajectory $x(t)$, we have $x(t) \to x_e$ as $t \to \infty$. (This implies x_e is the unique equilibrium point.)

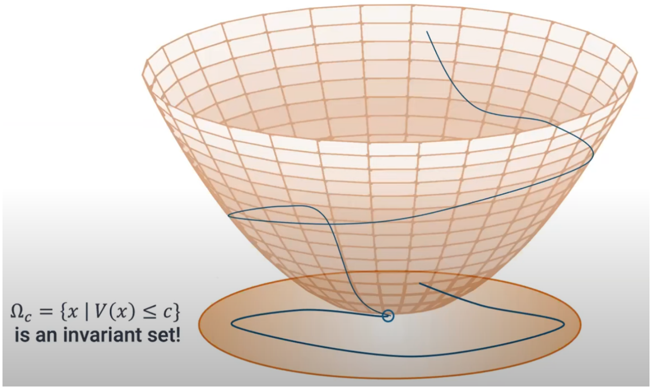
Lyapunov Function

If we can construct a function $V : \mathbb{R}^n \to \mathbb{R}$ for an invariant set $\dot{x} = f(x)$ and x_e as the *equilibrium point*, with the following conditions:

$$V(x_e) = 0, \tag{1a}$$

$$V(x) > 0 \quad \text{for } x \neq x_e, \tag{1b}$$

$$\dot{V}(x) = \frac{\partial V}{\partial x} f(x) < 0 \quad \text{for } x \neq x_e. \tag{1c}$$



Credit: *Lecture by Jason Choi [4]*

Control Lyapunov Function

Let $V(x) : \mathbb{R}^n \to \mathbb{R}$ be a continuously differentiable function. If there exists a constant $c > 0$ such that:

1. $\Omega_c := \{x \in \mathbb{R}^n : V(x) \leq c\}$, a sublevel set of $V(x)$ is bounded,
2. $V(x) > 0$ for all $x \in \mathbb{R}^n \setminus \{x_e\}$, $\quad V(x_e) = 0$,
3. $\min_{u \in U} \dot{V}(x, u) < 0$ for all $x \in \Omega_c \setminus \{x_e\}$,

Then $V(x)$ is a (local) Control Lyapunov Function (CLF), and Ω_c is a region of attraction (ROA), i.e., every state in Ω_c is asymptotically stabilizable to x_e.

$$(\forall x_0 \in \Omega_c, \text{ for } x(t) \text{ with } x(0) = x_0, \exists u : [0, t] \to U \text{ such that } \lim_{t \to \infty} x(t) = x_e).$$

CLF-QP

- Obtain a more stabilized control action based on some weights decided by matrix H.

$$\text{argmin}_{u,\delta} \quad (u - u_{\text{ref}})^T H(u - u_{\text{ref}}) + p\delta^2$$

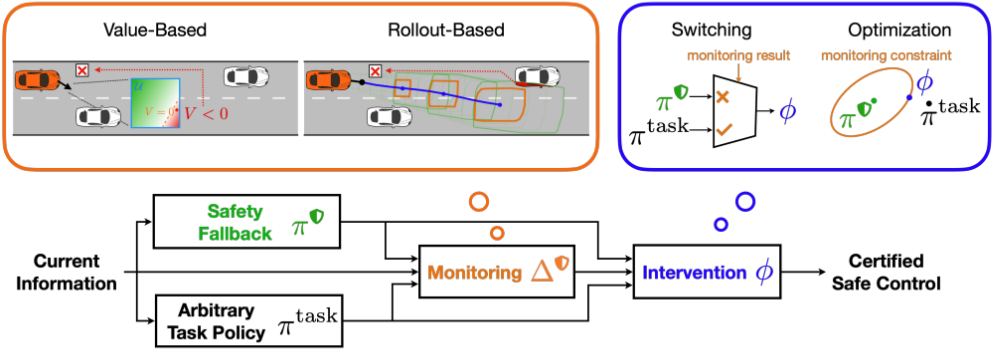
subject to:

$$L_f V(x) + L_g V(x)u + \gamma V(x) \leq \delta \qquad \text{CLF Constraint}$$

$$u \in U \qquad \text{Input Constraints}$$

Safety Filters

Introduction to Safety Filters



Credit: *Review Paper: Hsu et al. [1]*

Safety Filters

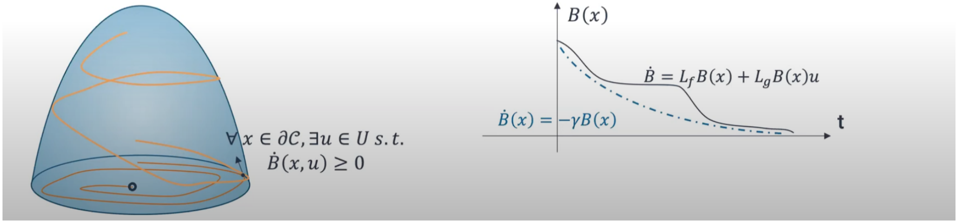
Safety Filters

Control Barrier Functions

Let $B(x)\colon \mathbb{R}^n \to \mathbb{R}$ a continuously differentiable function whose zero-superlevel set is $\mathcal{C}$ ($\mathcal{C} = \{x \mid B(x) \geq 0\}$) and $\nabla B(x) \neq 0$ for all $x \in \partial\mathcal{C}$.

If there exists a positive coefficient γ such that for all $x \in \mathcal{C}$

$$\sup_{u \in U} [L_f B(x) + L_g B(x) u] + \gamma B(x) \geq 0,$$



Credit: *Lecture by Jason Choi[4]*

Introduction

$$\mathcal{C} = \{x \in D \subset \mathbb{R}^n : h(x) \geq 0\},$$

$$\partial\mathcal{C} = \{x \in D \subset \mathbb{R}^n : h(x) = 0\},$$

$$\text{Int}(\mathcal{C}) = \{x \in D \subset \mathbb{R}^n : h(x) > 0\}.$$

We want to ensure the invariance of the set $\mathcal{C}$. We want to make sure we never leave the set $\mathcal{C}$.

CLF-CBF-QP

$$\text{argmin}_{u,\delta} \quad (u - u_{\text{ref}})^T H(u - u_{\text{ref}}) + p\delta^2$$

subject to:

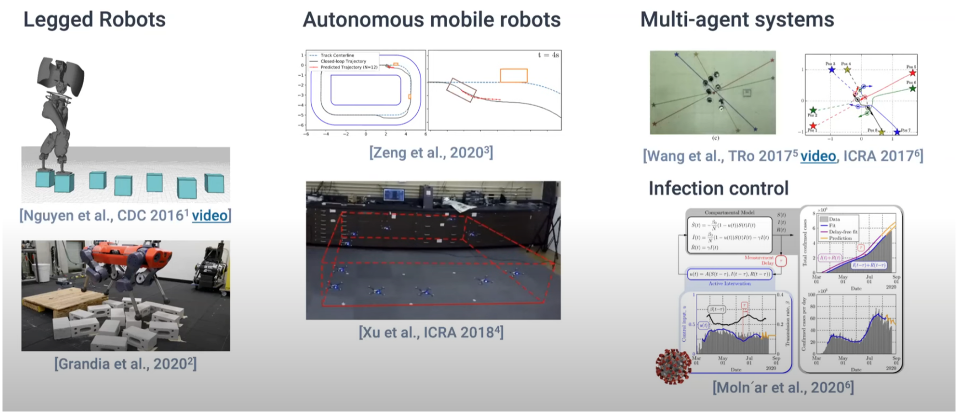
$L_f V(x) + L_g V(x)u + \gamma V(x) \leq \delta$	CLF Constraint
$L_f h(x) + L_g h(x)u + \alpha h(x) \geq 0$	CLF Constraint
$u \in U$	Input Constraints

Application to Ground Vehicles

Let's work out an easy example on how CBFs can be applied:

Distributed Safe Navigation of Multi-Agent Systems using Control Barrier Function-Based Optimal Controllers

Applications of CBFs

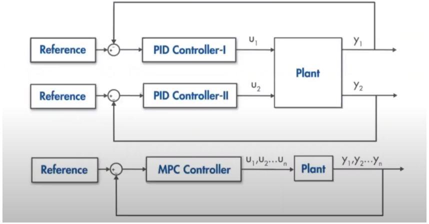


Credit: *Lecture by Jason Choi[4]*

Model Predictive Control

Benefits of using MPC

- Controllers such as PID are SISO, while MPC can handle MIMO
- Provide constraints on states and control inputs. It is essential for robotics.
- Optimise with respect to any objective function. This can be used for reference tracking, energy saving, etc.
- MPC is an optimization problem



Credit: *Lecture by Professor Yash Pant[5]*

MPC and PID

Applying MPC in closed loop format

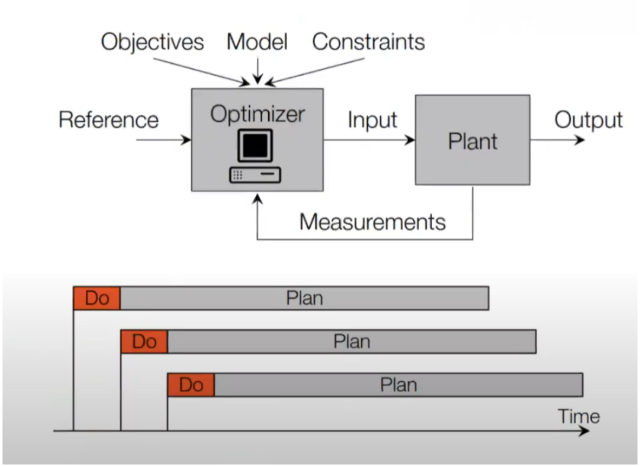
We have the output for the next N steps; what do we do afterwards?

Applying MPC in closed loop format

We have the output for the next N steps; what do we do afterwards?

Receding Horizon Control

Applying MPC in closed loop format



Credit: *Lecture by Professor Yash Pant[5]*

MPC as Optimization Problem

Generalised MPC Optimisation Problem

$$J^*_{0\to N}(x_0) = \min_{U_{0\to N-1}} \left(\sum_{k=0}^{N-1} q(x_k, u_k) + p(x_N) \right)$$

$$\begin{aligned}
\text{s.t.} \quad & x_{k+1} = g(x_k, u_k), && k = 0, 1, \ldots, N-1 && \text{[System Dynamics]} \\
& h(x_k, u_k) \leq 0, && k = 0, 1, \ldots, N-1 && \text{[System Constraints]} \\
& x_N \in X_f, && && \text{[Terminal State]} \\
& x_0 \text{ given} && && \text{[Initial State]}
\end{aligned}$$

Optimal solution: $U^*_{0\to N-1} = \{u^*_0, u^*_1, \ldots, u^*_{N-1}\}$

Solving an MPC Problem

How can we solve the above optimization problem?

Solving an MPC Problem

How can we solve the above optimization problem?

1. Dynamic Programming Approach
2. Solving a QP Problem

MPC as Optimization Problem

Linearized MPC Optimization Problem

$$J^*_{0\to N}(x_0) = \min_{U_{0\to N-1}} \left(\sum_{k=0}^{N-1} (x_k^T Q x_k + u_k^T R u_k) + x_N^T Q_F x \right)$$

$$\begin{aligned}
\text{s.t.} \quad & x_{k+1} = Ax_k + Bu_k, \quad k = 0, 1, \ldots, N-1 && \text{[System Dynamics]} \\
& Ax \leq b, && \text{[State Constraints]} \\
& u_{min} \leq u \leq u_{max}, && \text{[Input Constraints]} \\
& x_N \in X_f, && \text{[Terminal State]} \\
& x_0 \text{ given} && \text{[Initial State]}
\end{aligned}$$

Solving a QP Problem

$$\min_{z} \quad \frac{1}{2}z^T Hz + g^T z$$

$$\text{s.t.} \quad lb \leq A_c z \leq ub$$

where:

$z \in \mathbb{R}^{n \times 1}$ (optimization variable)

$H \in \mathbb{R}^{n \times n}$ (Hessian matrix)

$g \in \mathbb{R}^{n \times 1}$ (linear term)

$A_c \in \mathbb{R}^{m \times n}$ (constraint matrix, m constraints)

- This is a convex problem. Easier to solve. It becomes a problem when we have to solve a non-linear problem.
- Use solvers such as Python - (cvxopt, osqp, cvxpy), MATLAB - Casadi, C++ - CVXGen.

Other MPC Frameworks

- **Non-Linear MPC**: For nonlinear dynamics. We can apply Taylor expansion and linearize around the equilibrium point. Use solvers such as CASADI which can solve NMPC problems.
- **Learning MPC**: When the dynamics is not known. Dynamics can be learnt using ML and DL techniques.
- **Robust MPC**: When there is uncertainty in the model. In the form of the imperfect model, imperfect knowledge of the state, and disturbance in the environment.
- **Stochastic MPC**: Robust MPC can lead to conservative solutions. Incase, we know about the distribution of the noise or data. Can we use this information?

Applications

Applications

1. Learning-based MPC
   - Learning Model of Dynamic System
   - Learning Parameters for MPC
2. Control Barrier Functions
   - For changing environments
   - Learn CBFs from Demonstrations
   - For Multi-Agent Applications

MPCC for F1Tenth

An example for a video lecture on how to perform MPC:
MPC for F1Tenth

References

References

1. [Review Paper] A Unified View of Safety-Critical Control in Autonomous Systems
2. [Review Paper]Data-Driven Safety Filters
3. [Seminar]Planning with Confidence: Uncertainty Quantification for Safety-Critical Tasks by Professor Anushri Dixit
4. [Seminar] CLF and CBF by Jason Choi
5. [Seminar] MPC for Autonomous Racing by Professor Yash V. Pant
6. MATLAB Course on MPC for intuitions

Thank You