ECE 586 BH: Interplay between Control and Machine Learning **Fall 2023**

Lecture 10

Control Barrier Functions for Safety of Perception-Based Control, Part II

Lecturer: Bin Hu, Date:09/21/2023

In the last lecture, we briefly discuss the idea of the measurement-robust CBF approach. Let's first review the basic setup. Suppose a perception module p has been designed to produce the state estimation $\hat{x}(t) := p(y(t))$, which is then used for generating control actions. To guarantee safety, we need to design control actions u based on $\hat{x}$ to ensure that $\mathcal{C}$ becomes a forward invariant set, i.e. $x(t) \in \mathcal{C} = \{x : h(x) \geq 0\}\ \forall t$ if $x(0) \in \mathcal{C}$. For simplicity, we assume that some error bound $\|x - \hat{x}\| \leq r$ is available[1]. Then h is a measurement-robust CBF if it satisfies

$$\sup_{u\in\mathcal{U}} \inf_{x\in\mathcal{X}(\hat{x})} \left\{ \frac{\partial h}{\partial x}(x) f(x) + \frac{\partial h}{\partial x}(x) g(x) u + \alpha(h(x)) \right\} \geq 0,$$

where $\mathcal{X}(\hat{x}) = \{x : \|x - \hat{x}\| \leq r\}$ is a set capturing the error in the state estimation. If h is a measurement-robust CBF, then given the current state estimation $\hat{x}$, one can choose u satisfying

$$\inf_{x\in\mathcal{X}(\hat{x})} \left\{ \frac{\partial h}{\partial x}(x) f(x) + \frac{\partial h}{\partial x}(x) g(x) u + \alpha(h(x)) \right\} \geq 0 \tag{10.1}$$

such that the resultant controller ensures the forward invariance of $\mathcal{C}$ under the estimation error r. However, it is quite difficult to pose the above constraint in a computationally efficient manner. In this lecture, we discuss how to relax (10.1) for computational tractability. We will present some simplified version of the results in [1].

10.1 SOCP Relaxation for Measurement-Robust CBF

We needs to develop a lower bound $M(\hat{x}, u)$ for the left side of (10.1). If we can ensure

$$M(\hat{x}, u) \leq \inf_{x\in\mathcal{X}(\hat{x})} \left\{ \frac{\partial h}{\partial x}(x) f(x) + \frac{\partial h}{\partial x}(x) g(x) u + \alpha(h(x)) \right\},$$

then any u satisfying $M(\hat{x}, u) \geq 0$ will guarantee (10.1) to hold as desired. In addition, we want the dependence of M on u to be not relatively simple so that the following optimization problem can be efficiently solved in real time.

$$\begin{aligned} k_{\text{safe}}(\hat{x}) = &\arg\min_{u\in\mathcal{U}} \frac{1}{2}\|u - k_d(\hat{x})\|^2 \\ &\text{s.t.} M(\hat{x}, u) \geq 0 \end{aligned} \tag{10.2}$$

[1] We will elaborate on this bound later.

In [1], the lower bound $M(\hat{x}, u)$ is constructed using Lipschitz continuity. Let's explain the main idea. Specifically, suppose $\frac{\partial h}{\partial x}(\cdot)f(\cdot)$ is L_1-Lipschitz. Then we have

$$\frac{\partial h}{\partial x}(\hat{x})f(\hat{x}) - \frac{\partial h}{\partial x}(x)f(x) \leq \left|\frac{\partial h}{\partial x}(\hat{x})f(\hat{x}) - \frac{\partial h}{\partial x}(x)f(x)\right| \leq L_1\|\hat{x} - x\|, \tag{10.3}$$

which leads to

$$\frac{\partial h}{\partial x}(x)f(x) \geq \frac{\partial h}{\partial x}(\hat{x})f(\hat{x}) - L_1\|\hat{x} - x\|$$

For any $x \in \mathcal{X}(\hat{x})$, we have $\|\hat{x} - x\| \leq r$. Therefore, for any $x \in \mathcal{X}(\hat{x})$, we must have

$$\frac{\partial h}{\partial x}(x)f(x) \geq \frac{\partial h}{\partial x}(\hat{x})f(\hat{x}) - L_1 r. \tag{10.4}$$

Similarly, suppose $\frac{\partial h}{\partial x}(\cdot)g(\cdot)$ is L_2-Lipschitz. Then the following inequality can be proved using a combination of previous arguments and the Cauchy-Schwartz inequality:

$$\frac{\partial h}{\partial x}(x)g(x)u \geq \frac{\partial h}{\partial x}(\hat{x})g(\hat{x})u - L_2 r\|u\|, \quad \text{for any} \quad x \in \mathcal{X}(\hat{x}) \tag{10.5}$$

Finally, denote $\tilde{\alpha}(\hat{x}) = \inf_{x\in\mathcal{X}(\hat{x})} \alpha(h(x))$, which can be quickly searched since $\alpha(h(\cdot))$ is a 1-D function. Now we can combine (10.4) and (10.5) to obtain the following lower bound:

$$\begin{aligned} M(\hat{x}, u) &= \frac{\partial h}{\partial x}(\hat{x})f(\hat{x}) + \frac{\partial h}{\partial x}(\hat{x})g(\hat{x})u - L_2 r\|u\| - L_1 r + \tilde{\alpha}(\hat{x}) & (10.6)\\ &\leq \inf_{x\in\mathcal{X}(\hat{x})} \left\{\frac{\partial h}{\partial x}(x)f(x) + \frac{\partial h}{\partial x}(x)g(x)u + \alpha(h(x))\right\}. & (10.7) \end{aligned}$$

Therefore, if we can find u satisfying $\frac{\partial h}{\partial x}(\hat{x})f(\hat{x}) + \frac{\partial h}{\partial x}(\hat{x})g(\hat{x})u - L_2 r\|u\| - L_1 r + \tilde{\alpha}(\hat{x}) \geq 0$, we can ensure $\dot{h} \leq -\alpha(h)$ over the state trajectory such that $h(x(t)) \geq 0$ for all t. Eventually we can use the following safety filter to ensure safety (i.e. $x(t) \in \mathcal{C}$ for all t):

$$\begin{aligned} k_{\text{safe}}(\hat{x}) = &\arg\min_{u\in\mathcal{U}} \frac{1}{2}\|u - k_d(\hat{x})\|^2 \\ &\text{s.t.} \quad \frac{\partial h}{\partial x}(\hat{x})f(\hat{x}) + \frac{\partial h}{\partial x}(\hat{x})g(\hat{x})u - L_2 r\|u\| - L_1 r + \tilde{\alpha}(\hat{x}) \geq 0 \end{aligned} \tag{10.8}$$

Notice that there is a term on $\|u\|$ in the above constraint. Consequently, the safety filter k_{safe} needs to solve a second-order cone programming (SOCP) subproblem at every t to find the provably safe control actions. With the controller $k_{\text{safe}} \circ k_d$ embedded into the system, the set $\mathcal{C}$ becomes a forward invariant set for the closed-loop system. Based on the idea in [1], we can say h is a measurement-robust CBF if the following condition holds

$$\sup_{u\in\mathcal{U}} \left\{\frac{\partial h}{\partial x}(\hat{x})f(\hat{x}) + \frac{\partial h}{\partial x}(\hat{x})g(\hat{x})u - L_2 r\|u\| - L_1 r + \tilde{\alpha}(\hat{x})\right\} \geq 0. \tag{10.9}$$

Under the above condition, the SOCP problem in (10.8) is feasible. However, it is typically quite difficult to check the above condition. To address this issue, [1] has developed more tractable conditions which directly pose bounds on r to ensure the feasibility of (10.8). See [1, Theorem 3] for more details.

It is also pointed out in [1] that the practical implementation of (10.8) relies on introducing an extra slack variable for improving feasibility. The following optimization problem is actually implemented:

$$\begin{aligned} k_{\text{safe}}(\hat{x}) = &\arg\min_{u\in\mathcal{U}} \frac{1}{2}\|u - k_d(\hat{x})\|^2 + p\delta^2 \\ &\text{s.t.} \quad \frac{\partial h}{\partial x}(\hat{x})f(\hat{x}) + \frac{\partial h}{\partial x}(\hat{x})g(\hat{x})u - L_2 r\|u\| - L_1 r + \tilde{\alpha}(\hat{x}) + \delta \geq 0 \end{aligned} \tag{10.10}$$

where p is a large positive constant. The above formulation can also guarantee that the resultant controller is locally Lipschitz continuous.

10.2 More Discussions on the State Estimation Errors

Is it reasonable to pose a uniform state estimation error bound r? Based on some extra assumptions, some justifications can be provided [1]. Here we try to provide some simple intuitive explanations.

Assumptions. The assumptions in [1] are listed as follows.

- The environment is assumed to be static, and hence we have $y(t) = s(x(t))$.
- The image generation process s is assumed to be deterministic and locally Lipschitz.
- It is assumed that s can be inverted, i.e. there exists a locally Lipschitz function s^{-1} such that $s^{-1}(s(x(t))) = x(t)$. This is a very strong assumption. Notice that the perception module p can be viewed as an approximated version of s^{-1}.

Explanations. Let's say p is learned from data, and hence there is some difference between p and s^{-1}. We define the state estimation error as

$$e(x(t)) = \hat{x}(t) - x(t) = p(y(t)) - x(t) = p(s(x(t))) - x(t) \tag{10.11}$$

If $p = s^{-1}$, then we have $e(x(t)) = 0$. Since p is only an approximation of s^{-1}, we can think $e(\cdot)$ is a function mapping $x(t)$ to a state-dependent estimation error. Suppose p and s are continuous. If $\mathcal{C}$ is compact, then one can apply Weierstrass theorem to argue that a uniform bound on $\|e(x(t))\|$ exists over the set $\mathcal{C}$. This provides an intuitive explanation for the use of the uniform error bound. We want to emphasize that the above assumptions are very restrictive. In general, the environment can be uncertain, and $y(t)$ depends on external environmental factors. And the image generation process is typically not invertible. How to adapt CBFs for those scenarios is a good research question that requires further investigation.

Bibliography

[1] S. Dean, A. Taylor, R. Cosner, B. Recht, and A. Ames. Guaranteeing safety of learned perception modules via measurement-robust control barrier functions. In *Conference on Robot Learning*, pages 654–670. PMLR, 2021.