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

Lecture 9

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

Lecturer: Bin Hu, Date:09/19/2023

In this lecture, we will discuss the safety of learning-enabled perception-based control. For control engineering systems, safety is typically defined in terms of a safety set, within which the system state must remain at all times. The specifications of safety sets are task-dependent. For example, lane-keeping software is required to keep the car within designated lane boundaries. Similarly, an auto-landing system is designed to ensure that the aircraft touches down at a specific region of the landing strip with prescribed velocities. These safety requirements can be formulated rigorously via posing constraints on the system state variables such as position and velocity. The control barrier function (CBF) method is a popular technique that can be used to design controllers with safety guarantees. This note will cover the following items:

- A brief review of the CBF approach
- Setup of perception-based control
- Some high-level ideas for extending CBF to the perception-based control setting

9.1 A Brief Review of CBF

Consider a nonlinear control affine system:

$$\dot{x}(t) = f(x(t)) + g(x(t))u(t) \tag{9.1}$$

where $x \in \mathbb{R}^{n_x}$ is the state and $u \in \mathcal{U} \subset \mathbb{R}^{n_u}$ is the control action. The set $\mathcal{U}$ consists of all the feasible control inputs. We assume f and g to be locally Lipschitz. For simplicity, we first consider the state-feedback case where $u(t)$ is determined by $x(t)$. Safety is encoded by a safety set $\mathcal{C} \subset \mathbb{R}^{n_x}$. The condition $x(t) \in \mathcal{C}$ means that the system (9.1) is in a safe state at time t. Usually $\mathcal{C}$ is defined as the 0-superlevel set of some prescribed continuously differentiable function $h : \mathbb{R}^{n_x} \to \mathbb{R}$:

$$\mathcal{C} = \{x \in \mathbb{R}^{n_x} : h(x) \geq 0\} \tag{9.2}$$

We assume $\mathcal{C}$ is non-empty with no isolated points. Suppose $x(0) \in \mathcal{C}$, i.e. $h(x(0)) \geq 0$. Now the question becomes how to choose control actions such that $x(t) \in \mathcal{C}$ (or equivalently $h(x(t)) \geq 0$) for all t. To illustrate the idea of CBF, we first consider the following result.

Result 1 (The simplest version of CBF). Suppose $h(x(0)) \geq 0$. For some positive α, if the control action satisfies

$$\dot{h}(x) = \frac{\partial h}{\partial x}(x)f(x) + \frac{\partial h}{\partial x}(x)g(x)u \geq -\alpha h(x), \tag{9.3}$$

then $h(x(t)) \geq h(x(0))e^{-\alpha t} \geq 0$ (this step is based on the Gronwall-Bellman lemma) and hence $x(t) \in \mathcal{C}$ for all t.

Notice (9.3) is affine in u. Given f, g, h, α and $x(t)$, finding $u(t)$ to satisfy (9.3) reduces to a linear programming problem which can be easily solved in real time. This is exactly the basic idea of the CBF approach. Clearly we don't need $\dot{h}(x(t)) \geq 0$ for all t. It is OK to let the value of h decrease along the state trajectory. However, as long as $h(x(t))$ is lower bounded by the product of $h(x(0))$ and something positive, the system (9.1) is guaranteed to be safe. The right side of (9.3) can be replaced with other functions, and this leads to the following standard version of the CBF result.

Result 2 (The standard version of CBF). Suppose $h(x(0)) \geq 0$. If the control action satisfies

$$\dot{h}(x) = \frac{\partial h}{\partial x}(x)f(x) + \frac{\partial h}{\partial x}(x)g(x)u \geq -\alpha(h(x)), \tag{9.4}$$

where $\alpha(\cdot)$ is an extended class $\mathcal{K}$ function[1], then $h(x(t)) \geq 0$ and hence $x(t) \in \mathcal{C}$ for all t. The function h becomes a CBF.

The above result generalizes the previous result by allowing $\alpha(\cdot)$ to be ch^n, ce^{kh}, and $c\log(h+1)$. The above result can be proved using Nagumo's theorem and the comparison lemma.

When does there exist u satisfying (9.4)? If for any $x \in \mathcal{C}$, the following inequality holds

$$\sup_{u \in \mathcal{U}} \left\{ \frac{\partial h}{\partial x}(x)f(x) + \frac{\partial h}{\partial x}(x)g(x)u \right\} \geq -\alpha(h(x)), \tag{9.5}$$

then there exists a feasible solution u to satisfy (9.4). Whenever (9.5) is true, the function h is said to be a CBF.

How to use (9.4) for control design? The condition (9.4) is mainly used to develop the so-called safety filter that can be used to make any pre-designed controller safe. Suppose we have designed a controller k_d using some of our favorite design methods. At every time t, the controller generates a control action $u(t) = k_d(x(t))$. However, there is some possibility that the controller k_d can drive the system state to move outside the safety set $\mathcal{C}$. Based on

[1]The function α is strictly increasing and $\alpha(0) = 0$.

the above result, one can apply a controller k_{safe} which actually generates the control action $u(t)$ as

$$\begin{aligned} k_{\text{safe}}(x) = &\arg\min_{u\in\mathcal{U}} \frac{1}{2}\|u - k_d(x)\|^2 \\ &\text{s.t.} \frac{\partial h}{\partial x}(x)f(x) + \frac{\partial h}{\partial x}(x)g(x)u \geq -\alpha(h(x)) \end{aligned} \tag{9.6}$$

At every t, the controller k_{safe} solves a quadratic programming subproblem to find the control action that is closest to the signals generated by the original pre-designed controller k_d and satisfies the CBF condition (9.4) at the same time. Then the system is guaranteed to be safe. With the controller k_{safe} embedded into the system, the set $\mathcal{C}$ becomes a forward invariant set for the closed-loop system. In other words, the closed-loop system will stay in $\mathcal{C}$ if it is initialized from $\mathcal{C}$. In general, there is no guarantee that the controller k_{safe} will achieve the same performance as k_d. There is some general trade-off between performance and safety.

The above approach requires knowing f and g. Next, we will look at the perception-based control problem and see how safety becomes a more complicated issue.

9.2 Setup of Perception-Based Control

In this section, we briefly discuss the setup of a perception-based control problem. We consider the perception-based feedback loop as shown in Figure 9.1.

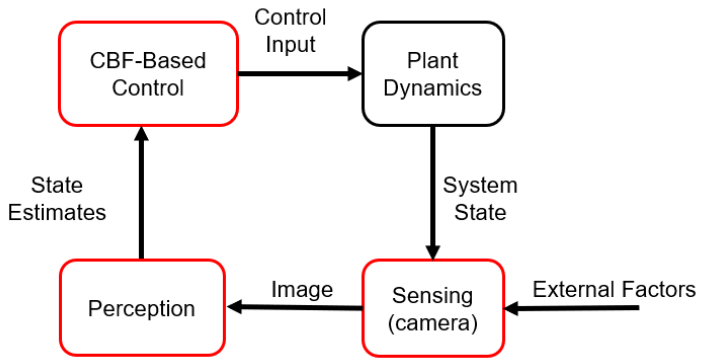

Figure 9.1. The feedback loop of a perception-based control system.

Suppose the plant is still governed by the dynamical system model (9.1). Now we assume that the system state measurement is not available, and a camera is used to generate image measurement $y(t) = s(x(t), e(t))$, where $s(\cdot)$ captures the image generation process, and $e(t)$ denotes the external environment factors (e.g. lighting). Now we ask the question: how can we generate control actions $u(t)$ based on the image measurement $y(t)$ to ensure $x(t) \in \mathcal{C}$?

End-to-end policy learning provides a general framework to achieve good performance. However, it is unclear how to enforce safety constraints for direct policy search. One possibility is to adapt a model-based approach. Suppose we know the model (9.1) (i.e. we know f and g), and the main difficulty is that we do not know $s(\cdot)$ and $e(t)$. Now it seems reasonable to apply standard computer vision techniques to estimate $x(t)$ from $y(t)$, and then use the CBF approach as if the state estimates are close to the true states. Due to the state estimation errors, such an approach may not be able to ensure safety. Is there a way to fix this?

9.3 Measurement-Robust CBF: Some High-Level Ideas

It is possible to "robustify" the CBF approach against the errors introduced by the perception module. Suppose that we have designed a perception module p to generate a state estimation $\hat{x}(t) := p(y(t))$ and then calculates $u(t)$ based on $\hat{x}(t)$. 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. $x(t) \in \mathcal{C}$ $\forall t$ if $x(0) \in \mathcal{C}$. Now we briefly explain the idea of measurement-robust CBF. Recall that we have $\mathcal{C} = \{x : h(x) \geq 0\}$. For simplicity, we assume that standard learning-theoretic tools have been used to construct some error bound $\|x - \hat{x}\| \leq r$ (one subtlety is that r is state-dependent). Then h is a measurement-robust CBF that can be used to extract a controller ensuring the forward invariance of $\mathcal{C}$ 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

$$\inf_{x \in \mathcal{X}(\hat{x})} \left\{ \frac{\partial h}{\partial x}(x) f(x) + \frac{\partial h}{\partial x}(x) g(x) u + \alpha(h(x)) \right\} \geq 0 \tag{9.7}$$

such that the resultant controller ensures the forward invariance of $\mathcal{C}$ under the estimation error r. However, it is quite difficult to solve the above problem efficiently. In the next lecture, we will discuss how to relax the above problem into more tractable convex programs.