Control Barrier Functions

Devesh Nath

December 17, 2024

Abstract

A brief overview of Control Barrier Functions.

1 Preliminaries

1.1 Safe Set's Invariance

Consider a dynamical system given by:

$$\dot{x} = f(x)$$

with $x \in \mathbb{R}^n$. Assume that the safe set C is the superlevel set of a smooth function $h : \mathbb{R}^n \to \mathbb{R}$, i.e.,

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

and that $\frac{\partial h}{\partial x}(x) \neq 0$ for all x such that $h(x) = 0$.

Nagumo's Theorem provides necessary and sufficient conditions for set invariance based on the derivative of h on the boundary of C:

$$C \text{ is invariant} \iff \dot{h}(x) \geq 0 \quad \forall x \in \partial C$$

To explain, the theorem states that for the set C to be invariant under the flow of the dynamical system, the derivative of the function h with respect to time, $\dot{h}(x)$, must be non-negative for all points x on the boundary of C (denoted as ∂C). This ensures that the trajectories of the system do not leave the set C, thereby maintaining safety.

1.2 Barrier Certificates

The choice of the term "barrier" was motivated by its use in the optimization literature where barrier functions are added to cost functions to avoid undesirable regions. In the case of barrier certificates, one considers an unsafe set C_u and a set of initial conditions C_0 together with a function $B : \mathbb{R}^n \to \mathbb{R}$ where $B(x) \leq 0$ for all $x \in C_0$ and $B(x) > 0$ for all $x \in C_u$. Then B is a barrier certificate if

$$\dot{B}(x) \leq 0 \implies C \text{ is invariant}$$

To elaborate, the term "barrier" is used because, similar to optimization problems where barrier functions prevent the solution from entering undesirable regions, barrier certificates ensure safety by preventing the system's state from entering unsafe regions. Specifically, for a given unsafe set C_u and initial conditions set C_0, a function B is defined such that $B(x) \leq 0$ for all $x \in C_0$ and $B(x) > 0$ for all $x \in C_u$. The function B acts as a certificate of safety if its time derivative $\dot{B}(x)$ is non-positive, which implies that the set C remains invariant and the system's state does not enter the unsafe set C_u.

If $\dot{B}(x) \leq 0$, then $B(x)$ cannot increase. This means that if the system starts in a state where $B(x) \leq 0$ (i.e., within the safe set C), it will remain in states where $B(x) \leq 0$ as time progresses. Consequently, the system's state will not enter the unsafe set C_u (where $B(x) > 0$) because $B(x)$ cannot increase to become positive.

1.3 Relation between Safe and Unsafe Sets

In the notation for C above, by picking the safe set to be the complement of the unsafe set $C = C_u^c$, with $B(x) = -h(x)$, the barrier certificate conditions become:

$$\dot{h}(x) \geq 0$$

which implies that C is invariant.

1.4 Lyapunov Functions as Barrier Functions

As a means to extend the safety guarantees beyond the boundary of the set, there have been a variety of approaches that can be best described as "Lyapunov-like." That is, Lyapunov functions yield invariant level sets so, if these level sets are contained in the safe set one can guarantee safety—importantly, these conditions can be applied over the entire set and not just on the boundary.

Lyapunov functions are used to prove the stability of equilibrium points in dynamical systems. A Lyapunov function $V : \mathbb{R}^n \to \mathbb{R}$ is a scalar function that is positive definite and decreases along the trajectories of the system. If $V(x)$ is positive definite and $\dot{V}(x) \leq 0$, then the equilibrium point is stable. The level sets of V (i.e., the sets where $V(x)$ is constant) are invariant under the flow of the system.

In this case, one constructs a "barrier Lyapunov function" B much as above but with the additional requirement that it is, for all intents and purposes, positive definite. Then, by enforcing the condition that $\dot{B} \leq 0$ over the set C, it ensures invariance of this set and thus safety.

A barrier Lyapunov function B is similar to a Lyapunov function but with an additional requirement that it is positive definite. This means that $B(x) > 0$ for all $x \neq 0$ and $B(0) = 0$. By ensuring that $\dot{B}(x) \leq 0$ for all $x \in C$, we can guarantee that the set C is invariant. This means that if the system starts in the set C, it will remain in C for all future times, thereby ensuring safety.

The major limitation is that, while these conditions ensure safety they also enforce invariance of every level set. Thus, they are overly strong and conservative.

The limitation of using barrier Lyapunov functions is that they enforce invariance of every level set of B. This means that not only is the set C invariant, but every sublevel set of B is also invariant. This can be overly conservative because it restricts the system's trajectories more than necessary to ensure safety.

1.5 Building up to Control Barrier Functions

The notion of a barrier certificate was extended to a "control" version to yield the first definition of a "control barrier function". In particular, given a control system and a safe set C as defined above by a function h, the conditions are effectively:

$$\exists u \text{ s.t. } \dot{h}(x,u) \geq 0 \implies C \text{ is invariant}$$

This means that for the set C to be invariant, there must exist a control input u such that the time derivative of h with respect to the system's dynamics is non-negative. This ensures that the system's state remains within the safe set C.

These ideas led to the development of control Lyapunov barrier functions, which unify safety and stability guarantees. However, the condition $\dot{h}(x,u) \geq 0$ is often too restrictive, motivating the modern version of control barrier functions.

Control Lyapunov barrier functions combine the properties of Lyapunov functions (ensuring stability) and barrier functions (ensuring safety). However, the condition $\dot{h}(x,u) \geq 0$ can be overly restrictive, leading to the development of more relaxed conditions.

The modern formulation of control barrier functions extends traditional barrier conditions to the entire safe set. For a control system and a safe set C defined by h, a control barrier function satisfies:

$$\exists u \text{ s.t. } \dot{h}(x,u) \geq -\alpha(h(x)) \implies C \text{ is invariant}$$

where α is a class $\mathcal{K}$ function. This condition is necessary and sufficient for compact sets, allowing for the synthesis of safe controllers through optimization-based methods that minimally adjust the desired control input.

A function $\alpha : [0,\infty) \to [0,\infty)$ is called a class $\mathcal{K}$ function if it satisfies the following properties:

- α is continuous,
- α is strictly increasing,
- $\alpha(0) = 0$.

The modern definition of control barrier functions requires that there exists a control input u such that the time derivative of h is greater than or equal to $-\alpha(h(x))$, where α is a class $\mathcal{K}$ function. This condition is both necessary and sufficient for the invariance of the set C, making it minimally restrictive. This allows for the synthesis of safe controllers using optimization-based methods that minimally alter the desired control input to ensure safety.

2 Control Barrier Functions

Consider a nonlinear affine control system given by:

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

where f and g are locally Lipschitz continuous functions, $x \in D \subset \mathbb{R}^n$, and $u \in U \subset \mathbb{R}^m$ is the set of admissible inputs.

2.1 Motivation: Control Lyapunov Functions

Suppose we have the control objective of asymptotically stabilizing the nonlinear control system to a point $x^* = 0$, i.e., driving $x(t) \to 0$. In a nonlinear context, this can be achieved by finding a feedback control law that drives a positive definite function $V : D \subset \mathbb{R}^n \to \mathbb{R}_{\geq 0}$ to zero. That is, if

$$\exists u = k(x) \text{ s.t. } \dot{V}(x, k(x)) \leq -\gamma(V(x)),$$

where

$$\dot{V}(x, k(x)) = L_f V(x) + L_g V(x) k(x),$$

Here, $L_f V(x)$ and $L_g V(x)$ are the Lie derivatives of the function V along the vector fields f and g, respectively. Specifically, $L_f V(x)$ represents the rate of change of V along the trajectories of the system defined by $\dot{x} = f(x)$, and $L_g V(x)$ represents the rate of change of V along the trajectories defined by $\dot{x} = g(x)$. These Lie derivatives help in understanding how the function V evolves over time under the influence of the system dynamics and the control input.

Geometrically:

- If $\dot{V}(x) < -\gamma(V(x))$, the energy dissipation is faster than required, meaning the trajectory is descending "steeply" across level sets of $V(x)$.
- If $\dot{V}(x) = -\gamma(V(x))$, the trajectory is exactly on the boundary where the dissipation matches the control objective.
- If $\dot{V}(x) > -\gamma(V(x))$ (e.g., approaching zero), the dissipation slows down, and stabilization may not be asymptotic.

Control Lyapunov Functions guarantee that the system's energy decreases at a rate that is at least as fast as required for stabilization.

then the system is stabilizable to $V(x^*) = 0$, i.e., $x^* = 0$. Note that here $\gamma : \mathbb{R}_{\geq 0} \to \mathbb{R}_{\geq 0}$ is a class $\mathcal{K}$ function defined on the entire real line for simplicity, i.e., γ maps zero to zero, $\gamma(0) = 0$, and it is strictly monotonic: for all $r_1, r_2 \in \mathbb{R}_{\geq 0}$, $r_1 < r_2$ implies that $\gamma(r_1) < \gamma(r_2)$. Thus, the process of stabilizing a nonlinear system can be understood as finding an input that creates a one-dimensional stable system given by the Lyapunov function: $\dot{V} \leq -\gamma(V)$, wherein the comparison lemma implies that the full-order nonlinear system is thus stable under the control law $u = k(x)$.

The above observations motivate the notion of a control Lyapunov function wherein a function V is shown to stabilize the system without the need to explicitly construct the feedback controller $u = k(x)$. That is, we only need a controller to exist that results in the desired inequality on $\dot{V}$. Concretely, V is a control Lyapunov function (CLF) if it is positive definite and satisfies:

$$\inf_{u \in U} [L_f V(x) + L_g V(x) u] \leq -\gamma(V(x)),$$

where γ is again a class $\mathcal{K}$ function. The importance of this definition is that it allows us to consider the set of all stabilizing controllers for every point $x \in D$:

$$K_{\text{clf}}(x) := \{u \in U : L_f V(x) + L_g V(x) u \leq -\gamma(V(x))\}.$$

$K_{clf}(x)$ is the set of stabilizing controls.

This is an affine constraint in u and thus will allow for the formulation of optimization-based controllers. It also elucidates conditions on when V is a CLF; for example, if $U = \mathbb{R}^m$, it is easy to verify that

$$L_g V(x) = 0 \implies L_f V(x) \leq -\gamma(V(x)) \implies K_{\text{clf}}(x) \neq \emptyset$$

and thus there are stabilizing controllers. More generally, we have the following central stabilization result for CLFs:

If $L_g V(x) = 0$, it means that the Lie derivative of V along the vector field g is zero. This implies that the control input u does not affect the rate of change of V. In other words, the function V evolves solely according to the dynamics defined by $f(x)$. Therefore, the inequality $L_f V(x) \leq -\gamma(V(x))$ must hold for the system to be stabilizable without any control input. This ensures that the set of stabilizing controllers $K_{\text{clf}}(x)$ is non-empty, as the system can be stabilized by the natural dynamics alone.

Theorem 1. *For the nonlinear control system $\dot{x} = f(x) + g(x)u$, if there exists a control Lyapunov function $V : D \to \mathbb{R}_{\geq 0}$, i.e., a positive definite function satisfying (3), then any Lipschitz continuous feedback controller $u(x) \in K_{clf}(x)$ asymptotically stabilizes the system to $x^* = 0$.*

Safe Set. We consider a set C defined as the superlevel set of a continuously differentiable function $h : D \subset \mathbb{R}^n \to \mathbb{R}$, yielding:

$$\begin{aligned} C &= \{x \in D \subset \mathbb{R}^n : h(x) \geq 0\}, \\ \partial C &= \{x \in D \subset \mathbb{R}^n : h(x) = 0\}, \\ \text{Int}(C) &= \{x \in D \subset \mathbb{R}^n : h(x) > 0\}. \end{aligned}$$

We refer to C as the safe set.

Safety Let $u = k(x)$ be a feedback controller such that the resulting dynamical system

$$\dot{x} = f_{\text{cl}}(x) := f(x) + g(x)k(x)$$

is locally Lipschitz. To formally define safety, due to the locally Lipschitz assumption, for any initial condition $x_0 \in D$ there exists a maximum interval of existence $I(x_0) = [0, \tau_{\max})$ such that $x(t)$ is the unique solution to the above equation on $I(x_0)$; in the case when f_{cl} is forward complete, $\tau_{\max} = \infty$.

This helps us define safety because it ensures that for any initial condition within the domain D, there exists a unique solution to the system's dynamics over a maximum interval of time. If the system is forward complete, the solution exists for all future times. This guarantees that the system's behavior is predictable and well-defined, which is a prerequisite for ensuring that the system remains within a safe set C over time.

Definition 1. *The set C is forward invariant if for every $x_0 \in C$, $x(t) \in C$ for $x(0) = x_0$ and all $t \in I(x_0)$. The system is safe with respect to the set C if the set C is forward invariant.*

2.2 Control Barrier Functions

Using control Lyapunov functions (CLFs) as motivation, we wish to generalize to the concept of safety. However, one must be careful about directly generalizing Lyapunov functions. If there exists a CLF V such that $V(x) = 0 \implies x \in C$ and V has a superlevel set $\Omega_c = \{x \in D : V(x) \leq c\} \subset C$, then the corresponding controllers will render Ω_c invariant, and hence C safe. Nevertheless, this is overly restrictive as it would render every sublevel set invariant, i.e., $\Omega_{c'}$ for all $c' < c$.

Rather, we wish to enforce set invariance without requiring a positive definite function. For h to be a control barrier function (CBF), it should render C invariant but not its sublevel sets. This allows for more flexibility in ensuring safety without the overly conservative constraints imposed by CLFs.

If a sublevel set is invariant, the system is restricted to stay there, that is why it gets restrictive. A CBF only has invariance under C which relaxes the constraints.

Class $\mathcal{K}_\infty$ Functions. An extended class $\mathcal{K}_\infty$ function is a function $\alpha : \mathbb{R} \to \mathbb{R}$ that is strictly increasing and satisfies $\alpha(0) = 0$. Unlike class $\mathcal{K}$ functions, which are defined on $[0, \infty)$, extended class $\mathcal{K}_\infty$ functions are defined on the entire real line $\mathbb{R} = (-\infty, \infty)$.

Definition 2. *Let $C \subset D \subset \mathbb{R}^n$ be the superlevel set of a continuously differentiable function $h : D \to \mathbb{R}$. Then h is a* ***control barrier function (CBF)*** *if there exists an extended class $\mathcal{K}_\infty$ function α such that for the control system:*

$$\sup_{u \in U}[L_f h(x) + L_g h(x)u] \geq -\alpha(h(x))$$

for all $x \in D$.

Guaranteed Safety via Control Barrier Functions: We consider the set of control values that render C safe:

$$K_{\text{cbf}}(x) = \{u \in U : L_f h(x) + L_g h(x)u + \alpha(h(x)) \geq 0\}.$$

As in the case of control Lyapunov functions, this set quantifies all control inputs at a point $x \in D$ that keep the system safe. The main result regarding control barrier functions is that the existence of a control barrier function implies the control system is safe.

2.3 Optimization Based Control

Having established that control barrier functions give necessary and sufficient conditions on safety, the question becomes: how does one synthesize controllers? Importantly, we wish to do so in a minimally invasive fashion, i.e., modify an existing controller in a minimal way so as to guarantee safety. This naturally leads to optimization-based controllers.

Safety-Critical Control: Suppose we are given a feedback controller $u = k(x)$ for the control system and we wish to guarantee safety. Yet it may be the case that $k(x) \notin K_{\text{cbf}}(x)$ for some $x \in D$. To modify this controller in a minimal way so as to guarantee safety, we start by noticing that the conditions on safety given are affine in u. Thus, we can consider the following Quadratic Program (QP) based controller that finds the minimum perturbation on u:

$$u(x) = \arg\min_{u \in \mathbb{R}^m} \frac{1}{2}\|u - k(x)\|^2$$

subject to

$$L_f h(x) + L_g h(x) u \geq -\alpha(h(x))$$

This optimization problem modifies the existing controller $k(x)$ in a minimal way to ensure safety by solving a QP. The objective is to minimize the difference between the new control input u and the original control input $k(x)$, subject to the safety constraint.

Unifying with Lyapunov: The QP-based formulation of safety-critical controllers suggests a means to unify safety and stability. Optimization-based controllers were first utilized in the context of CLFs for multi-objective nonlinear control. Concretely, we consider the following QP-based controller:

$$u(x) = \arg \min_{(u,\delta) \in \mathbb{R}^m \times \mathbb{R}} \frac{1}{2} u^T H(x) u + p\delta^2$$

subject to

$$L_f V(x) + L_g V(x) u \leq -\gamma(V(x)) + \delta$$
$$L_f h(x) + L_g h(x) u \geq -\alpha(h(x))$$

Here, $H(x)$ is any positive definite matrix, and δ is a relaxation variable that ensures the solvability of the QP. The term $p\delta^2$ penalizes the relaxation, ensuring that the stability condition is relaxed only as much as necessary to guarantee safety.

3 CBFs for Systems with Actuation Constraints

3.1 The Allowable Set A

The allowable set A is defined as:

$$A = \{x \in D \mid \rho(x) \geq 0\},$$

where $\rho(x)$ is a performance function that quantifies whether a state x lies in the allowable set. Specifically:

- $\rho(x) > 0$: The state x is strictly within the allowable set.
- $\rho(x) = 0$: The state x is on the boundary of A.
- $\rho(x) < 0$: The state x is outside the allowable set.

The goal is to ensure the system remains within this set. However, direct enforcement of invariance on A may not be possible due to limitations in the system dynamics or actuation constraints.

3.2 Motivation for Constructing a Safe Subset C

If the allowable set A cannot be made invariant, we aim to construct a safe subset C, defined as the superlevel set of a Control Barrier Function $h(x)$:

$$C = \{x \in D \mid h(x) \geq 0\}.$$

This subset C must satisfy $C \subseteq A$, meaning the safe set C is fully contained within the allowable set A. Thus, we need a function $h(x)$ that ensures the system remains within C, even if A cannot be rendered invariant globally.

3.3 The Role of the Nominal Controller β

A nominal control strategy $\beta : D \to U$, called an evading maneuver, is assumed to be known. Intuitively:

- β is a "best-effort" controller that attempts to keep the system in the allowable set A.
- However, there are no guarantees that β can keep the system in A indefinitely under all circumstances.

The system trajectory under β is denoted as $\varphi_\beta(t, x)$, where:

$$\dot{\varphi}_\beta(t, x) = f(\varphi_\beta(t, x)) + g(\varphi_\beta(t, x))\beta(\varphi_\beta(t, x)),$$

and $\varphi_\beta(0, x) = x$. This trajectory represents the evolution of the system starting from x under the nominal controller β.

3.4 Constructing the Barrier Function $h(x)$

To construct a barrier function $h(x)$ that guarantees safety, we evaluate the worst-case performance of ρ along the trajectory $\varphi_\beta(t, x)$ under the nominal controller β. Specifically:

$$h(x) = \inf_{\tau \in [0,\infty)} \rho(\varphi_\beta(\tau, x)).$$

Here's the intuition behind this definition:

- The trajectory $\varphi_\beta(\tau, x)$ evolves over time starting from x under the nominal controller β.
- Along this trajectory, the function $\rho(\varphi_\beta(\tau, x))$ measures how "close" the system is to violating the allowable set A.
- The infimum of ρ over all future times $\tau \in [0, \infty)$ identifies the worst-case value of ρ along the trajectory. This is the smallest value that ρ will attain.
- If $h(x) \geq 0$, the system trajectory under β will remain in the allowable set A or at least not violate it.
- If $h(x) < 0$, the trajectory under β will eventually exit A, indicating that x is not safe under the nominal controller.

Thus, $h(x)$ effectively measures how "safe" the state x is based on the performance of the nominal controller β.

3.5 Polynomial Parameterization and Sums-of-Squares (SOS) Programming

To overcome the computational challenges of direct simulation, an alternative approach is to parameterize $h(x)$ as a polynomial function of a given degree and enforce the required CBF conditions using Sums-of-Squares (SOS) programming.

3.5.1 Polynomial Parameterization

Assume $h(x)$ takes the form of a polynomial of fixed degree, e.g.:

$$h(x) = \sum_{i=1}^{N} c_i \phi_i(x),$$

where $\phi_i(x)$ are polynomial basis functions and c_i are the coefficients to be determined.

3.5.2 Sums-of-Squares (SOS) Polynomials

A polynomial $s(x)$ is said to be a SOS polynomial if it can be written as:

$$s(x) = \sum_{i=1}^{r} (g_i(x))^2,$$

where $g_i(x)$ are polynomials. Intuitively, SOS polynomials are always non-negative since they are sums of squared terms.

3.5.3 SOS Programming for CBFs

Using SOS programming, one can enforce the CBF conditions (e.g., $h(x) \geq 0$ and the derivative condition involving $\alpha(h(x))$) in a computationally tractable way. Specifically:

- Express the conditions for $h(x)$ (e.g., $L_f h(x) + L_g h(x) u \geq -\alpha(h(x))$) in polynomial form.
- Use SOS constraints to ensure these conditions are satisfied globally or locally within a domain D.

3.5.4 SOS Programming Optimization

SOS programming allows you to search for the coefficients c_i of the polynomial $h(x)$ such that it satisfies the required conditions:

$$\begin{aligned}
&\text{Find } h(x) = \text{polynomial such that:}\\
&h(x) \geq 0 \quad \text{(safety condition)}\\
&L_f h(x) + L_g h(x) u \geq -\alpha(h(x)) \quad \text{(CBF condition)}
\end{aligned}$$

This optimization can be solved efficiently using semidefinite programming (SDP) tools, as SOS constraints can be represented as linear matrix inequalities (LMIs).