Enforcing Mobile Robot Safety Under Input Constraints

SAPIENZA
UNIVERSITÀ DI ROMA

Tutor:
Tommaso Belvedere

Francesco D'Orazio	1806836
Lorenzo Govoni	1796934
Riccardo Riglietti	1811527

Outline

The objective of our project is finding suitable controllers for the safety critical problem of **Adaptive Cruise Control (ACC)** under input constraints.

Because of input limitations, methods that build a controller ignoring them are no more able to always guarantee safety, so new approaches taking into account such limitations are needed for safety critical problems such as **ACC**.

Our presentation is organized as follows:

- Theoretical introduction of different methods to build a controller
- Definition of the **Adaptive Cruise Control** problem
- Application of the different methods to the **ACC** problem and comparison
- Conclusions

Control Barrier Function

Safety can be framed in the context of enforcing invariance of a set, i.e. not leaving a safe set, defined as the zero superlevel set of a continuously differentiable function $h(x) : D \subset \mathbb{R}^n \to \mathbb{R}$

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

The function $h(x)$ is a Control Barrier Function (CBF) if there exists an extended class-K_∞ function α such that the following expression holds for the control system:

$$\sup_{u \in U} \left[L_f h(x) + L_g h(x) u\right] \geq -\alpha(h(x)) \;\; \forall x \in D \quad \longrightarrow \quad \begin{aligned} u^*(x) = \underset{u \in \mathbb{R}^m}{\operatorname{argmin}} \;& \frac{1}{2} ||u - k(x)||^2 \\ s.t. \;\; & L_f h(x) + L_g h(x) u \geq -\alpha(h(x)) \end{aligned}$$

Control Lyapunov Function - Control Barrier Function

Here we are going to consider the dual problem of safety that is the one of stabilizing the system.

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

Quadratic Programming

$$u^*(x) = \underset{(u,\delta)\in\mathbb{R}^{m+1}}{\operatorname{argmin}} \ \frac{1}{2} u^T H(x) u + p\delta^2$$

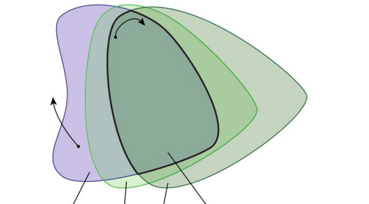
$$\text{s.t.} \quad L_f V(x) + L_g V(x) u \geq -\gamma(V(x)) + \delta$$

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

- *H(x)* is any positive definite matrix (pointwise in *x*)
- δ is a relaxation variable
- p > 0 penalty for the soft constraint

Input Constrained Control Barrier Function

Iterative procedure for computing the ICCBF



$$
\begin{aligned}
b_0(x) &= h(x) \\
b_1(x) &= \inf_{u \in U} [L_f b_0(x) + L_g b_0(x) u + \alpha_0(b_0(x))] \\
b_2(x) &= \inf_{u \in U} [L_f b_1(x) + L_g b_1(x) u + \alpha_1(b_1(x))] \\
&\vdots \\
b_{i+1}(x) &= \inf_{u \in U} [L_f b_i(x) + L_g b_i(x) u + \alpha_i(b_i(x))] \\
&\vdots \\
b_N(x) &= \inf_{u \in U} [L_f b_{N-1}(x) + L_g b_{N-1}(x) + \alpha_{N-1}(b_{N-1}(x))]
\end{aligned}
$$

$$
\longrightarrow
$$

$$
\begin{aligned}
C_0 &= \{x \in X : b_0(x) \geq 0\}, \\
C_1 &= \{x \in X : b_1(x) \geq 0\}, \\
&\vdots \\
C_i &= \{x \in X : b_i(x) \geq 0\}, \\
&\vdots \\
C_N &= \{x \in X : b_N(x) \geq 0\},
\end{aligned}
$$

Input Constrained Control Barrier Function

If there exists a class-K function α_N such that

$$\sup_{u \in U} \left[L_f b_N(x) + L_g b_N(x) u + \alpha_N(b_N(x)) \right] \geq 0 \quad \forall x \in C^*$$

then $b_N(x)$ is an Input Constrained Control Barrier Function (ICCBF).

Quadratic programming

$$\begin{aligned} u^*(x) = \underset{U \in \mathbb{R}^m}{\operatorname{argmin}} \quad & \frac{1}{2} u^T H(x) u + F(x)^T u \\ \text{subject to} \quad & L_f b_N(x) + L_g b_N(x) u \geq \alpha_N(b_N(x)) \\ & u \in U \end{aligned}$$

- $H : X \longrightarrow \mathbb{R}_+^{m \times m}$, where $\mathbb{R}_+^{m \times m}$ is the set of real $m \times m$ positive definite matrices
- $F : X \longrightarrow \mathbb{R}^m$

Safe Control Synthesis for ACC

The **Adaptive Cruise Control** problem considers two vehicles, a leader and a (controlled) follower, where the second one would like to reach its maximum speed while remaining safe. The controller must prevent the follower from colliding with the leading vehicle, but should also allow it to accelerate to the speed limit when this requirement does not conflict with safety.

We simulated this problem under input constraints applying the following methods:

- **Input Constrained Control Barrier Function (ICCBF)**
- **Control Lyapunov Function - Control Barrier Function (CLF-CBF)**
- **Optimal Control Barrier Function (Optimal CBF)**

ACC formalization for ICCBF and CLF-CBF

Dynamical model

$$\dot{x} = \begin{bmatrix} \dot{d} \\ \dot{v} \end{bmatrix} = \begin{bmatrix} v_0 - v \\ -F(v)/m \end{bmatrix} + \begin{bmatrix} 0 \\ g_0 \end{bmatrix} u$$

d represents the distance between the two vehicles.

v is the velocity of the follower.

F(v) $= f_0 + f_1 v + f_2 v^2$ represents the aerodynamic and rolling drag.

Input constraints

$$U = \{u : -u_{max} < u < u_{max}\}$$

The input control **u** is constrained because of limitations in the actuation system.

The controller based on **ICCBF** intrinsically solves the problem with saturation, whereas using **CLF-CBF** may lead the system to exceed the safe set boundary.

ACC formalization for ICCBF and CLF-CBF

Safety function and evaluation of the different controllers

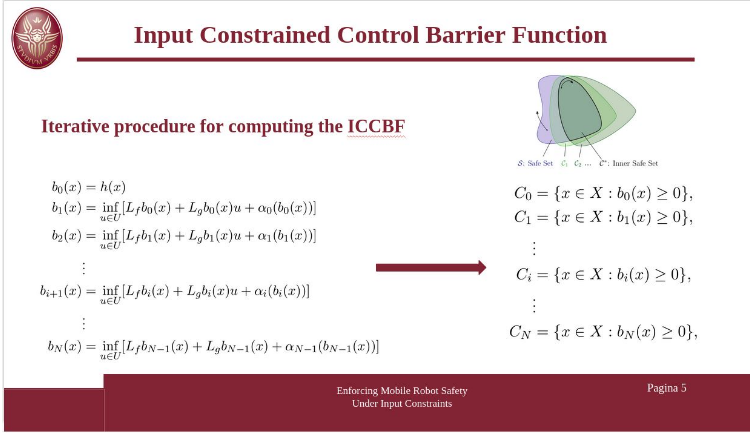
The safety function for the **ACC** system is defined as:

$$b_0(x) = h(x) = x_1 - 1.8x_2$$

For each controller we will graph this function as the system evolves to check if and when a certain procedure lets the system enter in an unsafe state and how conservative each method is (i.e. the higher the safety function, the more conservative the method).

Input Constrained Control Barrier Function

We build the **ICCBF** by following the iterative procedure already introduced in the theory slides



After the derivation of **b_2(x)** we need to check if the function is indeed an **ICCBF**, by solving the optimization problem below computing γ and checking if $\gamma \geq 0$

$$\begin{aligned}\gamma = \text{minimise}_{x\in X}\ & \sup_{u\in U}[\dot{b}_N(x,u) + \alpha_N(b_N(x))] \\ & \text{subject to } x \in C^*\end{aligned}$$

Input Constrained Control Barrier Function

The **QP problem** assumes the form:

$$u^* = \underset{u \in \mathbb{R}}{\operatorname{argmin}} \ \frac{1}{2} \ (u - u_d)^2$$

$$\text{subject to } L_f b_2(x) + L_g b_2(x) u \ \geq -2b_2$$
$$u \in U$$

u_d is the desired acceleration computed from the Control Lyapunov Function

$V(x) = (x_2 - v_{max})^2$ by solving the relation:

$$L_f V(x) + L_g V(x) u_d = -\gamma(V(x))$$

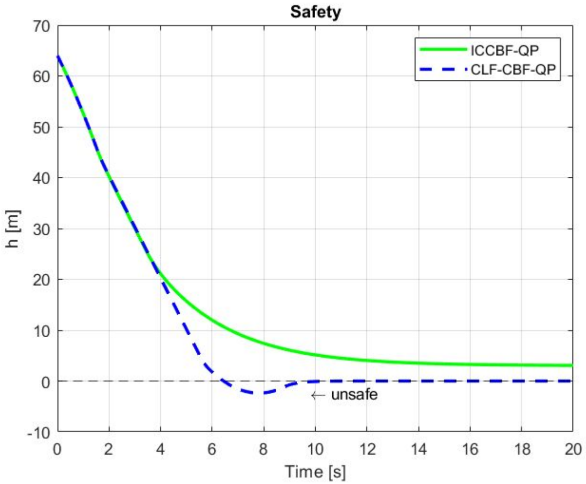
Control Lyapunov Function - Control Barrier Function

The **QP problem** in this case assumes the form:

$$
\begin{aligned}
u^*(x) = \underset{u\in\mathbb{R},\delta\in\mathbb{R}_+}{\operatorname{argmin}} \quad & \frac{1}{2}u^2 + 0.1\delta^2 \\
\text{subject to} \quad & L_fV(x) + L_gV(x)u \leq -10V(x) + \delta \\
& L_fh(x) + L_gh(x)u \geq -2h(x)
\end{aligned}
$$

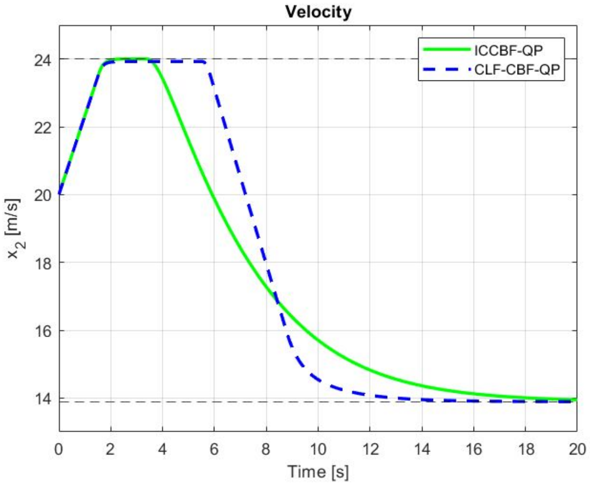
we need to clamp the solution of the **QP** such that $u^*(x)$ lies in the range of feasible control inputs, i.e. $u^*(x) \in U$

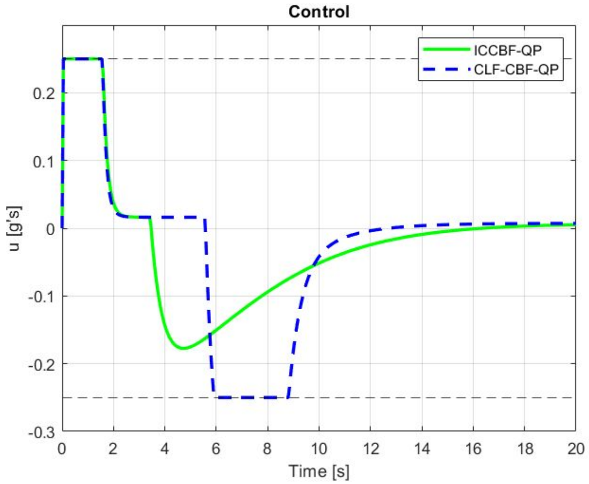
Comparison between ICCBF and CLF-CBF



Safety

Velocity



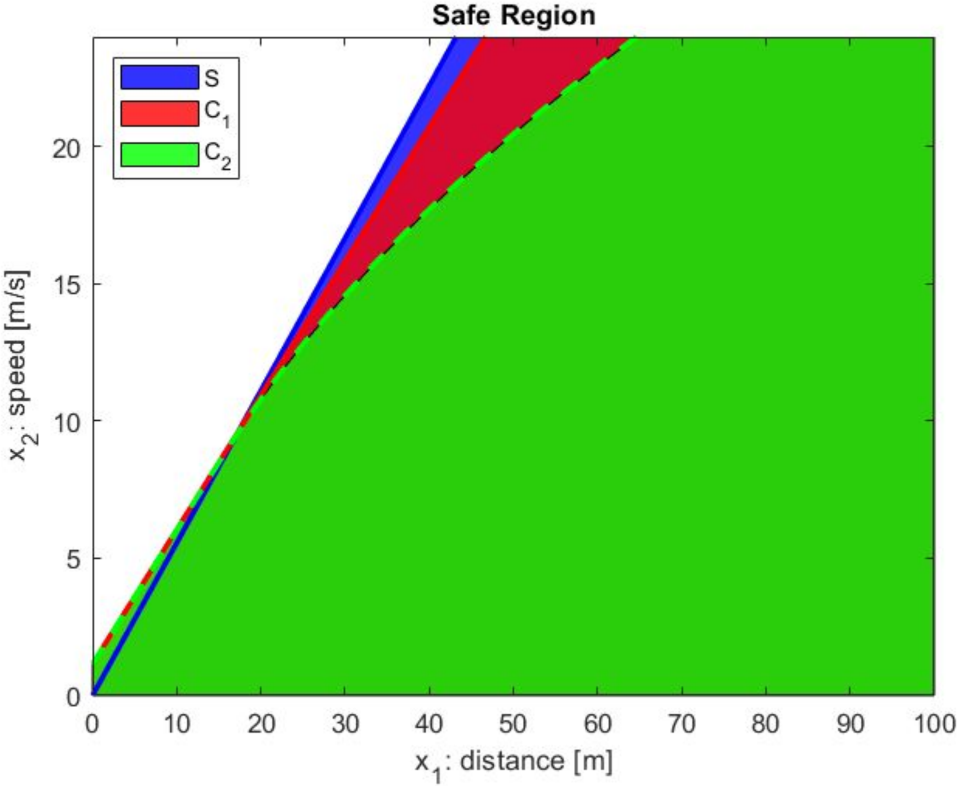


Control

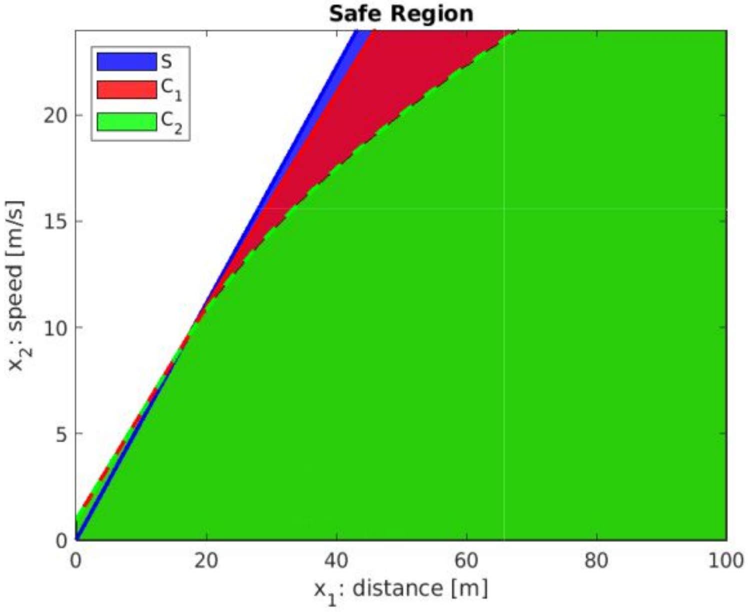
ICCBF - Safe Set

The safe regions are functions of the state and since we are dealing with a 2-dimensional system we are able to visualize them in a plane where on the *x-axis* we have the state *d* and on the *y-axis* the state *v*.

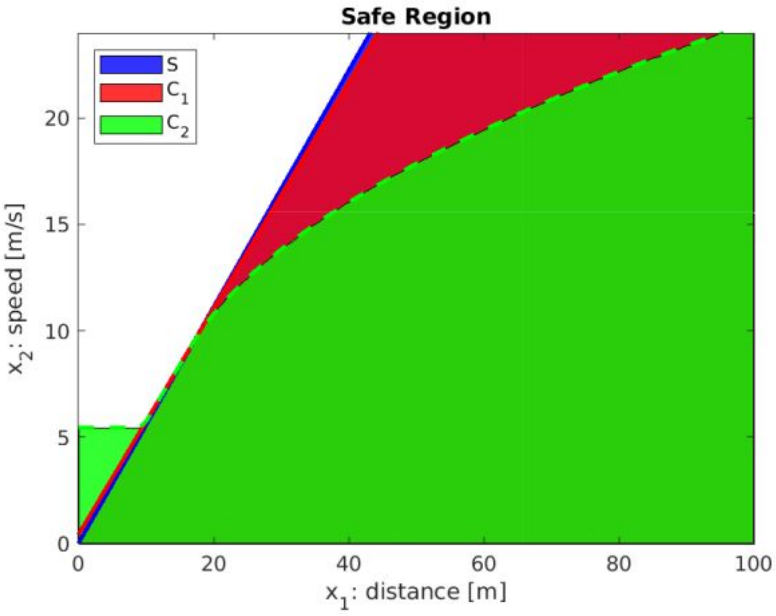
The **safe set** is the intersection between the sets S, C1, C2.



ICCBF - Safe Set: changing k_0



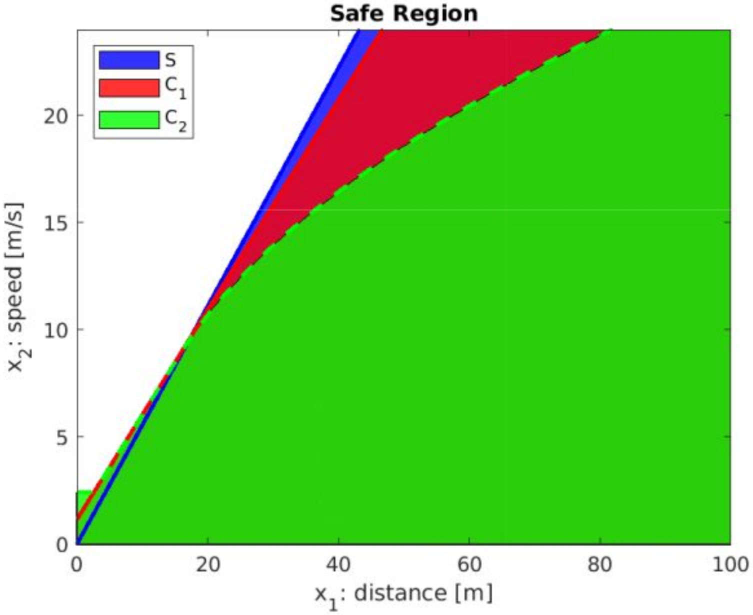
(a) $k_0 = 5\ k_1 = 7\ k_2 = 2$



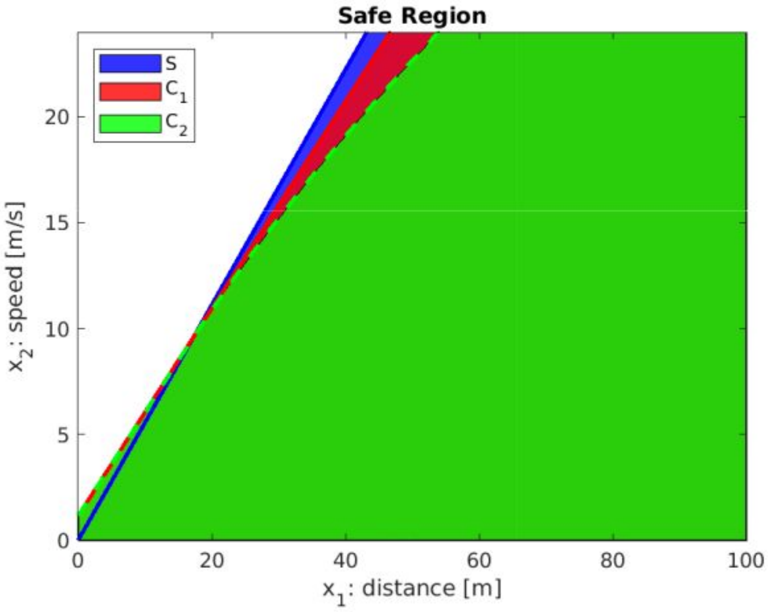
(b) $k_0 = 12\ k_1 = 7\ k2 = 2$

Increasing k_0 reduces the safety at higher speeds and distances.

ICCBF - Safe Set: changing k_1



(a) $k_0 = 4 \ k_1 = 5 \ k_2 = 2$

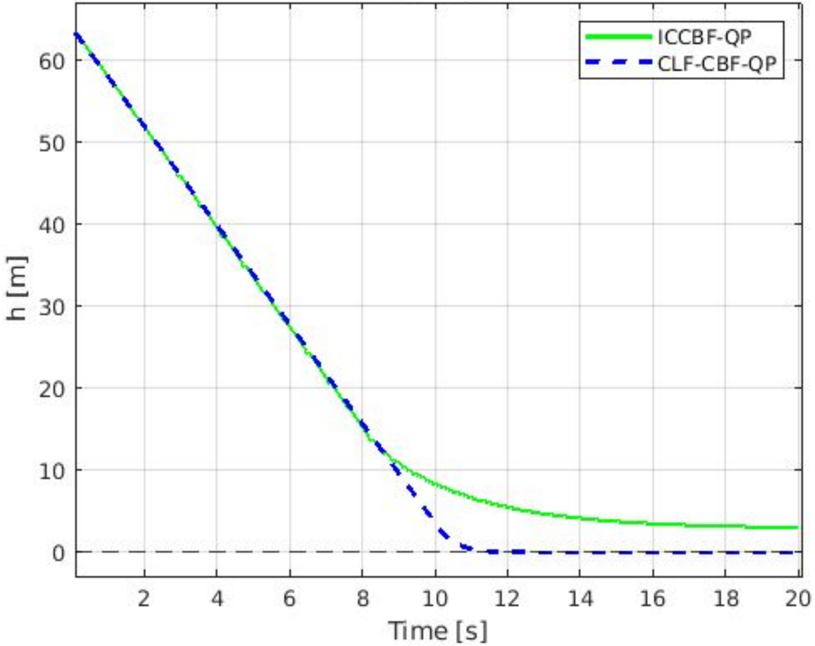


(b) $k_0 = 4 \ k_1 = 11 \ k_2 = 2$

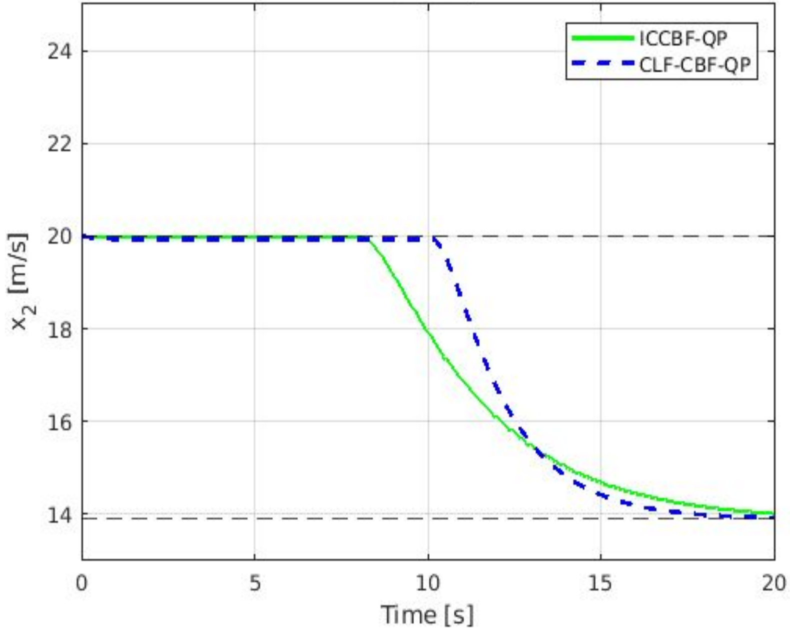
Increasing k_1 increases the safety at higher speeds.

Validity of ICCBF: $v_{max} = 20$ m/s

Safety



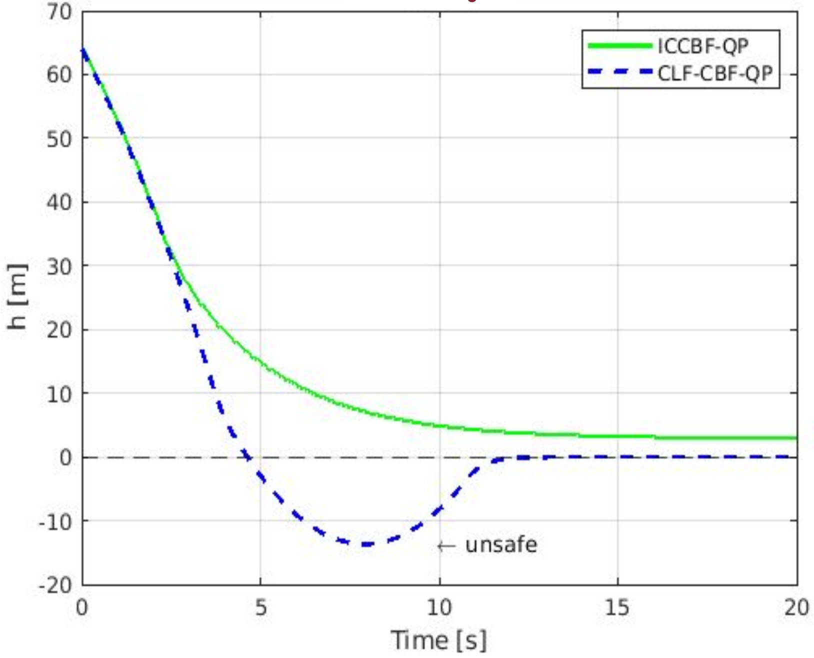
Velocity



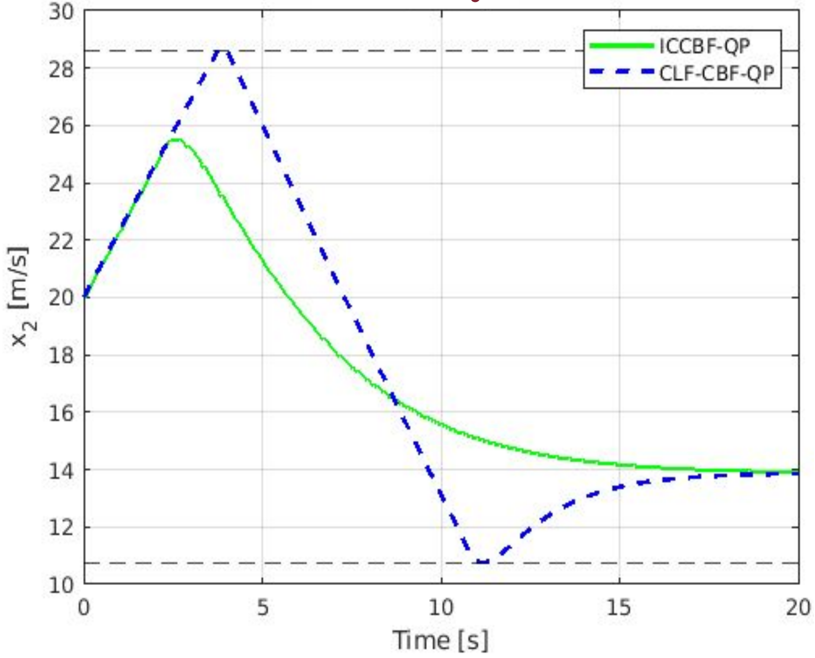
Reducing the target velocity makes CLF-CBF-QP safe

Validity of ICCBF: v_{max} = 40 m/s

Safety

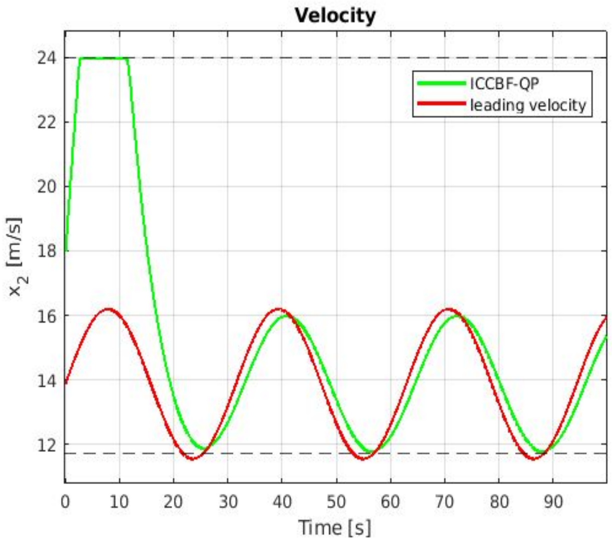


Velocity



A higher target velocity makes CLF-CBF-QP lose safety earlier and by a larger margin.

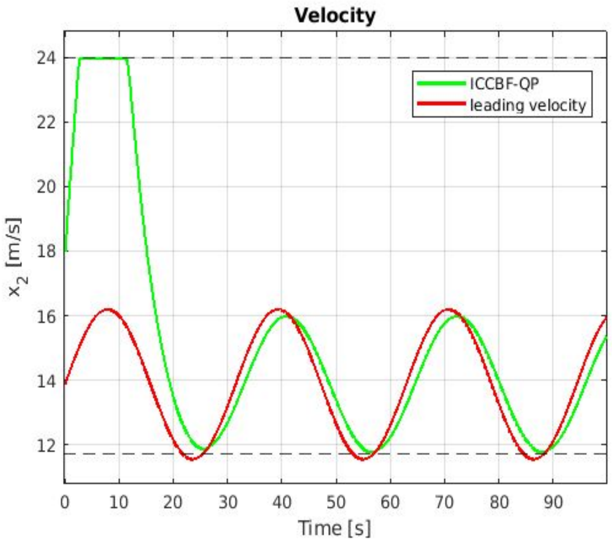
ICCBF - Sinusoidal Velocity Profile



Two state model

The presence of the leading acceleration in the **ICCBF** may suggest that it could act as a "feedforward", but we can see that the tracking the leading speed is unchanged.

Three state model



CLF-CBF With Optimal CBF

We consider also the evolution of the leading velocity and the control is given in terms of force, rather than in terms of acceleration

$$\dot{x} = \begin{bmatrix} \dot{v}_f \\ \dot{v}_\ell \\ \dot{d} \end{bmatrix} = \begin{bmatrix} -F(v_f)/m \\ a_\ell \\ x_2 - x_1 \end{bmatrix} + \begin{bmatrix} 1/m \\ 0 \\ 0 \end{bmatrix} u$$

The resulting optimal CBF takes the expression:

$$h^o(x) = D - \Delta^{o*} \longrightarrow \begin{cases} \Delta_1^{o*} = 1.8 v_f \\ \Delta_2^{o*} = \frac{1}{2}\frac{(1.8 a_f g_0 - v_f)^2}{a_f g_0} + 1.8 v_f - \frac{v_\ell^2}{2 a_\ell g_0} \\ \Delta_3^{o*} = \frac{1}{2}\frac{(v_\ell + 1.8 a_f g_0 - v_f)^2}{(a_f - a_\ell) g_0} + 1.8 v_f \end{cases}$$

CLF-CBF With Optimal CBF

Solution of the Optimization Procedure

1. When $a_\ell = a_f$

$$\Delta^{o*} = \begin{cases} \Delta_1^{o*} & \text{if } 0 < v_f < v_\ell + 1.8 a_f g_0 \\ \Delta_2^{o*} & \text{otherwise} \end{cases}$$

2. When $a_\ell < a_f$

$$\Delta^{o*} = \begin{cases} \Delta_1^{o*} & \text{if } 0 < v_f < v_\ell + 1.8 a_f g_0 \\ \Delta_2^{o*} & \text{if } v_f \geq \frac{a_f}{a_\ell} v_\ell + 1.8 a_f g_0 \\ \Delta_3^{o*} & \text{otherwise} \end{cases}$$

3. When $a_\ell > a_f$

$$\Delta^{o*} = \begin{cases} \Delta_1^{o*} & \text{if } 0 < v_f < \sqrt{\frac{a_f}{a_\ell}} v_\ell + 1.8 a_f g_0 \\ \Delta_2^{o*} & \text{otherwise} \end{cases}$$

CLF-CBF With Optimal CBF

QP problem formulation

$$u^*(x) = \underset{u=[u \;\; \delta]^T \in U_{acc} \times \mathbb{R}}{\operatorname{argmin}} \frac{1}{2} u^T H_{acc} u + F_{acc}^T u$$

$$\begin{aligned} \text{subject to} \quad & A_{clf} u \leq b_{clf} \\ & A_{cbf} u \leq b_{cbf} \\ & A_{fc} u \leq b_{fc} \end{aligned}$$

Matrices and Constraints

$$H_{acc} = 2 \begin{bmatrix} \frac{1}{m^2} & 0 \\ 0 & p_{sc} \end{bmatrix} \qquad F_{acc} = -2 \begin{bmatrix} F(v_f)/m^2 \\ 0 \end{bmatrix}$$

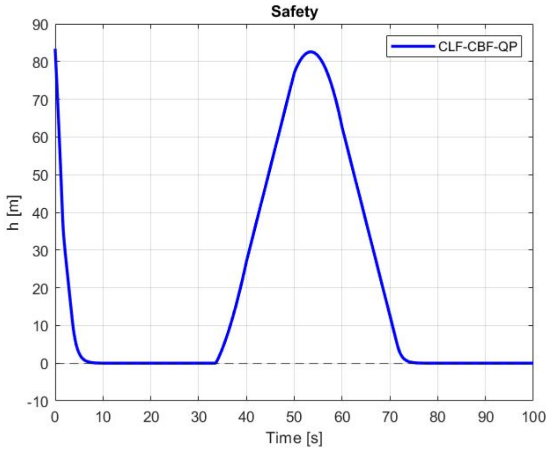
$$A_{clf} = \begin{bmatrix} L_g V(x) & -1 \end{bmatrix} \qquad b_{clf} = -LfV(x) - cV(x)$$

$$A_{cbf} = \begin{bmatrix} L_g h(x) & 0 \end{bmatrix} \qquad b_{cbf} = -L_f h(x) + h(x)$$

$$A_{fc} = \begin{bmatrix} 1 & 0 \\ -1 & 0 \end{bmatrix} \qquad b_{fc} = \begin{bmatrix} a'_f m g_0 \\ a_f m g_0 \end{bmatrix}$$

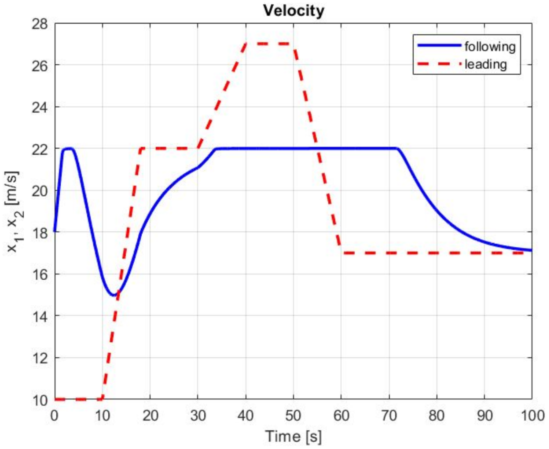
CLF-CBF with Optimal CBF

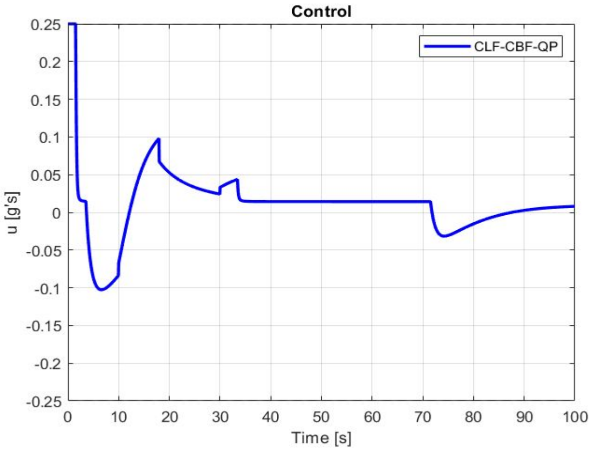
Implementation Results



Safety

Velocity

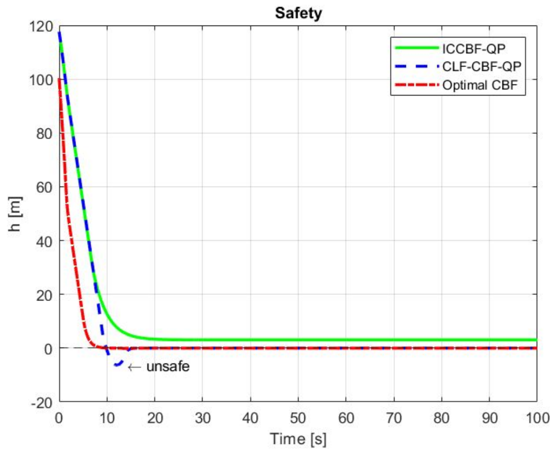




Control

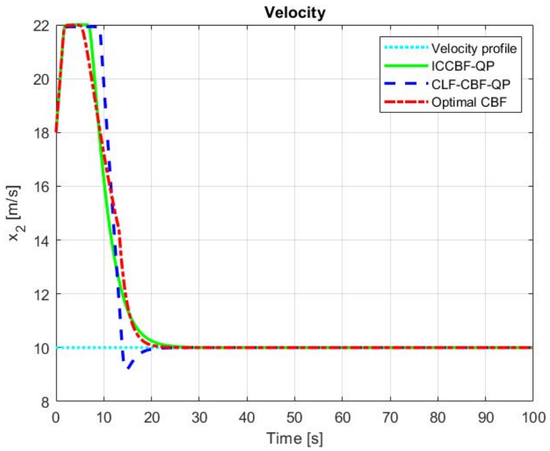
Comparison between Methods

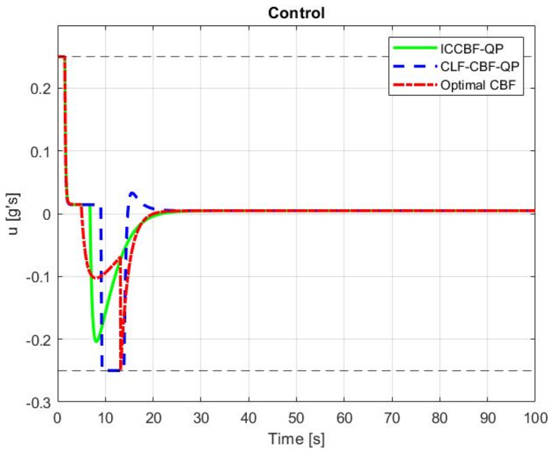
Constant Leader Speed



Safety

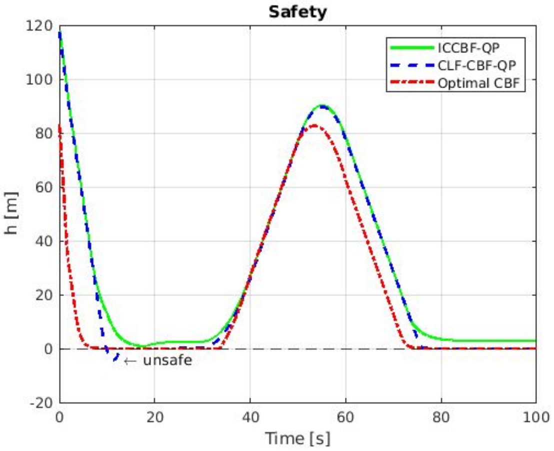
Velocity





Control

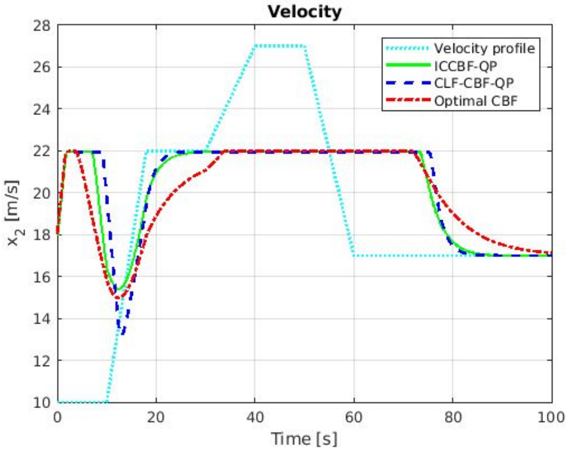
Comparison between Methods

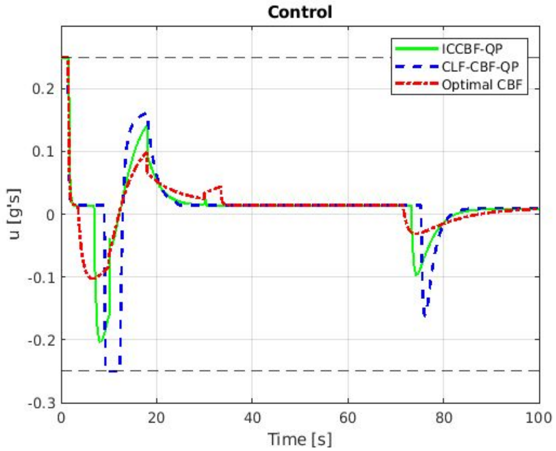


Safety

Trapezoidal Leader Speed

Velocity





Control

Conclusions

We simulated the **ACC** problem under input constraints applying the following methods:

- **Input Constrained Control Barrier Function (ICCBF)**
- **Control Lyapunov Function - Control Barrier Function (CLF-CBF)**
- **Optimal Control Barrier Function (Optimal CBF)**

We have found that the behaviour of the **CLF-CBF** method is unacceptable as it lets the system enter in an unsafe state. The **ICCBF** and **Optimal CBF** methods both manage to always keep the system in a safe state but the former is more conservative than the latter.

We can conclude that in more complex problems where an **Optimal CBF** method is infeasible the **ICCBF** method must be used to guarantee safety instead of the **CLF-CBF**.

References

1. Aaron D Ames, Samuel Coogan, Magnus Egerstedt, Gennaro Notomista, Koushil Sreenath, and Paulo Tabuada. Control barrier functions: Theory and applications. In 2019 18th European control conference (ECC), pages 3420–3431. IEEE, 2019.

2. Devansh Agrawal and Dimitra Panagou. Safe control synthesis via input constrained control barrier functions. arXiv preprint arXiv:2104.01704, 2021.

3. Aaron D Ames, Xiangru Xu, Jessy W Grizzle, and Paulo Tabuada. Control barrier function based quadratic programs for safety critical systems. IEEE Transactions on Automatic Control, 62(8):3861–3876, 2016

ACC for ICCBF and CLF-CBF

Simulation parameters

Variable	Description	Value
$d(0)$	initial distance between the vehicles	$100\ m$
$v(0)$	follower initial velocity	$20\ \frac{m}{s}$
m	mass of the following car	$1650\ Kg$
f_0	constant term of friction	$0.1\ N$
f_1	linear term of friction	$5\ N \cdot (\frac{m}{s})^{-1}$
f_2	quadratic term of friction	$0.25\ N \cdot (\frac{m}{s})^{-2}$
g_0	gravitational acceleration	$9.81\ \frac{m}{s^2}$
v_0	leader constant velocity	$13.89\ \frac{m}{s}$
v_{max}	maximum follower velocity	$24\ \frac{m}{s}$
u_{max}	maximum absolute value of input control as a fraction of g_0	0.25
$\gamma(V(x))$	class-$\mathcal{K}$ function for CLF-CBF	$10 \cdot V(x)$
k_0	parameter of α_0	4
k_1	parameter of α_1	7
k_2	parameter of α_2	2

CLF-CBF With Optimal CBF

Simulation parameters

Variable	Description	Value
$D(0)$	initial distance between the vehicles	$150\ m$
$v_f(0)$	follower initial velocity	$18\ \frac{m}{s}$
$v_\ell(0)$	leader initial velocity	$10\ \frac{m}{s}$
m	mass of the following car	$1650\ Kg$
f_0	constant term of friction	$0.1\ N$
f_1	linear term of friction	$5\ N \cdot (\frac{m}{s})^{-1}$
f_2	quadratic term of friction	$0.25\ N \cdot (\frac{m}{s})^{-2}$
g_0	gravitational acceleration	$9.81\ \frac{m}{s^2}$
v_{max}	maximum follower velocity	$24\ \frac{m}{s}$
$a'_f = a_f$	maximum follower acceleration/deceleration	$0.25\ \frac{m}{s^2}$
p_{sc}	weight for δ in `quadprog`	100
c	CLF constant	10