SAPIENZA
UNIVERSITÀ DI ROMA

DEPARTMENT OF COMPUTER, CONTROL AND
MANAGEMENT ENGINEERING ANTONIO RUBERTI

Model-free Control Barrier Functions for obstacle avoidance

AUTONOMOUS AND MOBILE ROBOTICS

MASTER PROGRAMS IN
ARTIFICIAL INTELLIGENCE AND ROBOTICS
& CONTROL ENGINEERING

Professor:
Giuseppe Oriolo

Students:

Dennis Rotondi	1834864
Mirko Mizzoni	1841476
Marco Montagna	1882418

Academic Year 2022/2023

Contents

Introduction

As the reality of autonomous machines is becoming increasingly prevalent, there is a growing interest in designing systems that are safe. In the past, Control Theory has primarily focused on the property of liveness, specifically asymptotic stability, rather than safety. Intuitively, the crucial difference between those two properties is that safety refers to the absence of "bad" outcomes, while liveness requires the eventual occurrence of "good" outcomes. This distinction suggests that prioritizing the avoidance of fatal events may be favored. Hovewer, the development of controllers for safety-critical systems (those systems for which safety is a major design consideration) typically requires the use of sophisticated and high-dimensional dynamical models. As a consequence, synthesizing control laws for these systems could be complex and challenging with a non-trivial implementaion. In this project, our focus is on Control Barrier Functions (CBFs), which can be viewed as the safety property in the form of invariance, in the sense that any trajectory starting inside an invariant set will never reach the complement of the set, which represents the area where undesirable events occur. We will compare the traditional approach outlined in [1] to the 2021 model-free alternative presented in [2] that tries to attenuate the need of a huge dynamic model in the control synthesis for a specific subset of them. We aim to explore the benefits and drawbacks of these methods in the context of obstacle avoidance simulation of robotics systems in MATLAB. The remainder of this report is organized as follows. The next Section (1) introduces some concepts from the analysis, stability, modeling and control of nonlinear systems together with the notion of CBFs which will be used throughout the report. Building upon this, in Section 2, we formulate the reduced optimization-based framework together with the model-free CBFs, required to ensure the safety of an invariant set, pointing out the importance of disturbances. Finally, the results of applying the developed approach to a point robot and to a unicycle are shown in Section 3.

1 Safety-Critical Control for Robotic Systems

A safety controller is not intended to function independently, but rather to support the primary, potentially unsafe controller that is designed to achieve standard control objectives such as stability, regulation, or tracking. To guarantee full comprehension of the tools used hereinafter, we will briefly review several concepts and provide the necessary definitions. Firstly, the robotic systems under examination are those in the nonlinear affine form:

$$\dot{\boldsymbol{x}} = \boldsymbol{f}(\boldsymbol{x}) + \boldsymbol{g}(\boldsymbol{x})\boldsymbol{u}, \tag{1}$$

where $\boldsymbol{x} \in X \subseteq \mathbb{R}^n$ is the state, $\boldsymbol{u} \in U \subseteq \mathbb{R}^m$ the control input, with $\boldsymbol{f} : X \to \mathbb{R}^n$ and $\boldsymbol{g} : X \to \mathbb{R}^{n \times m}$ locally Lipschitz continuous functions.

1.1 Control Lyapunov Functions

In Lyapunov theory, stabilizing (1) to a point $\boldsymbol{x}^* = \boldsymbol{0}$ (assuming w.l.o.g. $\boldsymbol{x}^*$ is an equilibrium and X is an open and connected neighborhood of $\boldsymbol{x}^*$) can be achieved by finding a feedback control law $\boldsymbol{u} = \boldsymbol{k}(\boldsymbol{x})$ that drives a positive definite function, $V : D \subset \mathbb{R}^n \to \mathbb{R}_{\geq 0}$, to zero. That is, if:

$$\forall \boldsymbol{x} \in X \qquad \inf_{\boldsymbol{u} \in U} [\dot{V}(\boldsymbol{x}, \boldsymbol{u})] \leq -\gamma(V(\boldsymbol{x})), \tag{2}$$

where

$$\dot{V}(\boldsymbol{x}, \boldsymbol{k}(\boldsymbol{x})) = \nabla V(\boldsymbol{x}) \cdot (\boldsymbol{f}(\boldsymbol{x}) + \boldsymbol{g}(\boldsymbol{x})\boldsymbol{u}) = L_{\boldsymbol{f}} V(\boldsymbol{x}) + L_{\boldsymbol{g}} V(\boldsymbol{x})\boldsymbol{u}. \tag{3}$$

Note: $\gamma : \mathbb{R}_{\geq 0} \to \mathbb{R}_{\geq 0}$ is a class-$\mathcal{K}$ function here, recalling that a continuous function $\zeta : [0, b) \to \mathbb{R}_{\geq 0},\ b \in \mathbb{R}_{\geq 0}$ is of class-$\mathcal{K}$ (or $\zeta : [-a, b) \to \mathbb{R}_{\geq 0},\ a, b \in \mathbb{R}_{\geq 0}$ is of extended class-$\mathcal{K}$) if it is strictly monotonically increasing and $\zeta(0) = 0$. We say to have an extended class-$\mathcal{K}_\infty$ if $(a, b) = (-\infty, \infty) = \mathbb{R}$, the entire real line.

Theorem 1. If V is a control Lyapunov function (CLF) for (1), i.e. a positive definite function satisfing (2), then any Lipschitz continuous feedback controller $\boldsymbol{u}(\boldsymbol{x})$ satisfying $\dot{V}(\boldsymbol{x}, \boldsymbol{u}) \leq -\gamma(V(\boldsymbol{x}))\ \forall \boldsymbol{x} \in X$ asymptotically stabilizes the system to $\boldsymbol{x}^*$.

1.1.1 Exponential Stability

Another very important stability property can be introduced from the following:

Definition 1. If there exist $c, k_1, k_2, \lambda \in \mathbb{R}_{>0}$ such that $\forall \boldsymbol{x} \in X$:

$$\begin{aligned} k_1 ||\boldsymbol{x}||^c \leq V(\boldsymbol{x}) \leq k_2 ||\boldsymbol{x}||^c, \\ \inf_{\boldsymbol{u} \in \mathbb{R}^m} [\dot{V}(\boldsymbol{x}, \boldsymbol{u})] \leq -\lambda V(\boldsymbol{x}), \end{aligned} \tag{4}$$

for a continuously differentiable function $V : X \to \mathbb{R}_{\geq 0}$, then V is a CLF for (1).

Theorem 2. If V is a CLF that respects Definition 1, then any locally Lipschitz continuous controller $\boldsymbol{u} = \boldsymbol{k}(\boldsymbol{x})$ satisfying

$$\dot{V}(\boldsymbol{x}, \boldsymbol{u}) \leq -\lambda V(\boldsymbol{x}), \tag{5}$$

$\forall \boldsymbol{x} \in X$ renders $\boldsymbol{x}^*$ exponentially stable i.e. there exist $a, A, \beta \in \mathbb{R}_{>0}$ such that $||\boldsymbol{x}_0|| \leq a \Rightarrow ||\boldsymbol{x}(t)|| \leq Ae^{-\beta t}||\boldsymbol{x}_0||,\ \forall t \geq 0$.

1.2 Control Barrier Functions

In contrast to stability, which is concerned with guiding a system towards a specific point or set, safety can be conceptualized as enforcing the invariance of a set. For a dynamic system, safety constraints can be viewed as defining a safe region within its state space where the state must always reside. This region is referred to as the safe set $\mathcal{S}$.

Definition 2. Let $\mathcal{S} \subset X \subseteq \mathbb{R}^n$ be a compact set that is the 0-superlevel set of a continuously differentiable function $h : X \to \mathbb{R}$, $\forall \boldsymbol{x} \in X$

$$\begin{aligned} \mathcal{S} =& \{\boldsymbol{x} \in \mathbb{R}^n : \ \ h(\boldsymbol{x}) \geq 0\}, \\ \partial\mathcal{S} =& \{\boldsymbol{x} \in \mathbb{R}^n : \ \ h(\boldsymbol{x}) = 0\}, \\ \text{Int}(\mathcal{S}) =& \{\boldsymbol{x} \in \mathbb{R}^n : \ \ h(\boldsymbol{x}) > 0\}. \end{aligned} \tag{6}$$

A matematical tool as a means to guarantee the invariance property of set $\mathcal{S}$ for (1) is represented by the use of Control Barrier Functions (CBF).

Definition 3. System (1) is *safe* w.r.t. $\mathcal{S}$, if $\mathcal{S}$ is forward invariant under (1), that is, $\boldsymbol{x}_0 \in \mathcal{S} \Rightarrow \boldsymbol{x}(t) \in \mathcal{S}, \forall t \geq 0$. Then, h is a CBF for (1) if $\frac{\partial h}{\partial \boldsymbol{x}}(\boldsymbol{x}) \neq 0$ for all $\boldsymbol{x} \in \partial\mathcal{S}$ and there exists $\alpha \in \mathbb{R}_{>0}$ such that $\forall \boldsymbol{x} \in \mathcal{S}$

$$\sup_{\boldsymbol{u} \in U} \underbrace{\Big[L_{\boldsymbol{f}} h(\boldsymbol{x}) + L_{\boldsymbol{g}} h(\boldsymbol{x}) \boldsymbol{u}\Big]}_{\dot{h}(\boldsymbol{x}, \boldsymbol{u})} \geq -\alpha h(\boldsymbol{x}). \tag{7}$$

Theorem 3. If $h : X \to \mathbb{R}$ is a CBF for (1), then any locally Lipschitz continuous controller $\boldsymbol{u} = \boldsymbol{k}(\boldsymbol{x})$ satisfying

$$\dot{h}(\boldsymbol{x}, \boldsymbol{k}(\boldsymbol{x})) \geq -\alpha h(\boldsymbol{x}), \tag{8}$$

$\forall \boldsymbol{x} \in \mathcal{S}$ renders (1) safe with respect to $\mathcal{S}$. Additionally, the set $\mathcal{S}$ is asymptotically stable in X, hence with the beneficial of drive the system back to the set $\mathcal{S}$ in case of noise or modeling errors.

Generically for these results $\alpha h(\boldsymbol{x})$ may be taken as any extended class-$\mathcal{K}_\infty$ function $\alpha(\cdot)$ of h. It interesting to point out that CLFs yield invariant level sets as well. If these level sets are contained in the safe set one can guarantee safety, however clearly

they are overly strong and conservative. On the other hand, the action of CBFs can be modulated by $\alpha(\cdot)$ depending on whether a more conservative or aggressive behavior is desired, while allowing the state to evolve freely within the safe set $\mathcal{S}$. Ideally, as $\alpha \to \infty$, the CBF would activate only on the boundary of the set.

1.2.1 Control synthesis

Theorem 3 establishes safety-critical controller synthesis by means of condition (8). However, satisfying it may require a complex analytic procedure. A minimally invasive strategy to obtain this result is to set up an optimization-based control problem whose objectives is to minimize the difference between the desired, possibly unsafe controller $\boldsymbol{k}_d(\boldsymbol{x})$ and the actuated one $\boldsymbol{u}$, so as to satisfy the CBF constraint

$$\begin{aligned} \boldsymbol{k}(\boldsymbol{x}) = \underset{\boldsymbol{u}\in U}{\arg\min}(\boldsymbol{u}-\boldsymbol{k}_d(\boldsymbol{x}))^T(\boldsymbol{u}-\boldsymbol{k}_d(\boldsymbol{x})), \\ \text{s.t. } \dot{h}(\boldsymbol{x},\boldsymbol{u}) \geq -\alpha h(\boldsymbol{x}). \end{aligned} \tag{9}$$

Additional requirements can be imposed and it can be shown that the controller obtained through this method is Lipschitz continuous.

1.3 Effect of disturbances

In reality, all robotics systems are typically subject to unknown disturbances that may affect and so compromise the stability and safety properties. For instance, a bounded disturbance $\boldsymbol{d} \in \mathbb{R}^m$ added to the input $\boldsymbol{u}$ leads to the system $\dot{\boldsymbol{x}} = \boldsymbol{f}(\boldsymbol{x}) + \boldsymbol{g}(\boldsymbol{x})(\boldsymbol{u}+\boldsymbol{d})$. To deal with these disturbances, the notion of exponential *input-to-state stability* (ISS) is of paramount importance. It comes by extending the definition of exponential stability by requiring the existency of a class-$\mathcal{K}$ function μ such that if $||\boldsymbol{x}_0|| \leq a \Rightarrow ||\boldsymbol{x}(t)|| \leq Le^{-\beta t}||\boldsymbol{x}_0|| + \mu(||\boldsymbol{d}||_\infty)$, $\forall t \geq 0$; where $||\cdot||_\infty$ is the maximum norm. Since $\beta > 0$, solutions converge to a neighborhood of the origin which depends on the size of the disturbance. Exponential ISS is achieved by requiring that

$$\dot{V}(\boldsymbol{x},\boldsymbol{u},\boldsymbol{d}) \leq -\lambda V(\boldsymbol{x}) + \iota(||\boldsymbol{d}||_\infty), \tag{10}$$

for some class-$\mathcal{K}$ function ι. Similarly, safety can be extended to *input-to-state safety* (ISSf) by requiring that the system stays within a neighbourhood $\mathcal{S}_d \supseteq \mathcal{S}$ of the safe set $\mathcal{S}$, which clearly depends on the entity of the disturbance. This neighbourhood is defined as a 0-superlevel set

$$S_d = \{\boldsymbol{x} \in X : h(\boldsymbol{x}) + \gamma(||\boldsymbol{d}||_\infty) \geq 0\}, \tag{11}$$

with some class-$\mathcal{K}$ function γ. ISSf is guaranteed by imposing over (8) in Theorem 3

$$\dot{h}(\boldsymbol{x},\boldsymbol{u},\boldsymbol{d}) \geq -\alpha h(\boldsymbol{x}) - \iota(||\boldsymbol{d}||_\infty). \tag{12}$$

2 Model Free Safety-Critical Control

The work in [2] proposes an approach in which a safe velocity is designed based on reduced-order kinematics, and then it is tracked by a velocity tracking controller that may be model aware. Once velocity tracking is established, enforcing safety does not require further consideration of the high-fidelity model, and this is referred to as *model-free safety-critical control*. Consider now a robotic system with configuration space Q of dimension n and generalized coordinates $\boldsymbol{q} \in Q$, set of admissible inputs $U \subseteq \mathbb{R}^m$, control input $\boldsymbol{u} \in U$, and dynamics

$$\boldsymbol{M}(\boldsymbol{q})\ddot{\boldsymbol{q}} + \boldsymbol{C}(\boldsymbol{q}, \dot{\boldsymbol{q}})\dot{\boldsymbol{q}} + \boldsymbol{G}(\boldsymbol{q}) = \boldsymbol{B}\boldsymbol{u}, \tag{13}$$

where $\boldsymbol{M}(\boldsymbol{q}) \in \mathbb{R}^{n\times n}$ is the inertia matrix, $\boldsymbol{C}(\boldsymbol{q}, \dot{\boldsymbol{q}}) \in \mathbb{R}^{n\times n}$ collects centrifugal and Coriolis forces, $\boldsymbol{G}(\boldsymbol{q}) \in \mathbb{R}^n$ is the gravity term, and $\boldsymbol{B} \in \mathbb{R}^{n\times m}$ is the input matrix. The presence of a constant matrix $\boldsymbol{B}$ and the absence of constraints do not significantly hinder this approach, as it is, for many cases of interest, possible to employ the reduced dynamic model that allows for a constant $\boldsymbol{B}$ to demonstrate the results in this section. For example, the unicycle ($n = 3,\ m = 2$) dynamic model can be written as

$$\boldsymbol{M}\ddot{\boldsymbol{q}} = \boldsymbol{B}(\boldsymbol{q})\boldsymbol{u} + \boldsymbol{A}(\boldsymbol{q})\lambda, \tag{14}$$

$$\boldsymbol{A}^T(\boldsymbol{q})\dot{\boldsymbol{q}} = 0, \tag{15}$$

where $\lambda \in \mathbb{R}$ represents the *Lagrangian multiplier* and $\boldsymbol{A}(\boldsymbol{q})$ is the transpose of the row vector characterizing the kinematic constraint. The kinematic model is

$$\dot{\boldsymbol{q}} = \boldsymbol{H}(\boldsymbol{q})\boldsymbol{v}, \tag{16}$$

where the columns of $\boldsymbol{H}(\boldsymbol{q})$ are a basis for the null space of $\boldsymbol{A}^T(\boldsymbol{q})$ i.e. $\boldsymbol{A}^T(\boldsymbol{q})\boldsymbol{H}(\boldsymbol{q}) = \boldsymbol{0}$. From the time derivative of (16), and by left multiplying (14) for $\boldsymbol{H}(\boldsymbol{q})^T$ we obtain

$$\boldsymbol{M}'\dot{\boldsymbol{v}} = \boldsymbol{I}_2\boldsymbol{u}, \tag{17}$$

with $\boldsymbol{M}' = \boldsymbol{H}^T(\boldsymbol{q})\boldsymbol{M}\boldsymbol{H}(\boldsymbol{q})$. As expected the $\boldsymbol{B}$ matrix is now the identity and the kinematic constraints no longer need to be considered in the dynamics. It's not hard to put (13) into (1) using a change of coordinates $\boldsymbol{x} = (\boldsymbol{q}, \dot{\boldsymbol{q}})^T$ keeping valid all the previous results, few of them will be rewritten to highlight the model-free aspect.

2.1 Model Free Control Barrier Functions

We will now trace the origins of the proposed model-free approach and outline the required assumptions. Let us consider the control law $\boldsymbol{k} : Q \times \mathbb{R}^n \to \mathbb{R}^m,\ \boldsymbol{u} = \boldsymbol{k}(\boldsymbol{q}, \dot{\boldsymbol{q}})$, initial conditions $\boldsymbol{q}(0) = \boldsymbol{q}_0,\ \dot{\boldsymbol{q}}(0) = \dot{\boldsymbol{q}}_0$ and assume that a unique solution $\boldsymbol{q}(t)$ exists for all $t \geq 0$.

Assumption 1. The safe set is defined as the 0-superlevel set of a continuously differentiable function $h : Q \to \mathbb{R}$

$$\mathcal{S} = \{\boldsymbol{q} \in Q : h(\boldsymbol{q}) \geq 0\}, \tag{18}$$

where the gradient of h is finite: $\exists C_h \in \mathbb{R}_{>0}$ such that $||\nabla h(\boldsymbol{q})|| \leq C_h, \forall \boldsymbol{q} \in \mathcal{S}$. This means that safety depends on the configuration $\boldsymbol{q}$ only and h is independent of $\dot{\boldsymbol{q}}$.
We want to achieve safety by generating a safe velocity $\dot{\boldsymbol{q}}_s \in \mathbb{R}^n$ that needs to satisfy:

$$\nabla h(\boldsymbol{q}) \cdot \dot{\boldsymbol{q}}_s \geq -\alpha h(\boldsymbol{q}), \tag{19}$$

for some $\alpha \in \mathbb{R}_{>0}$ to be selected. Note that (19) is a kinematic condition that does not depend on the full dynamics (13). An error to track the safe velocity is defined as:

$$\dot{\boldsymbol{e}} = \dot{\boldsymbol{q}} - \dot{\boldsymbol{q}}_s \tag{20}$$

Assumption 2. The velocity tracking controller $\boldsymbol{u} = \boldsymbol{k}(\boldsymbol{q}, \dot{\boldsymbol{q}})$ achieves exponential stable tracking: $||\dot{\boldsymbol{e}}(t)|| \leq A||\dot{\boldsymbol{e}}_0||e^{-\lambda t}$ for some $A, \lambda \in \mathbb{R}_{>0}$. Thus, if $\dot{\boldsymbol{e}}$ is differentiable there exists a continuously differentiable Lyapunov function $V : Q \times \mathbb{R}^n \to \mathbb{R}_{\geq 0}$ such that $\forall (\boldsymbol{q}, \dot{\boldsymbol{e}}) \in Q \times \mathbb{R}^n$:

$$k_1||\dot{\boldsymbol{e}}|| \leq V(\boldsymbol{q}, \dot{\boldsymbol{e}}) \leq k_2||\dot{\boldsymbol{e}}||, \tag{21}$$

for some $k_1, k_2 \in \mathbb{R}_{>0}$ such that $\forall (\boldsymbol{q}, \dot{\boldsymbol{e}}, \dot{\boldsymbol{q}}, \ddot{\boldsymbol{q}}_s) \in Q \times \mathbb{R}^n \times \mathbb{R}^n \times \mathbb{R}^n$ $\boldsymbol{u}$ satisfies:

$$\dot{V}(\boldsymbol{q}, \dot{\boldsymbol{e}}, \dot{\boldsymbol{q}}, \ddot{\boldsymbol{q}}_s, \boldsymbol{u}) \leq -\lambda V(\boldsymbol{q}, \dot{\boldsymbol{e}}). \tag{22}$$

In particular, for a tracking controller satisfying the previous assumption, stability translates into safety for the dynamic system (13) if $\lambda > \alpha$, thanks to:

Theorem 4. Consider system (13), safe set $\mathcal{S}$ (18), and velocity tracking controller satisfying (22). If $\lambda > \alpha$, safety is achieved such that $(\boldsymbol{q}_0, \dot{\boldsymbol{e}}_0) \in \mathcal{S}_V \Rightarrow \boldsymbol{q}(t) \in \mathcal{S}, \forall t \geq 0$, where

$$\mathcal{S}_V = \{(\boldsymbol{q}, \dot{\boldsymbol{e}}) \in Q \times \mathbb{R}^n : h_V(\boldsymbol{q}, \dot{\boldsymbol{e}}) \geq 0\}, \tag{23}$$

$$h_V(\boldsymbol{q}, \dot{\boldsymbol{e}}) = -V(\boldsymbol{q}, \dot{\boldsymbol{e}}) + \alpha_e h(\boldsymbol{q}), \tag{24}$$

with $\alpha_e = (\lambda - \alpha)k_1/C_h > 0$ and C_h, k_1 defined at (18) and (21).

Proof. Since $V(\boldsymbol{q}, \dot{\boldsymbol{e}}) \geq 0$, the implication $h_V(\boldsymbol{q}, \dot{\boldsymbol{e}}) \geq 0 \to h(\boldsymbol{q}) \geq 0$ holds. Thus it is sufficient to show that $h_V(\boldsymbol{q}, \dot{\boldsymbol{e}}) \geq 0$ to prove the theorem. By definition of the initial conditions $h_V(\boldsymbol{q}_0, \dot{\boldsymbol{e}}_0) \geq 0$ and its time derivative

$$\begin{aligned}
\dot{h}_V(\boldsymbol{q}, \dot{\boldsymbol{e}}, \dot{\boldsymbol{q}}, \ddot{\boldsymbol{q}}_s, \boldsymbol{u}) &= -\dot{V}(\boldsymbol{q}, \dot{\boldsymbol{e}}, \dot{\boldsymbol{q}}, \ddot{\boldsymbol{q}}_s, \boldsymbol{u}) + \alpha_e \nabla h(\boldsymbol{q}) \cdot \dot{\boldsymbol{q}} \\
&\geq \lambda V(\boldsymbol{q}, \dot{\boldsymbol{e}}) + \alpha_e \nabla h(\boldsymbol{q}) \cdot \dot{\boldsymbol{q}}_s + \alpha_e \nabla h(\boldsymbol{q}) \cdot \dot{\boldsymbol{e}} \\
&\geq \lambda V(\boldsymbol{q}, \dot{\boldsymbol{e}}) - \alpha_e \alpha h(\boldsymbol{q}) + \alpha_e \nabla h(\boldsymbol{q}) \cdot \dot{\boldsymbol{e}} \\
&\geq (\lambda - \alpha) V(\boldsymbol{q}, \dot{\boldsymbol{e}}) - \alpha h_V(\boldsymbol{q}, \dot{\boldsymbol{e}}) - \alpha_e ||\nabla h(\boldsymbol{q})|| \cdot ||\dot{\boldsymbol{e}}|| \\
&\geq (\lambda - \alpha) k_1 ||\dot{\boldsymbol{e}}|| - \alpha h_V(\boldsymbol{q}, \dot{\boldsymbol{e}}) - \alpha_e C_h ||\dot{\boldsymbol{e}}|| \\
&\geq -\alpha h_V(\boldsymbol{q}, \dot{\boldsymbol{e}}).
\end{aligned}$$

where in the second line was used (22) and definition (20). The next two inequalities follows from (19) and then from (24) together with the Cauchy-Schwarts inequality. For the second to last line the lower and upper bounds (21) and (18) were used. Lastly the definition for α_e from this theorem guarantees $h_V(\boldsymbol{q}(t), \dot{\boldsymbol{e}}(t)) \geq 0, \forall t \geq 0$ for the safety property ensured by Theorem 3. □

The parameter λ depends on how well/fast the actual robot is able to track the safe velocities commands coming from the reduced model. However, if the robot is able to exponentially track these velocities, then, such parameter exists, and therefore there exixts an $\alpha < \lambda$ and safety is achieved. This parameter may not be known in practice.

2.2 Importance of disturbances

Now consider that ideal exponential tracking of the safe velocity is not possible. This is often the case when velocity tracking is done in a model free frashion or when $\dot{\boldsymbol{q}}_s$ is generated without taking into account the dynamics. To mitigate this problem a bounded input disturbance $\boldsymbol{d}$ can be considered to capture the effects of modeling errors. When the input disturbance is introduced instead of safety one shall guarantee ISSf which is the invariance of the larger set $S_d \supseteq S$:

$$\begin{aligned} S_d &= \{\boldsymbol{q} \in Q : h_d(\boldsymbol{q}) \geq 0\}, \\ h_d(\boldsymbol{q}) &= h(\boldsymbol{q}) + \gamma(\|\|\boldsymbol{d}\|\|_\infty), \end{aligned} \tag{25}$$

where γ is a class-$\mathcal{K}$ function to be specified. The dynamic extension of set S_d, $S_{Vd} \supseteq S_V$ can be written as:

$$\begin{aligned} S_{Vd} &= \{(\boldsymbol{q}, \dot{\boldsymbol{e}}) \in Q \times \mathbb{R}^n : h_{Vd}(\boldsymbol{q}, \dot{\boldsymbol{e}}) \geq 0\}, \\ h_{Vd}(\boldsymbol{q}, \dot{\boldsymbol{e}}) &= h_V(\boldsymbol{q}, \dot{\boldsymbol{e}}) + \gamma(||\boldsymbol{d}||_\infty). \end{aligned} \tag{26}$$

Theorem 5 shows that ISSf is guaranteed by exponential ISS tracking. As already shown in (10), for exponential ISS instead of (22) the tracking controller satisfies

$$\dot{V}(\boldsymbol{q}, \dot{\boldsymbol{e}}, \dot{\boldsymbol{q}}, \ddot{\boldsymbol{q}}_s, \boldsymbol{u}, \boldsymbol{d}) \leq -\lambda V(\boldsymbol{q}, \dot{\boldsymbol{e}}) + \iota(||\boldsymbol{d}||_\infty). \tag{27}$$

Theorem 5. Consider system (13), sets S_d and S_{Vd}, in (25) and (26), safe velocity satisfying (19), and velocity tracking controller satisfying (27). If $\lambda > \alpha$, ISSf is achieved such that $(\boldsymbol{q}_0, \dot{\boldsymbol{e}}_0) \in S_{Vd} \rightarrow \boldsymbol{q}(t) \in S_d, \forall t \geq 0$, where α_e is given in Theorem 4 and $\gamma(||\boldsymbol{d}||_\infty) = \iota(||\boldsymbol{d}||_\infty)/\alpha$.

The proof follows the same steps as in Theorem 4, by replacing h and h_V with h_d and h_{Vd}. It implies that exponential ISS tracking of a safe velocity translates to ISSf for the full system.

2.2.1 Velocity Controller Example

As the simplest choice for a velocity tracking controller that provides ISS by (27), it is possible to take a model-free controller:

$$\boldsymbol{u} = -\boldsymbol{K}_D \dot{\boldsymbol{e}}, \tag{28}$$

where $\boldsymbol{K}_D \in \mathbb{R}^{m \times n}$ is selected so that $\boldsymbol{K} = \boldsymbol{B}\boldsymbol{K}_D$ is positive definite. This controller is characterized by the constant $\lambda \in \mathbb{R}_{>0}$:

$$\lambda = \frac{\sigma_{\min}(\boldsymbol{K})}{\sup_{\boldsymbol{q} \in Q} \sigma_{\max}(\boldsymbol{M}(\boldsymbol{q}))}, \tag{29}$$

where $\sigma_{\min}$ and $\sigma_{\max}$ denote the smallest and largest eigenvalues. The eigenvalues are positive real numbers since $\boldsymbol{M}(\boldsymbol{q})$ and $\boldsymbol{K}$ are positive definite. Accordingly, λ characterizes how fast a controller may track the desired velocity since it representes the smallest gain divided by the largest inertia. The proposed controller can be associated with the Lyapunov function candidate:

$$V(\boldsymbol{q}, \dot{\boldsymbol{e}}) = \sqrt{\frac{1}{2} \dot{\boldsymbol{e}}^T \boldsymbol{M}(\boldsymbol{q}) \dot{\boldsymbol{e}}}, \tag{30}$$

that has the bound (21) with

$$k_1 = \inf_{\boldsymbol{q} \in Q} \sqrt{\sigma_{\min}(\boldsymbol{M}(\boldsymbol{q}))/2}, \quad k_2 = \sup_{\boldsymbol{q} \in Q} \sqrt{\sigma_{\max}(\boldsymbol{M}(\boldsymbol{q}))/2}. \tag{31}$$

The linear class-$\mathcal{K}$ function ι can also be defined as $\iota(||\boldsymbol{d}||_\infty) = ||\boldsymbol{d}||_\infty / 2k_1$. With the reported controller, the parameters to be selected during control design are α and $\boldsymbol{K}_D$. Now it can be shown that the controller satisfies the required stability properties.

Theorem 6. Consider the system (13), the Lyapunov function V (30), the constant λ given by (29) and $\dot{\boldsymbol{e}} \neq 0$. Then, the controller (28) satisfies the ISS condition (27) with respect to $\boldsymbol{d} = -\boldsymbol{M}(\boldsymbol{q})\ddot{\boldsymbol{q}}_s - \boldsymbol{C}(\boldsymbol{q}, \dot{\boldsymbol{q}})\dot{\boldsymbol{q}}_s - \boldsymbol{G}(\boldsymbol{q})$.

Proof. To prove this theorem, firstly, V from (30) can be differentiated leading to

$$\dot{V}(\boldsymbol{q}, \dot{\boldsymbol{e}}, \dot{\boldsymbol{q}}, \ddot{\boldsymbol{q}}_s, \boldsymbol{u}, \boldsymbol{d}) = \frac{1}{2V(\boldsymbol{q}, \dot{\boldsymbol{e}})} \left(\frac{1}{2} \dot{\boldsymbol{e}}^T \dot{\boldsymbol{M}}(\boldsymbol{q}, \dot{\boldsymbol{q}}) \dot{\boldsymbol{e}} + \dot{\boldsymbol{e}}^T \boldsymbol{M}(\boldsymbol{q}) \ddot{\boldsymbol{e}} \right). \tag{32}$$

Looking at the dynamic model (13) and by definition of the error $\dot{\boldsymbol{e}}$ we obtain

$$\boldsymbol{M}(\boldsymbol{q})\ddot{\boldsymbol{e}} = -\boldsymbol{C}(\boldsymbol{q}, \dot{\boldsymbol{q}})\dot{\boldsymbol{e}} - \boldsymbol{C}(\boldsymbol{q}, \dot{\boldsymbol{q}})\dot{\boldsymbol{q}}_s - \boldsymbol{G}(\boldsymbol{q}) + \boldsymbol{B}\boldsymbol{u} - \boldsymbol{M}(\boldsymbol{q})\ddot{\boldsymbol{q}}_s. \tag{33}$$

For the considered controller (28), substituting (33) into (32) leads to

$$\dot{V}(\boldsymbol{q}, \dot{\boldsymbol{e}}, \dot{\boldsymbol{q}}, \ddot{\boldsymbol{q}}_s, \boldsymbol{u}, \boldsymbol{d}) = \frac{-\dot{\boldsymbol{e}}^T \boldsymbol{K} \dot{\boldsymbol{e}} + \dot{\boldsymbol{e}}^T \boldsymbol{d}}{2V(\boldsymbol{q}, \dot{\boldsymbol{e}})} = \frac{-\dot{\boldsymbol{e}}^T \boldsymbol{K} \dot{\boldsymbol{e}}}{2V(\boldsymbol{q}, \dot{\boldsymbol{e}})} + \frac{\dot{\boldsymbol{e}}^T \boldsymbol{d}}{2V(\boldsymbol{q}, \dot{\boldsymbol{e}})}, \tag{34}$$

where the term $\dot{\boldsymbol{e}}^T(\dot{\boldsymbol{M}}(\boldsymbol{q},\dot{\boldsymbol{q}}) - 2\boldsymbol{C}(\boldsymbol{q},\dot{\boldsymbol{q}}))\dot{\boldsymbol{e}}$ is dropped because $\dot{\boldsymbol{M}}(\boldsymbol{q},\dot{\boldsymbol{q}}) - 2\boldsymbol{C}(\boldsymbol{q},\dot{\boldsymbol{q}})$ is skew-symmetric. Since (29) implies that $\dot{\boldsymbol{e}}^T\boldsymbol{K}\dot{\boldsymbol{e}} - \lambda\dot{\boldsymbol{e}}^T\boldsymbol{M}(\boldsymbol{q})\dot{\boldsymbol{e}} \geq 0$ definition (30) of V leads to

$$\frac{-\dot{\boldsymbol{e}}^T\boldsymbol{K}\dot{\boldsymbol{e}}}{2V(\boldsymbol{q},\dot{\boldsymbol{e}})} \leq -\lambda V(\boldsymbol{q},\dot{\boldsymbol{e}}), \tag{35}$$

$\forall \boldsymbol{q} \in Q, \dot{\boldsymbol{e}} \in \mathbb{R}^n$. Furthermore, for the Cauchy-Schwartz inequality, the bound (21) on V and the definition of ι yield

$$\frac{\dot{\boldsymbol{e}}^T\boldsymbol{d}}{2V(\boldsymbol{q},\dot{\boldsymbol{e}})} \leq \frac{\|\dot{\boldsymbol{e}}\|\|\boldsymbol{d}\|_\infty}{2k_1\|\dot{\boldsymbol{e}}\|} = \iota(\|\boldsymbol{d}\|_\infty), \tag{36}$$

where $||\dot{\boldsymbol{e}}||$ drops, making the right-hand side independent of time. Substituting (35,36) into (34) yields (27). □

3 Simulations

We utilized MATLAB simulations with Simulink to validate the efficiency of CBFs in various robotic systems and to compare the performance of model-free techniques with classical ones. We made an effort to maintain as many simulation parameters as possible to be consistent throughout the process. The structure of all the control blocks for the model-free method follows the basic scheme reported in Figure 1. Here, we have always implemented a model-independent controller to track the safe velocity generated utilizing knowledge from a reduced-order model, as opposed to the real one representing the robotic system. The classical method does not utilize a safe velocity tracker, as the input is directly provided to the robot. It does not make use of an approximation of the dynamics and takes into account both the pose and

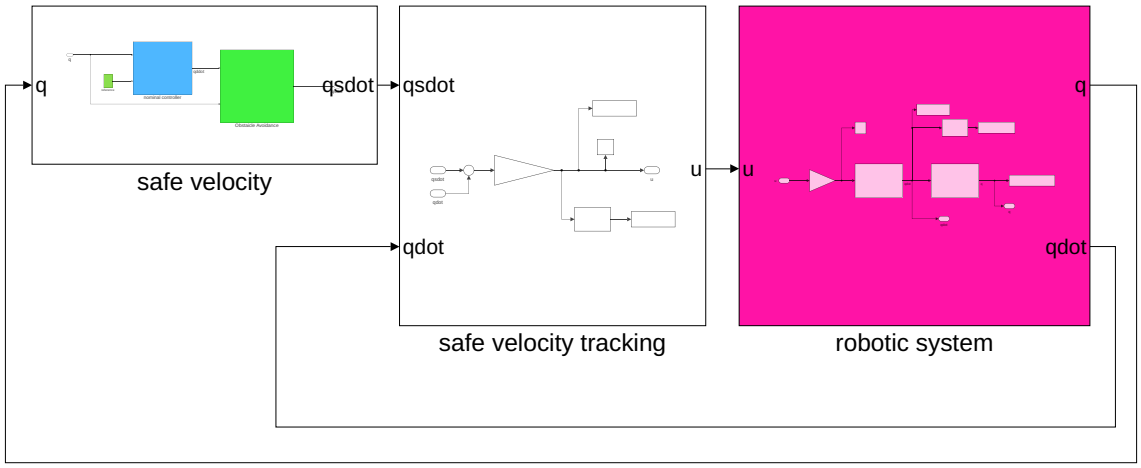


Figure 1: Control block structure of our model-free approach implementation. The robotic system (13) outputs the pose and velocities, only the former takes a role in the safe velocity generation while the latter are used to track it.

velocity in generating the safe input. Such input has been produced by solving (9) with the Barrier Certificate Enforcement block of the Optimization Toolbox. Appendix A contains additional details on the equations involved in the problems implemented.

3.1 Point robot

For the first simulations, as the simplest instance of (13), let us consider a point robot, which dynamics can be modeled by

$$\boldsymbol{M}\ddot{\boldsymbol{q}} = \boldsymbol{B}\boldsymbol{u}, \tag{37}$$

where $\boldsymbol{q} \in \mathbb{R}^2$ represents the planar position of the robot, $\boldsymbol{u} \in \mathbb{R}^2$, $\boldsymbol{B} = \boldsymbol{I}_2$ and $\boldsymbol{M} = \text{diag}\{m, m\}$ in this scenario. We want to solve the problem of guiding the system from an initial configuration $\boldsymbol{q}_{start}$ to a final one $\boldsymbol{q}_{goal}$ while avoiding circular obstacles thanks to the CBF, therefore remaining in the safe region.

3.1.1 Model-free CBF

A solution that allows navigating from $\boldsymbol{q}_{start}$ to $\boldsymbol{q}_{goal}$ without taking into account the obstacles consists in realizing the nominal velocity

$$\dot{\boldsymbol{q}}_{nom} = -K_P(\boldsymbol{q} - \boldsymbol{q}_{goal}),\ K_P \in \mathbb{R}_{>0}. \tag{38}$$

We can modify it, in a minimally invasive fashion attaining $\dot{\boldsymbol{q}}_{safe}$ by solving the optimization problem (9) using as CBF

$$h_i(\boldsymbol{q}) = d_i - R_{\mathcal{O}_i}, \tag{39}$$

where $R_{\mathcal{O}_i}$ is the radius of the obstacle centered at $\boldsymbol{q}_{\mathcal{O}_i}$ and $d_i = \|\boldsymbol{q} - \boldsymbol{q}_{\mathcal{O}_i}\|$. When $h_i(\boldsymbol{q})$ is positive it means that there is no collision with $\mathcal{O}_i$. In Simulation 1 to track $\dot{\boldsymbol{q}}_{safe}$ we used controller (28). Here, the robot starts at rest from $\boldsymbol{q}_{start} = (0, 0)^T$ and the goal is $\boldsymbol{q}_{goal} = (4, -1)^T$, while the obstacles positions are $\boldsymbol{q}_{\mathcal{O}_1} = (1.5, 0)^T$, $\boldsymbol{q}_{\mathcal{O}_2} = (3, -1.5)^T$ with common radius $R = 0.75$. Setting the gains $K_P = 0.2$, $\boldsymbol{K}_D = \boldsymbol{I}_{2x2}$, $m = 1$ [Kg] and considering at each time the closest object for the CBF (39), we have been able to reproduce exactly the simulation proposed in [2] varying α, as shown in Figure 2. Note that, even though $\alpha = 0.5 < \lambda = 1$, the CBF becomes negative because the robot is passing through the obstacle $\mathcal{O}_2$. This is an interesting result since it highlights how, when the exponential tracking of the safe velocity is not guaranteed Figure 3, Theorem 4 does not hold, meaning that safety is lost. For this reason, in Simulation 2 we decided to guarantee ISSf with respect to the full model disturbance $\boldsymbol{d} = -\boldsymbol{M}\ddot{\boldsymbol{q}}_{safe}$. We have estimated $\ddot{\boldsymbol{q}}_{safe}$ through Simulink with a backward derivative block after $\dot{\boldsymbol{q}}_{safe}$ used to solve the same problem. Setting $\alpha = 0.99$ and without changing the other Simulation 1 parameters, resulted in $||\boldsymbol{d}||_\infty = 0.5$, obtaining $\gamma = 0.3571$. In order to limit the values for γ the best approach is to set α as large as possible. Intuitively, this method can be

understood as ensuring safety with respect to a reduced set, which accurately reflects the true size of the safe set by incorporating the uncertainty from the disturbance $\boldsymbol{d}$. This is implemented as an additional clearance equal to γ that prevents the robot from hitting the obstacles even in the case of tracking errors. Therefore even if the CBF is lower than the γ bound in certain instants, meaning that the considered CBF is negative, safety is still maintained since collisions with the obstacles are avoided. This additional clearance has resulted in better performances as shown in Figures 4–7.

3.1.2 Model-based CBF

In Simulation 3, in order to approach the same problem directly through the full dynamic model, it is not possible to use CBF (39) since it misses the *relative-degree* one condition with respect to (37). Therefore, we augmented it by adding a velocity term that lets the new h decrease if the robot is moving towards the obstacle and increase on the contrary, as proposed in [3]

$$h_i(\boldsymbol{q}, \dot{\boldsymbol{q}}) = d_i - R_{\mathcal{O}_i} + \mu(\boldsymbol{q} - \boldsymbol{q}_{\mathcal{O}_i})^T \dot{\boldsymbol{q}}, \tag{40}$$

with μ a positive constant, set to 0.8 in this simulation. We used this in the optimization problem to correct the nominal input generated with:

$$\boldsymbol{u}_{nom} = -K_P(\boldsymbol{q} - \boldsymbol{q}_{goal}) - K_D\dot{\boldsymbol{q}}, \tag{41}$$

using $K_P = 0.2, K_D = 1$, the same gains of Simulation 1 and $\alpha = 1$. As depicted in Figure 8, it is clear that access to the dynamic system enables the robot to fully utilize the safe region without any concerns regarding uncertainties in the model.

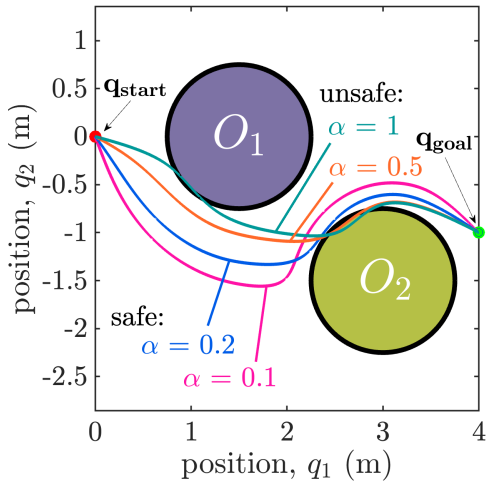


Figure 2: Simulation 1: evolution of the state $\boldsymbol{q}$ over the environment for the different values of parameter α.

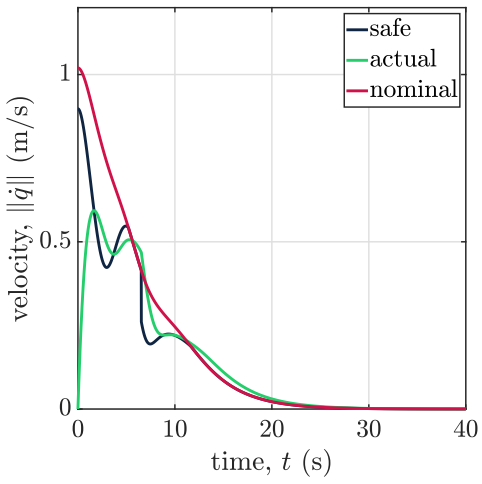


Figure 3: Simulation 1: evolution of the velocity norms $|\dot{\boldsymbol{q}}|$ for $\alpha = 0.5$.

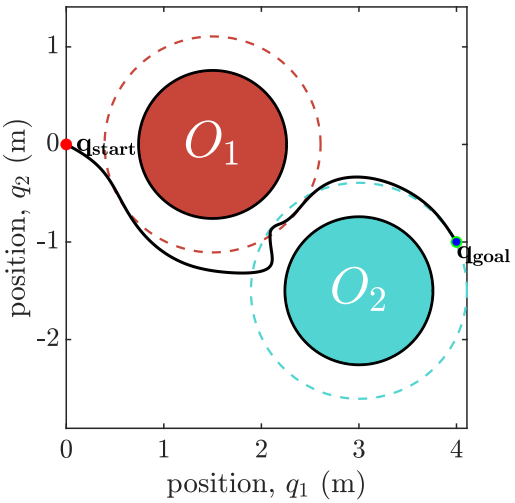


Figure 4: Simulation 2: evolution of the state $\boldsymbol{q}$ over the environment.

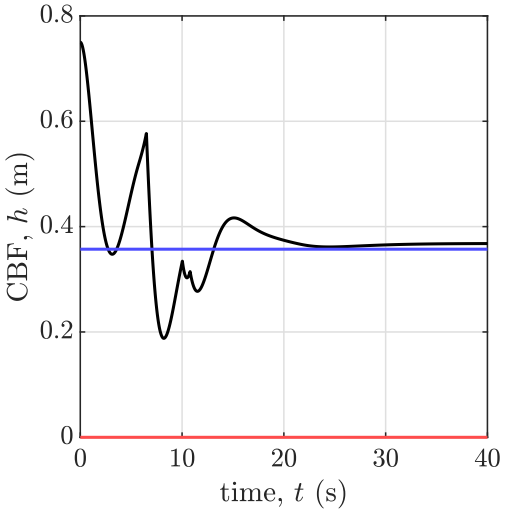
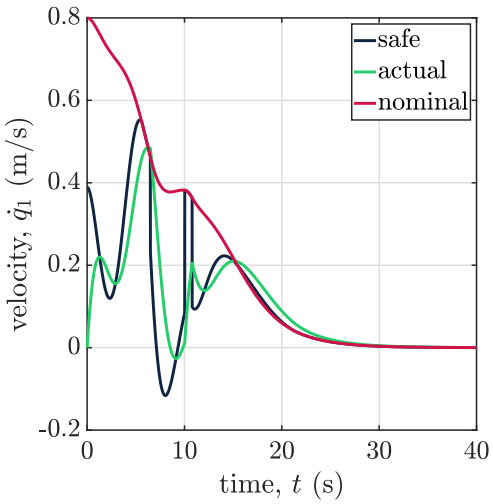


Figure 5: Simulation 2: evolution of h with $\gamma = 0.3571$ [m] (blue line).



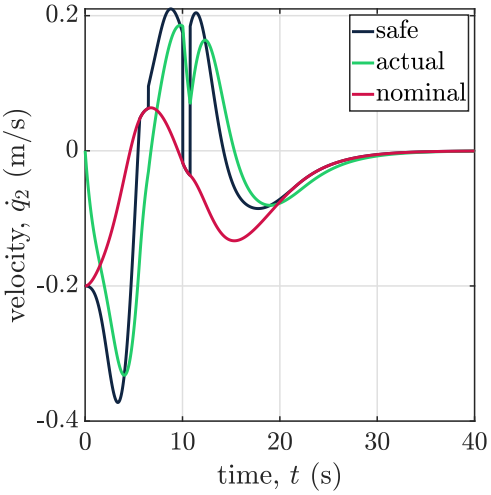
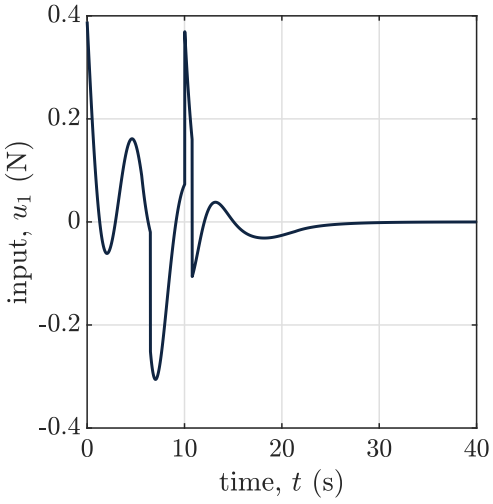


Figure 6: Simulation 2: Evolution of the velocity vector $\dot{\boldsymbol{q}}$ components.



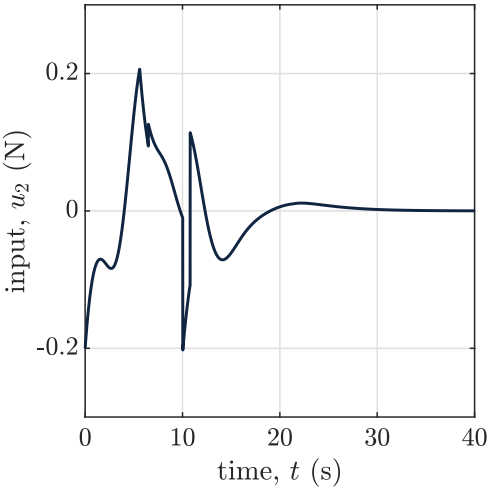


Figure 7: Simulation 2: Evolution of the input vector $\boldsymbol{u}$ components.

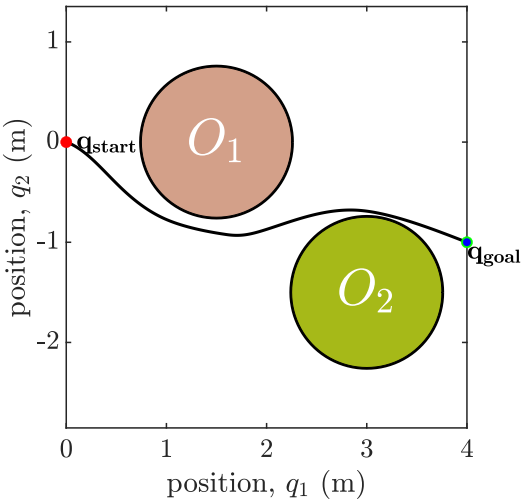


Figure 8: Simulation 3: evolution of the state $\boldsymbol{q}$ over the environment.

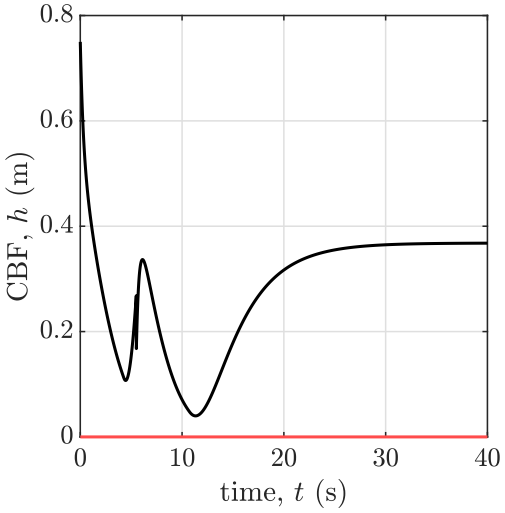


Figure 9: Simulation 3: evolution of h.

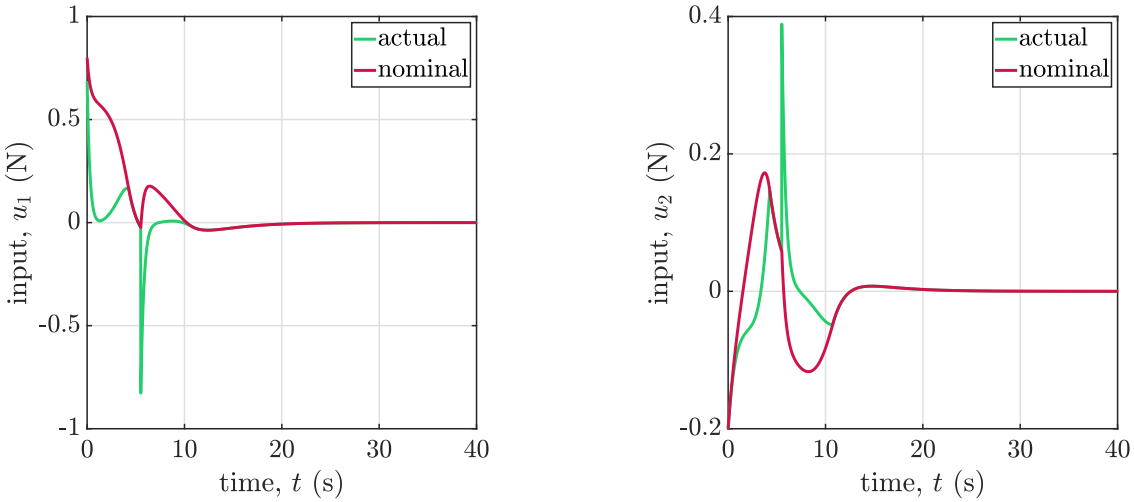


Figure 10: Simulation 3: evolution of the input vector $\boldsymbol{u}$ components.

3.2 Unicycle

The other robotic system implemented is that of a circular body unicycle with x and y coordinates of a robot point along its sagittal axis at distance $|a|$ from the wheel contact point. The dynamic model is given by

$$\begin{cases} \dot{x} = v\cos\theta - \omega a \sin\theta \\ \dot{y} = v\sin\theta + \omega a \cos\theta \\ \dot{\theta} = \omega \\ \dot{v} = 1/m\ u_1 \\ \dot{\omega} = 1/I_{cm}\ u_2 \end{cases} \tag{42}$$

The choice of controlling point a instead of the position of the wheel is made to ensure a fair comparison between the two approaches, in fact even though (42) can be feedback linearized when $a = 0$, the kinematic model cannot without a change of coordinates or other strategies. This difference would result in an unequal comparison between the two approaches for the Cartesian regulation task being analyzed. Regarding the CBF, for these simulations we aim to adopt a more sophisticated approach that can handle multiple obstacles more effectively. The current method of dealing with one obstacle at a time, during the CBF constraint evaluation may lead to discontinuities in h, as shown in Figure 9. Additionally, the unicycle may not be able to avoid an obstacle if it is considered too late due to the kinematic constraint. One thing to notice is that in the following plots we represent the unicycle as a point while the radius of each obstacle is augmented with the radius of the robot r plus the displacement $|a|$, so that if the trajectory of the point does not intersect any obstacle we can ensure no collisions. The unicycle parameters are $m = 5.3$ [kg], $r = 0.3$ [m], $I_{cm} = mr^2/2 = 0.24$ [kg m^2], $a = 0.1$ [m]. The unicycle starts at rest with pose $(\boldsymbol{p}_{start}^T, \theta_{start})^T = (x_{start}, y_{start}, \theta_{start})^T = (0, 0, \pi/2)^T$ and we want to reach $\boldsymbol{p}_{goal} = (8, -3)^T$. In this case, the considered obstacles are $\boldsymbol{p}_{\mathcal{O}_1} = (3.5, 0.5)^T$, $\boldsymbol{p}_{\mathcal{O}_2} = (3, -1.5)^T$, $\boldsymbol{p}_{\mathcal{O}_3} = (6, -1)^T$ with radius $R_{\mathcal{O}_1} = 0.45$, $R_{\mathcal{O}_2} = 0.45$, $R_{\mathcal{O}_3} = 0.75$ [m].

3.2.1 Model-free CBF

Let us consider the first three components $\dot{\boldsymbol{q}}$ of (42) i.e. the unicycle kinematic model. The velocity to reach the goal without taking into account any obstacles is

$$\boldsymbol{v}_{nom} = \begin{pmatrix} v_{nom} \\ \omega_{nom} \end{pmatrix} = - \begin{bmatrix} \cos\theta & \sin\theta \\ -\sin\theta/a & \cos\theta/a \end{bmatrix} K_P \cdot \boldsymbol{I}_2(\boldsymbol{p} - \boldsymbol{p}_{goal}), K_P \in \mathbb{R}_{>0}, \tag{43}$$

where $\boldsymbol{p}$ and $\boldsymbol{p}_{goal}$ are the cartesian coordinates of the robot and the goal respectively. The CBF used to avoid obstacles is similar to (39) but with the inclusion of a term that considers if the system is moving towards the obstacle $\mathcal{O}_i$

$$h_i(\boldsymbol{q}) = d_i - R_{\mathcal{O}_i} - \delta\cos(\theta - \psi_i), \tag{44}$$

where $\psi_i = \arctan((y_{\mathcal{O}_i} - y)/(x_{\mathcal{O}_i} - x))$ is the robot-$\mathcal{O}_i$ angle and $\delta \in \mathbb{R}_{>0}$ is a tunable parameter. This CBF is highly effective for the unicycle as it encourages the robot to steer away from obstacles. In Simulation 4, we used multiple CBF constraints such that each component i of $h(\boldsymbol{q}) = (h_1(\boldsymbol{q}),\ ...,\ h_l(\boldsymbol{q}))^T$ is defined as (44) and $l = 3$ is the number of obstacles. Here, controller

$$\boldsymbol{u} = -\boldsymbol{K}_D\dot{\boldsymbol{e}} = -\boldsymbol{K}_D(\boldsymbol{v} - \boldsymbol{v}_{safe}), \tag{45}$$

tracks the safe velocity $\boldsymbol{v}_{safe}$ from (9) with $\boldsymbol{K}_D = 2.7 \cdot \boldsymbol{I}_2$, then we set $K_P = 0.1$, $\alpha = 0.50 < \lambda = 0.51$, $\delta = 2$ and bound $|\dot{\boldsymbol{v}}_{safe}| \leq 0.5$. As shown in Figures 11–12 the

unicycle is capable of reaching its destination, but it has to circumvent the obstacles as the clearance, in this case, is $\gamma = 1.45$ [m]. In Figure 13 we can see how the nominal angular velocity generated tries to align the robot with the goal, but due to the safety mechanism that prevents pointing in the direction of the obstacle, a much lower command is produced making the robot take a longer route. Anyway, it is worth noticing how, increasing $\boldsymbol{K}_D$ eigenvalues to reduce the clearance, almost gives perfect tracking of the angular velocity.

3.2.2 Model-based CBF

In the classic solution, similar to the point robot, we cannot use directly (46) for each obstacle $\mathcal{O}_i$, but we need to add a term to have the relative degree equal to one. Using the same term added in (40) and considering as $\dot{\boldsymbol{q}}$ the full system (42) the new CBF is

$$h_i(\boldsymbol{q}) = d_i - R_{\mathcal{O}_i} - \delta\cos(\theta - \psi_i) + \mu(\boldsymbol{p} - \boldsymbol{p}_{\mathcal{O}_i})^T\dot{\boldsymbol{p}}, \tag{46}$$

with $\boldsymbol{p}$ the cartesian position vector. In Simulation 5, we have generated $\boldsymbol{u}_{nom}$ analogously to (43) so that the system is linearized and a simple PD controller can be employed

$$\boldsymbol{u}_{nom} = \begin{pmatrix} \cos\theta/m & -a\sin\theta/I_{cm} \\ \sin\theta/m & a\cos\theta/I_{cm} \end{pmatrix}^{-1} \left[-\begin{pmatrix} -v\omega\sin\theta - \omega^2 a\cos\theta \\ v\omega\cos\theta - \omega^2 a\sin\theta \end{pmatrix} + \ddot{\boldsymbol{p}}_{nom} \right], \tag{47}$$

with

$$\ddot{\boldsymbol{p}}_{nom} = -\boldsymbol{K}_P(\boldsymbol{p} - \boldsymbol{p}_{goal}) - \boldsymbol{K}_D\,\dot{\boldsymbol{p}}. \tag{48}$$

By solving the multi-CBF optimization problem for the same obstacles of Simulation 4 we produce the $\boldsymbol{u}_{safe}$, which is given directly in input to the model. Setting $\boldsymbol{K}_P = 0.2 \cdot \boldsymbol{I}_2$, $\boldsymbol{K}_D = \boldsymbol{I}_2$, $\delta = 0.2$, $\alpha = 0.8$, $\mu = 0.3$, the robot is now able to pass in between the obstacles to reach the goal since it does not need clearances to ensure safety, Figure 15. Nevertheless, in practice, having an additional clearance for classic CBFs could be beneficial as robot parameters may not be perfectly known, introducing disturbances that can be modeled similarly to the ones discussed in the model-free section. With the aim of giving a practical example, Simulation 6 repeats the previous one using a slightly different mass m_2, and so moment of inertia I_{cm2}, to compute nominal and safe inputs, which could represent an unknown payload whose weight is not precisely estimated. While the robot can still complete this task with a different mass, see Figure 18, navigating in a more complex environment like in Simulation 7 (only obstacles change) can cause problems when the dynamic model is inaccurate, Figures 19–21. Indeed, using $m_2 = 3$ [kg], meaning that the model underestimates the mass of the robot, at around 7 seconds into the simulation the unicycle turns to avoid obstacle $\mathcal{O}_3$, but, right after, when it needs to dodge $\mathcal{O}_2$ the generated safe input is not strong enough Figures 22–23 and the robot collides with the obstacle.

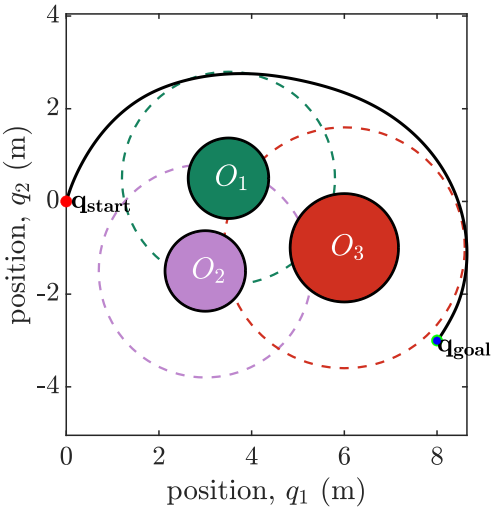


Figure 11: Simulation 4: evolution of the state $\boldsymbol{q}$ over the environment.

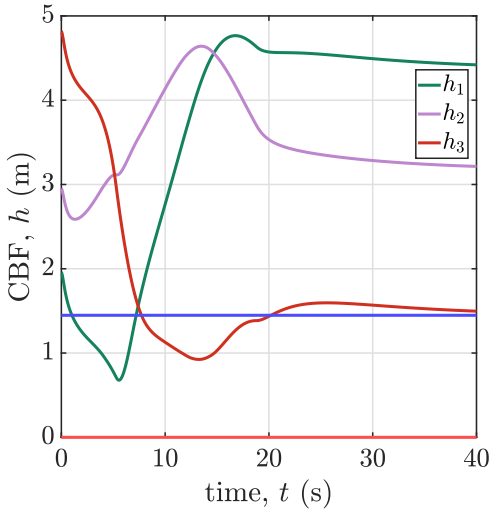
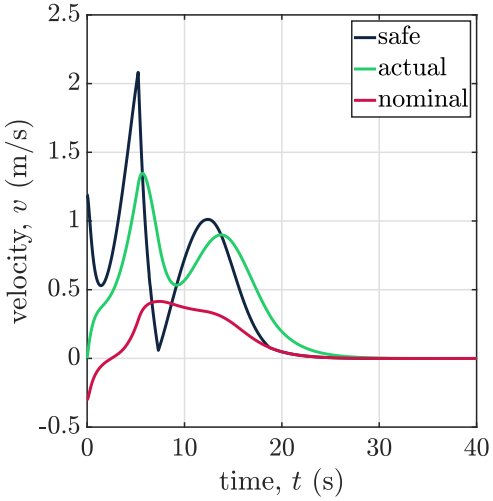


Figure 12: Simulation 4: evolution of h with $\gamma = 1.45$ [m] (blue line).



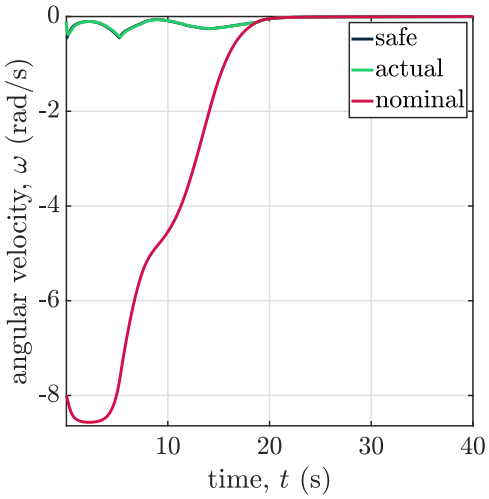
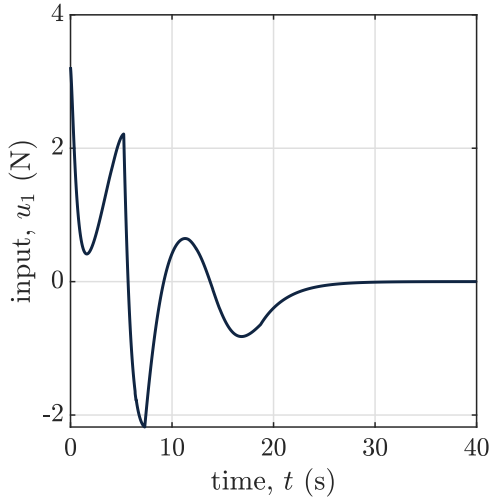


Figure 13: Simulation 4: Evolution of the velocity vector $\dot{\boldsymbol{q}}$ components.



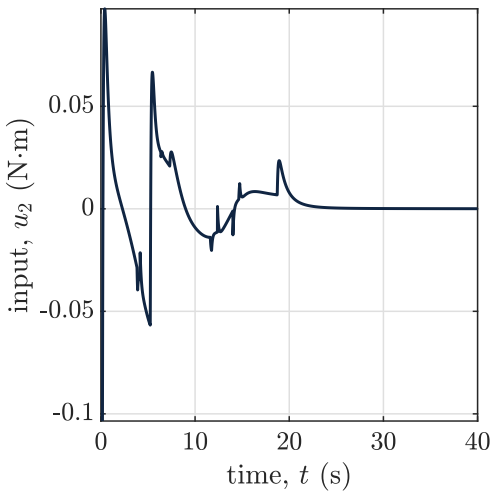


Figure 14: Simulation 4: Evolution of the input vector $\boldsymbol{u}$ components.

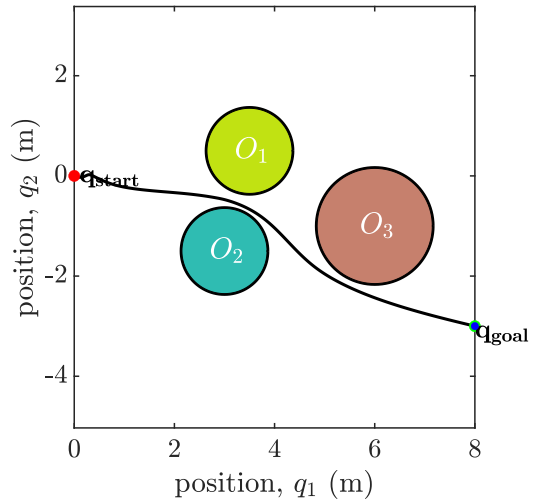


Figure 15: Simulation 5: evolution of the state $\boldsymbol{q}$ over the environment.

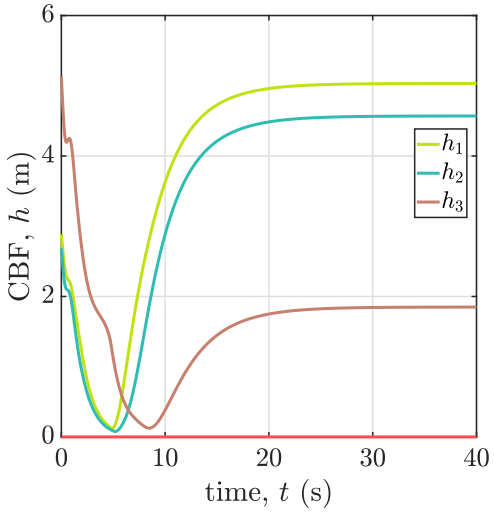
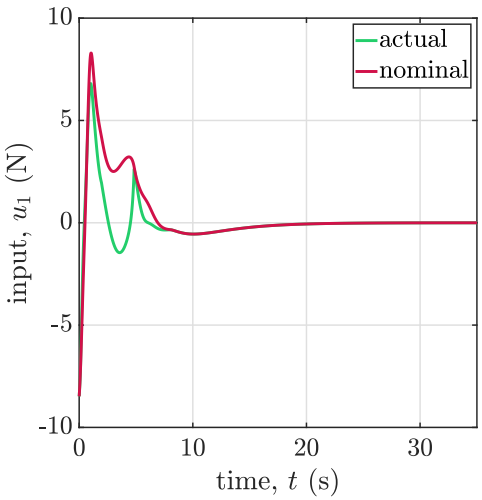


Figure 16: Simulation 5: evolution of h.



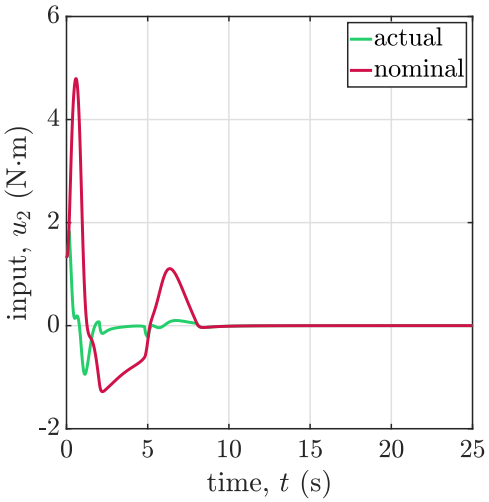


Figure 17: Simulation 5: evolution of the input vector $\boldsymbol{u}$ components.

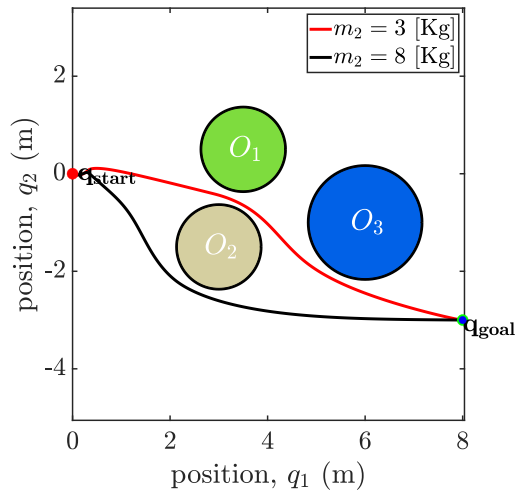


Figure 18: Simulation 6: trajectories for the two values of m_2.

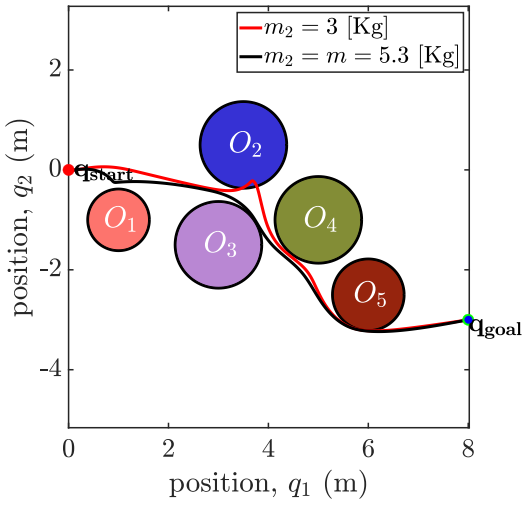


Figure 19: Simulation 7: trajectories for the two values of m_2.

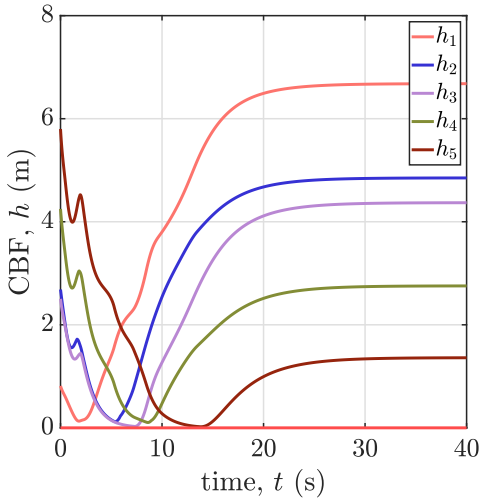


Figure 20: Simulation 7: evolution of h when the mass is estimated correctly, namely $m_2 = m = 5.3$ [Kg].

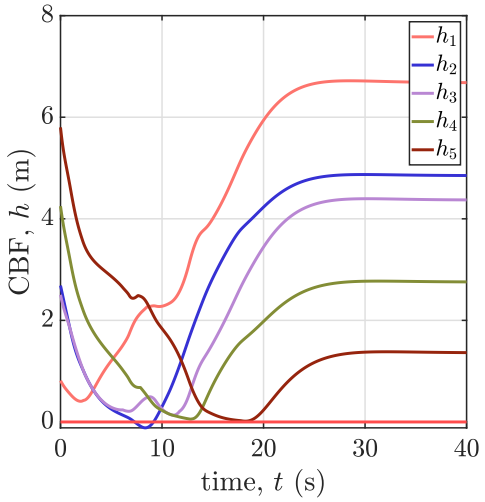
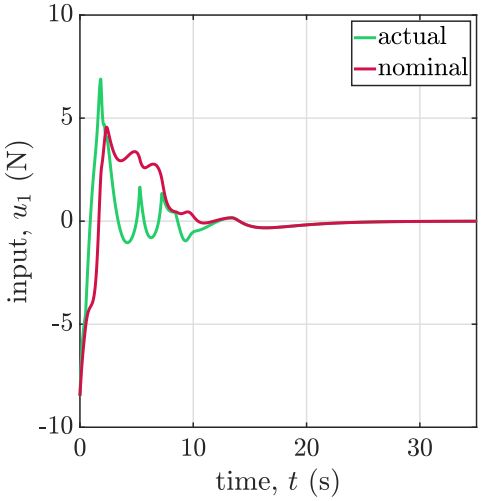


Figure 21: Simulation 7: evolution of h when $m_2 = 3$ [kg]. When the robot collides with $\mathcal{O}_2$, h_2 becomes negative.



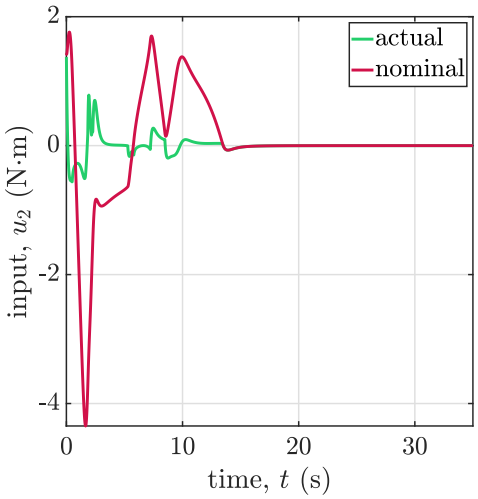
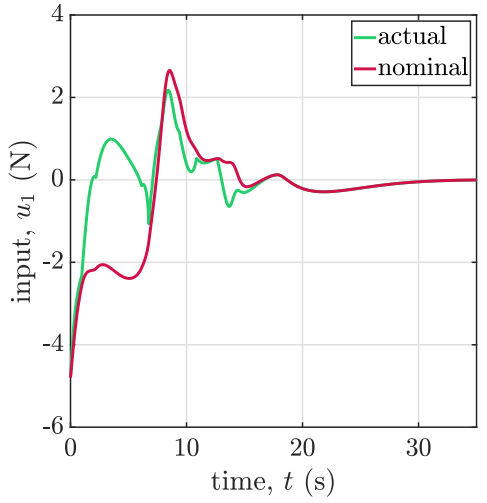


Figure 22: Simulation 7: Evolution of the input vector $\boldsymbol{u}$ components for $m_2 = m$.



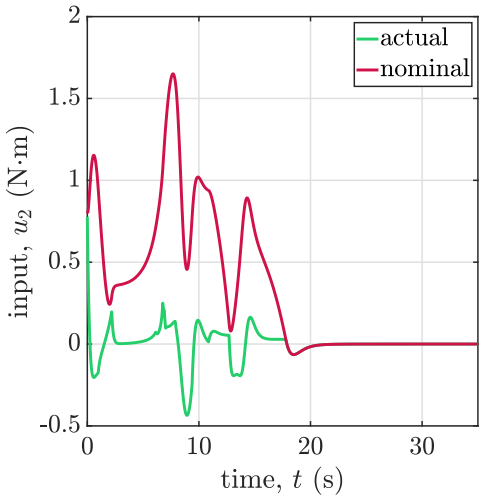


Figure 23: Simulation 7: Evolution of the input vector $\boldsymbol{u}$ components for $m_2 = 3$ [Kg].

Conclusions

This project has enabled us to examine and evaluate one of the cutting-edge control techniques that has garnered significant attention in recent years: Control Barrier Functions. With the aim of making them easier to apply, the model-free approach suggests the use of purely kinematic CBFs so to implement a much simpler robotic model and a less complex optimization problem. In particular, we have used the kinematic equations instead of the dynamic ones but, for example, the authors of [2] exploit the unicycle kinematic model to plan the motion of a spatial Segway and a quadruped. This approach has the limitation of needing exponential tracking of the generated velocities that can only be achieved with a model-based velocity tracker, for which the dynamics of the robot need to be accurately known. If one uses a simpler tracking algorithm, ensuring safety comes at the price of restricting the safe set with the use of a clearance $\gamma(||\boldsymbol{d}||_\infty)$. It would be desirable to reduce γ as much as possible by changing the control parameters. For the double integrator and the unicycle $\gamma = ||\boldsymbol{M}\ \dot{\boldsymbol{v}}_s||_\infty/(2\alpha\sqrt{\sigma_{\min}(\boldsymbol{M})/2})$. A way to decrease the clearance is to set α close to its maximum admissible value, given by $\alpha < \lambda$. High values of α encourages the system to get close to the boundaries of the safe region, but this is not an issue since these have been changed by the addition of the clearance. To reduce the clearance further one could diminish the maximum admissible safe acceleration $\dot{\boldsymbol{v}}_s$ by scaling down the control gains, but that reduces λ and subsequently α, meaning that it is not straightforward to extend the bounds of the safe region this way. In conclusion, our experience with model-free CBF points towards the limited usefulness of this approach. In fact, if the dynamic model is not known, exponential tracking of the velocities is not possible, and to ensure safety one needs to introduce clearances that are often quite large, and moreover should be estimated using the dynamic parameters. When the dynamic model is instead known, one could achieve exponential tracking of the velocities, but then the use of model-based CBFs is also possible. In this case, the main advantage of model-free CBFs is that the optimization problem is less complex, especially when more than one obstacle is considered at a time and there is no need to include higher-order terms in h to keep the relative degree one. Moreover, classic CBFs rely heavily on the model, and when the system parameters are imperfectly known adding a clearance may be necessary to ensure safety.

References

[1] A. D. Ames, S. Coogan, M. Egerstedt, G. Notomista, K. Sreenath, and P. Tabuada. Control barrier functions: Theory and applications. *CoRR*, abs/1903.11199, 2019.

[2] T. G. Molnár, R. K. Cosner, A. W. Singletary, W. Ubellacker, and A. D. Ames. Model-free safety-critical control for robotic systems. *CoRR*, abs/2109.09047, 2021.

[3] A. Cristofaro, M. Ferro, and M. Vendittelli. Safe trajectory tracking using closed-form controllers based on control barrier functions. In *2022 IEEE 61st Conference on Decision and Control (CDC)*, pages 3329–3334, 2022.

[4] G. Oriolo, B. Siciliano, L. Sciavicco, and L. Villani. *Robotics: Modelling, Planning and Control*. Springer Publishing Company, Incorporated, 1st edition, 2008.

A Optimization problems

For completeness, we report here the optimization problem instantiations of (9) used to calculate the safe input for our simulations. Firstly, let us denote with $\boldsymbol{q}_{\mathcal{O}} = (x_{\mathcal{O}}, y_{\mathcal{O}})^T$ the Cartesian position of the obstacle $\mathcal{O}$ and with $\boldsymbol{p} = (x, y)^T$ the Cartesian position of the robot. The nominal values are the input generated without taking into account the obstacles.

Model Free CBF for Double Integrator

The kinematic model considered in this case is

$$\begin{cases} \dot{x} = v_1 \\ \dot{y} = v_2 \end{cases}$$

with control barrier function

$$h(\boldsymbol{p}) = d - R_{\mathcal{O}} = \sqrt{(x - x_{\mathcal{O}})^2 + (y - y_{\mathcal{O}})^2} - R_{\mathcal{O}},$$

for the obstacle $\mathcal{O}$. The safe velocity is obtained from the optimization problem

$$\boldsymbol{v}_{safe} = \arg\min_{\boldsymbol{v}} (\boldsymbol{v} - \boldsymbol{v}_{nom})^T (\boldsymbol{v} - \boldsymbol{v}_{nom}),$$

$$\text{s.t. } \dot{h}(\boldsymbol{p}, \boldsymbol{v}) \geq -\alpha h(\boldsymbol{p}),$$

with

$$\dot{h}(\boldsymbol{p}, \boldsymbol{v}) = \frac{dh}{d\boldsymbol{p}} \dot{\boldsymbol{p}} = \frac{1}{d} \left[(x - x_{\mathcal{O}}) v_1 + (y - y_{\mathcal{O}}) v_2 \right].$$

Model Based CBF for Double Integrator

The model considered in this case is

$$\begin{cases} \ddot{x} = \dfrac{u_1}{m} \\ \ddot{y} = \dfrac{u_2}{m} \end{cases}$$

with control barrier function

$$h(\boldsymbol{q}, \dot{\boldsymbol{q}}) = d - R_{\mathcal{O}} + \mu (\boldsymbol{q} - \boldsymbol{q}_{\mathcal{O}})^T \dot{\boldsymbol{q}},$$

for the obstacle $\mathcal{O}$. The safe input is obtained from the optimization problem

$$\boldsymbol{u}_{safe} = \arg\min_{\boldsymbol{u}} (\boldsymbol{u} - \boldsymbol{u}_{nom})^T (\boldsymbol{u} - \boldsymbol{u}_{nom}),$$

$$\text{s.t. } \dot{h}(\boldsymbol{p}, \dot{\boldsymbol{p}}, \boldsymbol{u}) \geq -\alpha h(\boldsymbol{p}, \dot{\boldsymbol{p}}),$$

with

$$\dot{h}(\boldsymbol{p}, \dot{\boldsymbol{p}}, \boldsymbol{u}) = \left[\frac{1}{d}(x - x_{\mathcal{O}}) + \mu \dot{x}\right] \dot{x} + \left[\frac{1}{d}(y - y_{\mathcal{O}}) + \mu \dot{y}\right] \dot{y} + \frac{\mu}{m} \left[(x - x_{\mathcal{O}})u_1 + (y - y_{\mathcal{O}})u_2\right].$$

Model Free CBF for Unicycle

The kinematic model considered for the unicycle is

$$\begin{cases} \dot{x} = v \cos\theta - \omega a \sin\theta \\ \dot{y} = v \sin\theta + \omega a \cos\theta \\ \dot{\theta} = \omega \end{cases}$$

with control barrier function

$$h(\boldsymbol{q}) = d - R_{\mathcal{O}} - \delta \cos(\theta - \psi),$$

for the obstacle $\mathcal{O}$ where $\boldsymbol{q} = (x, y, \theta)^T$. Considering $\boldsymbol{v} = (v, \omega)^T$, the safe velocity is obtained from the optimization problem

$$\boldsymbol{v}_{safe} = \arg\min_{\boldsymbol{v}} (\boldsymbol{v} - \boldsymbol{v}_{nom})^T (\boldsymbol{v} - \boldsymbol{v}_{nom}),$$

$$\text{s.t. } \dot{h}(\boldsymbol{q}, \dot{\boldsymbol{q}}, \boldsymbol{v}) = \frac{\partial h}{\partial \boldsymbol{q}} \dot{\boldsymbol{q}} \geq -\alpha h(\boldsymbol{q}),$$

with

$$\frac{\partial h}{\partial \boldsymbol{q}} = \begin{bmatrix} \frac{\partial h}{\partial x} & \frac{\partial h}{\partial y} & \frac{\partial h}{\partial \theta} \end{bmatrix} = \begin{bmatrix} -\frac{\mathrm{x}_{\mathcal{O}} - \mathrm{x}}{d} - \frac{\delta\, \sigma_1\, (\mathrm{y}_{\mathcal{O}} - y)}{d^2} & -\frac{\mathrm{y}_{\mathcal{O}} - y}{d} + \frac{\delta\, \sigma_1\, (\mathrm{x}_{\mathcal{O}} - x)}{d^2} & \delta\, \sigma_1 \end{bmatrix},$$

where

$$\sigma_1 = \sin\left(\theta - \operatorname{atan2}\left(\mathrm{y}_{\mathcal{O}} - y, \mathrm{x}_{\mathcal{O}} - x\right)\right).$$

This leads to

$$\begin{aligned} \dot{h}(\boldsymbol{q}, \dot{\boldsymbol{q}}, \boldsymbol{v}) = & \left(-\frac{\mathrm{x}_{\mathcal{O}} - \mathrm{x}}{d} - \frac{\delta\, \sigma_1\, (\mathrm{y}_{\mathcal{O}} - y)}{d^2}\right) (v \cos\theta - \omega a \sin\theta) + \\ & + \left(-\frac{\mathrm{y}_{\mathcal{O}} - y}{d} + \frac{\delta\, \sigma_1\, (\mathrm{x}_{\mathcal{O}} - x)}{d^2}\right) (v \sin\theta + \omega a \sin\theta) + \delta\, \sigma_1 \omega. \end{aligned}$$

Model Based CBF for Unicycle

The dynamic model of the unicycle considered is :

$$\begin{cases} \dot{x} = v\cos\theta - \omega a \sin\theta \\ \dot{y} = v\sin\theta + \omega a \cos\theta \\ \dot{\theta} = \omega \\ \dot{v} = 1/m\ u_1 \\ \dot{\omega} = 1/I_{cm}\ u_2 \end{cases} \tag{49}$$

with control barrier function

$$h(\boldsymbol{q}) = d - R_{\mathcal{O}} - \delta\cos(\theta - \psi) + \mu(\boldsymbol{p} - \boldsymbol{p}_{\mathcal{O}})^T\dot{\boldsymbol{p}},$$

for the obstacle $\mathcal{O}$ where $\boldsymbol{q} = (x, y, \theta, v, \omega)$. The safe input is obtained from the optimization problem

$$\boldsymbol{u}_{safe} = \arg\min_{\boldsymbol{u}}(\boldsymbol{u} - \boldsymbol{u}_{nom})^T(\boldsymbol{u} - \boldsymbol{u}_{nom}),$$

$$\text{s.t. } \dot{h}(\boldsymbol{q}, \dot{\boldsymbol{q}}, \boldsymbol{u}) = \frac{\partial h}{\partial \boldsymbol{q}}\dot{\boldsymbol{q}} \geq -\alpha h(\boldsymbol{q}),$$

with

$$\frac{\partial h}{\partial \boldsymbol{q}} = \begin{bmatrix} \frac{\partial h}{\partial x} & \frac{\partial h}{\partial y} & \frac{\partial h}{\partial \theta} & \frac{\partial h}{\partial v} & \frac{\partial h}{\partial \omega} \end{bmatrix}$$

$$\begin{aligned} \frac{\partial h}{\partial x} &= \mu\,\sigma_3 - \frac{\mathrm{x}_{\mathcal{O}} - x}{\sqrt{\sigma_2}} - \frac{\delta\,\sigma_1\,(\mathrm{y}_{\mathcal{O}} - y)}{\sigma_2} \\ \frac{\partial h}{\partial y} &= \mu\,\sigma_4 - \frac{\mathrm{y}_{\mathcal{O}} - y}{\sqrt{\sigma_2}} + \frac{\delta\,\sigma_1\,(\mathrm{x}_{\mathcal{O}} - x)}{\sigma_2} \\ \frac{\partial h}{\partial \theta} &= \delta\,\sigma_1 + \mu\,\sigma_4\,(\mathrm{x}_{\mathcal{O}} - x) - \mu\,\sigma_3\,(\mathrm{y}_{\mathcal{O}} - y) \\ \frac{\partial h}{\partial v} &= -\mu\,\cos\theta\,(\mathrm{x}_{\mathcal{O}} - x) - \mu\,\sin\theta\,(\mathrm{y}_{\mathcal{O}} - y) \\ \frac{\partial h}{\partial \omega} &= a\,\mu\,\sin\theta\,(\mathrm{x}_{\mathcal{O}} - x) - a\,\mu\,\cos\theta\,(\mathrm{y}_{\mathcal{O}} - y) \end{aligned}$$

where

$$\begin{aligned} \sigma_1 &= \sin\left(\theta - \mathrm{atan2}\,(\mathrm{y}_{\mathcal{O}} - y, \mathrm{x}_{\mathcal{O}} - x)\right), \\ \sigma_2 &= (\mathrm{x}_{\mathcal{O}} - x)^2 + (\mathrm{y}_{\mathcal{O}} - y)^2, \\ \sigma_3 &= v\,\cos\theta - a\,\omega\,\sin\theta, \\ \sigma_4 &= v\,\sin\theta + a\,\omega\,\cos\theta. \end{aligned}$$

This leads to

$$\begin{aligned}\dot{h}(\boldsymbol{q},\dot{\boldsymbol{q}},\boldsymbol{u}) = &\left(\mu\,\sigma_3 - \frac{\mathrm{x}_{\mathcal{O}}-x}{\sqrt{\sigma_2}} - \frac{\delta\,\sigma_1\,(\mathrm{y}_{\mathcal{O}}-y)}{\sigma_2}\right)\dot{x} + \left(\mu\,\sigma_4 - \frac{\mathrm{y}_{\mathcal{O}}-y}{\sqrt{\sigma_2}} + \frac{\delta\,\sigma_1\,(\mathrm{x}_{\mathcal{O}}-x)}{\sigma_2}\right)\dot{y} + \\ &+ \left(\delta\,\sigma_1 + \mu\,\sigma_4\,(\mathrm{x}_{\mathcal{O}}-x) - \mu\,\sigma_3\,(\mathrm{y}_{\mathcal{O}}-y)\right)\omega + \\ &+ \left(-\mu\,\cos\theta\,(\mathrm{x}_{\mathcal{O}}-x) - \mu\,\sin\theta\,(\mathrm{y}_{\mathcal{O}}-y)\right)\frac{u_1}{m} + \\ &+ \left(a\,\mu\,\sin\theta\,(\mathrm{x}_{\mathcal{O}}-x) - a\,\mu\,\cos\theta\,(\mathrm{y}_{\mathcal{O}}-y)\right)\frac{u_2}{I_{cm}}.\end{aligned}$$