Enabling Safe Robotic Contact: A Velocity and Bounded Penetration Approach Using Control Barrier Functions

by

Miquel Rull Trinidad

MSc Thesis to be defended on

16/10/2025

Supervisor & Chair commitee:	C. Pek
Examination commitee:	M.C. Ebere
Examination commitee:	R.D. McAllister
Faculty:	Mechanical Engineering
Department:	Cognitive Robotics
Programme:	MSc. Robotics
Period:	November 2024 - October 2025

TU Delft

Enabling Safe Robotic Contact: A Velocity and Bounded Penetration Approach Using Control Barrier Functions

Miquel Rull Trinidad

***Abstract*— There has been a push for contact-rich manipulation in robotics, where meaningful and deliberate contact with the environment is required. By "meaningful," we refer to purposeful interactions such as pushing a button, inserting a humidity sensor into soil, probing the temperature of a cake, or squeezing a bottle to grasp it. These tasks highlight the need for contact while demanding strict safety guarantees to prevent damage to objects, humans, or the robot. Traditional robot safety assumes contact should be avoided altogether, which constrains contact-rich tasks. As a result, controllers often operate with either reduced safety margins or with reduced velocity, thus decreasing the capabilities of the robot. An ideal controller operates the robot within the full safety margins and with minimal restrictions to its capabilities. This paper uses Control Barrier Functions (CBFs) to ensure that contact with a surface is below a maximum safety velocity and the subsequent surface penetration is bounded, while allowing free movement when contact is not expected. We derive an exponentially decaying velocity–distance CBF and test it in a torque-controlled manipulator simulation. The simulation shows the manipulator slowing down to a safe contact speed and not exceeding the maximum allowed penetration, even when the nominal controller is unaware of the safety constraints. The results are promising and open the door to additional research in safety guarantees in contact with Control Barrier Functions.**

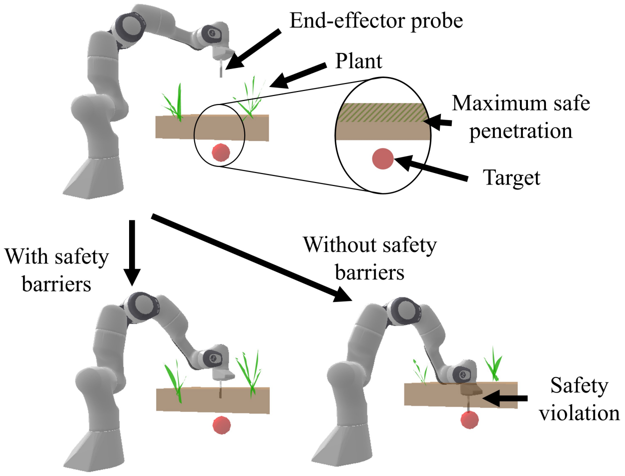


Fig. 1: Running example: probing a hydroponic medium to measure humidity and temperature. The objective of the end-effector is to get as close as possible to the target without violating safety constraints.

I. Introduction

Robotic manipulation increasingly requires contact-rich interactions, where purposeful contact with the environment is essential. Such tasks introduce safety challenges, as excessive contact velocity, force, or penetration can damage the object, the environment, or the robot itself. These challenges fall under the more general notion of contact safety. Although safety in the automotive industry has been a large research topic for decades [1]–[3], contact safety, particularly in robotic systems, remains an emerging and underdeveloped field [4], [5].

Safety in contact is affected by several physical factors, including velocity and mass (kinetic energy and momentum), shape and contact area (pressure), material hardness and compliance (energy dissipation), and the duration of force application (impulse) [4]. Many of these are fixed by design and cannot be changed on runtime, such as mass or shape. We therefore focus on the factors that can be directly controlled in a robot.

Generally, in robots, we can control velocity (through actuation) and force (through force or torque sensors). Force can be controlled so that, once contact has been achieved, the contact force does not go above a safety threshold [6]. However, force sensors are reactive sensors, meaning they cannot prevent unsafe contact when the robot moves with high velocities and inertias. A way to guarantee safety would be to slow down the robot enough such that contact is always safe with impedance control. However, this results in a reduction of the robots dynamic capabilities, as it is being artificially slowed down, even when contact is not expected. An ideal approach would prevent a robot from reaching unsafe contact without limiting its capabilities when contact is not expected.

To illustrate the problem, consider the practical case of probing hydroponic growing media, a common farming task, where a sensorised probe must contact a surface at a safe speed and must not exceed a certain penetration depth to avoid damaging the sensor or the medium. A visualization of the example can be seen in Fig. 1

In this paper, we address the safety and capability limitation problem by proposing a velocity-based Control Barrier Function, which can guarantee that contact with a surface will happen below a defined threshold, even in the presence of an unsafe nominal controller. Additionally, we will incorporate a maximum penetration depth into the Control Barrier Function, such that the movement is halted below a certain penetration in highly compliant surfaces.

II. Related Work

We narrow the scope to studies closely aligned with our setting: contact safety and Control Barrier Functions.

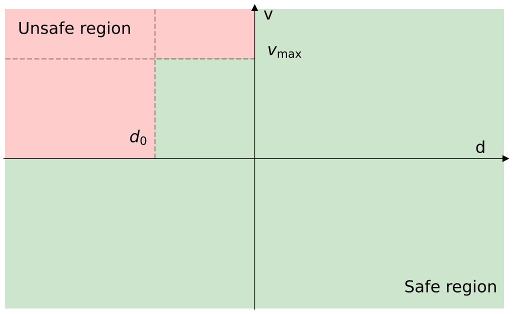


Fig. 2: Safe and unsafe regions of the running example. Unsafe regions (within the obstacle, $d < 0$) occur when the contact velocity exceeds $v_{\max}$, violating (i), or when the penetration depth exceeds d_0, violating (ii).

Contact safety: While safety has been extensively studied in the automotive domain, focusing on impact prevention and dynamics [2], [3], the study of contact safety in robotics is comparatively recent [4], [5]. In robotic systems, research has largely focused on regulating contact forces to prevent damage or injury during interaction [6], [7].

CBF: Control Barrier Functions (CBFs) provide a formal framework for enforcing safety constraints in control systems [8]. They define a set of safe states and adjust control inputs to guarantee the forward invariance of this set. In manipulation, their integration with Operational Space Control has been explored to maintain task consistency while ensuring safety [9], allowing safety filters to intervene minimally in task execution. However, applications of CBFs to contact-rich interaction remain limited. In the healthcare domain, CBFs have been used to constrain contact forces during physical interaction [6], and recent work has investigated self-contact safety in hybrid soft–rigid robots [10].

III. Problem Statement

We consider the problem of reaching a target position $[x, y, z]$ and orientation $[r, p, y]$ that lies within an obstacle region, without exceeding two safety bounds: (i) a maximum allowable contact velocity $v_{\max}$, and (ii) a maximum penetration depth d_0. The approach developed is general, but we use a running example for clarity.

Hydroponic media probing: The running example involves probing a hydroponic media surface for humidity and temperature measurement where the tip of the end effector (the probe) is controlled.

This task imposes two requirements for safe contact:

(i) The probe must contact the surface with a velocity below $v_{\max}$, and this bound must not be exceeded even after initial penetration.

(ii) The probe must not penetrate deeper than d_0, ensuring neither the sensor or the medium are damaged.

A visualization of the safe and unsafe regions for this recurring example is shown in Fig. 2. The objective of this paper is to guarantee that the end effector does not operate in the unsafe region, even under an unsafe nominal controller.

IV. Preliminaries

A. System model

Consider a torque-controlled robotic arm with n degrees of freedom. The control input is the joint torque vector

$$\mathbf{u} = \boldsymbol{\tau} \in \mathbb{R}^n,$$

and the joint state is represented by joint positions and velocities

$$\mathbf{z} = \begin{bmatrix} \mathbf{q} \\ \dot{\mathbf{q}} \end{bmatrix} \in \mathbb{R}^{2n}.$$

The control-affine dynamics of the manipulator [9] can be expressed as

$$\dot{\mathbf{z}} = \begin{bmatrix} \dot{\mathbf{q}} \\ \ddot{\mathbf{q}} \end{bmatrix} = \underbrace{\begin{bmatrix} \dot{\mathbf{q}} \\ -M^{-1}(\mathbf{q})\big(\mathbf{c}(\mathbf{q}, \dot{\mathbf{q}}) + \mathbf{g}(\mathbf{q})\big) \end{bmatrix}}_{f(\mathbf{z})} + \underbrace{\begin{bmatrix} 0 \\ M^{-1}(\mathbf{q}) \end{bmatrix}}_{g(\mathbf{z})} \mathbf{u} \tag{1}$$

where $M(\mathbf{q}) \in \mathbb{R}^{n \times n}$ is the positive-definite mass matrix, $\mathbf{c}(\mathbf{q}, \dot{\mathbf{q}})$ contains Coriolis and centrifugal terms, and $\mathbf{g}(\mathbf{q})$ represents gravitational effects.

B. Control Barrier Functions

Control Barrier Functions (CBFs) provide a formal mechanism for guaranteeing safety in control systems. Given a continuously differentiable function $h : \mathbb{R}^n \to \mathbb{R}$, the corresponding safe set is defined as

$$\mathcal{C} = \{\mathbf{z} \in \mathbb{R}^n \mid h(\mathbf{z}) \geq 0\}.$$

Forward invariance of $\mathcal{C}$ is guaranteed if there exists an extended class-$\mathcal{K}$ function α such that

$$\dot{h}(\mathbf{z}, \mathbf{u}) = L_f h(\mathbf{z}) + L_g h(\mathbf{z}) \mathbf{u} \geq -\alpha(h(\mathbf{z})), \tag{2}$$

for all $\mathbf{z} \in \mathcal{C}$. This condition ensures that if the system state starts in $\mathcal{C}$, it will remain in $\mathcal{C}$ for all future times.

In practice, α is often chosen as a positive constant multiplier, making it a linear class-$\mathcal{K}$ function. Larger values of α yield more aggressive corrections near the boundary $h \approx 0$, resulting in faster recovery from violations of the constraint.

The barrier condition can be enforced through a quadratic program (QP), where a nominal control input is modified as little as possible while satisfying the inequality constraint above.

$$\begin{aligned} \underset{\mathbf{u}}{\text{minimize}} \quad & \|\mathbf{u} - \mathbf{u}_{\text{nom}}\|_2^2 \\ \text{subject to} \quad & L_f h(\mathbf{z}) + L_g h(\mathbf{z})\, \mathbf{u} \geq -\alpha(h(\mathbf{z})) \end{aligned} \tag{3}$$

For rigorous proof and a more detailed theoretical treatment of CBFs, we refer the reader to [8].

We employ CBFs as our primary safety mechanism because they provide formal guarantees of forward invariance while remaining computationally efficient to implement in real time. This makes them practical for deployment on physical robotic systems, in contrast to alternative approaches such as reachability analysis, which are often too computationally expensive. CBFs therefore offer a balance between theoretical rigour and real-world feasibility.

V. Safe Contact Through CBFs

This section presents the formulation of the proposed Control Barrier Function. We begin by outlining the assumptions and modelling choices. The CBF is then derived to enforce both velocity and penetration constraints. Finally, task consistency is introduced to ensure that safety enforcement minimally affects task performance.

A. Assumptions

To make the CBF formulation applicable, we outline below the general assumptions required for implementation.

- The robot dynamics are accurately represented by a known state-space model.
- Reliable perception is available, including state estimation and obstacle distance measurement, with sufficiently low uncertainty.
- Object penetration produces negligible resistance, as in soft or porous materials such as hydroponic media.
- Control authority is available at the torque level. Position or velocity control alone cannot guarantee safety in a velocity CBF.

B. Problem modelling

- The problem is bounded in space and time. That is to say, safety constraints are enforced only when the robot intentionally approaches the surface.
- The motion of the end effector relative to the target surface is modelled as one-dimensional and perpendicular to the surface. In our case, the end effector moves down in the negative-z direction.
- The maximum allowable contact velocity $v_{\max}$ is known.
- The signed distance to the surface d and relative velocity v_{rel} can be measured or estimated with sufficient accuracy. For instance, d may be obtained from a distance sensor, while v_{rel} can be computed using the robot Jacobian and $\dot{\mathbf{q}}$. In our specific case, d is the Cartesian difference between the end effector z position and the surface z_{obs}, while v_{rel} is equal to the velocity in the negative-z direction.

C. CBF derivation

The problem statement introduced two requirements: (i) the probe must contact the surface with velocity not exceeding $v_{\max}$, and (ii) the penetration depth must not exceed d_0.

Let v_{rel} denote the relative Cartesian velocity of the probe with respect to the surface, and let d denote the signed distance from the probe to the surface ($d < 0$ indicates positive penetration). We define the initial CBF as

$$h(\mathbf{z}) = v_{\text{ref}}(d) - v_{\text{rel}}, \tag{4}$$

where $v_{\text{ref}}(d)$ is a distance-dependent reference velocity that we do not want to exceed. The system is safe when $h(\mathbf{z}) \geq 0$. If the relative velocity is higher than the reference, safety is violated. Therefore, we want to shape this $v_{\text{ref}}(d)$ to be close to the boundary between safe and unsafe regions found in Fig. 2, while enclosing all of the unsafe regions.

Since this boundary is not differentiable in its raw form, as there are discontinuous jumps in $d = 0$ and $d = -d_0$, we will have to approximate the desired shape with a smooth curve. In this paper, we choose to model v_{ref} as an exponentially decaying curve.

$$v_{\text{ref}}(d) = v_{\max} + \kappa \left(e^{\lambda d} - 1 \right), \tag{5}$$

where $\kappa > 0$ is a design constant and $\lambda > 0$ is the damping. This exponential form offers several advantages. It is smooth and analytically simple, with an easily tunable slope through λ. Increasing λ results in a rapid decay of velocity just after contact, producing a more conservative response; smaller λ values generate a near-linear decay, allowing faster penetration. The choice of λ depends on the desired behaviour in the task. This CBF slope ensures that the probe decelerates continuously as it penetrates the surface while being differentiable.

With Eq. 5, the velocity requirement (i) would be achieved. This is because as $d \Rightarrow 0$, the CBF becomes

$$\begin{aligned} h(\mathbf{z}) &= v_{\max} + \kappa \left(e^{\lambda d \Rightarrow 0} - 1 \right) - v_{\text{rel}} \\ &= v_{\max} - v_{\text{rel}} \end{aligned} \tag{6}$$

Which is negative when $v_{\text{rel}} > v_{\max}$ and positive otherwise. This confirms that the CBF is safe at the point of contact $d = 0$, as the CBF, when starting on the safe set $h > 0$, is guaranteed to remain positive for all times. Therefore we now satisfy (i). However, we do not yet satisfy (ii).

To satisfy the penetration requirement, we make use of the design constant κ. We require that $h(\mathbf{z}) \Rightarrow 0$ (safety boundary) with $v_{\text{rel}} \Rightarrow 0$ and $d \Rightarrow -d_0$ (the end effector must come to a stop at the maximum penetration amount), yielding

$$h(\mathbf{z}) \Rightarrow 0 = v_{\max} + \kappa \left(e^{\lambda d \Rightarrow -d_0} - 1 \right) - (v_{\text{rel}} \Rightarrow 0) \tag{7}$$

Rearranging results in

$$\kappa = \frac{v_{\max}}{1 - e^{-\lambda d_0}}. \tag{8}$$

Altogether, we have the following CBF:

$$h(\mathbf{z}) = v_{\max} + \frac{v_{\max}}{1 - e^{-\lambda d_0}} \left(e^{\lambda d} - 1 \right) - v_{\text{rel}} \tag{9}$$

This formulation reduces the maximum allowed velocity as the probe goes deeper, until it reaches zero at the maximum penetration depth. In effect, the CBF makes the end-effector move smoothly from free motion to a full stop at $d = -d_0$. This prevents both excessive speed and over-penetration, even if the nominal controller issues aggressive commands. The CBF can ve visualized, together with the safe and unsafe set in Fig. 3, with the boundary $h = 0$ marking the limit. Note that in practice, the constant κ is pre-computed.

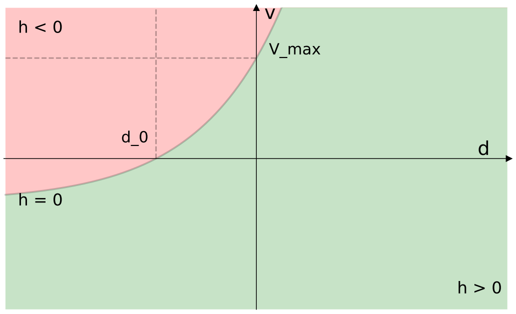


Fig. 3: Safe set (region where $h > 0$) and unsafe set ($h < 0$) of the derived CBF for the running example.

D. Task consistency

When the QP filter minimises the input (joint torque) while the task is expressed in Cartesian space, task performance can degrade. This happens as the QP modifies the input in the joint space while disregarding what happens in the Cartesian space. Task consistency ensures that the safety filter minimally alters the nominal task execution, preserving performance while enforcing safety. This is achieved by changing the QP objective, so that deviations from both joint and operational space commands are penalized. Following [9], we adopt their task-consistent quadratic program formulation:

$$\begin{aligned} \min_{\Gamma} \quad & \|W_j(\ddot{q}_N - \ddot{q}_{N,\text{nom}})\|_2^2 + \|W_o(\dot{\nu} - \dot{\nu}_{\text{nom}})\|_2^2 \\ \text{s.t.} \quad & L_f h(z) + L_g h(z)\,\mathbf{u} \geq -\alpha(h(z)) \end{aligned} \tag{10}$$

Here, $\ddot{q}_N$ and $\dot{\nu}$ denote the null joint-space and operational-space accelerations, with $\ddot{q}_{N,\text{nom}}$ and $\dot{\nu}_{\text{nom}}$ as their respective nominal values. The weighting matrices W_j and W_o are positive definite and allow tuning of the relative importance between joint and operational space objectives. In our paper, W_j and W_o are set to be identity matrices. This formulation ensures that the CBF modifies the control input only as much as necessary to maintain safety, while respecting the task. For a complete derivation and further discussion, we refer the reader to [9].

VI. Experimental Results

The experiments aim to validate that the proposed Control Barrier Function (CBF) ensures safe contact in simulation by constraining velocity and penetration during surface interaction. We also assess robustness to variations in key parameters (λ, d_0, and $v_{\max}$) and verify that safety is maintained across these conditions.

A. Simulation setup

The PyBullet simulation environment and CBF framework provided by OSCBF were used. This framework provides a Franka Emika Panda arm model with its dynamical model, as well as additional functionality to compute Jacobians and mass matrices outside PyBullet. Furthermore, most computations are accelerated using JAX, which gives an excellent base to implement our own CBFs. There are two changes made to the framework. The first change is the Panda gripper arm being removed and replaced with a ten centimetre long cylinder, simulating the probe. The second change was made to the CBFPy subclass, by removing the CBF relaxation. To balance simulation stability and computation speed, a simulation time-step of $0.002\,\text{s}$ was selected.

Nominal controller: The PD Pose-Task torque controller from OSCBF [9] was used, however, the desired-state function was simplified. The objective orientation is set to the -z direction (end effector looking down). The desired position is set to a target below the surface, beyond the penetration safety limit. The desired velocity and angular velocity are set to 0. The gains used can be seen in Table I. This simple PD controller is unaware of the safety constraints and will therefore serve as a comparison between using a CBF or not.

TABLE I: Nominal controller gains

Gain	Value	Gain	Value
K_p – position	20	K_p – orientation	100
K_d – position	20	K_d – orientation	100

The controller will be used to position the arm above the surface to penetrate (aligned with the z-axis) before the experiments begin. When the experiment begins, the CBF is activated and the arm moves down in the negative z direction using the nominal controller until 6 seconds pass.

Simulation run without safety: Fig. 4 summarizes the results of a simulation without active CBF safety. The nominal controller exceeds both the maximum contact velocity and penetration limits, confirming unsafe behaviour.

Base run: The base run is a simulation that remains constant across the result graphs, allowing cross-graph referencing. The colour of this run in the main comparison [Fig. 5] is orange. The parameters of this run are listed in Table II.

TABLE II: Base run values

Variable	Value	Variable	Value
$v_{\max}$	0.05 m/s	d_0	0.1 m
λ	10	z_{obstacle}	0.3 m
α	10	Start Position	[0.4, 0.4, 0.5] m
Target Position	[0.4, 0.4, 0.1] m		

B. Results

The following results illustrate how the active CBF performs under different damping, maximum contact velocity, and penetration parameters.

a) Damping behaviour: In Fig. 5a, the damping behaviour is shown. In the top-left figure, none of the damping values exceed the maximum penetration requirement, while in the top-centre figure all runs contact the surface below $v_{\max}$. Therefore, the method appears robust with respect to the task requirements. However, in the phase-space graph on the right, different trajectory behaviours are observed. Higher damping values allow more speed outside the surface, while small damping limits the speed outside the surface.

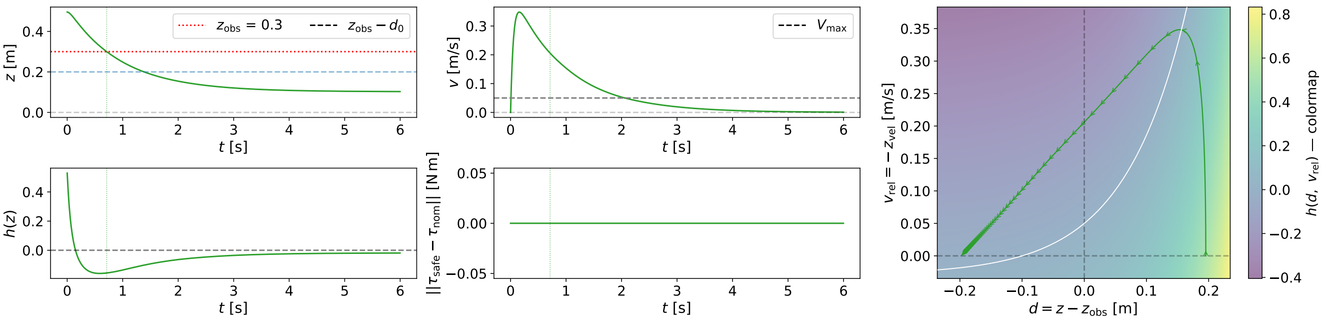
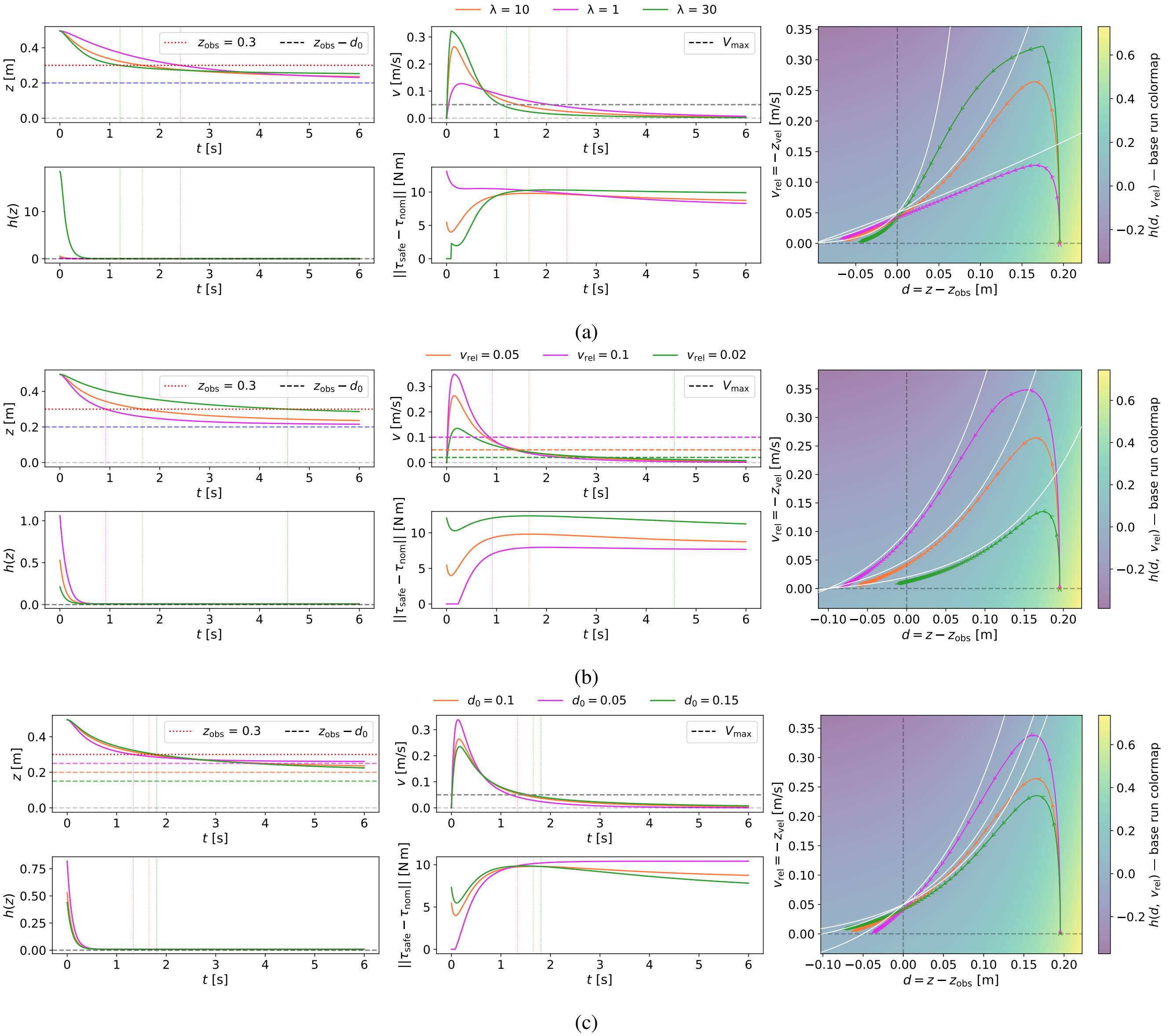


Fig. 4: Simulation run without an active CBF. Top-left: end-effector position relative to obstacle and maximum allowed penetration. Top-center: end-effector speed vs. time. Bottom-left: CBF value over time. Bottom-center: normalised torque difference between safe and nominal control. Right: phase-space diagram of distance vs. speed; the white line marks the $h = 0$ safety boundary.



(a)

(b)

(c)

Fig. 5: Parameter comparison. The orange line is consistent between the graphs, representing the base run. (a) Damping comparison. (b) Maximum contact speed comparison. (c) Penetration comparison.

This results from the exponential approximation of the task space, which extends past the $d = 0$ plane. We also note that a heavily damped CBF stops the arm from moving further in, while the less damped runs penetrate deeper within the same time frame, despite their slower speeds outside the surface. Another notable observation, visible in the green line, is the clear change from free movement to CBF intervention. This can be seen both in the phase-space and in the normalised change graph in the bottom-centre.

b) Maximum contact velocity behaviour: In Fig. 5b, three different contact velocities are compared. Again, all safety constraints are respected. The results show very similar trajectories with the expected exponential decays for each speed. The lower speed predictably does not reach as deep as the others within the six-second time frame. Notably, the pink line shows the transition from no intervention to CBF intervention, though this time it is only evident in the normalised change graph.

c) Penetration behaviour: Finally, the maximum penetration d_0 is compared in Fig. 5c. The phase-space graph shows a steeper trajectory decay with smaller allowed penetration, while larger penetration permits a slightly slower approach. This resembles the damping behaviour, although penetration does not affect the speeds as drastically as damping.

C. Experiment conclusions

All experiments remained within the safety boundary $h > 0$, and no oscillations or failed QP solutions were observed. The proposed CBF exhibited robustness to parameter variations and consistently enforced the safety constraints. The results demonstrate that the safety constraints derived in Section V were correctly enforced in simulation, even when the nominal controller was unaware of them. The computational efficiency of the proposed method was also evaluated to assess its suitability for real-time implementation.

D. Implementation details

The simulation experiments were run on a laptop with a Ryzen 7 8845HS, operating at maximum performance (5 GHz). The simulations, when run without real-time constraints, consistently ran at 1200 Hz ± 30 Hz.

Given the achieved simulation speed, real-time deployment at 100–200 Hz appears computationally feasible on standard embedded hardware.

VII. Discussion and Limitations

The results address the central challenge raised in the introduction: enabling purposeful contact without compromising safety and minimally affecting the dynamic capabilities of the robot. Traditional force-based or impedance controllers can only react to unsafe contact after it occurs, often by reducing overall speed or limiting task performance. In contrast, the proposed velocity-based CBF approach prevents unsafe contact proactively by constraining motion before impact, ensuring safety without unnecessarily restricting free-space behaviour. This demonstrates that formal safety guarantees can be brought to contact-rich manipulation without the conservative trade-offs that can limited robotic interaction.

Limitations

During the research, several limitations were observed. First, uncertainties in the distance to the surface, the robot's position, and the relative velocity to obstacles must be known with high accuracy. If there is a known error in velocity or position, safety can be maintained by reducing $v_{\max}$ and d_0 by the respective uncertainty amounts. However, if the uncertainty exceeds $v_{\max}$ or d_0, the Control Barrier Functions (CBFs) cannot guarantee safe contact. Second, after contact is achieved, the robot often stops penetrating further due to resistance, surface rebound and friction. The current CBF formulation applies torque as if the robot could move freely, which may limit tasks that require a specific penetration depth. Overcoming this limitation would require introducing a contact model into the state-space representation to capture penetration resistance and allow higher torque when the distance becomes negative. Finally, when all degrees of freedom are used, the quadratic programming (QP) solver can struggle to find a safe solution. For instance, during a diagonal trajectory instead of a linear one, the safety function h exceeded its limit by approximately 0.002, and contacted the surface 1.5 mm/s faster than $v_{\max}$, as seen in Fig. 6.

Challenges and future directions

The main obstacles in bridging the gap between simulation and real-world are the uncertainties and external resistances listed on the limitations. Additionally, the robot must be controlled with acceleration or torque, which restricts the set of robots able to benefit from this method. The approach also needs to be generalised for robots performing diverse tasks. Based on these challenges, several research directions emerge:

- Surface detection at runtime: The current method assumes a known surface position, while a deployed robot must be able to detect surfaces as required by the task.
- Generalisation to 3D: The current implementation is restricted to tasks where a manipulator moves downward along the z-axis. The CBFs v and d should instead be defined relative to the end-effector, enabling tasks on surfaces with arbitrary positions and orientations.
- Adaptation to varying surface properties: If surface properties vary during operation, the robot should be able to generate or switch between different CBFs, updating v_{max} and d_0 to reflect the new conditions.
- Blending velocity and force CBFs: A promising approach to combine the strengths of velocity-based and force-based contact CBFs [6] would be to use both simultaneously. This ensures safe contact velocities while also preventing excessive force or penetration.
- Pushing tasks: For tasks such as pushing, there may be no maximum penetration constraint, but a maximum contact or pushing velocity remains. The safe and unsafe regions of this task are illustrated in Fig. 7.

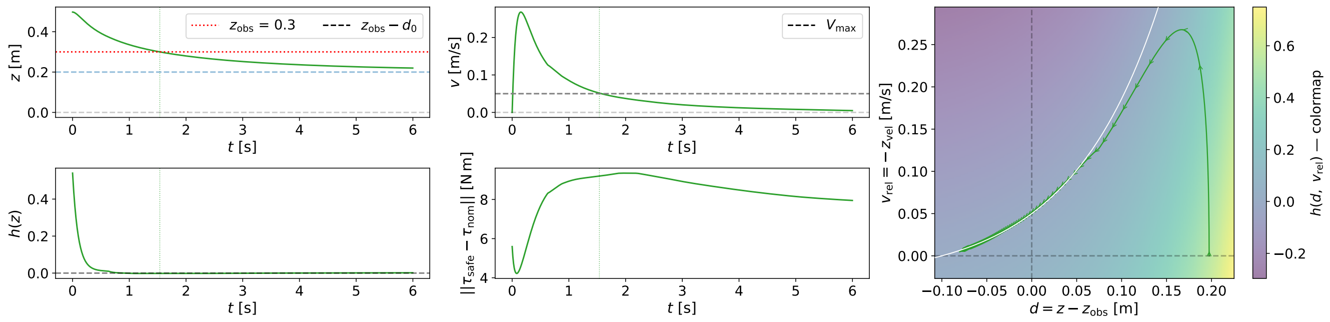


Fig. 6: Limitations observed when all degrees of freedom are in use.

- Improved safe set approximations: The current exponential form can limit the capabilities of the robot outside of the task, due to approximation errors. A different approximation, such as $\ln(1+x^7)$ (Fig. 8), may reduce conservatism.

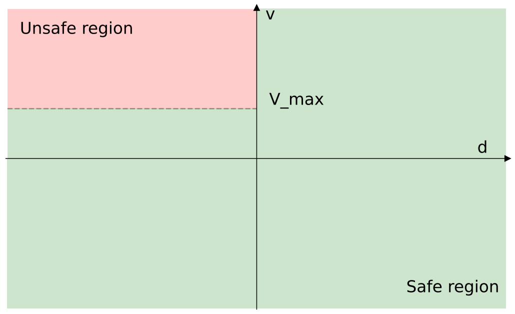


Fig. 7: Safe and unsafe regions for a pushing task.

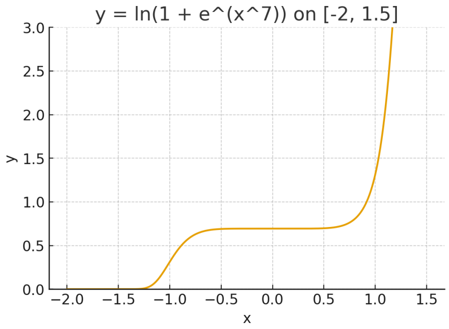


Fig. 8: Better approximation of the task space

VIII. Conclusion

This paper presents a novel method to achieve safe contact between a robot and a surface, while also enforcing a maximum penetration limit. To the best of our knowledge, this is the first application of CBFs that explicitly considers both velocity and penetration constraints in contact tasks. This contribution expands the scope of CBF-based safety methods and provides a foundation for tasks where bounded penetration is essential.

The results demonstrate that the method respects safety requirements under different damping, velocity, and penetration parameters. Within the stated assumptions, the approach shows potential for deployment on torque-controlled robots.

Future work should address uncertainty handling, compliance modeling, and generalisation to 3D and diverse tasks, as outlined in the discussion. By tackling these challenges, the method could move from simulation toward real-world deployment, enabling safe and versatile physical interaction with the environment.

Acknowledgment

I would like to thank my supervisor Chris Pek, as well as Moses Ebere, for their extensive guidance throughout this Master's thesis.

Additionally, I would like to thank the creators of CBFPy and OSCBF for providing frameworks that enabled efficient implementation of new control barrier functions which significantly reduced development time.

References

[1] R. van der horst and J. Hogema, "Time-to-collision and collision avoidance systems," 01 1994.

[2] K. Schweizerhof, L. Nilsson, and J.O.Hallquist, "Crashworthiness analysis for the automotive industry," *International Journal of Computer Applications in Technology*, vol. 5, pp. 134–156, 01 1992.

[3] L. Wech and U. Seiffert, *Automotive Safety Handbook*, ser. R.: Society of Automotive Engineers. SAE International, 2007. [Online]. Available: https://books.google.nl/books?id=bOKbEAAAQBAJ

[4] C. Fischer, M. Neuhold, M. Steiner, T. Haspl, M. Rathmair, and S. Schlund, "Collision tests in human-robot collaboration: Experiments on the influence of additional impact parameters on safety," *IEEE Access*, vol. 11, pp. 118 395–118 413, 2023.

[5] S. Haddadin, A. Albu-Schäffer, and G. Hirzinger, "Safety Evaluation of Physical Human-Robot Interaction via Crash-Testing," Tech. Rep.

[6] F. Vinter-Hviid, C. Sloth, T. R. Savarimuthu, and I. Iturrate, "Safe contact-based robot active search using bayesian optimization and control barrier functions," *Frontiers in Robotics and AI*, vol. 11, 2024. [Online]. Available: https://www.frontiersin.org/journals/robotics-and-ai/articles/10.3389/frobt.2024.1344367

[7] X. Zhu, W. Lian, B. Yuan, C. D. Freeman, and M. Tomizuka, "Allowing safe contact in robotic goal-reaching: Planning and tracking in operational and null spaces," 2022. [Online]. Available: https://arxiv.org/abs/2211.08199

[8] A. D. Ames, S. Coogan, M. Egerstedt, G. Notomista, K. Sreenath, and P. Tabuada, "Control barrier functions: Theory and applications," 2019. [Online]. Available: https://arxiv.org/abs/1903.11199

[9] D. Morton and M. Pavone, "Safe, task-consistent manipulation with operational space control barrier functions," 2025. [Online]. Available: https://arxiv.org/abs/2503.06736

[10] Z. J. Patterson, W. Xiao, E. Sologuren, and D. Rus, "Safe control for soft-rigid robots with self-contact using control barrier functions," 2024. [Online]. Available: https://arxiv.org/abs/2311.03189