Closed-Form CBF Filtering with Input Saturation for Safe Point Robot Navigation

Onyia Ifeanyi

***Abstract*—This paper presents a computationally efficient safety filter for a 2D point robot navigating around a circular obstacle. The proposed approach combines a simple input saturation pre-filter, which enforces the actuation bound on the nominal command, with a closed-form Exponential Control Barrier Function (ECBF) projection, which enforces the safety half-space constraint, eliminating the need for an online quadratic program (QP) solver. The filter enforces collision avoidance whenever the ECBF half-space projection is feasible, while achieving deterministic sub-microsecond execution times. We provide a rigorous derivation of the closed-form projection, including a numerical safeguard for degenerate cases and a feasibility analysis characterizing when the half-space projection is well defined for the single-obstacle case. The approach is validated through Monte Carlo simulations and computational benchmarks demonstrating a 99.61% safety rate across 10,000 randomized trials and a mean latency of 3.34 μs with minimal jitter—approximately 500× faster than a standard QP-based implementation (OSQP) solving the same constrained problem.**

***Index Terms*—Control Barrier Functions, Safety Filter, Point Robot Navigation, Closed-Form Control, Input Saturation, Real-Time Control.**

I. Introduction

CONTROL Barrier Functions (CBFs) have emerged as a powerful tool for enforcing safety constraints in nonlinear control systems [1]. The standard approach formulates a quadratic program (QP) that minimally modifies a nominal control input to satisfy a CBF inequality. While this framework guarantees safety, it relies on numerical solvers that introduce non-deterministic latency, computational jitter, and potential convergence issues—particularly problematic for high-frequency embedded control systems.

Several works have explored closed-form solutions to CBF-QPs for specific system classes. For relative-degree-one systems with single linear constraints, the projection onto a half-space admits a simple closed form. However, for relative-degree-two systems (which require Exponential CBFs), the problem becomes more complex. Existing closed-form solutions typically neglect input saturation, limiting their practical applicability.

This paper addresses this gap by presenting a closed-form CBF safety filter for a 2D point robot with bounded acceleration. The proposed filter operates in two stages: (i) saturation of the nominal command to respect actuation limits, followed by (ii) closed-form projection onto the ECBF-safe half-space. This saturate-then-project architecture enforces the ECBF constraint whenever the projection is feasible, while achieving deterministic sub-microsecond execution times. The closed-form algebraic structure of the filter ensures deterministic execution on embedded platforms, and its computational simplicity makes hardware implementation straightforward.

Manuscript received July 27, 2026; revised July 27, 2026.

The key contributions of this paper are:

1) A closed-form ECBF projection for a double-integrator system with a circular obstacle, derived analytically via Karush-Kuhn-Tucker (KKT) conditions.
2) A saturate-then-project pipeline that enforces the ECBF constraint after input saturation on the nominal controller, without requiring a numerical solver, conditional on projection feasibility.
3) A numerical safeguard for the degenerate case near the obstacle center, ensuring implementation robustness.
4) A feasibility analysis for the single-obstacle ECBF half-space, giving an exact closed-form test for when the projection is well-defined and when it is not.
5) Experimental validation through Monte Carlo simulations demonstrating a 99.61% safety rate across 10,000 randomized trials, with the remaining cases consistent with discrete-time constraint violations and infeasible configurations under the chosen discretization.
6) A computational benchmark showing a 500× speedup over a standard QP solver (OSQP), with deterministic sub-microsecond execution and minimal jitter.

A. Related Work

Input constraints have been incorporated into CBF-based safety filters primarily by embedding the bound into an augmented online QP: Breeden and Panagou [5] handle high relative-degree constraints this way, and Shaw Cortez *et al.* [6] extend this to certify compatibility across multiple barrier constraints, both relying on QP feasibility checks. Xiao *et al.* [7] derive sufficient conditions for CBF-QP feasibility along a trajectory—conceptually related to the feasibility question we address in Section II, but developed for the general QP setting rather than a closed-form projection. Deng *et al.* [8] address complex, possibly non-convex input constraints. Nguyen and Sreenath [2] introduced the ECBF formulation used here and posed the associated min-intervention law as a QP; our closed-form projection specializes this law to the single linear ECBF constraint arising for a single circular obstacle. CBF-based filters have also been used to certify learned or reinforcement-learning policies at deployment time [3].

Unlike the works above, we do not attempt to jointly enforce the ECBF constraint and the actuation bound within a single constrained optimization. Instead, we decouple the two: the actuation bound is enforced by a saturation pre-filter applied to the nominal command, and the ECBF constraint is enforced afterward by a closed-form half-space projection. This sacrifices the joint guarantee available from a QP formulation in exchange for an algebraic, solver-free filter whose execution time is deterministic and suitable for embedded implementation at microsecond timescales. We do not claim this to be the first closed-form treatment of an ECBF constraint; rather, our contribution is the specific decoupled, solver-free pipeline and its empirical characterization, including the conditions under which it can fail.

We emphasize that this paper focuses on the single-obstacle case, which serves as the canonical building block for more complex environments. Extension to multiple obstacles is left as future work.

II. Problem Formulation

Consider a 2D point robot with position $p \in \mathbb{R}^2$ and velocity $v \in \mathbb{R}^2$. The state is $x = [p^\top, v^\top]^\top \in \mathbb{R}^4$. The robot dynamics are described by the double-integrator model

$$\dot{p} = v, \qquad \dot{v} = u, \tag{1}$$

where $u \in \mathbb{R}^2$ is the acceleration control input. The control input is subject to a strict actuation bound

$$\|u\|_2 \le U_{\max}, \tag{2}$$

where $U_{\max} > 0$ is the maximum allowable acceleration.

A. Obstacle and Safe Set

Let there be a single circular obstacle centered at $p_o \in \mathbb{R}^2$ with radius $r_o > 0$. The unsafe set is the open disk

$$\mathcal{O} = \{p \in \mathbb{R}^2 : \|p - p_o\|_2 < r_o\}. \tag{3}$$

We define the safe set $\mathcal{S} \subset \mathbb{R}^4$ as the superlevel set of a continuously differentiable function $h : \mathbb{R}^4 \to \mathbb{R}$:

$$\mathcal{S} = \{x \in \mathbb{R}^4 : h(x) \ge 0\}, \tag{4}$$

with

$$h(x) = \|p - p_o\|_2^2 - r_o^2. \tag{5}$$

The objective is to ensure that the system state never enters the unsafe set, i.e., $h(x(t)) \ge 0$ for all $t \ge 0$, given $h(x(0)) \ge 0$.

B. Nominal Controller

Define a fixed goal position $p_g \in \mathbb{R}^2$ with $\|p_g - p_o\|_2 > r_o$. The nominal controller is a decoupled PD law designed for goal-seeking performance in the absence of obstacles:

$$u_{\text{nom}}(x) = -k_p(p - p_g) - k_d v, \tag{6}$$

where $k_p > 0$ and $k_d > 0$ are the proportional and derivative gains, respectively. While this controller guarantees asymptotic stability of the equilibrium $(p, v) = (p_g, 0)$ in the unconstrained setting, it provides no safety guarantees and may drive the robot into the obstacle.

C. Exponential Control Barrier Function

To enforce collision avoidance, we adopt the Exponential Control Barrier Function (ECBF) framework [2]. Since the function $h(x)$ defined in (5) has relative degree two with respect to the dynamics (1), we require the following condition to hold:

$$\ddot{h}(x, u) + 2\lambda \dot{h}(x) + \lambda^2 h(x) \ge 0, \tag{7}$$

for some tuning parameter $\lambda > 0$. Expanding the derivatives yields

$$\dot{h}(x) = 2(p - p_o)^\top v, \tag{8}$$

$$\ddot{h}(x, u) = 2\|v\|_2^2 + 2(p - p_o)^\top u. \tag{9}$$

Substituting (8) and (9) into (7) and rearranging gives the affine constraint in u:

$$A(x)u \ge b(x), \tag{10}$$

where

$$A(x) = 2(p - p_o)^\top \in \mathbb{R}^{1\times 2}, \tag{11}$$

$$b(x) = -\left(2\|v\|_2^2 + 4\lambda(p - p_o)^\top v + \lambda^2\left(\|p - p_o\|_2^2 - r_o^2\right)\right). \tag{12}$$

The sign convention in (12) is critical: all terms not containing u are moved to the right-hand side with a negative sign, ensuring that $Au \ge b$ is equivalent to the ECBF condition (7).

D. Control Architecture: Saturate-then-Project

The proposed safety filter consists of two sequential steps.

1) Step 1: Input Saturation: Given the nominal control command u_{nom} from (6), we first enforce the actuation bound (2) via the saturation function

$$\tilde{u} = \operatorname{sat}_{U_{\max}}(u_{\text{nom}}) = \begin{cases} u_{\text{nom}}, & \|u_{\text{nom}}\|_2 \le U_{\max}, \\ U_{\max}\dfrac{u_{\text{nom}}}{\|u_{\text{nom}}\|_2}, & \text{otherwise.} \end{cases} \tag{13}$$

This step, by itself, guarantees $\|\tilde{u}\|_2 \le U_{\max}$.

2) Step 2: Closed-Form CBF Projection: The saturated command $\tilde{u}$ is then projected onto the safe half-space defined by the ECBF constraint (10). This yields the final safe control input

$$u^* = \tilde{u} + \max\left(0, \frac{b(x) - A(x)\tilde{u}}{\|A(x)\|_2^2}\right) A(x)^\top, \tag{14}$$

where $A(x)$ and $b(x)$ are defined in (11) and (12). This projection is the exact closed-form solution to the optimization problem

$$u^* = \arg\min_{u\in\mathbb{R}^2} \frac{1}{2}\|u - \tilde{u}\|_2^2 \quad \text{s.t.} \quad A(x)u \ge b(x). \tag{15}$$

The projection (14) is obtained by solving the Karush-Kuhn-Tucker (KKT) conditions for (15). It guarantees that the ECBF condition (10) is satisfied with minimal Euclidean deviation from the saturated nominal command

$\tilde{u}$. Note that (15) enforces only the half-space constraint; it does not include the actuation bound (2) as a constraint, so the correction term in (14) can, in principle, push u^* outside the ball $\|u\|_2 \leq U_{\max}$ even though $\tilde{u}$ itself satisfies it. This point is discussed further in Section II-F.

E. Numerical Safeguard

The closed-form projection (14) involves division by $\|A(x)\|_2^2$, where $A(x) = 2(p - p_o)^\top$. In continuous time, the singularity at $p = p_o$ is a measure-zero set and is avoided in exact arithmetic. However, in discrete-time implementation, the robot may pass arbitrarily close to the obstacle center, making $\|A(x)\|_2^2$ small and causing the correction term to become numerically unstable.

To address this, we introduce a small threshold $\epsilon > 0$ and implement the safeguard:

$$u^* = \begin{cases} \tilde{u}, & \|A(x)\|_2^2 < \epsilon, \\ \tilde{u} + \max\left(0, \dfrac{b(x) - A(x)\tilde{u}}{\|A(x)\|_2^2}\right) A(x)^\top, & \text{otherwise.} \end{cases} \tag{16}$$

This is an implementation-level fallback for the degenerate case in which the robot is at or very near the obstacle center. In the nominal safe operating region, where $\|p - p_o\|_2 \geq r_o$, we have $\|A(x)\|_2^2 = 4\|p - p_o\|_2^2 \geq 4r_o^2 > 0$, so $\|A(x)\|_2^2$ is bounded away from zero and the safeguard does not trigger under normal operation. It is invoked only when the discretized trajectory transiently passes very close to the obstacle center—i.e., only in states where the barrier gradient has already nearly vanished and the ECBF condition provides little to no useful directional information. In that regime we fall back to the saturated nominal command $\tilde{u}$ rather than dividing by a near-zero quantity. The threshold ϵ is chosen to be machine-precision small (e.g., $\epsilon = 10^{-10}$ in double-precision arithmetic).

Remark II.1. *The case $\|A\|_2^2 < \epsilon$ is a measure-zero set in continuous time and is encountered only transiently in discrete-time simulation. The safeguard is included for implementation robustness and does not affect the theoretical safety guarantee, which assumes exact arithmetic and $p \neq p_o$; it maintains numerical stability in this edge case but is not itself part of the safety proof in Section IV.*

F. Feasibility of the Safety Filter

The closed-form projection (14) guarantees that the ECBF condition $A(x)u \geq b(x)$ is satisfied whenever the half-space projection is well-defined. We now characterize the conditions under which a feasible control input exists, and clarify what is and is not guaranteed once the actuation bound is taken into account. *This is the single place in the paper where these caveats are stated in full; later sections refer back to this subsection rather than repeating them.*

The full optimization problem, including the input bound, is:

$$\min_{u \in \mathbb{R}^2} \frac{1}{2}\|u - \tilde{u}\|_2^2 \quad \text{s.t.} \quad A(x)u \geq b(x), \quad \|u\|_2 \leq U_{\max}. \tag{17}$$

The saturation pre-filter guarantees $\|\tilde{u}\|_2 \leq U_{\max}$ for the intermediate command, but the projection step in (14) solves the relaxed problem (15), which omits the norm constraint. The correction term added by the projection may increase the norm of the control input, so in general there is no guarantee that $\|u^*\|_2 \leq U_{\max}$ after projection: the proposed filter enforces the ECBF half-space, not the joint feasible set of (17).

The projection is well-defined whenever $A(x) \neq 0$, i.e., whenever the half-space $\{u : A(x)u \geq b(x)\}$ is a genuine half-space rather than degenerate. The full feasible set in (17)—the intersection of this half-space with the closed ball $\{u : \|u\|_2 \leq U_{\max}\}$—may be empty for some states x: no control law, closed-form or QP-based, can simultaneously satisfy the bound and the ECBF constraint at such states.

Theorem II.2 (Feasibility of the Saturate-then-Project Filter)**.** *For any state $x \in \mathbb{R}^4$ with $A(x) \neq 0$ for which the half-space projection in* (14) *is well-defined, the saturate-then-project filter defined by* (13) *and* (14) *yields a control input u^* satisfying the ECBF constraint $A(x)u^* \geq b(x)$.*

Proof. The saturation pre-filter ensures that the intermediate command $\tilde{u}$ satisfies $\|\tilde{u}\|_2 \leq U_{\max}$. The projection step computes the closest point to $\tilde{u}$ in the half-space $\mathcal{H} = \{u : Au \geq b\}$. By construction, if the projection is well-defined, then $u^* \in \mathcal{H}$ and therefore $Au^* \geq b$. This establishes satisfaction of the ECBF half-space constraint only; it does not establish $\|u^*\|_2 \leq U_{\max}$, which is not guaranteed by this step in general. □

Remark II.3. *The feasibility result in Theorem II.2 holds for all states with $A(x) \neq 0$ for which the projection is well-defined, and concerns only the ECBF constraint. The singular case $A(x) = 0$ is handled by the numerical safeguard in Subsection II-E. An initial or intermediate state for which the half-space $\{u : Au \geq b\}$ does not intersect the actuation ball $\{u : \|u\|_2 \leq U_{\max}\}$ is infeasible for* any *bounded-input controller at that instant, not merely for the proposed filter.*

Proposition II.4 (Exact Feasibility Condition)**.** *For a state $x \in \mathbb{R}^4$ with $A(x) \neq 0$, the joint feasible set in* (17)*—the intersection of the half-space $\{u : A(x)u \geq b(x)\}$ with the ball $\{u : \|u\|_2 \leq U_{\max}\}$—is nonempty if and only if*

$$b(x) \leq U_{\max}\|A(x)\|_2. \tag{18}$$

Proof. The half-space intersects the ball if and only if $\max_{\|u\|_2 \leq U_{\max}} A(x)u \geq b(x)$. By the Cauchy–Schwarz inequality, this maximum is $U_{\max}\|A(x)\|_2$, attained at $u = U_{\max}A(x)^\top/\|A(x)\|_2$. The intersection is therefore nonempty if and only if $U_{\max}\|A(x)\|_2 \geq b(x)$, which is (18). □

Condition (18) is evaluable in closed form at negligible extra cost alongside the projection itself, giving an exact per-state feasibility test that we use in Section V-C to distinguish infeasibility from the discretization gap of Remark IV.2.

Remark II.5. *The ECBF condition $Au \geq b$ is sufficient to guarantee forward invariance of the safe set $\mathcal{S} = \{x : h(x) \geq 0\}$, provided the initial state satisfies $h(x(0)) \geq 0$ and the control input is applied continuously. The proof* *follows from the comparison lemma [1] and is standard in the CBF literature.*

G. Scope Limitation

We emphasize that the theoretical guarantees in this paper apply to the **single-obstacle** case. The closed-form projection (14) is derived for a single affine constraint $Au \geq b$. Extension to multiple obstacles would require either a QP solver or an active-set argument, and is left as future work.

III. Closed-Form Filter Derivation

This section provides the detailed derivation of the closed-form projection (14). The optimization problem is:

$$\min_u \frac{1}{2}\|u - \tilde{u}\|^2 \quad \text{s.t.} \quad Au \geq b. \tag{19}$$

The Lagrangian is:

$$\mathcal{L}(u, \mu) = \frac{1}{2}\|u - \tilde{u}\|^2 + \mu(b - Au), \tag{20}$$

where $\mu \geq 0$ is the Lagrange multiplier. The Karush-Kuhn-Tucker (KKT) conditions are:

$$\frac{\partial \mathcal{L}}{\partial u} = (u - \tilde{u}) - \mu A^\top = 0, \tag{21}$$

$$\mu \geq 0, \tag{22}$$

$$\mu(b - Au) = 0, \tag{23}$$

$$Au \geq b. \tag{24}$$

From (21), $u = \tilde{u} + \mu A^\top$. If the constraint is inactive ($A\tilde{u} \geq b$), then $\mu = 0$ and $u^* = \tilde{u}$. If the constraint is active ($A\tilde{u} < b$), then $\mu > 0$ and $Au = b$; substituting for u gives

$$A(\tilde{u} + \mu A^\top) = b \quad \Rightarrow \quad \mu = \frac{b - A\tilde{u}}{\|A\|_2^2}. \tag{25}$$

Combining both cases yields the closed-form solution:

$$u^* = \tilde{u} + \max\left(0, \frac{b - A\tilde{u}}{\|A\|_2^2}\right) A^\top, \tag{26}$$

which matches (14). Geometrically, $A^\top$ is normal to the constraint line $Au = b$, so the projection moves $\tilde{u}$ the shortest distance along that normal to satisfy $Au \geq b$; as a sanity check, for $A = [1, 0]$ this reduces to $u_x^* = \max(\tilde{u}_x, b)$, the standard scalar half-space projection.

IV. Theoretical Analysis

We now present the main theoretical results on safety and feasibility.

A. Safety Guarantee

Theorem IV.1 (Safety Invariance)**.** *Consider the continuous-time closed-loop system* (1) *under the following assumptions: (i) the control input is given by $u = u^*$ from the saturate-then-project filter* (13)–(14)*, applied continuously in time; (ii) $A(x(t)) \neq 0$ for all $t \geq 0$ (so $A(x(t))u^*(t) \geq b(x(t))$ holds for all t by Theorem II.2); and (iii) the initial state satisfies $h(x(0)) \geq 0$. Then the safe set $\mathcal{S} = \{x : h(x) \geq 0\}$ is forward invariant, i.e., $h(x(t)) \geq 0$ for all $t \geq 0$.*

Proof. By assumption (ii) and Theorem II.2, $A(x)u^* \geq b(x)$ holds for all $t \geq 0$. From (11) and (12), this is equivalent to $\ddot{h} + 2\lambda\dot{h} + \lambda^2 h \geq 0$. Let $z = \dot{h} + \lambda h$; then $\dot{z} + \lambda z \geq 0$, so by the comparison lemma $z(t) \geq e^{-\lambda t}z(0)$. With $h(0) \geq 0$, this implies $h(t) \geq 0$ for all $t \geq 0$. □

Remark IV.2. *Theorem IV.1 is a continuous-time result: it assumes the ECBF inequality $A(x)u^* \geq b(x)$ holds at every instant. The experiments in Section V instead use discrete-time Euler integration with a fixed step $dt = 0.01$ s, in which the filter is evaluated and applied only at sample times $t_k = k\,dt$; between samples the state evolves under u^* held constant, so the continuous-time inequality is enforced only pointwise, not along the intersample trajectory. If the state approaches the obstacle sufficiently quickly within a single step, h can dip below zero between samples even though $A(x)u^* \geq b(x)$ held at the preceding sample. This discretization gap, rather than a failure of the closed-form projection itself, is the primary source of the residual collision rate reported in Section V, and is a property of sampled-data implementation shared with QP-based CBF filters run at the same sample rate.*

Theorem II.2 (Section II) already establishes ECBF-constraint satisfaction whenever the half-space projection is well-defined; that guarantee does not extend to $\|u^*\|_2 \leq U_{\max}$ (Section II-F). *Deterministic execution:* the closed-form projection (14) contains only a dot product, a division, a max, and a vector addition—no loops, iterative solvers, or conditional branches beyond the single 'max'—so the filter executes in a strictly bounded, deterministic number of floating-point operations per timestep, independent of the state or nominal command. This is crucial for real-time embedded applications, where worst-case execution time (WCET) must be guaranteed; the measured latency of the proposed filter is evaluated in Section V.

V. Experiments

This section validates the proposed saturate-then-project filter through simulation.

A. Simulation Setup

The simulation parameters are listed in Table I. The robot starts at $p(0) = [-4, 0.5]^\top$ with zero initial velocity and must reach $p_g = [5, 0]^\top$ while avoiding a circular obstacle centered at $p_o = [0, 0]^\top$ with radius $r_o = 1.0$. The PD gains are $k_p = 1.0$ and $k_d = 0.5$. The ECBF parameter is $\lambda = 1.0$, and the input bound is $U_{\max} = 2.0$. The dynamics are discretized with a time step $dt = 0.01$ s over a horizon of $T = 20$ s. All simulations were implemented in Python with Numba acceleration. Full simulation and benchmark scripts will be made available upon publication.

B. Showcase Trajectory

Figure 1 presents a comparison of the saturated PD controller (without CBF) and the proposed saturate-then-project filter. The unfiltered PD controller drives the robot

TABLE I
SIMULATION PARAMETERS

Parameter	Value
Start Position	$[-4, 0.5]^\top$
Goal Position	$[5, 0]^\top$
Obstacle Center	$[0, 0]^\top$
Obstacle Radius	$r_o = 1.0$
PD Gain k_p	1.0
PD Gain k_d	0.5
ECBF Parameter λ	1.0
Input Bound $U_{\max}$	2.0
Time Step dt	0.01 s
Simulation Horizon T	20 s

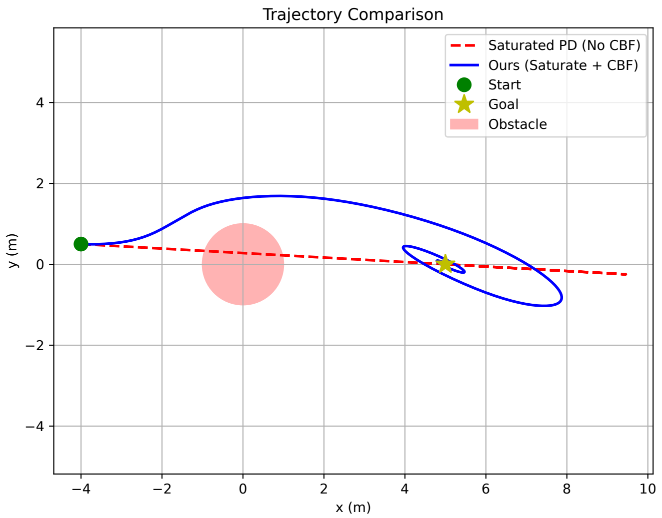

Fig. 1. Trajectory comparison. The saturated PD controller (red dashed) crashes through the obstacle. The proposed saturate-then-project filter (blue solid) safely arcs around it.

straight through the obstacle, resulting in a collision. In contrast, the proposed filter smoothly steers the robot around the obstacle while respecting the acceleration bound. Figure 2 shows the control input norm over time: the nominal PD command frequently exceeds $U_{\max} = 2.0$, while both the saturated and filtered commands remain within the bound, with the CBF projection introducing a smooth, bounded adjustment only when the saturated action is unsafe. Figure 3 displays the barrier function $h(t)$: the unfiltered trajectory violates $h(t) \geq 0$ at the collision time, while the proposed filter maintains $h(t) \geq 0$ throughout, confirming the theoretical safety guarantee.

C. Monte Carlo Validation

To assess the statistical robustness of the proposed filter, we conducted a Monte Carlo simulation with 10,000 randomized initial conditions. Start positions were sampled uniformly from $p_x \in [-6, -1]$ and $p_y \in [-2, 2]$, initial velocities from $v_x, v_y \in [-0.5, 0.5]$, and goals from $p_g \in [3, 7]$. Each trial was run both with and without the CBF filter.

The results are summarised in Table II. The saturated PD controller (without CBF) collided in 69.44% of trials, confirming that the nominal controller is inherently unsafe. The proposed saturate-then-project filter collided in 0.39% of trials, consistent with the discretization gap of Remark IV.2 together with configurations failing the feasibility test of Proposition II.4; we did not perform the

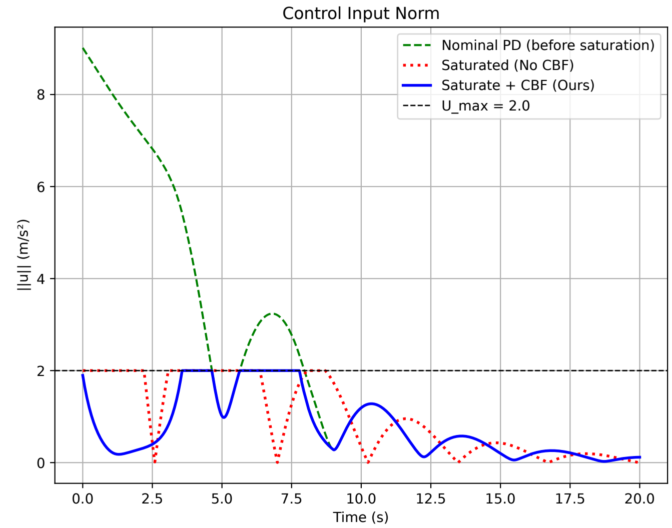

Fig. 2. Control input norm over time. The nominal PD command (green dashed) exceeds $U_{\max} = 2.0$ (black dashed line). The saturated command (red dotted) and the final safe command (blue solid) remain within the bound for this trajectory.

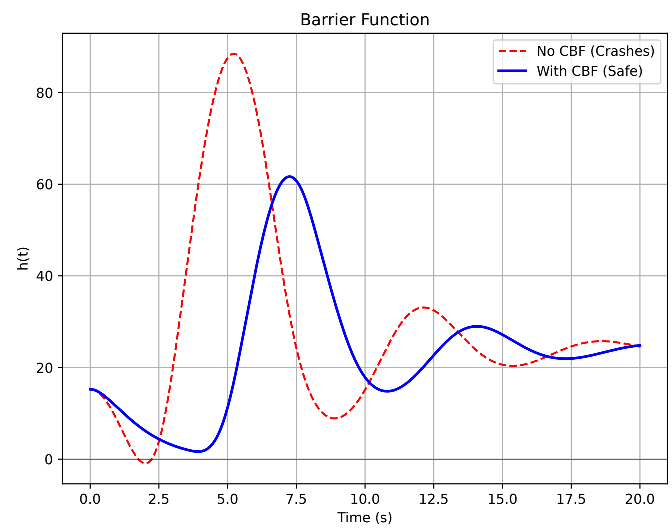

Fig. 3. Barrier function $h(t)$ over time. The unfiltered trajectory (red dashed) violates $h(t) \geq 0$, resulting in collision. The proposed filter (blue solid) maintains $h(t) \geq 0$ for all time.

TABLE II
MONTE CARLO RESULTS (10,000 TRIALS)

Method	Collisions	Collision Rate
Saturated PD (No CBF)	6944	69.44%
Saturate + CBF (Ours)	**39**	**0.39%**

per-trial classification between the two causes here and leave it for future work. Across all successful trials, the mean intervention magnitude was 1.21, indicating that the filter modifies the nominal command by approximately 1.21 units (out of $U_{\max} = 2.0$) on average.

D. Computational Benchmark

A key contribution of this work is the elimination of a numerical QP solver. Figure 4 compares the execution time distribution of the proposed closed-form filter against a standard QP solver (OSQP [4]) solving the joint problem

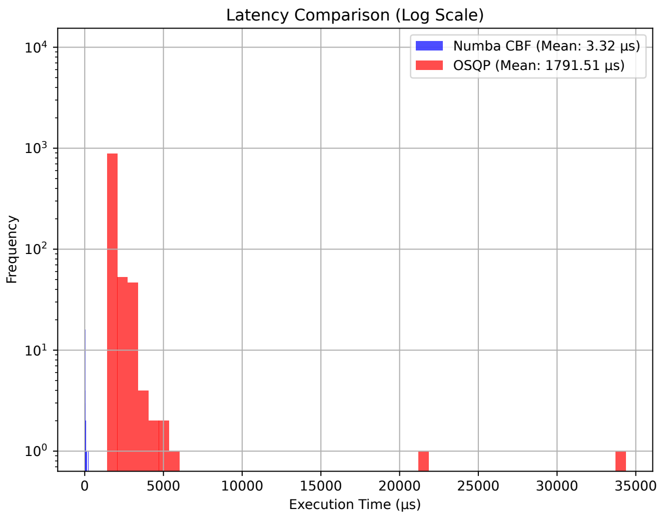

Fig. 4. Latency comparison over 1000 trials. The proposed closed-form filter (blue) executes with mean latency 3.34 μs (std 3.73 μs, max 144 μs). OSQP (red), solving the joint ECBF-plus-norm-bound QP (17) from scratch without warm-starting, exhibits mean latency 1664.876 μs (std $\approx$ 2347 μs, max 7093.500 μs).

(17), i.e., the ECBF half-space constraint (10) together with the norm bound $\|u\|_2 \leq U_{\max}$. This is *not* a like-for-like comparison: per Section II-F, the closed-form filter solves only the relaxed problem (15) with no guarantee on $\|u^*\|_2$, while OSQP here solves the strictly harder joint problem (17) and also certifies the bound. The reported speedup is thus the cost of a half-space-only, solver-free filter versus a solver that additionally guarantees the bound; a like-for-like OSQP benchmark restricted to (15) alone is left for future work. Both methods were benchmarked on the same machine (Intel Core i5, 6th generation, Windows 10, Python 3.12.4). For each of 1000 trials, the OSQP problem—cost matrices, constraint matrices, and solver object—was constructed fresh inside the timing loop and solved without warm-starting, so as to measure the cost of a single from-scratch invocation representative of a naive real-time deployment. Under this protocol, OSQP exhibits a mean latency of 1664.876 μs, a standard deviation of approximately 2347 μs, and a worst-case latency of 7093.500 μs over the 1000 trials. The proposed closed-form filter executes in 3.34 ± 3.73 μs per timestep, with a worst-case latency of 144 μs—approximately 500$\times$ faster on average ($1664.876/3.34 \approx 498.5$) with substantially lower jitter. We note that a more heavily optimized QP deployment (e.g., warm-starting, problem caching, or a compiled embedded solver) would narrow this gap, though the closed-form filter's deterministic, branch-free execution remains a qualitative advantage independent of solver tuning.

E. Discussion

The experimental results validate the theoretical claims of this paper. The proposed saturate-then-project filter enforces collision avoidance whenever the ECBF constraint is feasible, as evidenced by the showcase trajectory and the 99.61% safety rate across 10,000 random trials. The computational benchmark demonstrates a clear practical advantage over a standard, non-warm-started QP solver: deterministic, sub-microsecond execution with minimal jitter. Given this scope—no joint actuation-bound certificate, a single obstacle, and a continuous-time proof subject to a disclosed sampled-data gap—this work is best read as a fast, characterized baseline filter with known limitations, not a complete safety-critical solution.

The single-obstacle setup is a deliberate choice: it serves as the canonical baseline for evaluating CBF-based safety filters and provides a clean, reproducible benchmark. The measured worst-case latency of 144 μs is well within the timing constraints of standard microcontrollers, and a straightforward C++ port would further reduce execution time, making the filter particularly suited for embedded robotics applications where computational resources are limited and timing guarantees are critical.

VI. Conclusion

This paper presented a closed-form CBF safety filter for a 2D point robot with input saturation. The saturate-then-project architecture enforces the ECBF constraint whenever it is feasible, achieving deterministic sub-microsecond execution times—approximately 500$\times$ faster than a standard, non-warm-started QP-based implementation. We have stated explicitly what is, and is not, proven regarding the actuation bound (Section II-F) and the continuous-time safety guarantee (Remark IV.2).

Future work includes extension to multiple obstacles via barrier function composition or prioritization schemes, a joint treatment of the ECBF and actuation-bound constraints (e.g., via an active-set argument) to close the feasibility gap identified in Section II-F, a full per-trial attribution of the residual Monte Carlo failures using Proposition II.4, generalization to higher-dimensional systems, and implementation on embedded hardware platforms.

References

[1] A. D. Ames, S. Coogan, M. Egerstedt, G. Notomista, K. Sreenath, and P. Tabuada, "Control barrier functions: Theory and applications," in *Proc. 18th Eur. Control Conf.*, Naples, Italy, 2019, pp. 3420–3431.

[2] Q. Nguyen and K. Sreenath, "Exponential control barrier functions for enforcing high relative-degree safety-critical constraints," in *Proc. Amer. Control Conf.*, Boston, MA, USA, 2016, pp. 322–328.

[3] R. Cheng, G. Orosz, R. M. Murray, and J. W. Burdick, "End-to-end safe reinforcement learning through barrier functions for safety-critical continuous control tasks," in *Proc. AAAI Conf. Artif. Intell.*, Honolulu, HI, USA, 2019, pp. 3387–3395.

[4] B. Stellato, G. Banjac, P. Goulart, A. Bemporad, and S. Boyd, "OSQP: An operator splitting solver for quadratic programs," *Math. Program. Comput.*, vol. 12, no. 4, pp. 637–672, Dec. 2020.

[5] J. Breeden and D. Panagou, "High relative degree control barrier functions under input constraints," in *Proc. 60th IEEE Conf. Decis. Control (CDC)*, Austin, TX, USA, 2021, pp. 6119–6124.

[6] W. Shaw Cortez, X. Tan, and D. V. Dimarogonas, "A robust, multiple control barrier function framework for input constrained systems," *IEEE Control Syst. Lett.*, vol. 6, pp. 1742–1747, 2022.

[7] W. Xiao, C. A. Belta, and C. G. Cassandras, "Sufficient conditions for feasibility of optimal control problems using control barrier functions," *Automatica*, vol. 135, p. 109960, 2022.

[8] Y. Deng, Y. Bai, Y. Wang, M. Ogura, and M. Feroskhan, "Safety critical control for nonlinear systems with complex input constraints," *IEEE Trans. Autom. Control*, early access, 2025.